

---

# **Adversarial Content Manipulation for Analyzing and Improving Model Robustness**

---

A dissertation submitted towards the degree  
Doctor of Engineering (Dr.-Ing.)  
of the Faculty of Mathematics and Computer Science  
of Saarland University

by  
**Rakshith Shetty, M.Sc.**

Saarbrücken, 2021

Day of Colloquium                      27<sup>th</sup> of July, 2021

Dean of the Faculty                      Univ.-Prof. Dr. Thomas Schuster

**Examination Committee**

Chair    Prof. Dr. Vera Demberg

Reviewer, Advisor                      Prof. Dr. Bernt Schiele

Reviewer, Co-Advisor                      Prof. Dr. Mario Fritz

Reviewer                                      Prof. Dr. Tobias Ritschel

Reviewer                                      Prof. Dr. Antonio Torralba

Academic Assistant                      Dr. Paul Swoboda



# ABSTRACT

---

The recent rapid progress in machine learning systems has opened up many real-world applications — from recommendation engines on web platforms to safety critical systems like autonomous vehicles. A model deployed in the real-world will often encounter inputs far from its training distribution. For example, a self-driving car might come across a black stop sign in the wild. To ensure safe operation, it is vital to quantify the robustness of machine learning models to such out-of-distribution data before releasing them into the real-world. However, the standard paradigm of benchmarking machine learning models with fixed size test sets drawn from the same distribution as the training data is insufficient to identify these corner cases efficiently. In principle, if we could generate all valid variations of an input and measure the model response, we could quantify and guarantee model robustness locally. Yet, doing this with real world data is not scalable.

In this thesis, we propose an alternative, using generative models to create synthetic data variations at scale and test robustness of target models to these variations. We explore methods to generate semantic data variations in a controlled fashion across visual and text modalities. We build generative models capable of performing controlled manipulation of data like changing visual context, editing appearance of an object in images or changing writing style of text. Leveraging these generative models we propose tools to study robustness of computer vision systems to input variations and systematically identify failure modes. In the text domain, we deploy these generative models to improve diversity of image captioning systems and perform writing style manipulation to obfuscate private attributes of the user.

Our studies quantifying model robustness explore two kinds of input manipulations, *model-agnostic* and *model-targeted*. The model-agnostic manipulations leverage human knowledge to choose the kinds of changes without considering the target model being tested. This includes automatically editing images to remove objects not directly relevant to the task and create variations in visual context. Alternatively, in the *model-targeted* approach the input variations performed are directly adversarially guided by the target model. For example, we adversarially manipulate the appearance of an object in the image to fool an object detector, guided by the gradients of the detector. Using these methods, we measure and improve the robustness of various computer vision systems — specifically image classification, segmentation, object detection and visual question answering systems — to semantic input variations.



# ZUSAMMENFASSUNG

---

Der schnelle Fortschritt von Methoden des maschinellen Lernens hat viele neue Anwendungen ermöglicht – von Recommender-Systemen bis hin zu sicherheitskritischen Systemen wie autonomen Fahrzeugen. In der realen Welt werden diese Systeme oft mit Eingaben außerhalb der Verteilung der Trainingsdaten konfrontiert. Zum Beispiel könnte ein autonomes Fahrzeug einem schwarzen Stoppschild begegnen. Um sicheren Betrieb zu gewährleisten, ist es entscheidend, die Robustheit dieser Systeme zu quantifizieren, bevor sie in der Praxis eingesetzt werden. Aktuell werden diese Modelle auf festen Eingaben von derselben Verteilung wie die Trainingsdaten evaluiert. Allerdings ist diese Strategie unzureichend, um solche Ausnahmefälle zu identifizieren. Prinzipiell könnte die Robustheit “lokal” bestimmt werden, indem wir alle zulässigen Variationen einer Eingabe generieren und die Ausgabe des Systems überprüfen. Jedoch skaliert dieser Ansatz schlecht zu echten Daten.

In dieser Arbeit benutzen wir generative Modelle, um synthetische Variationen von Eingaben zu erstellen und so die Robustheit eines Modells zu überprüfen. Wir erforschen Methoden, die es uns erlauben, kontrolliert semantische Änderungen an Bild- und Textdaten vorzunehmen. Wir lernen generative Modelle, die kontrollierte Manipulation von Daten ermöglichen, zum Beispiel den visuellen Kontext zu ändern, die Erscheinung eines Objekts zu bearbeiten oder den Schreibstil von Text zu ändern. Basierend auf diesen Modellen entwickeln wir neue Methoden, um die Robustheit von Bilderkennungssystemen bezüglich Variationen in den Eingaben zu untersuchen und Fehlverhalten zu identifizieren. Im Gebiet von Textdaten verwenden wir diese Modelle, um die Diversität von sogenannten Automatische Bildbeschriftung-Modellen zu verbessern und Schreibstil-Manipulation zu erlauben, um private Attribute des Benutzers zu verschleiern.

Um die Robustheit von Modellen zu quantifizieren, werden zwei Arten von Eingabemanipulationen untersucht: Modell-agnostische und Modell-spezifische Manipulationen. Modell-agnostische Manipulationen basieren auf menschlichem Wissen, um bestimmte Änderungen auszuwählen, ohne das entsprechende Modell miteinzubeziehen. Dies beinhaltet das Entfernen von für die Aufgabe irrelevanten Objekten aus Bildern oder Variationen des visuellen Kontextes. In dem alternativen Modell-spezifischen Ansatz werden Änderungen vorgenommen, die für das Modell möglichst ungünstig sind. Zum Beispiel ändern wir die Erscheinung eines Objekts um ein Modell der Objekterkennung täuschen. Dies ist durch den Gradienten des Modells möglich. Mithilfe dieser Werkzeuge können wir die Robustheit von Systemen zur Bildklassifizierung oder -segmentierung, Objekterkennung und Visuelle Fragenbeantwortung quantifizieren und verbessern.



# ACKNOWLEDGEMENTS

---

I would like to thank my supervisors Prof. Mario Fritz and Prof. Bernt Schiele, for giving me the opportunity to work with you. I have learned so much in these four years in all aspects of doing research, from structuring experiments to how to communicate ideas effectively. I am grateful for all the guidance, great feedback and the support you provided during these years.

I would like to thank my committee members - Prof. Vera Demberg, Prof. Antonio Torralba, Prof. Tobias Ritschel and Dr. Paul Swoboda for taking the time to review my thesis and for the interesting discussion during the defense.

I would also like to thank all the people I have had a chance to work with and learn from during my PhD, Dr. Marcus Rohrbach, Dr. Lisa Anne Hendricks, Vedika Agarwal, Prof. Vera Demberg, Dr. Assad Sayeed, Tony Xudong, Farzaneh Rezaeianaran and Aymen Mir.

I would like to thank the fantastic colleagues and friends; Bharat Lal Bhatnagar, Mohamed Omran, Yongxin Xian, Hossein Hajipour, Tribhuvanesh Orekondy, Julian Steil, Alina Dima, Philipp Müller, Seong Joon Oh, Hosnieh Sattar, Qianru Sun, Apratim Bhattacharyya, Jan-Hendrik Lange, Verica Lazova, David Stutz, Moritz Böhle and Paul Swoboda. I learned a lot from all the interesting long discussions, fun coffee chats, Friday seminars and from all the great work you do. More importantly thank you for making life in Saarbrücken fun and filled with happy memories.

Special thanks to Connie Balzert, for all the work you do to keep the department running, being so welcoming and the numerous times you have helped me with official processes.

Thank you to my parents and my brother for being a constant source of support and love. Finally, to Klaara, thank you for going on this journey with me and all the million ways you have helped make this possible.



## CONTENTS

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contributions of the Thesis . . . . .	3
1.1.1	Automatic content manipulation . . . . .	4
1.1.2	Measuring model robustness . . . . .	7
1.2	Outline of the thesis . . . . .	9
<b>2</b>	<b>Related Work</b>	<b>13</b>
2.1	Models for Controlled Data Manipulation . . . . .	13
2.1.1	Generative adversarial networks . . . . .	14
2.1.2	Generative adversarial networks for discreet data . . . . .	14
2.1.3	Image manipulation with unpaired data . . . . .	15
2.2	Bias and Robustness in Machine Learning Models . . . . .	16
2.2.1	Robustness of machine learning models . . . . .	17
2.2.2	Generic language production in image captioning . . . . .	19
2.2.3	Overfitting in visual question answering . . . . .	21
2.3	Data Manipulation for Privacy and Robustness . . . . .	22
2.3.1	Privacy preserving data manipulation . . . . .	22
2.3.2	Improving robustness through data manipulation . . . . .	24
<b>I</b>	<b>Generative Language Models for Diversity and Privacy</b>	<b>27</b>
<b>3</b>	<b>Improving Diversity in Image Captioning</b>	<b>28</b>
3.1	Introduction . . . . .	29
3.2	Adversarial Caption Generator . . . . .	31
3.2.1	Caption generator . . . . .	31
3.2.2	Discriminator model . . . . .	34
3.2.3	Adversarial training . . . . .	35
3.3	Experimental Setup . . . . .	36
3.3.1	Insights in training the GAN . . . . .	36
3.4	Results . . . . .	37
3.4.1	Measuring if captions are human-like . . . . .	37
3.4.2	Comparing caption accuracy . . . . .	37
3.4.3	Comparing vocabulary statistics . . . . .	39
3.4.4	Ablation study . . . . .	42
3.5	Conclusions . . . . .	43
<b>4</b>	<b>Author Anonymization via Text Style Transfer</b>	<b>45</b>
4.1	Introduction . . . . .	46
4.2	Threat Model . . . . .	48

4.3	Author Attribute Anonymization . . . . .	48
4.3.1	Author attribute classifiers . . . . .	49
4.3.2	The A <sup>4</sup> NT network . . . . .	51
4.3.3	Style loss with GAN . . . . .	53
4.3.4	Preserving semantics . . . . .	54
4.3.5	Smoothness with language loss . . . . .	56
4.4	Experimental Setup . . . . .	56
4.4.1	Datasets . . . . .	57
4.4.2	Evaluation methods . . . . .	58
4.4.3	Baselines . . . . .	59
4.5	Experimental Results . . . . .	60
4.5.1	Quantitative evaluation . . . . .	60
4.5.2	Qualitative analysis . . . . .	66
4.6	Conclusions . . . . .	72

## **II Analyzing Model Robustness Through Image Manipulation 73**

<b>5</b>	<b>Adversarial Editing for Object Removal 74</b>
5.1	Introduction . . . . . 75
5.2	Learning to Remove Objects . . . . . 76
5.2.1	Two-staged editor architecture . . . . . 76
5.2.2	Mask priors . . . . . 77
5.2.3	Optimizing the in-painting network for removal . . . . . 78
5.3	Experimental Setup . . . . . 80
5.4	Results . . . . . 81
5.4.1	Qualitative results . . . . . 81
5.4.2	Quantitative evaluation of removal performance . . . . . 83
5.4.3	Ablation studies . . . . . 85
5.5	Conclusions . . . . . 86
<b>6</b>	<b>Measuring Context Sensitivity with Scene Editing 87</b>
6.1	Introduction . . . . . 88
6.2	Quantifying the Role of Context . . . . . 89
6.2.1	Object removal . . . . . 89
6.2.2	Measuring context dependency . . . . . 90
6.2.3	Data augmentation with object removal . . . . . 91
6.3	Experiments and Results . . . . . 93
6.3.1	Image level classification . . . . . 93
6.3.2	Semantic segmentation . . . . . 98
6.4	Conclusions . . . . . 101
<b>7</b>	<b>Measuring Spurious Correlations in VQA 103</b>
7.1	Introduction . . . . . 104



7.2	Synthetic Dataset for Variances and Invariances in VQA . . . . .	105
7.2.1	InVariant VQA (IV-VQA) . . . . .	106
7.2.2	CoVariant VQA (CV-VQA) . . . . .	108
7.3	Experiments: Consistency Analysis . . . . .	108
7.4	Robustification by Data Augmentation . . . . .	113
7.5	Conclusions . . . . .	115
<b>8</b>	<b>Semantic Adversarial Attacks on Appearance</b>	<b>117</b>
8.1	Introduction . . . . .	118
8.2	Synthesizing Semantic Adversarial Objects . . . . .	119
8.2.1	Synthesizer design . . . . .	120
8.2.2	Synthesizing semantic adversaries . . . . .	122
8.3	Experiments and Results . . . . .	124
8.3.1	Setup and datasets . . . . .	124
8.3.2	Semantic adversary for automated testing . . . . .	125
8.3.3	Semantic adversary for data augmentation . . . . .	128
8.4	Conclusions . . . . .	130
<b>9</b>	<b>Adverse Weather Testing</b>	<b>131</b>
9.1	Introduction . . . . .	132
9.2	Adversarial Weather Optimization . . . . .	133
9.2.1	Testing with simulator in loop . . . . .	133
9.2.2	Adversarially optimizing weather . . . . .	134
9.3	Experiments and Results . . . . .	136
9.3.1	Experimental setup . . . . .	136
9.3.2	Quantitative results . . . . .	137
9.3.3	Qualitative analysis of the failure modes . . . . .	138
9.4	Conclusions . . . . .	141
<b>10</b>	<b>Conclusions and Future Perspectives</b>	<b>143</b>
10.1	Key Contributions and Insights . . . . .	143
10.1.1	Automatic content manipulation . . . . .	143
10.1.2	Analyzing and improving robustness of computer vision systems .	144
10.2	Future Perspectives . . . . .	145
10.2.1	Short-term research questions . . . . .	145
10.2.2	Long-term perspectives . . . . .	147
	<b>List of Figures</b>	<b>149</b>
	<b>List of Tables</b>	<b>151</b>
	<b>Bibliography</b>	<b>153</b>



**Contents**

1.1	Contributions of the Thesis . . . . .	<b>3</b>
1.1.1	Automatic content manipulation . . . . .	4
1.1.2	Measuring model robustness . . . . .	7
1.2	Outline of the thesis . . . . .	<b>9</b>

---

What I cannot create, I do not understand

---

Richard Feynman

IMAGINATION plays an important role in human intelligence. It enables us to simulate new scenarios in our mind and plan how we can act in these scenarios. It is the key driver of the human ability to innovate. But it also serves an introspective function. Through imagination, we can relive previous experiences and examine the effect of the action we took. This kind of testing within our imagined scenarios helps us to identify how we could improve our behavior to get a better outcome. Visualization is an important component of imagination, where we can picture a scene/object in various configurations, including previously unseen ones. This enables us to be unfazed when encountering a novel object in the real world. For example, our vision system does not break down and is easily able to recognize the blue stop sign in Figure 1.1, although we might not have seen it before.

Generative models are the counterpart of human imagination in the field of machine learning. They aim to learn the data distribution and allow us to sample new data points from this distribution. Recent years have seen great progress in generative modeling in both image and text domains. This includes the rise of training frameworks like generative adversarial networks (GAN) and variational auto-encoders (VAE), and architectures improvements like instance normalization (Ulyanov *et al.*, 2016; Huang and Belongie, 2017) and transformers Vaswani *et al.* (2017). With these improvements, machine learning models are able to generate highly realistic looking images of structured domains like human faces<sup>1</sup> (Karras *et al.*, 2019) and coherent text paragraphs (Brown *et al.*, 2020). They have been widely used in creative applications like photo colorization (Nazeri *et al.*, 2018), animating humans (Chan *et al.*, 2019) and even generating art (Klingemann, 2018). Can we also use these generative models to build an introspective function similar to human imagination and use it to improve our machine learning models? We explore this question in the thesis and present works which positively answer this question in

---

<sup>1</sup><https://thispersondoesnotexist.com/>, created by Phil Wang, last accessed 22.05.2021



Figure 1.1: Atypical data points often cause failures in machine learning models. These failures are not often caught they often underrepresented in i.i.d. test sets.

different settings.

We study controlled content editing models in the thesis – i.e. generative models which edit the input data to create different variants of the input. This includes a model which edits the style of the input text in Chapter 4, a model which edits input images to remove a desired object class in Chapter 5 and a model which changes the appearance of an object in Chapter 8 (rightmost stop-sign in Figure 1.1 is created by this model). By focusing on editing inputs, rather than creating whole new samples, these models can create realistic synthetic variations in domains where unconditional generative models often struggle (e.g. crowded scenes). As a result, we can leverage these models to tackle problems where sample quality is important. There is however a critical challenge in building these content editing models – the lack of paired training data. Often it is not possible to find datasets with two variants of a sample which can supervise these models. Throughout the thesis we show that with a combination of generative adversarial network training and the right domain specific constraints we can tackle this challenge and train content editing models with unpaired data.

The thesis is divided into two parts. In Part I we study generative models for text. Our first work presented in Chapter 3 aims to improve diversity of image captioning systems. We propose a GAN based training framework to better leverage multiple human captions available for each image and show that it can significantly improve the diversity of caption generators. This work lays the technical foundations for our next work in Chapter 4, which builds a text style transfer model with the aim to edit the input sentence to obfuscate private attributes of the author, like age and gender. To learn this style transfer from unpaired data, we adversarially train our model against private attribute classifiers, and impose language smoothness and semantic consistency constraints on the output.

Part II of the thesis explores the broad theme of using image editing models to test and improve robustness of computer vision systems. Modern computer vision models have seen rapid improvement in performance over the last decade powered by deep learning and large annotated datasets. Benchmarks like ImageNet (Deng *et al.*, 2009) which helped drive this progress are nearly saturated (Beyer *et al.*, 2020). However this performance improvement has largely been on test sets which are independent and identically distributed (i.i.d.) to the training data. I.I.D testing is the dominant paradigm

in computer vision, with most benchmarks following this setup — e.g. ImageNet for classification, COCO (Lin *et al.*, 2014a) dataset for image captioning, object detection, Cityscapes (Cordts *et al.*, 2016) for semantic segmentation. When this i.i.d. assumption is broken – either by small changes like addition of noise, adversarial perturbation or by semantic changes like changes in context, unusual object appearances like in Figure 1.1 – computer vision models show significant drop in performance. This is a major bottleneck for real-world deployment of vision models. The problem is exacerbated by the fact that the standard i.i.d. benchmarks do not capture this problem and we do not have systematic methods to quantify the robustness of our vision models to such input distribution shifts.

In Part II, we propose creating controlled image variants through synthesis and use them to test robustness of computer vision systems. We start by building an object removal model in Chapter 5. In Chapters 6 and 7 this model is used to create context variations by removing co-occurring objects and test the robustness of image classification, semantic segmentation and visual question answering models. By comparing the target model’s responses on both original and edited images, we can measure how much it relies on spurious context correlations to make its predictions. In Chapter 8, we build a generative model which edits the appearance of an object while keeping its pose intact. This model is then used to create tailored hard test samples for an object detector model, by finding adversarial appearances. Finally, in Chapter 9, we aim to find worst-case weather configurations for a scene which break the semantic segmentation model, by adversarially optimizing simulator parameters.

The chapter is organized as follows. In Section 1.1 we will detail the contributions of the thesis, discussing the challenges and our solutions to overcome them. Then, Section 1.2 provides an outline of each chapter in the thesis.

## 1.1 CONTRIBUTIONS OF THE THESIS

This thesis makes contributions in two main directions: *generative models for content manipulation* and *measuring and improving robustness of computer vision systems*. We develop generative models of text and image content, focusing on controlled automatic editing. We apply the text generative models to improve diversity of image captioning systems and build a privacy preserving text editing tool. The image generative models are leveraged to automatically generate hard test samples to stress-test computer vision systems and find failure cases. In the course of our studies we make contributions in both generative modelling and measuring as well as improving model robustness. The following subsections will detail the challenges in the two above directions and the contributions we make to address these challenges. Summary of the chapter-wise contributions can be found in Section 1.2.

### 1.1.1 Automatic content manipulation

Content manipulation models are a particular class of generative models where the aim is to synthesize variations of an input data point. This problem has been studied for a long-time in the context of interactive image editing with the user in the loop to provide critical inputs (Oh *et al.*, 2001). These systems power many real-world applications with the most popular being the Photoshop tool<sup>2</sup>, enabling creative uses of interactive image editing. If similar content editing could be performed without the need of human interaction, it could potentially enable a wider array of applications. For example a model which can remove all occurrences of an object category from the input image can be used to create automatic filters for removing privacy violating (Orekondy *et al.*, 2018), copyright infringing or inappropriate content. This can also enable automatic testing suites, which systematically cover different variations in the input image and test downstream models for robustness to these variations. In this thesis we explore two such applications, privacy preserving text editing (Chapter 4) and large-scale testing of computer vision systems (Chapters 6 to 9), by developing automatic content manipulation systems. Generally, these systems start from an input data point, and automatically edit one desired attribute of this input while retaining the rest intact. Specifically, we target three different kinds of content manipulation. In Chapter 4, we build models to edit input text to change the writing style to mask private attributes of the author while retaining the semantic content. In Chapters 5 and 8, we focus on local image editing where we learn to remove an object and change appearance of an object respectively, without affecting the rest of the image. And finally in Chapter 9, we manipulate the weather in a scene while the objects are retained identical.

#### 1.1.1.1 Learning content editing models

**Challenges.** Generally, building machine-learning based content editing models is challenging due to the lack of paired training data. This is true for all content editing models discussed in this thesis (Chapters 4, 5 & 8). Specifically, we do not have access to two versions of the input where only the desired attribute changes. For example, we do not have different variants of the same *street scene* image with only the *cars* removed. Such paired data is prohibitively expensive to obtain, requiring significant manual effort. Hence, simple supervised training cannot be applied to train these generative models.

**Contributions.** Despite lack of paired data, we have another source of supervisory signal available. There is a lot of data showing the presence or absence of an attribute (e.g. different street scenes with and without cars). We can use this data to train attribute classifiers, which can in-turn supervise our generator via adversarial training. Specifically, we can adversarially train the generator to modify an attribute by training it to “fool” the corresponding attribute classifier. We exploit this same underlying principle, with necessary domain specific modifications, to build our text style transfer model in Chapter 4, object removal model in Chapter 5 and object appearance editor in Chapter 8. Exploiting adversarial training to overcome lack of supervision is not a new idea in itself.

---

<sup>2</sup><https://www.adobe.com/products/photoshop.html>, Adobe Inc., last accessed 22.05.2021

In particular, it has been applied in tasks like in domain adaptation (Ganin *et al.*, 2016; Shrivastava *et al.*, 2017) to align representations between source and target domains, and in semi-supervised learning (Dong and Lin, 2019). In this thesis, we demonstrate the usefulness and versatility of this approach for content editing, by building adversarially trained models for text and visual modalities. However, adversarial supervision is often under-constrained and alone is not sufficient to arrive at the right solution. In the following subsections we will discuss how combining adversarial losses with the additional task-specific constraints enables us to reach the desired solutions.

### 1.1.1.2 *Adversarially trained text generators*

**Challenges.** While adversarial training lets us learn from unpaired data, it is not straightforward to apply it on text generators due to the discrete output space. The output of text generators are discrete words which are sampled from a distribution over the vocabulary. This sampling process is not differentiable and therefore we cannot directly apply backpropagation to compute gradients of the generator weights w.r.t. the discriminator/attribute classifier output. While Yu *et al.* (2016) attempt to address this by using the REINFORCE rule (Williams, 1992) to approximate the gradients, it was only demonstrated on simple synthetic sequences.

**Contributions.** Our first work in this direction is the image captioning model in Chapter 3, aimed at improving caption diversity by using generative adversarial network (GAN) based training. Human written captions in image captioning datasets like COCO exhibit significant diversity in terms of sentence structure, vocabulary and topics referred to. However, most state-of-the-art image captioning systems tend to use generic language, overusing frequent words and n-grams. We address this by utilizing the multiple human-captions available for each image, and training our caption generator to better match the distribution of this caption set, by using adversarial training. Our discriminator network is designed to collectively score a set of generated captions, considering both similarity to the input image and mutual similarity of the captions in the set. This allows the discriminator to easily detect lack of diversity, and penalize it if it deviates from the human captions. Unlike prior work, we use the Gumbel-Softmax approximation (Jang *et al.*, 2016) to estimate the gradients for the discrete output. We demonstrate that our model significantly improves the diversity of generated captions, while maintaining the accuracy as judged by humans.

Next, using similar adversarial training techniques, in Chapter 4 we build a translation model (A<sup>4</sup>NT) which edits the input text to obfuscate the private attributes of the author. The goal of the obfuscation is to prevent authorship attribution methods from inferring private attributes like age, gender or identity of the author from the writing style. Our A<sup>4</sup>NT model learns to perform this style transformation using only unpaired data, by adversarially training to fool private attribute classifiers. Using just an adversarial loss to train the generator leads to the model outputting incoherent sentences which fool attribute classifiers. To address this, we propose two additional losses to measure unconditional language coherence and semantic consistency with the original input sentence. Combined, these losses teach our A<sup>4</sup>NT model to make small



changes which fool private attribute classifiers, while mostly preserving the semantics of the input text, as verified by automatic metrics and human evaluation.

### 1.1.1.3 *Controlled image manipulation*

**Challenges.** While there has been great progress in generative modelling of images, often the focus is on building models capable of sampling new data points either unconditionally (Goodfellow *et al.*, 2014; Karras *et al.*, 2019) or conditioned on certain attributes (Choi *et al.*, 2018; Chen and Koltun, 2017). This works well in structured domains like human faces or street scenes, but generating images of general crowded scenes is still very challenging (Johnson *et al.*, 2018; Ashual and Wolf, 2019). The complexity of the task increases greatly with unstructured scenes due to diversity in camera viewpoint, object location, poses and free-form interactions between objects. As a result, current state-of-the-art generative models still have low fidelity on general image datasets like COCO (Chen *et al.*, 2015), exhibiting artifact-filled image outputs with objects lacking clearly defined parts (e.g. see figure 4 in Hinz *et al.* (2020)). This is especially of concern in our target application, where we are trying to synthesize hard test samples. We do not want image quality to be a confounder when drawing conclusions from testing on generated data.

**Contributions.** Controlled image manipulation gets around this complexity problem, by focusing on editing only one aspect of the input image. Our approach enables us to generate realistic variations of complex scenes found in large datasets like COCO and ADE20k (Zhou *et al.*, 2017). This allows us to conduct our image manipulation based robustness analysis on the same standard benchmarks, instead of on a toy subset. Our first contribution in controlled image manipulation is the object removal model presented in Chapter 5. We develop a weakly-supervised image editor which, using only image-level labels, learns to remove all instances of the desired object class from the input image. We train the object removal model using unpaired data by adversarially training it to fool object classifiers. To avoid degenerate solutions where the generator uses adversarial patterns to fool the classifiers, we propose a two-staged generator architecture. It consists of a mask generator and an in-painter network which jointly learn to mask-out objects and fill in the background to produce realistic image respectively. To incorporate knowledge about object shapes, we encourage the mask generator to match a prior distribution by using a second unconditional mask discriminator. Our experiments show that this adversarial prior makes the masks more compact and improves the removal quality. We also show that our model achieves good removal performance without using any location annotations, matching the performance of a fully-supervised Mask-RCNN (He *et al.*, 2017) based solution.

Our second contribution in this direction is the object appearance editing model we discuss in Chapter 8. Our model learns to disentangle shape and appearance of an object, allowing it to synthesize new object appearance while maintaining the original pose. This disentanglement is achieved by deploying two image encoders, one to represent shape and other to represent appearance, while a decoder learns to reconstruct the object using these latent codes. A binary part-segmentation layer acts as a bottleneck



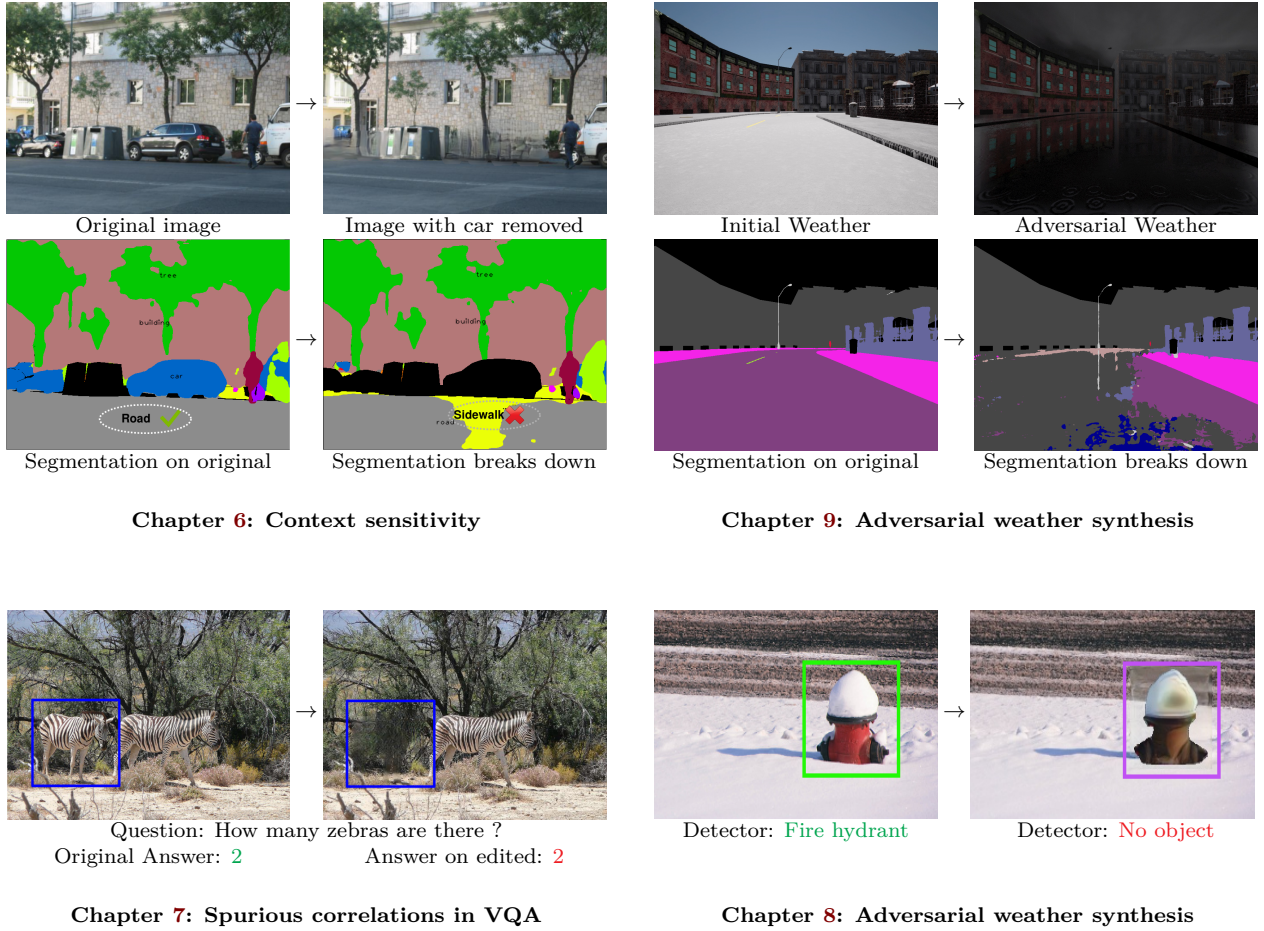


Figure 1.2: Overview of the different types of visual variations which are created in the thesis. We study the robustness of different computer vision systems to these variations.

at the shape encoder’s output, forcing it to focus only on representing the objects geometry. Qualitative results show that our model is able to smoothly interpolate object appearance without affecting the pose. Our model enables us to synthesize targeted corner cases for an object detector as discussed in the next section.

### 1.1.2 Measuring model robustness

To confidently deploy computer vision systems in the real world, especially in safety-critical applications like autonomous driving, we need methods to measure their robustness to expected input variations w.r.t. the training data. These variations can be surface level, such as from the camera noise, changes in lighting or adversarial perturbations. It can also be semantic, such as uncommon appearance of an object instance, or changes in visual context. Current fixed dataset driven model development and evaluation has drawbacks when it comes to measuring and improving model robustness to input variations. In the next subsections we will discuss the challenges and our contributions in this direction. Note that, when we refer to *model robustness* or *robustness* in rest of

the thesis, we mean robustness to input variations.

**Challenges.** The common approach to test machine learning models is to use shared fixed test sets, drawn from independent and identically distribution (i.i.d.) as the training data. Often, when benchmarking models on finite i.i.d. test-sets, input variations can be hard to capture and control for. For example, while our test set might contain a black car in day-light setting, it might not have a black car in night-setting. Using this finite test set, we will not capture the failure mode where our object detector fails to find black cars in the night, due to low-contrast. While this particular example can be addressed by collecting more data, it becomes infeasible to identify such failure cases at-scale through data collection. Possible test-cases explode combinatorially with the number of objects in the scene and their attributes.

We can categorize the efforts to study robustness into two main directions: a) dataset-level and b) surface-level variations. In dataset-level robustness, we study how well a model trained on dataset  $\mathcal{A}$  works on a related dataset  $\mathcal{B}$  or a set of related datasets, which contain some changes of the input distribution. Domain adaptation (Ben-David *et al.*, 2010; Wang and Deng, 2018) and domain generalization (Muandet *et al.*, 2013) can be classified under this umbrella. Works like Zendel *et al.* (2018) also fall under this category where a curated test-set for semantic segmentation is created to measure robustness to a specific set of hazard criteria. They often study semantic variations like context (Beery *et al.*, 2018), or appearance distribution changes (e.g. MNIST to SVHN). While dataset-level approaches allow us to obtain average estimates for performance variations under input shifts, it is hard to measure sample-level robustness; i.e. will my detector recognize this black car in the night.

The second class of works look at sample-level robustness to surface-level variations. Commonly studied are adversarial robustness (Carlini and Wagner, 2017) and robustness to a set of noise patterns (Hendrycks and Dietterich, 2019). They allow us to test robustness for every sample, but they are often restricted to easy-to-implement surface-level changes. Robustness to semantic variations like context or appearance is not explored at a sample-level.

**Contributions** We explore employing image editing models to automatically create test samples and find weaknesses in computer vision systems. The key idea here is to automatically create multiple variants of test samples using the controlled image editing models and measure the sensitivity of the target vision system to these variations. We explore two approaches to create these edits – first being *model-agnostic* which relies on human knowledge to decide what to change, and second being *model-specific* where the editing is guided by the particular model being tested. We study three variations in particular: 1) visual context (Chapters 6 and 7), 2) object appearance (Chapter 8) and 3) scene-level variations caused by weather (Chapter 9). See Figure 1.2 for a preview of these variations. Visual context editing is performed in a model-agnostic manner, by removing selected objects from the images. On the other hand, object appearance and weather variations are tailored to the target models by adversarially optimizing the synthesis.

Our first contribution in Chapter 6 studies the robustness of image classification and semantic segmentation models to changes in context created by co-occurring objects.

We employ the object removal model from Chapter 5 to remove one category at a time from an image, and measure the effect on the recognition/segmentation performance on other objects. Examples of this object removal based test-cases are shown in Figure 1.2. This analysis helps discover several undesirable associations learned by the model. For example, performance of road and sidewalk segmentation drops significantly when cars are absent. We extend this analysis to visual-question answering systems in Chapter 7. In addition to removing objects which are unrelated to the question-answer pair, we also explore removing objects which are referred to in the question-answer pair. First case is simple to measure, we expect the model’s answer to remain unchanged when we remove unrelated objects. However, the second case is only applied to specific question categories where we can use simple rules to determine how a model’s answer should change when a critical object is removed. For example, this applies to questions involving counting objects or inquiring of presence of an object. Applying this object removal based analysis to three state-of-the-art VQA approaches with different architectures, we show that modular architectures are more robust to contextual changes. In both Chapters 6 and 7 we show that robustness can be improved by augmenting the training data with edited samples.

In Chapter 8, we build an object appearance editing model, and use this to synthesize objects with rare appearance targeted to fool the object detector being tested. The targeting is achieved by adversarially optimizing the appearance latent code in the generator to minimize the detector’s confidence. In order to avoid degenerate solutions and keep the appearance plausible to a human observer, we propose to constrain the latent code optimization to the stay within the convex-hull spanned by guiding object instances. Our experiments show that this approach can create hard test samples which cause significant drop in detector performance, while still looking realistic to the human eye. An example of such a test case is shown in Figure 1.2. Adding these semantic adversarial examples to training, helps improve robustness and generalization to related datasets.

In Chapter 9, we study the robustness of semantic segmentation models to global visual changes caused by weather variations. Due to the lack of large datasets with good coverage of weather variations, we explore synthesizing these using the CARLA simulator (Dosovitskiy *et al.*, 2017). We find the worst-case weather setting for a given scene targeted to the segmentation model being tested, by adversarially optimizing the weather parameters. Since we cannot backpropagate through the simulator, we study effectiveness of finite-differences based gradient estimation and gradient-free black-box optimization techniques, to perform the adversarial optimization. Our experiments show that semantic segmentation models show a significant drop in performance on adversarially crafted weather conditions, compared to those seen in standard fixed test sets.

## 1.2 OUTLINE OF THE THESIS

We now summarize each chapter of the thesis, noting the publications to which each chapter corresponds. The thesis is divided into two parts. Part I (Chapters 3 - 4)

discusses two generative models in the text domain, focusing on diversity and privacy. Part II (Chapters 5 - 9) deals with generative models of images and leveraging them to study robustness of various computer vision systems.

**Chapter 2: Related work.** This chapter reviews the prior works relating to different topics covered in this thesis. In particular, we review related works on generative modeling for images and text, studies examining the bias and robustness issues in machine learning models and research utilizing data manipulation as a mechanism to improve model robustness and protect author privacy.

**Chapter 3: Improving Diversity in Image Captioning.** In this chapter we propose a GAN based training framework to improve the diversity of image captioning systems. Learning from a discriminator designed to operate on sets of captions, our caption generator learns to better match the statistics of human written captions. This approach leads to marked improvement in diversity metrics while maintaining the accuracy of captions.

The content in this chapter was published in ICCV 2017 with the title *Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training* (Shetty et al., 2017). Rakshith Shetty was the lead author of this publication. Marcus Rohrbach contributed with extracting visual features and running the human evaluation.

**Chapter 4: Author Anonymization via Text Style Transfer.** This chapter presents a model to protect privacy sensitive attributes of the author from automatic authorship attribution methods. Our model regenerates the input text in a different style, thereby hiding authors identity, while preserving the semantics. By training this model in a GAN framework we overcome the need for paired training of the same text in different styles.

The content in this chapter was published in USENIX 2018 with the title *A<sup>4</sup>NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation* (Shetty et al., 2018b). Rakshith Shetty was the lead author of this publication.

**Chapter 5: Adversarial Editing for Object Removal.** In this chapter we discuss a method for automatically removing objects from the input image with using only image-level supervision. Our main contributions here are a two-staged architecture with a mask generator and an inpainter, which avoids degenerate solutions and an adversarial mask prior which enables the model to learn object shapes. We demonstrate that our weakly-supervised model is able to achieve same levels of removal accuracy as a model fully supervised with pixel-level segmentation.

The content in this chapter was published in NeurIPS 2018 with the title *Adversarial Scene Editing: Automatic Object Removal from Weak Supervision* (Shetty et al., 2018a). Rakshith Shetty was the lead author of this publication.

**Chapter 6: Measuring Context Sensitivity with Scene Editing.** We study the robustness of image classification and semantic segmentation systems to changes in context in this chapter. We use the model developed in Chapter 5 to remove selected objects by image editing and measure the response of target computer vision system. Our analysis reveals that vision systems sometimes exploit spurious co-occurrence relationship found in the data and fail when these co-occurring objects are removed. We demonstrate data augmentation with our generated data can be used to alleviate this

problem to an extent.

The content in this chapter was published in CVPR 2020 with the title *Not Using the Car to See the Sidewalk—Quantifying and Controlling the Effects of Context in Classification and Segmentation* (Shetty *et al.*, 2019). Rakshith Shetty was the lead author of this publication.

**Chapter 7: Studying spurious correlations in VQA.** Similar to the previous chapter, we study the robustness of visual question answering models to changes in context through object removal. But going beyond Chapter 6, we also explore how consistent VQA models are under image editing which changes the answer. This analysis is conducted on three VQA variants and reveal that modular architectures are more robust to such contextual changes.

The content in this chapter was published in CVPR 2020 with the title *Towards Causal VQA: Revealing and Reducing Spurious Correlations by Invariant and Covariant Semantic Editing* (Agarwal *et al.*, 2020). Rakshith Shetty was the second author of this work and contributed the image editing model, planning the experiments and writing the paper. The work was carried out as part of the Master’s Thesis of Vedika Agarwal, under the instruction of Rakshith Shetty and Mario Fritz.

**Chapter 8: Semantic Adversarial Attacks on Appearance.** In this chapter, we quantify the robustness of object detectors to variations in object appearance. First we design a generative model capable of altering appearance of an object without affecting its pose. This model is used to adversarially edit the appearance of an object to fool the target detector, thereby synthesizing unusual appearances. We devise constraints to keep the generated appearance plausible to the human eye. This gives us a method to automatically mine difficult samples for testing and training the detector.

The content in this chapter was published in ECCV 2020 with the title *Towards Automated Testing and Robustification by Semantic Adversarial Data Generation* (Shetty *et al.*, 2020). Rakshith Shetty was the lead author of this publication.

**Chapter 9: Semantic Adversarial Attacks on Weather.** In this chapter, we study robustness of semantic segmentation models to global visual changes caused by weather variations in a simulator setting. To find the worst-case weather setting for a scene which breaks segmentation models, we adversarially optimize the weather configuration parameters of the CARLA simulator against the target model. We show that through gradient-free optimization we can find weather settings which cause catastrophic segmentation failures, causing performance drop of 20-30 mIoU points on different models.

Rakshith Shetty was the lead contributor of the work presented in this chapter.

**Chapter 10: Conclusions.** We summarize the key-takeaways from the thesis and discuss future directions of research in automated testing through generative models.





## Contents

2.1	Models for Controlled Data Manipulation . . . . .	<b>13</b>
2.1.1	Generative adversarial networks . . . . .	14
2.1.2	Generative adversarial networks for discrete data . . . . .	14
2.1.3	Image manipulation with unpaired data . . . . .	15
2.2	Bias and Robustness in Machine Learning Models . . . . .	<b>16</b>
2.2.1	Robustness of machine learning models . . . . .	17
2.2.2	Generic language production in image captioning . . . . .	19
2.2.3	Overfitting in visual question answering . . . . .	21
2.3	Data Manipulation for Privacy and Robustness . . . . .	<b>22</b>
2.3.1	Privacy preserving data manipulation . . . . .	22
2.3.2	Improving robustness through data manipulation . . . . .	24

THIS thesis develops generative models for controlled editing of image and text data modalities. We study the application of these generative models to measure and improve robustness of computer vision systems, and build privacy preserving tools. In this chapter we will discuss important prior works relating to each of these aspects. Section 2.1 presents the work in generative model research, focusing on the directions closely related to text and image editing models developed in this thesis. Section 2.2 discusses prior works which highlight the sensitivity of machine learning models to data distribution shifts. We will contrast prior works with our approach of using generative models to simulate these distribution shifts and quantify model robustness at scale. Finally Section 2.3 will discuss the applications of data manipulation through generative models to build privacy preserving tools for authorship obfuscation and improving robustness of computer vision models.

## 2.1 MODELS FOR CONTROLLED DATA MANIPULATION

In machine learning, given a training data distribution, a generative model aims to sample new data points from this distribution. Some approaches aim to explicitly model the data likelihood, for example Boltzmann machines (Hinton *et al.*, 1986), Variational Autoencoders (VAE) (Kingma and Welling, 2013), PixelRNN (Van Den Oord *et al.*, 2016), PixelCNN (Van den Oord *et al.*, 2016) and Normalizing Flows (Rezende and Mohamed, 2015). Another class of methods focuses only on sampling from the distribution and model the data likelihood implicitly, for example Generative Adversarial Networks (GAN) (Goodfellow *et al.*, 2014). Until recently, GANs handily outperformed explicit likelihood methods like VAEs on high dimensional image data in terms of sample fidelity.

Since sample quality is more important than the explicit measure of likelihood for the applications considered in this thesis, we utilize the GAN framework to build our generative models.

### 2.1.1 Generative adversarial networks

Generative adversarial networks (GAN) (Goodfellow *et al.*, 2014) are a framework where a generator learns by competing in an adversarial game against a discriminator network. The discriminator learns to distinguish between the real data samples and the “fake” generated samples. The generator is optimized to fool the discriminator into classifying generated samples as real. The generator can be conditioned on additional information to learn conditional generative models (Mirza and Osindero, 2014). GANs have been very effective for image generation with rapid progress in recent years. While early works (Denton *et al.*, 2015; Radford *et al.*, 2016) were only able to generate low resolution images, improvements in loss functions and architectures have led to models which can generate very high resolution images. For example, the Wasserstein loss proposed in Arjovsky and Bottou (2017) led to more stable training and the architectural improvements like progressive growing (Karras *et al.*, 2018) and style-based generators (Karras *et al.*, 2019, 2020) enabled scaling up to higher resolutions. This rapid progress has enabled various applications of image generation including photo colorization (Nazeri *et al.*, 2018), facial attribute manipulation (Choi *et al.*, 2018) and even contemporary art (Klingemann, 2018). In this thesis, we develop GAN based generative models for various tasks. In Chapters 3 and 4 we develop GAN based text generation models to improve captioning diversity and perform privacy preserving editing respectively. In Chapter 5, a GAN based editor model is trained to remove objects from images. Finally in Chapter 8, a GAN based model is trained to edit appearance of objects without affecting their pose.

### 2.1.2 Generative adversarial networks for discrete data

While GAN models are popular in image generation tasks, they are not often used in text generation. In image generation the output domain is continuous and one can backpropagate through the generator output in a straightforward manner. However, when generating text, output is discrete and it is unclear how to best back-propagate the loss through the sampling mechanism. A few works have looked at generating discrete distributions using GANs. Luc *et al.* (2016) aim to generate a semantic image segmentation with discrete semantic labels at each pixel by operating the discriminator directly on the generator softmax probabilities. Yu *et al.* (2016) use the REINFORCE trick to train an unconditional text generator using the GAN framework but diversity of the generated text is not considered. Contemporarily to our work in Chapter 3, a discrete GAN based model is used in dialogue generation in Li *et al.* (2017), achieving good results in human evaluation compared to standard maximum likelihood based training. Another contemporary work by Dai *et al.* (2017) applies this idea to image caption generation, improving caption diversity. While Li *et al.* (2017); Yu *et al.* (2016);



Dai *et al.* (2017) rely on the REINFORCE rule (Williams, 1992), our works in Chapters 3 and 4 use the Gumbel Softmax approximation (Jang *et al.*, 2016; Maddison *et al.*, 2016) to enable backpropagation through the discrete samples. This idea was introduced by Kusner and Hernández-Lobato (2016) for generating simple discrete sequences. However, our work in Chapter 3 extends this idea to natural text in the form of human written captions and shows that GAN training improves the diversity of captions generated (see Section 3.2.1 for further discussion). Follow up work by Fedus *et al.* (2018) further improves GAN based text generation by using actor-critic training and fill-in-the-blank tasks.

Adversarial training also enables learning text manipulations like style or sentiment transfer without the need for paired training data, by directly training to fool an attribute classifier (Shen *et al.*, 2017; Fu *et al.*, 2018). We exploit this idea in Chapter 4 to build a model to perform preserving transformations on input text, with minimal changes to the semantics.

### 2.1.3 Image manipulation with unpaired data

Conditional GANs have led to significant recent progress in many image manipulation tasks. Here, the generator takes an additional input apart from the noise distribution and learns to sample from the conditional distribution of the data given the inputs. Early works applied this idea to generate class conditional digits (Mirza and Osindero, 2014), text conditioned images (Reed *et al.*, 2016) and semi-supervised learning (Springenberg, 2016). A conditional GAN based image-to-image translation system was developed by Isola *et al.* (2016) to manipulate images using paired supervision data. Zhu *et al.* (2017) alleviate the need for paired supervision using cycle constraints and demonstrated translation between two different domains of unpaired images including (horse $\leftrightarrow$ zebras) and (summer $\leftrightarrow$ winter). Similar cyclic reconstruction constraints were extended to multiple domains to achieve facial attributes manipulation without paired data (Choi *et al.*, 2018). Nevertheless, these image manipulation works have been limited to object centric images like faces (Choi *et al.*, 2018) or constrained images like street scenes from a single point of view (Zhu *et al.*, 2017).

In our work in we move towards general scene-level manipulation, focusing on controlled editing to remove objects (Chapter 5) and edit appearance of an object (Chapter 8). Many prior works on scene-level images have focused on synthesizing entire images conditioned on text (Reed *et al.*, 2016; Huang *et al.*, 2017b; Xu *et al.*, 2018) and scene-graphs (Johnson *et al.*, 2018). But, the generated image quality on scene-level images (Johnson *et al.*, 2018) is still significantly worse than on structured data like faces (Karras *et al.*, 2018, 2019). In contrast, we focus on the manipulation of only parts of images rather than full image synthesis and achieve better image quality and control. This approach has been followed up in recent works and extended to more general manipulations like adding objects (Ntavelis *et al.*, 2020) and editing scene-graphs (Dhamo *et al.*, 2020).

**Object removal.** Our model proposed in Chapter 5 is a two-staged architecture with a mask-generator and image in-painter which jointly learn to remove the target

object class. Older works specifically targeting object removal focus on algorithmic improvements to in-painting while assuming users provide the object mask (Criminisi *et al.*, 2004; Hays and Efros, 2007; Mirkamali and Nagabhushan, 2015). One could argue that object segmentation masks can be obtained by a standalone segmenter like Mask-RCNN (He *et al.*, 2017) and just in-paint this masked region to achieve removal. However, this needs expensive mask annotation to supervise the segmentation networks for every category of image entity one wishes to remove for example objects or brand logos. Additionally, as we show in our experiments, even perfect segmentation masks are not sufficient for perfect removal. They tend to trace the object shapes too closely and leave object silhouettes giving away the object class. In contrast, our model learns to perform removal by jointly optimizing the mask generator and the in-painter for the removal task with only weak supervision from image-level labels. This joint optimization allows the two components to cooperate and achieve removal performance on par with a fully supervised segmenter based removal.

**Unsupervised disentangling of appearance and pose.** The architecture of our appearance editing synthesizer network in Chapter 8 is based on unsupervised generative models for disentangling object appearance and pose (Jakab *et al.*, 2018; Lorenz *et al.*, 2019; Siarohin *et al.*, 2019; Li *et al.*, 2019). These works aim to manipulate pose and appearance of objects independently by designing appropriate bottlenecks in representation to induce this disentanglement. However, they have so far been limited to single object categories like persons or birds. Most similar to our design is the model by Lorenz *et al.* (2019). Lorenz *et al.* (2019) use two separate encoders to create latent vectors of pose and appearance, with a Gaussian keypoint bottleneck regulating the pose encoding to carry only spatial information. The key difference in our work is that we propose binary segmentation maps as the bottleneck, which scales better to a larger number of diverse object classes seen in our experiments on COCO dataset.

## 2.2 BIAS AND ROBUSTNESS IN MACHINE LEARNING MODELS

In this thesis we leverage generative models to study robustness of various machine learning systems under data variations. This line of enquiry stems from prior works which have highlighted sensitivity of machine learning models to different perturbations to data, including addition of noise, simple transformations etc. This sensitivity arises from models overfitting to the biases found in the finite datasets they are trained on. In the following subsections, we will review prior works discussing robustness issues in machine learning models. Starting from general machine learning systems, we will narrow down on issues relating to the computer vision models that we deal with in the thesis, including image classifiers, object detectors, image captioning and visual question answering models.

### 2.2.1 Robustness of machine learning models

While generalization to new data has always been the primary concern in machine learning, often this new data is assumed to be drawn from the same distribution as the training data (i.i.d.). I.I.D assumption is at the heart of generalization guarantees provided by the Probably Approximately Correct (PAC) learning framework (Valiant, 1984). However, with the advent of deep learning, generalization to i.i.d. test sets has seen rapid progress in many computer vision tasks (He *et al.*, 2015). This has led to a growing interest in understanding and improving the performance of these models under data distribution shifts.

This problem has been studied from various perspectives, differing in the kind of distribution shifts considered. In domain adaptation (Ben-David *et al.*, 2010), the focus is on transferring a model learned from a source domain to a target data collected from a different domain (e.g. adapting an MNIST digit classifier to SVHN data). While domain adaptation often focusses on overt changes (Wang and Deng, 2018), machine learning models are also susceptible to more subtle changes like addition of a small amount of noise, geometric transformations in images or test set resampling (Recht *et al.*, 2019, 2018). This is the focus of robustness literature and it is closely related to our work. We refer the reader to Geirhos *et al.* (2020) for a comprehensive overview of robustness related problems in deep learning from the perspective of short-cut learning. Next we will discuss three categories of robustness — robustness to small noise patterns, to changes in context and to semantic variations in data.

#### 2.2.1.1 *Robustness to corruptions and adversarial noise*

Performance on different image corruptions has been proposed as a benchmark to quantify robustness of image classifiers by Hendrycks and Dietterich (2019). This benchmark was extended to include object detectors by Michaelis *et al.* (2019). These works demonstrate that models which perform well on standard i.i.d. benchmarks like ImageNet (Deng *et al.*, 2009) and COCO (Lin *et al.*, 2014a), show a large drop in performance under different input corruptions like additive gaussian noise, shot noise, changes in brightness etc. While these benchmarks consider fixed noise distributions, independent of the model being tested, model specific adversarial noise can also cause failures in machine learning models (Szegedy *et al.*, 2014). By directly optimizing against the target model, an adversary can craft a small additive noise pattern which is imperceptible to human eye, but causes catastrophic failures in state-of-the art image classifiers (Szegedy *et al.*, 2014; Goodfellow *et al.*, 2015; Carlini and Wagner, 2017). Adversarial attacks have also shown to be effective against a wide variety of machine learning models including object detectors (Xie *et al.*, 2017), image segmenters (Hendrik Metzen *et al.*, 2017) and even text classifiers (Liang *et al.*, 2018). In majority of adversarial attacks, the patterns are designed to be small and invisible to humans. In our work in Chapter 8, we show that the adversarial framework combined with a generative model can be used to craft semantic adversarial examples, where large and visible changes are made to the appearance of an object to fool an object detector while still remaining plausible to a

human observer.

### 2.2.1.2 *Robustness to semantic variations*

Apart from small noise patterns, computer vision models have been shown to be sensitive to geometric transformations of the input like translation and scaling (Azulay and Weiss, 2019) and rotations (Hamdi and Ghanem, 2019). Engstrom *et al.* (2019); Dumont *et al.* (2018) show that these translations and rotations can also be crafted adversarially to fool image classifiers. This is generalized to adversarial spatial deformations by Xiao *et al.* (2018a); Alaifari *et al.* (2019). With simulated data and 3D rendering, models can also be fooled by unusual object poses (Alcorn *et al.*, 2019) and lighting (Liu *et al.*, 2019). Difficult natural “adversarial” examples where ImageNet classifiers fail was collected by Hendrycks *et al.* (2021); Shankar *et al.* (2019). While Hendrycks *et al.* (2021) utilize model confidence to mine failure cases through image search on the web, Shankar *et al.* (2019) look for failure cases in long video frames. Both these approaches require manual curation and verification, and hence are expensive. In an effort to create a unified benchmark to measure robustness to semantic variations like rotations, viewpoint and background, Barbu *et al.* (2019) propose a new test set called *Objectnet* which controls for these attributes. Attempts to find failures by semantically changing object appearance have been limited to parametric color distortions (Hosseini and Poovendran, 2018) and using a generative model for faces (Song *et al.*, 2018) and digits (Stutz *et al.*, 2019).

Our work in Chapter 8, we go beyond these prior works in both scale and scope. We target real data and propose a method to adversarially manipulate object appearances to fool object detectors. Our approach leverages a generative model trained on the same datasets and is fully automatic, allowing us to efficiently synthesize failure cases across entire datasets. This gives us a method to quantify the robustness of an object detector to appearance variations. Additionally, we demonstrate the effectiveness of our approach on three large datasets with diverse object classes.

### 2.2.1.3 *Overfitting to context*

The importance of semantic context in visual recognition is well established with studies showing that context can help humans recognize objects faster e.g. when dealing with difficult low resolution images (Parikh *et al.*, 2012; Barenholtz, 2014). In computer vision, incorporating context information has been shown to improve performance in various tasks including object recognition (Marszalek and Schmid, 2007; Torralba *et al.*, 2010; Rabinovich *et al.*, 2007) and action recognition (Jain *et al.*, 2015), object detection (Bell *et al.*, 2016) and segmentation (Zhang *et al.*, 2018a). Early approaches built explicit context models by incorporating co-occurrences (Rabinovich *et al.*, 2007) and spatial location statistics (Desai *et al.*, 2011). Recently, explicit context modeling has been replaced by deep convolutional neural network (CNN) encoders which summarize the whole image into compact features. Classification and segmentation models, built on top of these deep features, can exploit information about object and context to achieve good performance (Long *et al.*, 2015; Durand *et al.*, 2016; Oquab *et al.*, 2015). Approaches

to improve the use of context in CNNs have been explored including using spatial pyramids (Zhao *et al.*, 2017a), atrous convolutions (Chen *et al.*, 2018) and learning context encoding with a separate neural network (Zhang *et al.*, 2018a). While this implicit context encoding with deep CNNs improves performance, it is less interpretable, and it is hard to know if the models’ decisions are based on object or contextual evidence.

To understand the decision process in CNNs and the role of context, methods have been proposed to inspect CNNs by visualizing salient regions for classification decisions (Ribeiro *et al.*, 2016; Zeiler and Fergus, 2014), and quantifying interpretability of individual units (Bau *et al.*, 2017). Petsiuk *et al.* (2018) propose erasing randomly sampled pixels to visualize important regions for a black-box model’s decision. While these visualization methods show salient regions, it is hard to reason about the importance of individual context objects in models prediction. By adding out-of-context objects into images Rosenfeld *et al.* (2018) show that object detection networks are brittle to the presence of out-of-context objects. In Chapter 6, we propose a method to quantify how much image classifiers and segmenters rely on context objects present in the image. Our approach utilizes the object removal model we develop in Chapter 5, to edit input images and remove objects. Comparing the network output under this change to its prediction on original image enables us to quantify the sensitivity of classification and segmentation models to context objects. Since it is an automated method, we perform this analysis on entire datasets and discover interesting and undesirable dependencies between classes. This approach has been extended to add more controls to the analysis in recent work by Xiao *et al.* (2021).

### 2.2.2 Generic language production in image captioning

Image captioning is the task of generating a natural language description of an input image. Early captioning models rely on first recognizing visual elements, such as objects, attributes, and activities, and then generating a sentence using language models such as a template model (Farhadi *et al.*, 2010), n-gram model (Kulkarni *et al.*, 2013), or statistical machine translation (Rohrbach *et al.*, 2013). Advances in deep learning have led to end-to-end trainable models that combine deep convolutional networks to extract visual features and recurrent networks to generate sentences (Donahue *et al.*, 2015; Vinyals *et al.*, 2015; Karpathy and Fei-Fei, 2015). Though modern description models are capable of producing coherent sentences which accurately describe an image, they tend to produce generic sentences which are often replicated from the train set (Devlin *et al.*, 2015). Furthermore, an image can correspond to many valid descriptions and this is reflected in the diversity of human written captions. However, at test time, sentences generated with methods such as beam search are generally very similar. This lack of diversity in language production partly occurs due to model amplifying the biases in the dataset and overfitting to the most common words and phrases. This bias-amplification can be dangerous, with models echoing and magnifying the gender biases present in the current dataset (Hendricks *et al.*, 2018; Zhao *et al.*, 2017b).

Reviewing the training methods of image captioning models gives us a clue on the source of this bias. The most common method is learning to predict a word  $w_t$



conditioned on an image and all previous *ground truth* words. At test time, each word is predicted conditioned on an image and previously *predicted* words. Consequently, at test time predicted words may be conditioned on words that were incorrectly predicted by the model. By only training on ground truth words, the model suffers from *exposure bias* (Ranzato *et al.*, 2016) and cannot effectively learn to recover when it predicts an incorrect word during training. To avoid this, Bengio *et al.* (2015) propose a scheduled sampling training scheme which begins by training with ground truth words, but then slowly conditions generated words on words previously produced by the model. However, Huszar (2015) shows that the scheduled sampling algorithm is inconsistent and the optimal solution under this objective does not converge to the true data distribution. Taking a different direction, Ranzato *et al.* (2016) propose to address the exposure bias by gradually mixing a sequence level loss (BLEU score) using REINFORCE rule with the standard maximum likelihood training. Several other works have followed this up with using reinforcement learning based approaches to directly optimize the evaluation metrics like BLEU, METEOR and CIDER (Rennie *et al.*, 2016; Liu *et al.*, 2017). However, directly optimizing for the evaluation metrics further reduces diversity (Wang and Chan, 2019). Since all current evaluation metrics use n-gram matching to score the captions, captions using more frequent n-grams are likely to achieve better scores than the ones using rarer and more diverse n-grams.

We study this problem in Chapter 3 and focus on improving diversity of generated captions. Our method achieves this by altering the training procedure and formulating the caption generator as a generative adversarial network. We design a discriminator that explicitly encourages generated captions to be diverse and indistinguishable from human captions. The generator is trained with an adversarial loss with this discriminator. Consequently, our model generates captions that better reflect the way humans describe images while maintaining similar correctness as determined by human evaluation.

Some prior works by Vijayakumar *et al.* (2016); Li *et al.* (2016) attempt to increase sentence diversity during inference, by integrating a diversity promoting heuristic into beam search. Taking a different approach, Wang *et al.* (2016b) increase the diversity in caption generation by training an ensemble of caption generators each specializing in different portions of the training set. Most similar to our work are concurrent works which use GANs for dialogue generation (Li *et al.*, 2017) and image caption generation (Dai *et al.*, 2017). While Li *et al.* (2017); Yu *et al.* (2016); Dai *et al.* (2017) rely on the reinforcement rule (Williams, 1992) to handle backpropagation through the discrete samples, we use the Gumbel Softmax approximation (Jang *et al.*, 2016). See Section 3.2.1 for further discussion. Li *et al.* (2017) aim to generate a diverse dialogue of multiple sentences, while we aim to produce diverse sentences for a single image. Additionally, Li *et al.* (2017) use both the adversarial and the maximum likelihood loss in each step of the generator training. In contrast we train the generator with only adversarial loss after pre-training. Concurrent work by Dai *et al.* (2017) also applies GANs to diversify generated image captions. Apart from using the Gumbel Softmax as discussed above, our work differs from Dai *et al.* (2017) in the discriminator design and quantitative evaluation of the generator diversity.

### 2.2.3 Overfitting in visual question answering

Amplification of data bias and lack of robustness to visual changes is also observed in the popular Visual question answering (VQA) task. VQA task requires a model to answer a question based on the input image. There has been a growing interest in VQA (Kafle and Kanan, 2017; Wu *et al.*, 2016a) recently, driven by the availability of large-scale datasets (Goyal *et al.*, 2017; Hudson and Manning, 2019; Antol *et al.*, 2015; Malinowski and Fritz, 2014; Deng *et al.*, 2009) and deep learning based advances in both vision and natural language processing. This interest has led to development of diverse model architectures for VQA (Lu *et al.*, 2015; Malinowski *et al.*, 2015; Ma *et al.*, 2016; Gao *et al.*, 2015), including simple models based on CNN+LSTMs (Lecun *et al.*, 1998; He *et al.*, 2016; Hochreiter and Schmidhuber, 1997), models using attention networks (Lu *et al.*, 2016; Kazemi and Elqursh, 2017; Yang *et al.*, 2016) and compositional module networks (Andreas *et al.*, 2016; Hu *et al.*, 2017, 2018; Hudson and Manning, 2018). In our work, we pick a representative model from each of these three design philosophies and study their robustness to semantic visual variations.

Existing VQA models often exploit language and contextual priors to predict the answers (Zhang *et al.*, 2016; Manjunatha *et al.*, 2019; Goyal *et al.*, 2017; Agrawal *et al.*, 2018). To understand how much do these models actually see and understand, various works have been proposed to study the robustness of models under different variations in the input modalities. Agrawal *et al.* (2018) show that changing the prior distributions for the answers across training and test sets significantly degrades models' performance. Ray *et al.* (2019); Shah *et al.* (2019) study the robustness of the VQA models towards linguistic variations in the questions. They show how different re-phrasings of the questions can cause the model to switch their answer predictions. In contrast, we study the robustness of VQA models to semantic manipulations in the image and propose a data augmentation technique to make the models robust. In order to counter the language priors in the VQA v1 dataset, Goyal *et al.* (2017) balance every question by collecting complementary images with a different answer. By construction, language priors are significantly weaker in the VQA v2 dataset.

While the above works study sensitivity of VQA models to language variations, sensitivity of these models to visual variations is not explored well. In Chapter 7 we study this problem, with the aid of a generative image editing model. By removing objects unrelated to the answer in a controlled fashion, we test if the VQA models still correctly answer the question. Similarly, by removing objects which predictably change the answer, we can test if a model's answer is consistent with the new image. This allows us to quantify model robustness and compare different architectures on both performance and robustness metrics. A recent work (Gokhale *et al.*, 2020) has extended this idea to include color editing as well as question manipulation into the testing arsenal.

## 2.3 DATA MANIPULATION FOR PRIVACY AND IMPROVING MODEL ROBUSTNESS

In this section we will discuss two applications of controlled data manipulations using generative models. First we will look at the task of authorship obfuscation in Section 2.3.1, where one tries to alter the writing style in order to make it harder for an adversary to identify the author. This relates to our work in Chapter 4. Then we look at how data manipulation can be used to create useful training data and improve model robustness in Section 2.3.2. This relates to our works in Chapters 6, 7 and 8.

### 2.3.1 Privacy preserving data manipulation

In Chapter 4, we develop an authorship obfuscation method A<sup>4</sup>NT, using a machine translation network trained in a generative adversarial network framework. The model performs controlled rewriting of the input text, altering its style to mask the authors private attributes, while trying to preserve its meaning. In this subsection we will provide the necessary context by reviewing related works for author attribute detection (our adversaries), authorship obfuscation (prior work) and machine translation (basis of our A<sup>4</sup>NT network).

**Authorship and attribute detection.** Authorship attribution methods use the stylistic properties of input text (e.g. grammar, synonym preference etc.) to infer the author’s identity or some private attributes of the author like age and gender (Mosteller and Wallace, 1963). Machine learning based approaches, where a set of text features are input to a classifier which learns to predict the author, have been popular in recent author attribution works (Stamatatos, 2009). These methods have been shown to work well on large datasets (Narayanan *et al.*, 2012), for duplicate author detection (Afroz *et al.*, 2014) and even on non-textual data like code (Caliskan-Islam *et al.*, 2015). Stylometric models can also be applied to determine private author attributes like age or gender (Argamon *et al.*, 2009). Classical author attribution methods rely on a predefined set of features extracted from the input text (Abbasi and Chen, 2008). Recently deep-learning methods have been applied to learn to extract the features directly from data (Bagnall, 2015; Ruder *et al.*, 2016). Bagnall (2015) use a multi-headed recurrent neural network (RNN) to train a generative language model on each author’s text and use the model’s perplexity on the test document to predict the author. Alternatively, Ruder *et al.* (2016) use convolutional neural network (CNN) to train author classifiers. To show generality of our A<sup>4</sup>NT network, we test it against both RNN and CNN based author attribute classifiers in Chapter 4.

**Authorship obfuscation.** Authorship obfuscation methods are adversarial in nature to stylometric methods of author attribution; they try to change the style of the input text so that the author identity is not discernible. The majority of prior works on author attribution are semi-automatic (Kacmarcik and Gamon, 2006; McDonald *et al.*, 2012), where the system suggests authors to make changes to the document by analyzing the stylometric features. The few available automatic obfuscation methods have relied



on general rephrasing methods like generic machine translation (Keswani *et al.*, 2016) or on predefined text transformations (Karadzhov *et al.*, 2017). Round-trip machine translation, where input text is translated to multiple languages one after the other until it is translated back to the source language, is proposed as an automatic method of obfuscation in Keswani *et al.* (2016). In a recent work Karadzhov *et al.* (2017) obfuscate text by moving the stylometric features towards the average values on the dataset by applying pre-defined transformations on input text. In Chapter 4, we propose the first method to achieve fully automatic obfuscation using text style transfer. This style transfer is not pre-defined but learnt directly from data optimized for fooling attribute classifiers. This allows us to apply our model across datasets without extra engineering effort.

**Machine translation.** The task of style-transfer of text data shares similarities with the machine translation problem. Both involve mapping an input text sequence onto an output text sequence. Style transfer can be thought of as machine translation on the same language.

Large end-to-end trainable neural networks have become a popular choice in machine translation (Bahdanau *et al.*, 2014; Wu *et al.*, 2016b). These methods are generally based on sequence-to-sequence recurrent models (Sutskever *et al.*, 2014) consisting of two networks, an encoder which encodes the input sentence into a fixed size vector and a decoder which maps this encoding to a sentence in the target language. We base our A<sup>4</sup>NT network architecture on the word-level sequence-to-sequence language model (Sutskever *et al.*, 2014). Neural machine translation systems are trained with large amounts of paired training data. However, in our setting, obtaining paired data of the same text in different writing styles is not viable. We overcome the lack of paired data by casting the task as matching style distributions instead of matching individual sentences. Specifically, our A<sup>4</sup>NT network takes an input text from a source distribution and generates text whose style matches the target attribute distribution. This is learnt without paired data using distribution matching methods. This reformulation allows us to demonstrate the first successful application of the machine translation models to the obfuscation task.

**Adversarial attacks in text domain.** Since we are training a network to fool attribute classifiers, it is related to methods which perform adversarial attacks against text classifiers. Recent works have shown that one can also fool NLP classifiers by deleting, adding or replacing few salient words (Samanta and Mehta, 2017; Liang *et al.*, 2018) and by adding whole sentences unrelated to the topic of the document (Jia and Liang, 2017). However, while the focus of these works is to fool the NLP classifiers with producing realistic text, there is no consideration to whether the meaning of the input text is preserved. Additionally the transformations performed are restricted to the predefined classes like add, remove or replace, with independently tuned heuristics for each of these transformations. In contrast, we propose a machine translation model which automatically learns to transform the input text appropriately to fool the attribute classifiers, while aiming to preserve the meaning of the input text.

### 2.3.2 Improving robustness through data manipulation

Controlled data manipulation can also be used to improve model robustness to data variations. A simple approach is to create hard examples through manipulation which can be added to augment the training data and improve the model performance to these variations. We apply this approach of generative model aided data augmentation in our work in Chapters 6, 7, 8 and 9. This is similar in spirit to standard data augmentation techniques, popular in deep learning since its early days (Krizhevsky *et al.*, 2012). But, transformations applied in standard data augmentations are usually global (applies to entire image), assume label invariance (i.e. image label does not change with the transformation) and are agnostic to the target model. In our work we break these assumptions. Chapter 6 explores data augmentation with local editing by removing objects. Chapter 7 breaks label-invariance assumption by removing objects to change the answer in VQA. Finally, Chapter 8 performs model-specific data augmentation by creating hard examples tailored to the target model. In the next subsections we review some recent data augmentation methods which aim to improve model generalization and robustness. We would like to note here that there are other paradigms to improve robustness of machine learning models which focus on the model training and loss functions used. This includes distributionally robust optimization (Sagawa *et al.*, 2020) where one tries to minimize the worst-case training error over a set of allowed distributional shifts, invariant risk minimization (Arjovsky *et al.*, 2019) which attempts to learn data representations that are simultaneously optimal across different data subsets and causal regularization (Heinze-Deml and Meinshausen, 2020) which penalizes network variance across different data points that share the same label and identity. Our work, in contrast, focusses on the data side; finding difficult variants of the data and exposing the model to them through data augmentation.

#### 2.3.2.1 Data augmentation for VQA

Data Augmentation has been used in VQA to improve a model’s performance either in terms of accuracy (Kafle *et al.*, 2017) or robustness against linguistic variations (Ray *et al.*, 2019; Shah *et al.*, 2019). Kafle *et al.* (2017) generated new questions by using existing semantic annotations and a generative approach via recurrent neural network. They showed that augmenting these questions gave a boost of around 1.5% points in accuracy. Shah *et al.* (2019) propose a cyclic-consistent training scheme where they generate different rephrasing of the question (based on the answer predicted by the model) and train the model such that answer predictions across the generated and the original question remain consistent. Ray *et al.* (2019) propose a data augmentation module that automatically generates entailed (or similar-intent) questions for a source QA pair and fine-tunes the VQA model if the VQA’s answer to the entailed question is consistent with the source QA pair.

In our work in Chapter 7, we augment the training data with images where context objects are removed through automatic image editing. While some of these removals are designed not to affect the answer, for specific question types like counting, we also

remove objects which change the answer in a controllable way. There have been some recent works which follow up and extend this idea. Gokhale *et al.* (2020) perform a broader range of image editing to include color-based questions in the augmentation. Teney *et al.* (2020) propose gradient supervision to better exploit the counterfactual samples created by image editing. Liang *et al.* (2020) apply contrastive losses between the original and counterfactual samples to learn better cross-modal embeddings and improve VQA performance on the VQA-CP dataset (Agrawal *et al.*, 2018).

### 2.3.2.2 Data augmentation for classification and object detection

Data augmentation techniques like random-flipping, cropping, affine transforms, brightness and contrast variations have been part of computer vision toolbox for a long-time (Lecun *et al.*, 1998; Simard *et al.*, 2003). While many of these transformations are applied globally to the image, DeVries and Taylor (2017) propose cutting out random boxes from the image to force the network to use all the available information. Cubuk *et al.* (2019) develop a meta-learning framework to learn the best series of data augmentation transformations directly from the data. Focus here is still improving the model’s performance on i.i.d. test sets. Hendrycks *et al.* (2019) leverage the data transformations learnt by AutoAugment (Cubuk *et al.*, 2019) to improve model robustness. By mixing different augmentations to create different versions of the same data point, they regularize the network response on these image versions to build invariances. This approach is shown to improve model robustness to common corruptions (Hendrycks and Dietterich, 2019). Geirhos *et al.* (2019) apply generative model based data augmentation, where style-transfer is applied on training data to improve robustness of the network. Hendrycks *et al.* (2020); Taori *et al.* (2020) provide a critical analysis of these approaches to improve model robustness. In Chapter 6 we apply our object removal tool to create hard examples with changes in context, and augment the training data. We demonstrate that this approach improves model generalization to a new dataset with objects in unusual contexts.

A related research area is the data augmentation for object detection that focuses on altering individual objects (Wang *et al.*, 2017; Dwibedi *et al.*, 2017; Dvornik *et al.*, 2018; Tripathi *et al.*, 2019; Wang *et al.*, 2019a). An early work by Wang *et al.* (2017) uses an adversary to partially mask objects to create hard occlusions. Objects are transferred onto new backgrounds for data augmentation with a cut-paste mechanism in the work by Dwibedi *et al.* (2017). Dvornik *et al.* (2018) refine this by also heeding to the context for picking a location to paste objects. Yet, this does not take the object pose into account. Tripathi *et al.* (2019) take the cut and paste approach further by training a network to predict the worst case position, rotation and scale of the added object to fool the detector. In our work in Chapter 8, we utilize a generative model to resynthesize the appearance of entire objects to fool the targeted detector, while preserving original context and pose. Thus our synthesis process allows wider range of semantic changes compared to occlusions, and better preserves realism and image context compared to the cut-and-paste approaches. We compare our approach to a recent work on data augmenting the object detector by switching instances of objects (Wang *et al.*, 2019a).

While Wang *et al.* (2019a) circumvent the context issue by switching instances in-place through shape matching, it does not allow generating targeted hard examples. We also find that our approach is complementary to global data augmentations learnt in Cubuk *et al.* (2019).

# Part I

## Generative Language Models for Diversity and Privacy

Generative modeling has seen great progress in recent years with the popularization of models like Generative Adversarial Networks (GANs) and Variational Auto Encoders (VAEs). Starting from blurry low-resolution images generated in the original work by Goodfellow *et al.* (2014), models which can generate high fidelity two mega-pixel images have been developed (Karras *et al.*, 2019). However, these techniques have not widely been used in the text domain due the output space being discrete, making it difficult to backpropagate through sampling. In this part of the thesis, we develop GAN-based text generation models, which overcome the discreteness problem by adopting the gumbel-softmax reparameterization trick (Jang *et al.*, 2016). We employ these models for two applications which exploit the strengths of GAN-based training, improving diversity of image captioning systems and learning to perform style editing with unpaired data.

In Chapter 3, we develop an image captioning model trained in the GAN framework. By designing the discriminator to score a set of generated captions, instead of a single one, it learns penalize the lack of diversity or accuracy in the generator. The generator learns to better match the diversity statistics of the human captions, leading to significant improvement over the baseline. In Chapter 4, we use similar techniques to design a text style translation network which is adversarially trained to rewrite input text to mask private author attributes. By using the GAN framework we overcome the need for paired data of the same text in different styles. Instead, our A<sup>4</sup>NT network learns to perform the style transfer by fooling authorship attribution networks, and additional loss functions designed to keep the output text natural and maintain the semantics of the input text.

## Contents

3.1	Introduction . . . . .	29
3.2	Adversarial Caption Generator . . . . .	31
3.2.1	Caption generator . . . . .	31
3.2.2	Discriminator model . . . . .	34
3.2.3	Adversarial training . . . . .	35
3.3	Experimental Setup . . . . .	36
3.3.1	Insights in training the GAN . . . . .	36
3.4	Results . . . . .	37
3.4.1	Measuring if captions are human-like . . . . .	37
3.4.2	Comparing caption accuracy . . . . .	37
3.4.3	Comparing vocabulary statistics . . . . .	39
3.4.4	Ablation study . . . . .	42
3.5	Conclusions . . . . .	43

IN this chapter we see how language diversity of image captioning systems can be improved through an adversarial training framework. While accuracy of image captioning systems has improved greatly in recent years, machine and human captions are still quite distinct. A closer look reveals that this is due to the deficiencies in the generated word distribution, vocabulary size, and strong bias in the generators towards frequent phrases. Furthermore, humans – rightfully so – generate multiple, diverse captions, due to the inherent ambiguity in the captioning task which is not often reflected in automatic captioning systems.

To address these challenges, we change the training objective of the caption generator from reproducing ground-truth captions to generating a set of captions that is indistinguishable from human generated captions. Instead of handcrafting such a learning target, we employ adversarial training in combination with an approximate Gumbel-Softmax sampler to implicitly match the generated distribution to the human one. While our method achieves comparable performance to the state-of-the-art in terms of the correctness of the captions, we generate a set of diverse captions, that are significantly less biased and match the word statistics better in several aspects.

### 3.1 INTRODUCTION

Image captioning systems have a variety of applications ranging from media retrieval and tagging to assistance for the visually impaired. In particular, models which combine state-of-the-art image representations based on deep convolutional networks and deep recurrent language models have led to ever increasing performance on evaluation metrics such as CIDEr (Vedantam *et al.*, 2015) and METEOR (Denkowski and Lavie, 2014a) as can be seen e.g. on the COCO image Caption challenge leaderboard (COCO, 2017).

Despite these advances, it is often easy for humans to differentiate between machine and human captions – particularly when observing multiple captions for a single image. As we analyze in this chapter, this is likely due to artifacts and deficiencies in the statistics of the generated captions, which is more apparent when observing multiple samples. Specifically, we observe that state-of-the-art systems frequently “reveal themselves” by generating a different word distribution and using smaller vocabulary. Further scrutiny reveals that generalization from the training set is still challenging and generation is biased to frequent fragments and captions.

Also, today’s systems are evaluated to produce a single caption. Yet, multiple potentially distinct captions are typically correct for a single image – a property that is reflected in human ground-truth. This diversity is not equally reproduced by state-of-the-art caption generators (Vijayakumar *et al.*, 2016; Li *et al.*, 2016).

Therefore, our goal is to make image captions less distinguishable from human ones – similar in the spirit to a Turing Test. We also embrace the ambiguity of the task and extend our investigation to predicting sets of captions for a single image and evaluating their quality, particularly in terms of the diversity in the generated set. In contrast, popular approaches to image captioning are trained with an objective to reproduce the captions as provided by the ground-truth.

Instead of relying on handcrafting loss-functions to achieve our goal, we propose an adversarial training mechanism for image captioning. For this we build on Generative Adversarial Networks (GANs) (Goodfellow *et al.*, 2014), which have been successfully used to generate mainly continuous data distributions such as images (Denton *et al.*, 2015; Radford *et al.*, 2016), although exceptions exist (Luc *et al.*, 2016). In contrast to images, captions are discrete, which poses a challenge when trying to backpropagate through the generation step. To overcome this obstacle, we use a Gumbel sampler (Jang *et al.*, 2016; Maddison *et al.*, 2016) that allows for end-to-end training.

We address the problem of caption set generation for images and discuss metrics to measure the caption diversity and compare it to human ground-truth. We contribute a novel solution to this problem using an adversarial formulation. The evaluation of our model shows that accuracy of generated captions is on par to the state-of-the-art, but we greatly increase the diversity of the caption sets and better match the ground-truth statistics in several measures. Qualitatively, our model produces more diverse captions across images containing similar content (Figure 3.1) and when sampling multiple captions for an image (Figure 3.2).





**Ours:** a person on skis jumping over a ramp



**Ours:** a skier is making a turn on a course



**Ours:** a cross country skier makes his way through the snow



**Ours:** a skier is headed down a steep slope

---

**Baseline:** a man riding skis down a snow covered slope

---

Figure 3.1: Four images from the test set, all related to skiing, shown with captions from our adversarial model and a baseline. Baseline model describes all four images with one generic caption, whereas our model produces diverse and more image specific captions.



Ours

a bus that has pulled into the side of the street  
 a bus is parked at the side of the road  
 a white bus is parked near a curb with people walking by

Base  
line

• a bus is parked on the side of the road  
 • a bus that is parked in the street  
 a bus is parked in the street next to a bus



a group of people standing outside in a old museum

an airplane show where people stand around  
 a line of planes parked at an airport show

a group of people standing around a plane  
 a group of people standing around a plane  
 a group of people standing around a plane

Figure 3.2: Two examples comparing multiple captions generated by our adversarial model and the baseline. Bi-grams which are top-20 frequent bi-grams in the training set are marked in red (e.g., “a group” and “group of”). Captions which are replicas from training set are marked with • .



## 3.2 ADVERSARIAL CAPTION GENERATOR

The image captioning task can be formulated as follows: given an input image  $x$  the generator  $G$  produces a caption,  $G(x) = [w_0, \dots, w_{n-1}]$ , describing the contents of the image. There is an inherent ambiguity in the task, with multiple possible correct captions for an image, which is also reflected in diverse captions written by human annotators (we quantify this in Table 3.4). However, most image captioning architectures ignore this diversity during training. The standard approach to model  $G(x)$  is to use a recurrent language model conditioned on the input image  $x$  (Donahue *et al.*, 2015; Vinyals *et al.*, 2015), and train it using a maximum likelihood (ML) loss considering every image-caption pair as an independent sample. This ignores the diversity in the human captions and results in models that tend to produce generic and commonly occurring captions from the training set, as we will show in Section 3.4.3.

We propose to address this by explicitly training the generator  $G$  to produce multiple diverse captions for an input image using the adversarial framework (Goodfellow *et al.*, 2014). In adversarial frameworks, a generative model is trained by pairing it with adversarial discriminator which tries to distinguish the generated samples from true data samples. The generator is trained with the objective to fool the discriminator, which is optimal when  $G$  exactly matches the data distribution. This is well-suited for our goal because, with an appropriate discriminator network we could coax the generator to capture the diversity in the human written captions, without having to explicitly design a loss function for it.

To enable adversarial training, we introduce a second network,  $D(x, s)$ , which takes as input an image  $x$  and a caption set  $S_p = \{s_1, \dots, s_p\}$  and classifies it as either real or fake. Providing a set of captions per image as input to the discriminator allows it to factor in the diversity in the caption set during the classification. The discriminator can penalize the generator for producing very similar or repeated captions and thus encourage the diversity in the generator.

Specifically, the discriminator is trained to classify the captions drawn from the reference captions set,  $R(x) = \{r_0, \dots, r_{k-1}\}$ , as real while classifying the captions produced by the generator,  $G(x)$ , as fake. The generator  $G$  can now be trained using an adversarial objective, i.e.  $G$  is trained to fool the discriminator to classify  $G(x)$  as real.

### 3.2.1 Caption generator

We use a near state-of-the art caption generator model based on Shetty *et al.* (2016). It uses the standard encoder-decoder framework with two stages: the encoder model which extracts feature vectors from the input image and the decoder which translates these features into a word sequence.

**Image features.** Images are encoded as activations from a pre-trained convolutional neural network (CNN). Captionin models also benefit from augmenting the CNN features with explicit object detection features (Shetty *et al.*, 2016). Accordingly, we extract a feature vector containing the probability of occurrence of an object and provide it as

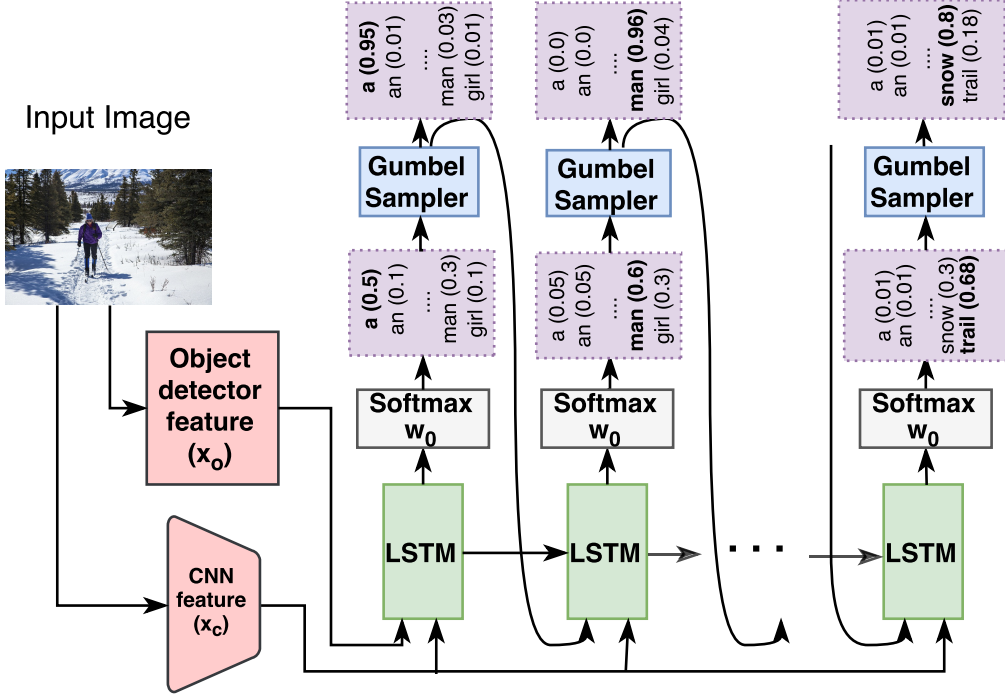


Figure 3.3: Caption generator model. Deep visual features are input to an LSTM to generate a sentence. A Gumbel sampler is used to obtain *soft* samples from the softmax distribution, allowing for backpropagation through the samples.

input to the generator.

**Language Model.** Our decoder shown in Figure 3.3, is adopted from a Long-Short Term Memory (LSTM) based language model architecture presented in Shetty *et al.* (2016) for image captioning. It consists of a three-layered LSTM network with residual connections between the layers. The LSTM network takes two features as input. First is the object detection feature,  $x_o$ , which is input to the LSTM at only 0th time step and shares the input matrix with the word vectors. Second is the global image CNN feature,  $x_c$ , and is input to the LSTM at all time-steps through its own input matrix.

The softmax layer at the output of the generator produces a probability distribution over the vocabulary at each step.

$$y_t = \text{LSTM}(w_{t-1}, x_c, y_{t-1}, c_{t-1}) \quad (3.1)$$

$$p(w_t | w_{t-1}, x) = \text{softmax}[\beta W_d * y_t], \quad (3.2)$$

where  $c_t$  is the LSTM cell state at time  $t$  and  $\beta$  is a scalar parameter which controls the peakyness of the distribution. Parameter  $\beta$  allows us to control how large a hypothesis space the generator explores during adversarial training. An additional uniform random noise vector  $z$ , is input to the LSTM in adversarial training to allow the generator to use the noise to produce diversity.

**Discreteness Problem.** To produce captions from the generator we could simply sample from this distribution  $p(w_t | w_{t-1}, x)$ , recursively feeding back the previously sampled word at each step, until we sample the END token. One can generate multiple sentences by sampling and pick the sentence with the highest probability as done

in Donahue *et al.* (2016). Alternatively we could also use greedy search approaches like beam-search. However, directly providing these discrete samples as input to the discriminator does not allow for backpropagation through them as they are discontinuous. Alternatives to overcome this are the reinforce rule/trick (Williams, 1992), using the softmax distribution, or using the Gumbel-Softmax approximation (Jang *et al.*, 2016; Maddison *et al.*, 2016).

Using policy gradient algorithms with the reinforce rule/trick (Williams, 1992) allows estimation of gradients through discrete samples (Hendricks *et al.*, 2016; Andreas and Klein, 2016; Yu *et al.*, 2016; Li *et al.*, 2017). However, learning using reinforce trick can be unstable due to high variance (Sutton and Barto, 1998) and some mechanisms to make learning more stable, like estimating the action-value for intermediate states by generating multiple possible sentence completions (e.g used in Yu *et al.* (2016); Dai *et al.* (2017)), can be computationally intensive.

Another option is to input the softmax distribution to the discriminator instead of samples. We experimented with this, but found that the discriminator easily distinguishes between the softmax distribution produced by the generator and the sharp reference samples, and the GAN training fails.

The last option, which we rely on in this work, is to use a continuous relaxation of the samples encoded as one-hot vectors using the Gumbel-Softmax approximation proposed in Jang *et al.* (2016) and Maddison *et al.* (2016). This continuous relaxation combined with the re-parametrization of the sampling process allows backpropagation through samples from a categorical distribution. The main benefit of this approach is that it plugs into the model as a differentiable node and does not need any additional steps to estimate the gradients. Whereas most previous methods to applying GAN to discrete output generators use policy gradient algorithms, we show that Gumbel-Softmax approximation can also be used successfully in this setting. An empirical comparison between the two approaches can be found in Jang *et al.* (2016).

The Gumbel-Softmax approximation consists of two steps. First Gumbel-Max trick is used to re-parametrize sampling from a categorical distribution. Given a random variable  $r$  drawn from a categorical distribution parametrized by  $\Theta = \theta_0, \dots, \theta_{v-1}$ ,  $r$  can be expressed as:

$$r = \text{one\_hot} \left[ \arg \max_i (g_i + \log \theta_i) \right], \quad (3.3)$$

where  $g_i$ 's are i.i.d. random variables from the standard gumbel distribution. Next the argmax in Equation (3.3) is replaced with softmax to obtain a continuous relaxation of the discrete random variable  $r$ .

$$r' = \text{softmax} \left[ \frac{g_i + \log \theta_i}{\tau} \right], \quad (3.4)$$

where  $\tau$  is the temperature parameter which controls how close  $r'$  is to  $r$ , with  $r' = r$  when  $\tau = 0$ .

We use straight-through variation of the Gumbel-Softmax approximation (Jang *et al.*, 2016) at the output of our generator to sample words during the adversarial

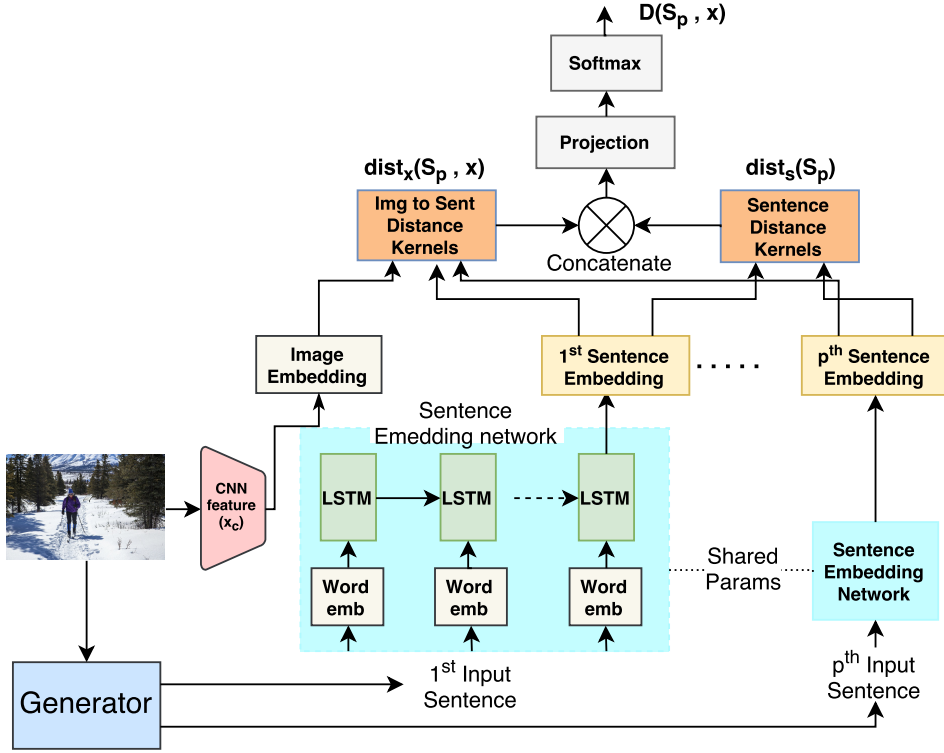


Figure 3.4: Discriminator Network. Caption set sampled from the generator is used to compute image to sentence ( $\text{dist}_x(S_p, x)$ ) and sentence-to-sentence ( $\text{dist}_s(S_p)$ ) distances. They are used to score the set as real/fake.

training. In the straight-through variation, sample  $r$  is used in the forward path and soft approximation  $r'$  is used in the backward path to allow backpropagation.

### 3.2.2 Discriminator model

The discriminator network,  $D$  takes an image  $x$ , represented using CNN feature  $x_c$ , and a set of captions  $S_p = \{s_1, \dots, s_p\}$  as input and classifies  $S_p$  as either real or fake. Ideally, we want  $D$  to base this decision on two criteria: a) do  $s_i \in S_p$  describe the image correctly? b) is the set  $S_p$  is diverse enough to match the diversity in human captions?

To enable this, we use two separate distance measuring kernels in our discriminator network as shown in Figure 3.4. The first kernel computes the distances between the image  $x$  and each sentence in  $S_p$ . The second kernel computes the distances between the sentences in  $S_p$ . The architecture of these distance measuring kernels is based on the minibatch discriminator presented in Salimans *et al.* (2016). However, unlike Salimans *et al.* (2016), we only compute distances between captions corresponding to the same image and not over the entire minibatch.

Input captions are encoded into a fixed size sentence embedding vector using an LSTM encoder to obtain vectors  $f(s_i) \in \mathbb{R}^M$ . The image feature,  $x_c$ , is also embedded into a smaller image embedding vector  $f(x_c) \in \mathbb{R}^M$ . The distances between  $f(s_i), i \in$

$\{1, \dots, p\}$  are computed as

$$K_i = T_s \cdot f(s_i) \quad (3.5)$$

$$c_l(s_i, s_j) = \exp(-\|K_{i,l} - K_{j,l}\|_{L_1}) \quad (3.6)$$

$$d_l(s_i) = \sum_{j=1}^p c_l(s_i, s_j) \quad (3.7)$$

$$\text{dist}_s(S_p) = [d_1(s_1), \dots, d_O(s_1), \dots, d_O(s_p)] \in \mathbb{R}^{p \times O} \quad (3.8)$$

where  $T_s$  is a  $M \times N \times O$  dimensional tensor and  $O$  is the number of different  $M \times N$  distance kernels to use.

Distances between  $f(s_i), i \in 1, \dots, p$  and  $f(x_c)$  are obtained with similar procedure as above, but using a different tensor  $T_x$  of dimensions  $M \times N \times O$  to yield  $\text{dist}_x(S_p, x) \in \mathbb{R}^{p \times O}$ . These two distance vectors capture the two aspects we want our discriminator to focus on.  $\text{dist}_x(S_p, x)$  captures how well  $S_p$  matches the image  $x$  and  $\text{dist}_s(S_p)$  captures the diversity in  $S_p$ . The two distance vectors are concatenated and multiplied with a output matrix followed by softmax to yield the discriminator output probability,  $D(S_p, x)$ , for  $S_p$  to be drawn from reference captions.

### 3.2.3 Adversarial training

In adversarial training both the generator and the discriminator are trained alternatively for  $n_g$  and  $n_d$  steps respectively. The discriminator tries to classify  $S_p^r \in R(x)$  as real and  $S_p^g \in G(x)$  as fake. In addition to this, we found it important to also train the discriminator to classify few reference captions drawn from a random image as fake, i.e.  $S_p^f \in R(y), y \neq x$ . This forces the discriminator to learn to match images and captions, and not just rely on diversity statistics of the caption set. The complete loss function of the discriminator is defined by

$$L(D) = -\log(D(S_p^r, x)) - \log(1 - D(S_p^g, x)) - \log(1 - D(S_p^f, x)) \quad (3.9)$$

The training objective of the generator is to fool the discriminator into classifying  $S_p^g \in G(x)$  as real. We found helpful to additionally use the feature matching loss (Salimans et al., 2016). This loss trains the generator to match activations induced by the generated and true data at some intermediate layer of the discriminator. In our case we use an  $l_2$  loss to match the expected value of distance vectors  $\text{dist}_s(S_p)$  and  $\text{dist}_x(S_p, x)$  between real and generated data. The generator loss function is given by

$$\begin{aligned} L(G) = & -\log(D(S_p^g, x)) + \left\| \mathbb{E}[\text{dist}_s(S_p^g)] - \mathbb{E}[\text{dist}_s(S_p^r)] \right\|_2 \\ & + \left\| \mathbb{E}[\text{dist}_x(S_p^g, x)] - \mathbb{E}[\text{dist}_x(S_p^r, x)] \right\|_2, \end{aligned} \quad (3.10)$$

where the expectation is over a training mini-batch.

### 3.3 EXPERIMENTAL SETUP

We conduct all our experiments on the MS-COCO dataset (Chen *et al.*, 2015). The training set consists of 83k images with five human captions each. We use the publicly available test split of 5000 images (Karpathy and Fei-Fei, 2015) for all our experiments. Section 3.4.4 uses a validation split of 5000 images.

For image feature extraction, we use activations from *res5c* layer of the 152-layered *ResNet* (He *et al.*, 2016) convolutional neural network (CNN) pre-trained on ImageNet. The input images are scaled to  $448 \times 448$  dimensions for *ResNet* feature extraction. Additionally we use features from the VGG network (Simonyan and Zisserman, 2015) in our ablation study in Section 3.4.4. Following Shetty *et al.* (2016), we additionally extract 80-dimensional object detection features using a Faster Region-Based Convolutional Neural Network (RCNN) (Ren *et al.*, 2015) trained on the 80 object categories in the COCO dataset. The CNN features are input to both the generator (at  $x_p$ ) and the discriminator. Object detection features are input only to the generator at the  $x_i$  input and is used in all the generator models reported here.

#### 3.3.1 Insights in training the GAN

As is well known (Arjovsky and Bottou, 2017), we found GAN training to be sensitive to hyper-parameters. Here we discuss some settings which helped stabilize the training of our models.

We found it necessary to pre-train the generator using standard maximum likelihood training. Without pre-training, the generator gets stuck producing incoherent sentences made of random word sequences. We also found pre-training the discriminator on classifying correct image-caption pairs against random image-caption pairs helpful to achieve stable GAN training. We train the discriminator for 5 iterations for every generator update. We also periodically monitor the classification accuracy of the discriminator and train it further if it drops below 75%. This prevents the generator from updating using a bad discriminator.

Without the feature matching term in the generator loss, the GAN training was found to be unstable and needed additional maximum likelihood update to stabilize it. This was also reported in Li *et al.* (2017). However with the feature matching loss, training is stable and the ML update is not needed.

A good range of values for the Gumbel temperature was found to be (0.1, 0.8). Beyond this range training was unstable, but within this range the results were not sensitive to it. We use a fixed temperature setting of 0.5 in the experiments reported here. The softmax scaling factor,  $\beta$  in (3.2), is set to value 3.0 for training of all the adversarial models reported here. The sampling results are also with  $\beta = 3.0$ .

## 3.4 RESULTS

We conduct experiments to evaluate our adversarial caption generator w.r.t. two aspects: how human-like the generated captions are and how accurately they describe the contents of the image. Using diversity statistics and word usage statistics as a proxy for measuring how closely the generated captions mirror the distribution of the human reference captions, we show that the adversarial model is more human-like than the baseline. Using human evaluation and automatic metrics we also show that the captions generated by the adversarial model performs similar to the baseline model in terms of correctness of the caption.

Henceforth, *Base* and *Adv* refer to the baseline and adversarial models, respectively. Suffixes *bs* and *samp* indicate decoding using beamsearch and sampling respectively.

### 3.4.1 Measuring if captions are human-like

**Diversity.** We analyze  $n$ -gram usage statistics, compare vocabulary sizes and other diversity metrics presented below to understand and measure the gaps between human written captions and the automatic methods and show that the adversarial training helps bridge some of these gaps. To measure the corpus level diversity of the generated captions we use:

- *Vocabulary Size* - number of unique words used in all generated captions
- *% Novel Sentences* - percentage of generated captions not seen in the training set.

To measure diversity in a set of captions,  $S_p$ , corresponding to a single image we use:

- *Div-1* - ratio of number of unique unigrams in  $S_p$  to number of words in  $S_p$ . Higher is more diverse.
- *Div-2* - ratio of number of unique bigrams in  $S_p$  to number of words in  $S_p$ . Higher is more diverse.
- *mBleu* - Bleu score is computed between each caption in  $S_p$  against the rest. Mean of these  $p$  Bleu scores is the mBleu score. Lower values indicate more diversity.

**Correctness.** Just generating diverse captions is not useful if they do not correctly describe the content of an image. To measure the correctness of the generated captions we use two automatic evaluation metrics Meteor (Denkowski and Lavie, 2014a) and SPICE (Anderson *et al.*, 2016). However since it is known that the automatic metrics do not always correlate very well with human judgments of the correctness, we also report results from human evaluations comparing the baseline model to our adversarial model.

### 3.4.2 Comparing caption accuracy

Table 3.1 presents the comparison of our adversarial model to the baseline model. Both the baseline and the adversarial models use *ResNet* features. The beamsearch results are with beam size 5 and sampling results are with taking the best of 5 samples. Here the best caption is obtained by ranking the captions as per probability assigned by the model.



Method	Meteor	Spice
ATT-FCN (You <i>et al.</i> , 2016)	0.243	–
MSM (Yao <i>et al.</i> , 2017)	0.251	–
KWL (Lu <i>et al.</i> , 2017)	0.266	<b>0.194</b>
Ours Base-bs	<b>0.272</b>	0.187
Ours Base-samp	0.265	0.186
Ours Adv-bs	0.239	0.167
Ours Adv-samp	0.236	0.166

Table 3.1: Comparing captioning correctness of baseline and adversarial models using Meteor and Spice metrics.

Method	Spice					
	Color	Attribute	Object	Relation	Count	Size
Base-bs	<b>0.101</b>	<b>0.085</b>	0.345	0.049	0.025	0.034
Base-samp	0.059	0.069	<b>0.352</b>	<b>0.052</b>	0.032	0.033
Adv-bs	0.079	0.082	0.318	0.034	<b>0.080</b>	0.052
Adv-samp	0.078	0.082	0.316	0.033	0.076	<b>0.053</b>

Table 3.2: Detailed look at different categories of the Spice metric.

Table 3.1 also shows the metrics from some recent methods from the image captioning literature. The purpose of this comparison is to illustrate that we use a strong baseline and that our baseline model is competitive to recent published work, as seen from the Meteor and Spice metrics.

Comparing baseline and adversarial models in Table 3.1 the adversarial model does worse in-terms of Meteor scores and overall spice metrics. When we look at Spice scores on individual categories shown in Table 3.2 we see that adversarial models excel at counting relative to the baseline and describing the size of an object correctly.

However, it is well known that automatic metrics do not always correlate with human judgments on correctness of a caption. A primary reason the adversarial models do poorly on automatic metrics is that they produce significantly more unique sentences using a much larger vocabulary and rarer  $n$ -grams, as shown in Section 3.4.3. Thus, they are less likely to do well on metrics relying on  $n$ -gram matches.

To verify this claim, we conduct human evaluations comparing captions from the baseline and the adversarial model. Human evaluators from Amazon Mechanical Turk are shown an image and a caption each from the two models and are asked “Judge which of the two sentences is a better description of the image (w.r.t. correctness and relevance)!”. The choices were either of the two sentences or to report that they are the same. Results from this evaluation are presented in Table 3.3. We can see that both adversarial and baseline models perform similarly, with adversarial models doing slightly better. This shows that despite the poor performance in automatic evaluation metrics, the adversarial models produce captions that are similar, or even slightly better, in accuracy to the baseline model.



Comparison	Adversarial - Better	Adversarial - Worse
Beamsearch	36.9	34.8
Sampling	35.7	33.2

Table 3.3: Human evaluation on caption correctness comparing adversarial model vs the baseline model on 482 random samples. With agreement of at least 3 out of 5 judges in %. Humans agreed in 89.2% and 86.7% of images in beamsearch and sampling cases respectively.

Method	n	Div-1	Div-2	mBleu-4	Vocab- ulary	% Novel Sentences
Base-bs	1 of 5	–	–	–	756	34.18
	5 of 5	0.28	0.38	0.78	1085	44.27
Base-samp	1 of 5	–	–	–	839	52.04
	5 of 5	0.31	0.44	0.68	1460	55.24
Adv-bs	1 of 5	–	–	–	1508	68.62
	5 of 5	0.34	0.44	0.70	2176	72.53
Adv-samp	1 of 5	–	–	–	1616	73.92
	5 of 5	<b>0.41</b>	<b>0.55</b>	<b>0.51</b>	<b>2671</b>	<b>79.84</b>
Human captions	1 of 5	–	–	–	3347	92.80
	5 of 5	0.53	0.74	0.20	7253	95.05

Table 3.4: Diversity Statistics described in Section 3.4.1. Higher values correspond to more diversity in all except mBleu-4, where lower is better.

### 3.4.3 Comparing vocabulary statistics

To characterize how well the captions produced by the automatic methods match the statistics of the human written captions, we look at  $n$ -gram usage statistics in the generated captions. Specifically, we compute the ratio of the actual count of an  $n$ -gram in the caption set produced by a model to the expected  $n$ -gram count based on the training data.

Given that an  $n$ -gram occurred  $m$  times in the training set we can expect that it occurs  $m * |\text{test-set}|/|\text{train-set}|$  times in the test set. However actual counts may vary depending on how different the test set is from the training set. We compute these ratios for reference captions in the test set to get an estimate of the expected variance of the count ratios. The left side of Figure 3.5 shows the mean count ratios for uni-, bi- and tri-grams in the captions generated on test-set plotted against occurrence counts in the training set. Histogram of these ratios are shown on the right side.

Count ratios for the reference captions from the test-set are shown in green. We see that the  $n$ -gram counts match well between the training and test set human captions and the count ratios are spread around 1.0 with a small variance.

The baseline model shows a clear bias towards more frequently occurring  $n$ -grams.

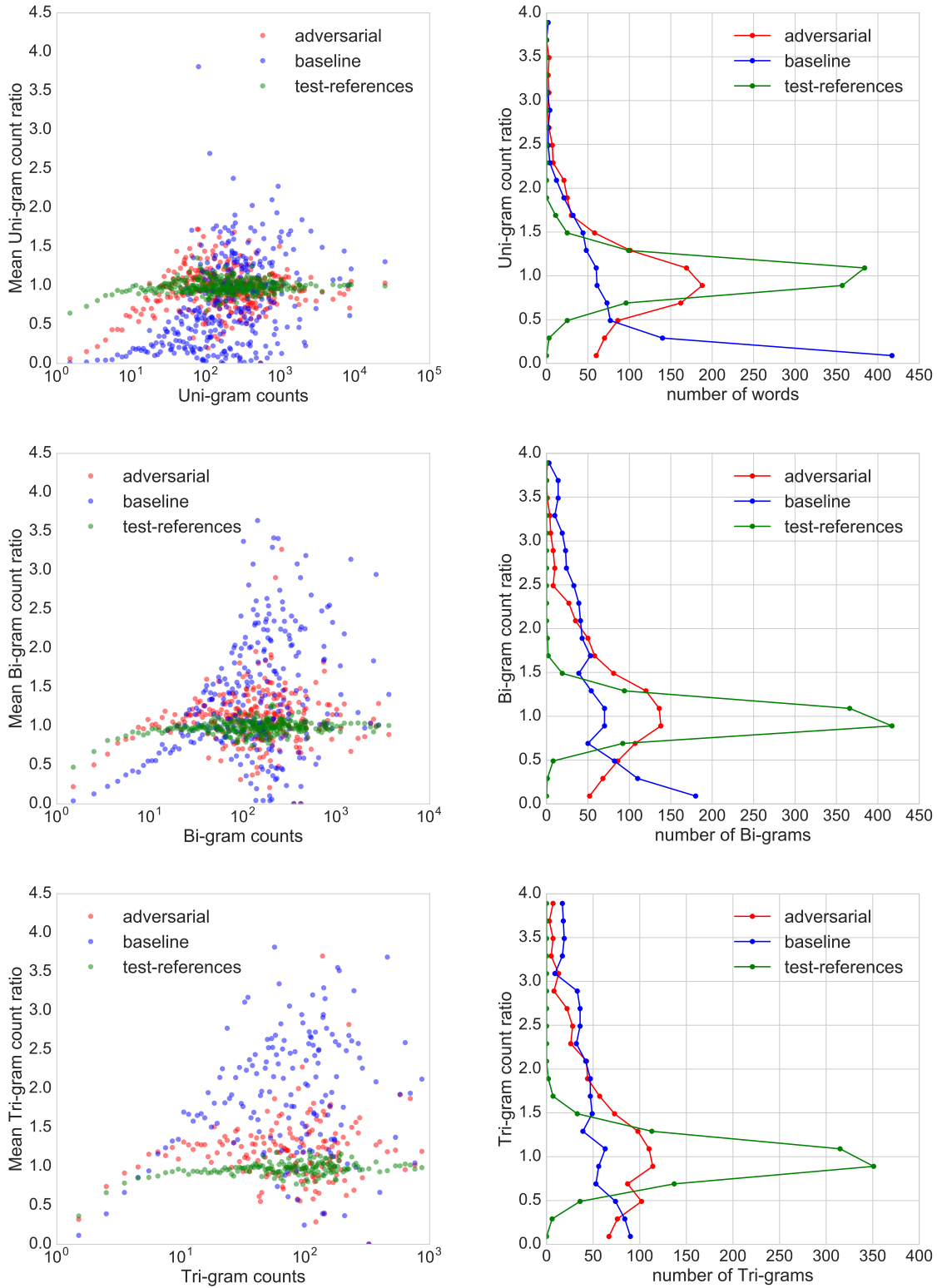


Figure 3.5: Comparison of  $n$ -gram count ratios in generated test-set captions by different models. Left side shows the mean  $n$ -gram count-ratios as a function of counts on training set. Right side shows the histogram of the count-ratios.

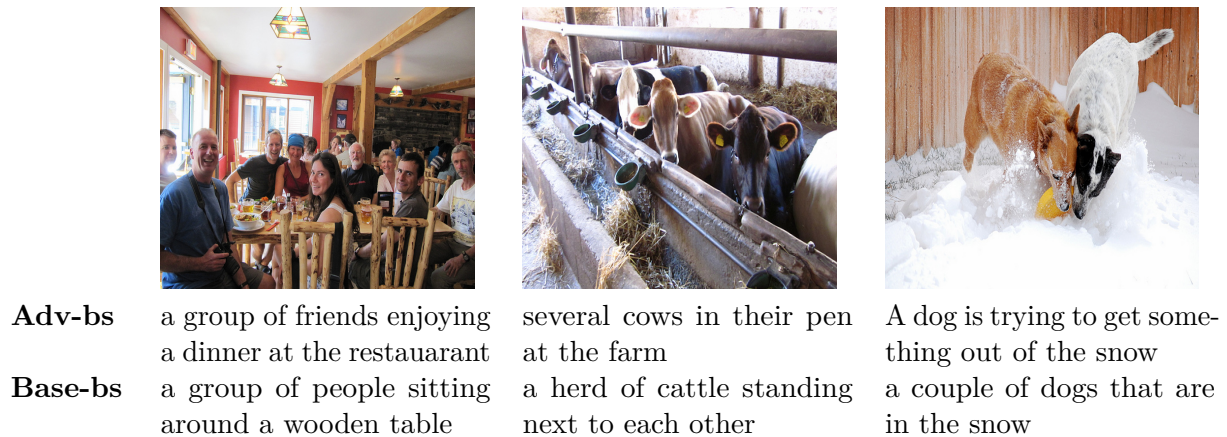


Figure 3.6: Qualitative comparison of captions generated by our model and the baseline model.

It consistently overuses more frequent n-grams (ratio $>1.0$ ) from the training set and under-uses less frequent ones (ratio $<1.0$ ). This trend is seen in all the three plots, with more frequent tri-grams particularly prone to overuse. It can also be observed in the histogram plots of the count ratios, that the baseline model does a poor job of matching the statistics of the test set.

Our adversarial model does a much better job in matching these statistics. The histogram of the uni-gram count ratios are clearly closer to that of test reference captions. It does not seem to be significantly overusing the popular words, but there is still a trend of under utilizing some of the rarer words. It is however clearly better than the baseline model in this aspect. The improvement is less pronounced with the bi- and tri-grams, but still present.

Another clear benefit from using the adversarial training is observed in terms of diversity in the captions produced by the model. The diversity in terms of both global statistics and per image diversity statistics is much higher in captions produced by the adversarial models compared to the baseline models. This result is presented in Table 3.4. We can see that the vocabulary size approximately doubles from 1085 in the baseline model to 2176 in the adversarial model using beamsearch. A similar trend is also seen comparing the sampling variants. As expected more diversity is achieved when sampling from the adversarial model instead of using beamsearch with vocabulary size increasing to 2671 in *Adv-samp*. The effect of this increased diversity can be in the qualitative examples shown in Figure 3.6.

We can also see that the adversarial model learns to construct significantly more novel sentences compared to the baseline model with *Adv-bs* producing novel captions 72.53% of the time compared to just 44.27% by the *beam-bs*. All three per-image diversity statistics also improve in the adversarial models indicating that they can produce a more diverse set of captions for any input image.

Table 3.4 also shows the diversity statistics on the reference captions on the test set. This shows that although adversarial models do considerably better than the baseline, there is still a gap in diversity statistics when compared to the human written captions, especially in vocabulary size.

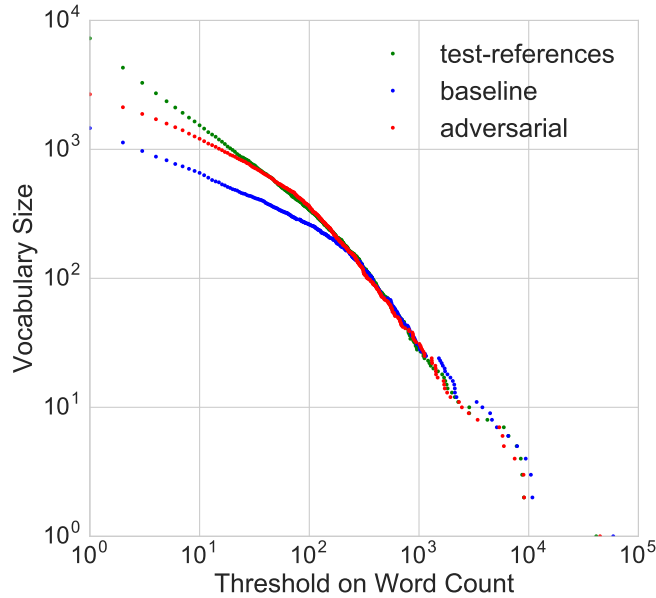


Figure 3.7: Vocabulary size as a function of word counts.

Finally, Figure 3.7 plots the vocabulary size as a function of word count threshold,  $k$ . We see that the curve for the adversarial model better matches the human written captions compared to the baseline for all values of  $k$ . This illustrates that the gains in vocabulary size in adversarial models does not arise from using words with specific frequency, but is instead distributed evenly across word frequencies.

#### 3.4.4 Ablation study

We conducted experiments to understand the importance of different components of our architecture. The results are presented in Table 3.5. The baseline model for this experiment uses VGG (Simonyan and Zisserman, 2015) features as  $x_p$  input and is trained using maximum likelihood loss and is shown in the first row of Table 3.5. The other four models use adversarial training.

Comparing rows 1 and 2 of Table 3.5, we see that adversarial training with a discriminator evaluating a single caption does badly. Both the diversity and Meteor score drop compared to the baseline. In this setting the generator can get away with producing one good caption (mode collapse) for an image as the discriminator is unable to penalize the lack of diversity in the generator.

However, comparing rows 1 and 3, we see that adversarial training using a discriminator evaluating 5 captions simultaneously does much better in terms of Div-2 and vocabulary size. Adding feature matching loss further improves the diversity and also slightly improves accuracy in terms of Meteor score. Thus simultaneously evaluating multiple captions and using feature matching loss allows us to alleviate mode collapse generally observed in GANs.

Upgrading to the *ResNet*(He *et al.*, 2016) increases the Meteor score greatly and

Image Feature	Evalset size (p)	Feature Matching	Meteor	Div-2	Vocab. Size
VGG (baseline)			0.247	0.44	1367
VGG (adversarial)	1	No	0.179	0.40	812
VGG (adversarial)	5	No	0.197	0.52	1810
VGG (adversarial)	5	yes	0.207	<b>0.59</b>	2547
ResNet (adversarial)	5	yes	<b>0.236</b>	0.55	<b>2671</b>

Table 3.5: Performance comparison of various configurations of the adversarial caption generator on the validation set.

slightly increases the vocabulary size. *ResNet* features provide richer visual information which is used by the generator to produce diverse but still correct captions.

We also notice that the generator learns to ignore the input noise. This is because there is sufficient stochasticity in the generation process due to sequential sampling of words and thus the generator doesn't need the additional noise input to increase output diversity. Similar observation was reported in other conditional GAN architectures (Isola *et al.*, 2016; Mathieu *et al.*, 2016)

### 3.5 CONCLUSIONS

We have presented an adversarial caption generator model which is explicitly trained to generate diverse captions for images. We achieve this by utilizing a discriminator network designed to promote diversity and use the adversarial learning framework to train our generator. Results show that our adversarial model produces captions which are diverse and match the statistics of human generated captions significantly better than the baseline model. The adversarial model also uses a larger vocabulary and is able to produce significantly more novel captions. The increased diversity is achieved while preserving accuracy of the generated captions, as shown through a human evaluation.



---

**Contents**

4.1	Introduction . . . . .	<b>46</b>
4.2	Threat Model . . . . .	<b>48</b>
4.3	Author Attribute Anonymization . . . . .	<b>48</b>
4.3.1	Author attribute classifiers . . . . .	49
4.3.2	The A <sup>4</sup> NT network . . . . .	51
4.3.3	Style loss with GAN . . . . .	53
4.3.4	Preserving semantics . . . . .	54
4.3.5	Smoothness with language loss . . . . .	56
4.4	Experimental Setup . . . . .	<b>56</b>
4.4.1	Datasets . . . . .	57
4.4.2	Evaluation methods . . . . .	58
4.4.3	Baselines . . . . .	59
4.5	Experimental Results . . . . .	<b>60</b>
4.5.1	Quantitative evaluation . . . . .	60
4.5.2	Qualitative analysis . . . . .	66
4.6	Conclusions . . . . .	<b>72</b>

---

IN this chapter, we extend the adversarial training for language models developed in Chapter 3, to build a privacy preserving tool to mask authors identity. Text-based analysis methods enable an adversary to reveal privacy relevant author attributes such as gender, age and can identify the text’s author. Such methods can compromise the privacy of an anonymous author even when the author tries to remove privacy sensitive content. In this chapter, we propose an automatic method, called Adversarial Author Attribute Anonymity Neural Translation (A<sup>4</sup>NT), to combat such text-based adversaries. Unlike prior works on obfuscation, we propose a system that is fully automatic and learns to perform obfuscation entirely from the data. This allows us to easily apply the A<sup>4</sup>NT system to obfuscate different author attributes. Our A<sup>4</sup>NT is a sequence-to-sequence language model and is trained in an adversarial framework, allowing it to learn obfuscating text transforms without paired data. We also propose and evaluate techniques to impose constraints on the A<sup>4</sup>NT model to preserve the semantics of the input text. A<sup>4</sup>NT learns to make minimal changes to the input to successfully fool author attribute classifiers, while preserving the meaning of the input text. We present experiments on two datasets and three settings which show that the proposed method is effective in fooling the attribute classifiers and thus improves the anonymity of authors.

## 4.1 INTRODUCTION

Natural language processing (NLP) methods including stylometric tools enable identification of authors of anonymous texts by analyzing stylistic properties of the text (Juola *et al.*, 2008; Stamatatos, 2009; Ruder *et al.*, 2016). NLP-based tools have also been applied to profiling users by determining their private attributes like age and gender (Argamon *et al.*, 2009). These methods have been shown to be effective in various settings like blogs, reddit comments, twitter text (Overdorf and Greenstadt, 2016) and in large scale settings with up to 100,000 possible authors (Narayanan *et al.*, 2012). In a recent famous case, authorship attribution tools were used to help confirm J.K Rowling as the real author of *A Cuckoo’s Calling* which was written by Ms. Rowling under pseudonymity (Juola, 2013). This case highlights the privacy risks posed by these tools.

Apart from the threat of identification of an anonymous author, the NLP-based tools also make authors susceptible to profiling. Text analysis has been shown to be effective in predicting age group (Morgan-Lopez *et al.*, 2017), gender (Ikeda *et al.*, 2013) and to an extent even political preferences (Makazhanov *et al.*, 2014). By determining such private attributes an adversary can build user profiles which have been used for manipulation through targeted advertising, both for commercial and political goals (Grassegger and Krogerus, 2017).

Since the NLP based profiling methods utilize the stylistic properties of the text to break the authors anonymity, they are immune to defense measures like pseudonymity, masking the IP addresses or obfuscating the posting patterns. The only way to combat them is to modify the content of the text to hide stylistic attributes. Prior work has shown that while people are capable of altering their writing styles to hide their identity (Brennan *et al.*, 2012), success rate depends on the authors skill and doing so consistently is hard for even skilled authors (Afroz *et al.*, 2012). Currently available solutions to obfuscate authorship and defend against NLP-methods has been largely restricted to semi-automatic solutions which suggest possible changes to the user (McDonald *et al.*, 2012) or hand-crafted transformations to text (Castro *et al.*, 2017) which need re-engineering on different datasets. This however limits the applicability of these defensive measures beyond the specific dataset it was designed on. To the best of our knowledge, text rephrasing using generic machine translation tools (Keswani *et al.*, 2016) is the only prior work offering a fully automatic solution to author obfuscation which can be applied across datasets. But as found in prior work Caliskan and Greenstadt (2012) and further demonstrated with our experiments, generic machine translation based obfuscation fails to sufficiently hide the identity and protect against attribute classifiers.

Additionally the focus in prior research has been towards protecting author identity. However, obfuscating identity does not guarantee protection of private attributes like age and gender. Determining attributes is generally easier than predicting the exact identity for NLP-based adversaries, mainly due to former being small closed-set prediction task compared to later which is larger and potentially open-set prediction task. This makes obfuscating attributes a difficult but an important problem.

**Our work.** We propose an unified automatic system (A<sup>4</sup>NT) to obfuscate authors



text and defend against NLP adversaries. A<sup>4</sup>NT follows the imitation model of defense discussed in Brennan *et al.* (2012) and protects against various attribute classifiers by learning to imitate the writing style of a target class. For example, A<sup>4</sup>NT learns to hide the gender of a female author by re-synthesizing the text in the style of the male class. This imitation of writing style is learned by adversarially training (Goodfellow *et al.*, 2014) our style-transfer network against the attribute classifier. Our A<sup>4</sup>NT network learns the target style by learning to fool the authorship classifiers into misclassifying the text it generates as target class. This style transfer is accomplished while aiming to retain the semantic content of the input text.

Unlike many prior works on authorship obfuscation (McDonald *et al.*, 2012; Castro *et al.*, 2017), we propose an end-to-end learnable author anonymization solution, allowing us to apply our method not only to authorship obfuscation but to the anonymization of different author attributes including identity, gender and age with a *unified approach*. We illustrate this by successfully applying our model on three different attribute anonymization settings on two different datasets. Through empirical evaluation, we show that the proposed approach is able to fool the author attribute classifiers in all three settings effectively and better than the baselines. While there are still challenges to overcome before applying the system to multiple attributes and situations with very little data, we believe that A<sup>4</sup>NT offers a new data driven approach to authorship obfuscation which can easily adapt to improving NLP-based adversaries.

**Technical challenges.** We design our A<sup>4</sup>NT network architecture based on the sequence-to-sequence neural machine translation model (Sutskever *et al.*, 2014). A key challenge in learning to perform style transfer, compared to other sequence-to-sequence mapping tasks like machine translation, is the lack of paired training data. Here, paired data refers to datasets with both the input text and its corresponding ground-truth output text. In obfuscation setting, this means having a large dataset with semantically same sentences written in different styles corresponding to the attributes we want to hide. Such paired data is infeasible to obtain and this has been a key hurdle in developing automatic obfuscation methods. Some prior attempts to perform text style transfer required paired training data (Xu *et al.*, 2012) and hence were limited in their applicability beyond toy-data settings. We overcome this by training our A<sup>4</sup>NT network within a generative adversarial networks (GAN) (Goodfellow *et al.*, 2014) framework. GAN framework enables us to train the A<sup>4</sup>NT network to generate samples that match the target distribution without need for paired data.

We characterize the performance of our A<sup>4</sup>NT network along two axes: privacy effectiveness and semantic similarity. Using automatic metrics and human evaluation to measure semantic similarity of the generated text to the input, we show that A<sup>4</sup>NT offers a better trade-off between privacy effectiveness and semantic similarity. We also analyze the effectiveness of A<sup>4</sup>NT for protecting anonymity for varying degrees of input text “difficulty”.

**Contributions.** In summary, our main contributions are. **(1):** We propose a novel approach to authorship obfuscation that uses a style-transfer network (A<sup>4</sup>NT) to automatically transform the input text to a target style and fool the attribute classifiers. The network is trained without paired data by adversarial training. **(2):** The proposed

obfuscation solution is end-to-end trainable, and hence can be applied to protect different author attributes and on different datasets with no changes to the overall framework. **(3):** Quantifying the performance of our system on privacy effectiveness and semantic similarity to input, we show that it offers a better trade-off between the two metrics compared to baselines.

## 4.2 THREAT MODEL

In our target scenario, our user is faced with an adversary who can access the text written by the user and the adversary wishes to determine the user’s private attributes for identification or for profiling. We assume that the author has taken care to remove obvious identifiable features from the text like name, zip code, IP address etc. The adversary has to rely on stylistic properties of the text for the analysis. To aid with this analysis, adversary can train NLP models on large amount of publicly available data, for example blog dataset (Schler *et al.*, 2006), twitter dataset (Morgan-Lopez *et al.*, 2017). In this scenario, the proposed A<sup>4</sup>NT system enables automatic obfuscation of user’s writing style to hide any desired private attribute like age group, gender or identity.

## 4.3 AUTHOR ATTRIBUTE ANONYMIZATION

We propose an author adversarial attribute anonymizing neural translation (A<sup>4</sup>NT) network to defend against NLP-based adversaries. The proposed solution includes the A<sup>4</sup>NT Network, the adversarial training scheme, and semantic and language losses to learn to protect private attributes. The A<sup>4</sup>NT network transforms the input text from a source attribute class to mimic the style of a different attribute class, and thus fools the attribute classifiers.

Technically, A<sup>4</sup>NT network is essentially solving a sequence to sequence mapping problem — from text sequence in the source domain to text in the target domain — similar to machine translation. Exploiting this similarity, we design our A<sup>4</sup>NT network based on the sequence-to-sequence neural language models (Sutskever *et al.*, 2014), widely used in neural machine translation (Bahdanau *et al.*, 2014). These models have proven effective when trained with large amounts of paired data and are also deployed commercially (Wu *et al.*, 2016b). If there were paired data in source and target attributes, we could train our A<sup>4</sup>NT network exactly like a machine translation model, with standard supervised learning. However, such paired data is infeasible to obtain as it would require the same text written in multiple styles.

To address the lack of paired data, we cast the anonymization task as learning a generative model,  $Z_{xy}(s_x)$ , which transforms an input text sample  $s_x$  drawn from source attribute distribution  $s_x \sim X$ , to look like samples from the target distribution  $s_y \sim Y$ . This formulation enables us to train the A<sup>4</sup>NT network  $Z_{xy}(s_x)$  with the GAN framework to produce samples close to the target distribution  $Y$ , using only unpaired samples from  $X$  and  $Y$ . Figure 4.1 shows this overall framework.

The GAN framework consists of two models, a generator producing synthetic samples

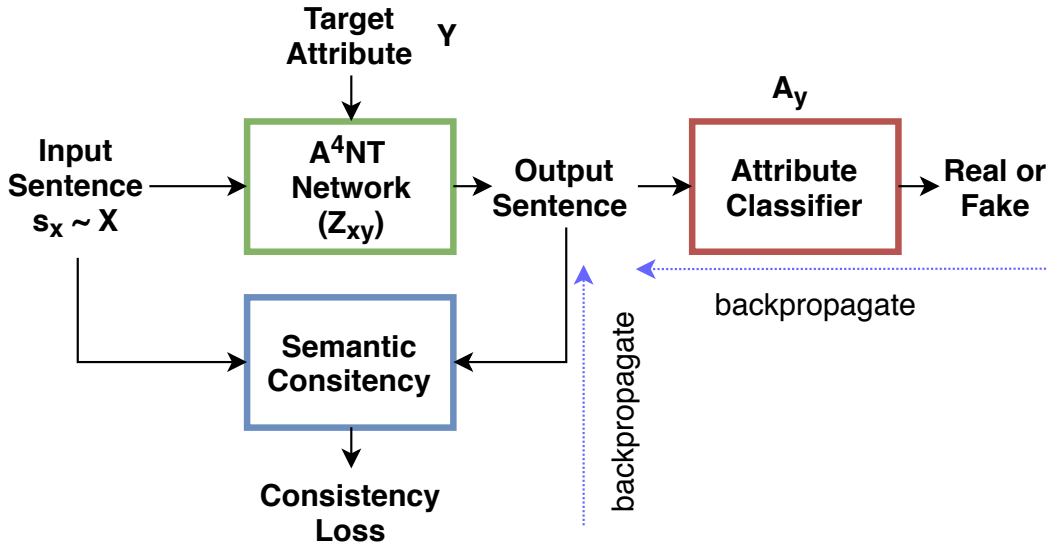


Figure 4.1: GAN framework to train our  $A^4NT$  network. Input sentence is transformed by  $A^4NT$  to match the style of the target attribute. This output is evaluated using the attribute classifier and semantic consistency loss.  $A^4NT$  is trained by backpropagating through these losses.

to mimic the target data distribution, and a discriminator which tries to distinguish real data from the synthesized “fake” samples from the generator. The two models are trained adversarially, i.e. the generator tries to fool the discriminator and the discriminator tries to correctly identify the generated samples. We use an attribute classifier as the discriminator and the  $A^4NT$  network as the generator. The  $A^4NT$  network, in trying to fool the attribute classification network, learns to transform the input text to mimic the style of the target attribute and protect the attribute anonymity.

For our  $A^4NT$  network to be a practically useful defensive measure, the text output by this network should be able to fool the attribute classifier while also preserving the meaning of the input sentence. If we could measure the semantic difference between the generated text and the input text it could be used to penalize deviations from the input sentence semantics. Computing this semantic distance perfectly would need true understanding of the meaning of input sentence, which is beyond the capabilities of current natural language processing techniques. To address this aspect of style transfer, we experiment with various proxies to measure and penalize changes to input semantics, which will be discussed in Section 4.3.4. Following subsections will describe each module in detail.

#### 4.3.1 Author attribute classifiers

We build our attribute classifiers using neural networks that predict the attribute label by directly operating on the text data. This is similar to recent approaches in authorship recognition (Bagnall, 2015; Ruder *et al.*, 2016) where, instead of hand-crafted features used in classical stylometry, neural networks are used to directly predict author

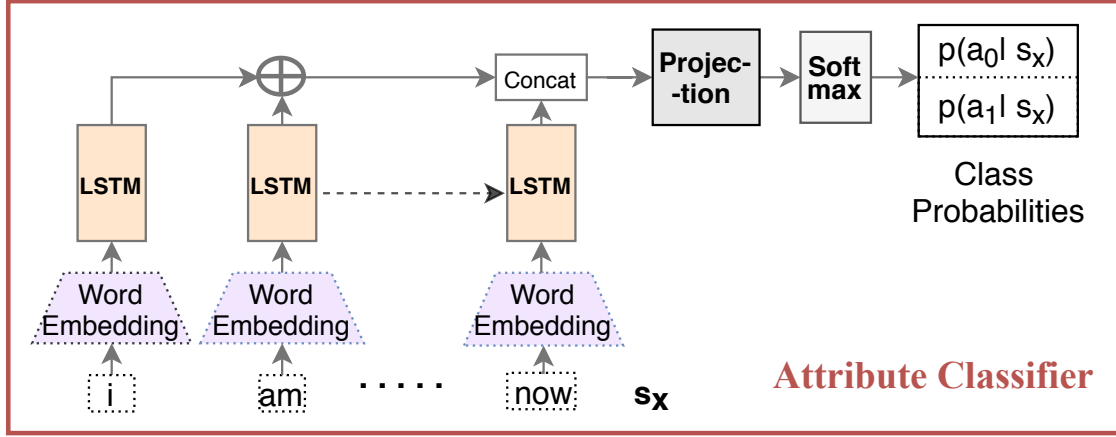


Figure 4.2: Block diagram of the attribute classifier network. The LSTM encoder embeds the input sentence into a vector. Sentence encoding is passed to linear projection followed by softmax layer to obtain class probabilities

identity from raw text data. However, unlike in these prior works, our focus is attribute classification and obfuscation. We train our classifiers with recurrent networks operating at word-level, as opposed to character-level models used in [Bagnall \(2015\)](#); [Ruder et al. \(2016\)](#) for two reasons. We found that the word-level models give good performance on all three attribute-classification tasks we experiment with (see Section 4.5.1). Additionally, they are much faster than character-level models, making it feasible to use them in GAN training described in Section 4.3.2.

Specifically, our attribute classifier  $A_x$  to detect attribute value  $x$  is shown in Figure 4.2. It consists of a Long-Short Term Memory (LSTM) ([Hochreiter and Schmidhuber, 1997](#)) encoder network to compute an embedding of the input sentence into a fixed size vector. It learns to encode the parts of the sentence most relevant to the classification task into the embedding vector, which for attribute prediction is mainly the stylistic properties of the text. This embedding is input to a linear layer and a softmax layer to output the class probabilities.

Given an input sentence  $s_x = \{w_0, w_1, \dots, w_{n-1}\}$ , the words are one-hot encoded and then embedded into fixed size vectors using the word-embedding layer shown in Figure 4.2 to obtain vectors  $\{v_0, v_1, \dots, v_{n-1}\}$ . The word embedding layer is simply a matrix of  $V \times d_{wv}$  containing the word vectors of  $d_{wv}$  dimensions for each word in the vocabulary of size  $V$ . This matrix is multiplied with the one-hot encoding of the word to obtain the representation of the corresponding word. The learned word vectors encode the similarities between words and can help deal with large vocabulary sizes. The word vectors are randomly initialized and then learned from the data during the training of the model. This approach works better than using pre-trained word vectors like word2vec ([Mikolov et al., 2013](#)) or Glove ([Pennington et al., 2014](#)) since the learned word-vectors can encode similarities most relevant to the attribute classification task at hand.

This sequence of word vectors is recursively passed through an LSTM to obtain a sequence of outputs  $\{h_0, h_1, \dots, h_{n-1}\}$ . We refer the reader to [Hochreiter and Schmid-](#)

huber (1997) for the exact computations performed to get the LSTM output.

Sentence embeddings are obtained by concatenating the final LSTM output and the mean of the LSTM outputs from other time-steps.

$$E(s_x) = \left[ h_{n-1}; \frac{1}{n-1} \sum h_{n-1} \right] \quad (4.1)$$

At the last time-step the LSTM network has seen all the words in the sentence and can encode a summary of the sentence in its output. However, using LSTM outputs from all time-steps, instead of just the final one, speeds up training due to improved flow of gradients through the network. Finally,  $E(s_x)$  is passed through linear and softmax layers to obtain class probabilities, for each class  $c_i$ . The network is then trained using cross-entropy loss.

$$p_{\text{auth}}(c_i | s_x) = \text{softmax}(W \cdot E(s_x)) \quad (4.2)$$

$$\text{Loss}(A_x) = \sum_i t_i(s_x) \log(p_{\text{auth}}(c_i | s_x)) \quad (4.3)$$

where  $t(s_x)$  is the one-hot encoding of the true class of  $s_x$ .

The same network architecture is applied for all our attribute prediction tasks including identity, age and gender.

#### 4.3.2 The A<sup>4</sup>NT network

A key design goal for the A<sup>4</sup>NT network is that it is trainable purely from data to obfuscate the author attributes. This is a significant departure from prior works on author obfuscation (McDonald *et al.*, 2012; Karadzhov *et al.*, 2017) that rely on hand-crafted rules for text modification to achieve obfuscation. The methods relying on hand-crafted rules are limited in applicability to specific datasets they were designed for.

To achieve this goal, we base our A<sup>4</sup>NT network  $Z_{xy}$ , shown in Figure 4.3, on a recurrent sequence-to-sequence neural translation model (Sutskever *et al.*, 2014) (*Seq2Seq*) popular in many sequence mapping tasks. As seen from the wide-range of applications mapping text-to-text (Bahdanau *et al.*, 2014), speech-to-text (Weiss *et al.*, 2017), text-to-part of speech (Ma and Hovy, 2016), the *Seq2Seq* models can effectively learn to map input sequences to arbitrary output sequences, with appropriate training. They operate on raw text data and alleviate the need for hand-crafted features or rules to transform the style of input text, predominantly used in prior works on author obfuscation (McDonald *et al.*, 2012; Karadzhov *et al.*, 2017). Instead, appropriate text transformations can be learnt directly from data. This flexibility allows us to easily apply the same A<sup>4</sup>NT network and training scheme to different datasets and settings.

The A<sup>4</sup>NT network  $Z_{xy}$  consists of two components, an encoder and a decoder modules, similar to standard sequence-to-sequence models. The encoder embeds the variable length input sentence into a fixed size vector space. The decoder maps the vectors in this embedding space to output text sequences in the target style. The encoder is an LSTM network, sharing the architecture of the sentence encoder in Section 4.3.1.

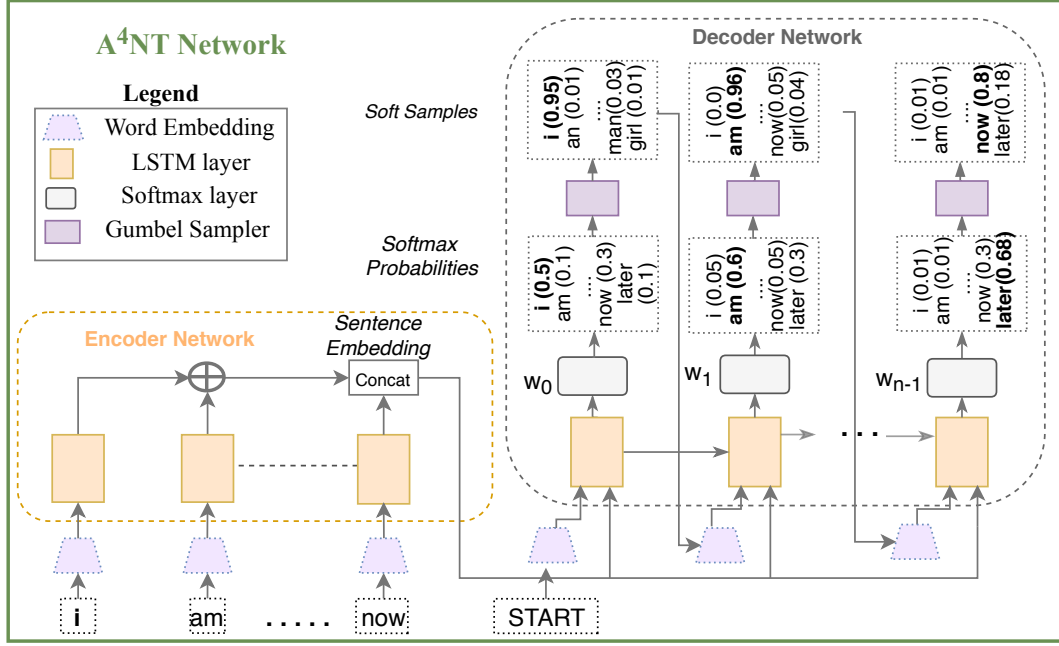


Figure 4.3: Block diagram of the  $A^4NT$  network. First LSTM encoder embeds the input sentence into a vector. The decoder maps this sentence encoding to the output sequence. Gumbel sampler produces “soft” samples from the softmax distribution to allow backpropagation.

The same architecture applies here as the task here is also to embed the input sentence  $s_x$  into a fixed size vector  $E_G(s_x)$ . However,  $E_G(s_x)$  should learn to represent the semantics of the input sentence allowing the decoder network to generate a sentence with similar meaning but in a different style.

The sentence embedding from the encoder is input to the decoder LSTM which generates the output sentence one word at a time. At each step  $t$ , the decoder LSTM takes  $E_G(s_x)$  and the previous output word  $w_{t-1}^o$  to produce a probability distribution over the vocabulary. Sampling from this distribution outputs the next word.

$$h_t^{\text{dec}}(s_x) = \text{LSTM}[E_G(s_x), W_{\text{emb}}(\tilde{w}_{t-1})] \quad (4.4)$$

$$p(\tilde{w}_t | s_x) = \text{softmax}_V(W_{\text{dec}} \cdot h_t^{\text{dec}}(s_x)) \quad (4.5)$$

$$\tilde{w}_t = \text{sample}(p(\tilde{w}_t | s_x)) \quad (4.6)$$

where  $W_{\text{emb}}$  is the word embedding,  $W_{\text{dec}}$  matrix maps the LSTM output to vocabulary size and  $V$  is the vocabulary.

In most applications of *Seq2Seq* models, the networks are trained using parallel training data, consisting of input and ground-truth output sentence pairs. A sentence is input to the encoder and propagated through the network and the network is trained to maximize the likelihood of generating the paired ground-truth output sentence. However, in our setting, we do not have access to such parallel training data of text in different styles and the  $A^4NT$  network  $Z_{xy}$  is trained in an unsupervised setting.

We address the lack of parallel training data by using the GAN framework to train



the A<sup>4</sup>NT network. In this framework, the A<sup>4</sup>NT network  $Z_{xy}$  learns by generating text samples and improving itself iteratively to produce text that the attribute classifier,  $A_y$ , classifies as target attribute. A benefit of GANs is that the A<sup>4</sup>NT network is directly optimized to fool the attribute classifiers. It can hence learn to make transformations to the parts of the text which are most revealing of the attribute at hand, and so hide the attribute with minimal changes.

However, to apply the GAN framework, we need to differentiate through the samples generated by  $Z_{xy}$ . The word samples from  $p(\tilde{w}_t|s_x)$  are discrete tokens and are not differentiable. Following Shetty *et al.* (2017), we apply the Gumbel-Softmax approximation (Jang *et al.*, 2016) to obtain differentiable soft samples and enable end-to-end GAN training.

**Splitting decoder.** To transfer styles between attribute pairs,  $x$  and  $y$ , in both directions, we found it ineffective to use the same network  $Z_{xy}$ . A single network  $Z_{xy}$  is unable to sufficiently switch its output word distributions solely on a binary condition of target attribute. Nonetheless, using a separate network for each ordered pair of attributes is prohibitively expensive. A good compromise we found is to share the encoder to embed the input sentence but use different decoders for style transfer between each ordered pair of attributes. Sharing the encoder allows the two networks to share a significant number of parameters and enables the attribute specific decoders to deal with the words found only in the vocabulary of the other attribute group using shared sentence and word embeddings.

### 4.3.3 Style loss with GAN

We train the two A<sup>4</sup>NT networks  $Z_{xy}$  and  $Z_{yx}$  in the GAN framework to produce samples which are indistinguishable from samples from distributions of attributes  $y$  and  $x$  respectively, without having paired sentences from  $x$  and  $y$ . Figure 4.4 shows this training framework.

Given a sentence  $s_x$  written by author with attribute  $x$ , the A<sup>4</sup>NT network outputs a sentence  $\tilde{s}_y = Z_{xy}(s_x)$ . This is passed to the attribute classifier for attribute  $y$ ,  $A_y$ , to obtain probability  $p_{\text{auth}}(y|\tilde{s}_y)$ .  $Z_{xy}$  tries to fool the classifier  $A_y$  into assigning high probability to its output, whereas  $A_y$  tries to assign low probability to sentences produced by  $Z_{xy}$  while assigning high probability to real sentences  $s_y$  written by  $y$ . The same process is followed to train the A<sup>4</sup>NT network from  $y$  to  $x$ , with  $x$  and  $y$  swapped. The loss functions used to train the A<sup>4</sup>NT network and the attribute classifiers in this setting is given by:

$$L(A_y) = -\log(p_{\text{auth}}(y|s_y)) - \log(1 - p_{\text{auth}}(y|\tilde{s}_y)) \quad (4.7)$$

$$L_{\text{style}}(Z_{xy}) = -\log(p_{\text{auth}}(y|\tilde{s}_y)) \quad (4.8)$$

The two networks  $Z_{xy}$  and  $A_y$  are adversarially competing with each other when minimizing the above loss functions. At optimum it is guaranteed that the distribution of samples produced by  $Z_{xy}$  is identical to the distribution of  $y$  (Goodfellow *et al.*, 2014).

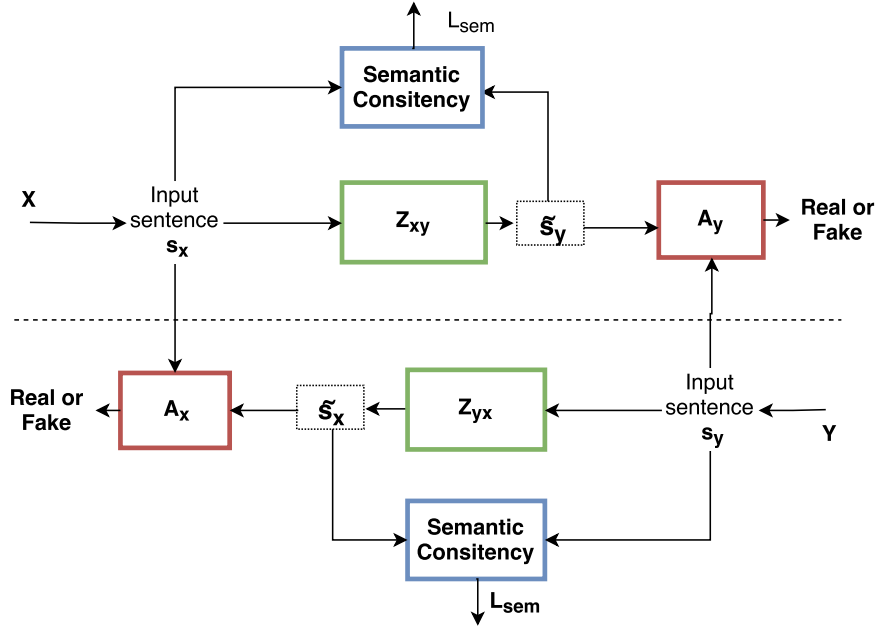


Figure 4.4: Illustrating use of GAN framework and cyclic semantic loss to train a pair of  $A^4NT$  networks.

However, we want the  $A^4NT$  network to only imitate the style of  $y$ , while keeping the content from  $x$ . Thus, we explore methods to enforce the semantic consistency between the the input sentence and the  $A^4NT$  output.

#### 4.3.4 Preserving semantics

We want the output sentence,  $\tilde{s}_y$ , produced by  $Z_{xy}(s_x)$  to not only fool the attribute classifier, but also to preserve the meaning of the input sentence  $s_x$ . We propose a semantic loss  $L_{sem}(\tilde{s}_y, s_x)$  to quantify the meaning changed during the anonymization by  $A^4NT$ . Simple approaches like matching words in  $\tilde{s}_y$  and  $s_x$  can severely limit the effectiveness of anonymization, as it penalizes even synonyms or alternate phrasing. In the following subsection we will discuss two approaches to define  $L_{sem}$ , and later in Section 4.5 we compare these approaches quantitatively.

##### 4.3.4.1 Cycle constraints

One could evaluate how semantically close is  $\tilde{s}_y$  to  $s_x$  by evaluating how easy it is to reconstruct  $s_x$  from  $\tilde{s}_y$ . If  $\tilde{s}_y$  means exactly the same as  $s_x$ , there should be no information loss and we should be able to perfectly reconstruct  $s_x$  from  $\tilde{s}_y$ . We could use the  $A^4NT$  network in the reverse direction to obtain a reconstruction,  $\hat{s}_x = Z_{yx}(\tilde{s}_y)$  and compare it to input sentence  $s_x$ . Such an approach, referred to as cycle constraint, has been used in image style transfer (Zhu *et al.*, 2017), where  $l_1$  distance is used to compare the reconstructed image and the original image to impose semantic relatedness penalty. However, in our case  $l_1$  distance is not meaningful to compare  $\hat{s}_x$  and  $s_x$ , as



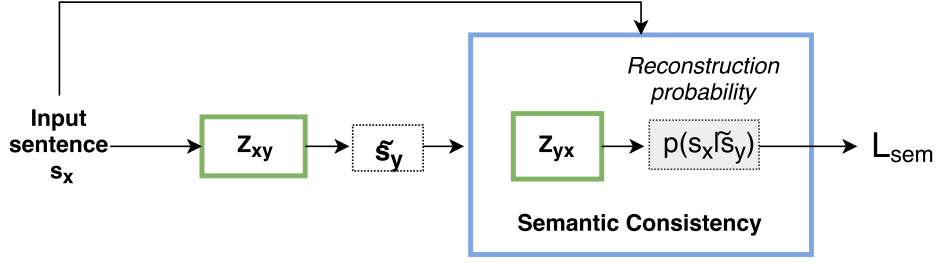


Figure 4.5: Semantic consistency in  $A^4NT$  networks is enforced by maximizing cyclic reconstruction probability.

they are sequences of possibly different lengths. Even a single word insertion or deletion in  $\tilde{s}_x$  can cause the entire sequence to mismatch and be penalized by the  $l_1$  distance.

A simpler and more stable alternative we use is to forgo the reconstruction and just computing the likelihood of reconstruction of  $s_x$  when applying reverse style-transfer on  $\tilde{s}_y$ . This likelihood is simple to obtain from the reverse  $A^4NT$  network  $Z_{yx}$  using the word distribution probabilities at the output. This cyclic loss computation is illustrated in Figure 4.5. Duly, we compute reconstruction probability  $P_r(s_x|\tilde{s}_y)$  and define the semantic loss as:

$$P_r(s_x|\tilde{s}_y) = \prod_{t=0}^{n-1} p_{z_{yx}}(w_t|\tilde{s}_y) \quad (4.9)$$

$$L_{sem}(\tilde{s}_y, s_x) = -\log P_r(s_x|\tilde{s}_y) \quad (4.10)$$

The lower the semantic loss  $L_{sem}$ , the higher the reconstruction probability and thus more meaning of the input sentence  $s_x$  is preserved in the style-transfer output  $\tilde{s}_y$ .

#### 4.3.4.2 Semantic embedding loss

An alternative approach to measuring the semantic loss is to embed the two sentences,  $\tilde{s}_y$  and  $s_x$ , into a semantic space and compare the two embedding vectors using  $l_1$  distance. The idea is that a semantic embedding method puts similar meaning sentences close to each other in this vector space. This approach is used in many natural language processing tasks, for example in semantic entailment (Conneau *et al.*, 2017)

Since we do not have annotations of semantic relatedness on our datasets, it is not possible to train a semantic embedding model but instead we have to rely on pre-trained models known to have good transfer learning performance. Several such semantic sentence embeddings are available in the literature (Kiros *et al.*, 2015; Conneau *et al.*, 2017). We use the universal sentence embedding model from Conneau *et al.* (2017), pre-trained on the Stanford natural language inference dataset (Bowman *et al.*, 2015).

We embed the two sentences using this semantic embedding model  $F$  and use the  $l_1$  distance to compare the two embeddings and define the semantic loss as:

$$L_{sem}(\tilde{s}_y, s_x) = \sum_{dim} |F(s_x) - F(\tilde{s}_y)| \quad (4.11)$$

### 4.3.5 Smoothness with language loss

The A<sup>4</sup>NT network can minimize the style and the semantic losses, while still producing text which is broken and grammatically incorrect. To minimize the style loss the A<sup>4</sup>NT network needs to add words typical of the target attribute style. While minimizing the semantic loss, it needs to retain the semantically relevant words from the input text. However neither of these two losses explicitly enforces correct grammar and word order of  $\tilde{s}$ .

On the other hand, unconditional neural language models are good at producing grammatically correct text. The likelihood of the sentence produced by our A<sup>4</sup>NT model  $\tilde{s}$  under an unconditional language model,  $M_y$ , trained on the text by target attribute authors  $y$ , is a good indicator of the grammatical correctness of  $\tilde{s}$ . The higher the likelihood, the more likely the generated text  $\tilde{s}$  has syntactic properties seen in the real data. Therefore, we add an additional language smoothness loss on  $\tilde{s}$  in order to enforce  $Z$  to produce syntactically correct text.

$$L_{\text{lang}}(\tilde{s}) = -\log M_y(\tilde{s}) \quad (4.12)$$

**Overall loss function.** The A<sup>4</sup>NT network is trained with a weighted combination of the three losses: style loss, semantic consistency and language smoothing loss.

$$L_{\text{tot}}(Z_{xy}) = w_{\text{sty}}L_{\text{style}} + w_{\text{sem}}L_{\text{sem}} + w_lL_{\text{lang}} \quad (4.13)$$

We chose the above three weights so that the magnitude of the weighted loss terms are approximately equal at the beginning of training. Model training was not sensitive to exact values of the weights chosen that way.

**Implementation details.** We implement our model using the PyTorch framework (Paszke *et al.*, 2019). The networks are trained by optimizing the loss functions described with stochastic gradient descent using the RMSprop algorithm (Tieleman and Hinton, 2012). The A<sup>4</sup>NT network is pre-trained as an autoencoder, i.e to reconstruct the input sentence, before being trained with the loss function described in (4.13). During the GAN training, the A<sup>4</sup>NT network and the attribute classifiers are trained for one minibatch each alternatively. We will open source our code, models and data at the time of publication.

## 4.4 EXPERIMENTAL SETUP

We test our A<sup>4</sup>NT network on obfuscation of three different attributes of authors on two different datasets. The three attributes we experiment with include author’s age (under 20 vs over 20), gender (male vs female authors), and author identities (setting with two authors).

#### 4.4.1 Datasets

We use two real world datasets for our experiments: Blog Authorship corpus (Schler *et al.*, 2006) and Political Speech dataset. The datasets are from very different sources with distinct language styles, the first being from mini blogs written by several anonymous authors, and the second from political speeches of two US presidents Barack Obama and Donald Trump. This allows us to show that our approach works well across very different language corpora.

**Blog dataset.** The blog dataset is a large collection of micro blogs from blogger.com collected by Schler *et al.* (2006). The dataset consists of 19,320 “documents” along with annotation of author’s age, gender, occupation and star-sign. Each document is a collection of all posts by a single author. We utilize this dataset in two different settings; split by gender (referred to as blog-gender setting) and split by age annotation (blog-age setting). In the blog-age setting, we group the age annotations into two groups, teenagers (age between 13-18) and adults (age between 23-45) to obtain data with binary age labels. Age-groups 19-22 are missing in the original dataset. Since the dataset consists of free form text written while blogging with no proper sentence boundaries markers, we use the Stanford CoreNLP tool to segment the documents into sentences. All numbers are replaced with the NUM token. For training and evaluation, the whole dataset is split into training set of 13,636 documents, validation set of 2,799 documents and test set of 2,885 documents.

**Political speech dataset.** To test the limits of how far style imitation based anonymization can help protect author identity, we also test our model on two well known political figures with very different verbal styles. We collected the transcriptions of political speeches of Barack Obama and Donald Trump made available by the The American Presidency Project (Woolley and Peters, 1999). While the two authors talk about similar topics they have highly distinctive styles and vocabularies, making it a challenging dataset for our A<sup>4</sup>NT network. The dataset consists of 372 speeches, with about 65,000 sentences in total as shown in Table 4.1. We treat each speech as a separate document when evaluating the classification results on the document-level. This dataset contains a significant amount of references to named entities like people, organizations, etc. To avoid that both attribute classifiers and the style transfer model rely on these references to specific people, we use the Stanford Named Entity Recognizer tool (Finkel *et al.*, 2005) to identify and replace these entities with entity labels. The dataset is split into training set of 250 speeches, validation set of 49 speeches and test set of 73 speeches.

The comparison of the two datasets can be found in Table 4.1. The blog dataset is much larger and therefore we run most of our evaluation on it. Using these two datasets, we evaluate our model in three different attribute obfuscation settings, namely age (blog-age), gender (blog-gender) and identity obfuscation (speech dataset). Detailed analysis of our model presented in Section 4.5.2 is done on the validation split of the blog dataset, in the blog-age setting, containing 2,799 documents and 518,268 sentences.

Dataset	Attributes	# Documents	# Sentences	# Vocabulary
Speech	Identity	372	65k	5.6k
Blog	Age, Gender	19320	3.38 Mil	22k

Table 4.1: Comparing statistics of the two datasets.

#### 4.4.2 Evaluation methods

We evaluate our models with according to two different aspects: **(a)** *Privacy effectiveness*, to measure how well the A<sup>4</sup>NT network can fool the attribute classifiers and **(b)** *Semantic similarity*, quantifying how much of the input sentence semantics is preserved after style transfer.

To quantify privacy effectiveness we compare the F1-score of the attribute classifiers on the original data and the generated text. To mimic a real-world setting where we will not have access to the parameters of the attribute-classifier used by the adversary to train our A<sup>4</sup>NT network, we test it against attribute classifiers which have been separately trained on real data.

The F1-score for the attribute classifiers is defined as:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (4.14)$$

With successful style transfer, the attribute classifier will misclassify generated text and the F1-scores will drop. We consider the obfuscation to be successful if the F1-score drops below that of a random guessing classifier. Note that since the datasets are not perfectly balanced with respect to the binary attributes, the F1-score for random classifier is not 0.5. The expected document-level F1-scores of random guessing are 0.54 for the blog-age setting, 0.5 for the blog-gender setting, and 0.61 for the speech data.

For achieving obfuscation, having the attribute classifier around random chance-level is more desirable than pushing the classifier to 0% accuracy. Zero accuracy means the classifier has perfectly misclassified every sample. In a binary classification setting like age-group or gender, if the adversary detects there has been obfuscation he can fully recover the original classes by just flipping the decisions of the classifier. However if the obfuscation is to the random chance-level, even upon detection, the best the adversary can do is to get the random chance accuracy. This also holds true for multi-class situation, as can be seen from the information theoretic point of view. To achieve perfect obfuscation, we want the attribute classifier output to contain minimum information about the true class of the input text. When the classifier accuracy of the  $k$ -class attribute classifier is at the random chance-level, it is guessing the class labels with uniform probability  $p(y|c) \sim \text{Uniform}(1, 2, \dots, k)$ . In this case the mutual information between the classifier predicted label  $y$  and true label  $c$  is zero, since the  $p(y|c) = p(y)$ . However, the prediction of classifier  $p(y|c)$  at 0% accuracy is not independent of the input class-label since it cannot take the correct class value  $c$ , i.e  $p(y|c) \sim \text{Uniform}(1, 2, \dots, c-1, c+1, \dots, k)$ . This leads to non-zero mutual information between  $y$  and  $c$ . Hence, we use the random chance-level as our success criteria for obfuscation instead of targeting 0% classifier accuracy.

To quantify semantic similarity, we use the meteor metric (Denkowski and Lavie, 2014b). It is used in machine translation and image captioning to evaluate the similarity between the candidate text and a reference text. Meteor compares the candidate text to one or more references by matching n-grams, while allowing for soft matches using synonym and paraphrase tables. Meteor score lies between zero and one with zero indicating no similarity and one indicating identical sentences. For a point of reference, the state-of-the-art methods for paraphrase generation task achieve meteor scores between 0.35-0.4 (Li *et al.*, 2018) and for multimodal machine translation task achieve meteor score in the range 0.5-0.55 (Elliott *et al.*, 2017). We use the meteor score between the generated and input text as the measure of semantic similarity.

However, the automatic evaluation for semantic similarity is not perfectly correlated with human judgments, especially with few reference sentences. To address this, we additionally conduct two user studies on a subset of the test data of 745 sentences, first to compare the semantic similarity between different obfuscation methods relatively, and second to measure the semantic similarity between the model output and input text on an absolute scale. We ask human annotators on Amazon Mechanical Turk (AMT) to judge the semantic similarity of the generated text from our models. No other information was collected from the annotators, thereby keeping them anonymous. The annotators were compensated for their work through the AMT system. We manually screened the text shown to the annotators to make sure it contained no obvious offensive content.

#### 4.4.3 Baselines

We use the two baseline methods below to compare our model with. Both chosen baselines are automatic obfuscation methods not relying on hand-crafted rules.

**Autoencoder.** We train our A<sup>4</sup>NT network  $Z$  as an autoencoder, where it takes as input  $s_x$  and tries to reproduce it from the encoding. The autoencoder is trained similar to a standard neural language model with cross entropy loss. We train two such auto-encoders  $Z_{xx}$  and  $Z_{yy}$  for the two attributes. Now simple style transfer can be achieved from  $x$  to  $y$  by feeding the sentence  $s_x$  to the autoencoder of the other attribute class  $Z_{yy}$ . Since  $Z_{yy}$  is trained to output text in the  $y$  domain, the sentence  $Z_{yy}(s_x)$  tends to look similar to sentences in  $y$ . This model sets the baseline for style transfer that can be achieved without cross domain training using GANs, with the same network architecture and the same number of parameters.

**Google machine translation.** A simple and accessible approach to change writing style of a piece of text without hand designed rules is to use generic machine translation software. The input text is translated from a source language to multiple intermediate languages and finally translating back to the source language. The hope is that through this round-trip the style of the text has changed, with the meaning preserved. This approach was used in the PAN authorship obfuscation challenge recently (Keswani *et al.*, 2016).

We use the Google machine translation service<sup>3</sup> to perform the round-trip translation of our input sentences. We have tried a varying number of intermediate languages, results of which will be discussed in Section 4.5. Since Google limits the api calls and imposes character limits on manual translation, we use this baseline only on the subset of 745 sentences from the test set for human evaluation.

## 4.5 EXPERIMENTAL RESULTS

We test our model on the three settings discussed in Section 4.4 with the goal to understand if the proposed A<sup>4</sup>NT network can fool the attribute classifiers to protect the anonymity of the author attributes. Through quantitative evaluation done in Section 4.5.1, we show that this is indeed the case: our A<sup>4</sup>NT network learns to fool the attribute classifiers across all three settings. We compare the two semantic loss functions presented in Section 4.3.4 and show that the proposed reconstruction likelihood loss does better than pre-trained semantic encoding.

However, this privacy gain comes with a trade-off. The semantics of the input text is sometimes altered. In Section 4.5.2, using qualitative examples, we analyze the failure modes of our system and identify limits up to which style-transfer can help preserve anonymity.

We use three variants of our model in the following study. The first model uses the semantic encoding loss described in Section 4.3.4.2 and is referred to as *FBsem*. The second uses the reconstruction likelihood loss discussed in Section 4.3.4.1 instead, and is denoted by *CycML*. Finally, *CycML+Lang* uses both cyclic maximum likelihood and the language smoothing loss described in Section 4.3.5.

### 4.5.1 Quantitative evaluation

Before analyzing the performance of our A<sup>4</sup>NT network, we evaluate the attribute classifiers on the three settings we use. For this, we train the attribute classifier model in Section 4.3.1 on all three settings. Table 4.2 shows the F1-scores of the attribute classifiers on the training and the validation splits of the blog and the speech datasets. Document-level scores are obtained from accumulating the class log-probability scores on each sentence in a document before picking the maximum scoring class as the output label. We also tried hard voting to accumulate sentence level decisions, and observed that the hard voting results follow the same trend across datasets and splits.

On the smaller political speech dataset, the attribute classifier is able to easily discriminate between the two authors, Barack Obama and Donald Trump, achieving perfect F1-score of 1.0 on both the training and the validation splits. The model also performs well on the age-group classification, achieving F1-score of 0.88 on the validation set at the document-level. Gender classification turns out to be the hardest to generalize, with a significant drop in F1-score on the validation set compared to the training set (down to 0.75 from 0.93). However, we note that our gender classifier achieves similar

---

<sup>3</sup><https://translate.google.com/>

Setting	Training Set		Validation Set	
	Sentence	Document	Sentence	Document
Speechdata	0.84	1.00	0.68	1.00
Blog-age	0.76	0.92	0.74	0.88
Blog-gender	0.64	0.93	0.52	0.75

Table 4.2: F1-scores of the attribute classifiers. All of them do well and better than the document-level random chance (0.62 for speech), (0.53 for age), and (0.50 for gender).

Model	Blog-age data			Blog-gender data			Speech dataset		
	Sent	F1	Doc F1 Meteor	Sent	F1	Doc F1 Meteor	Sent	F1	Doc F1 Meteor
Random classifier	0.54	0.54	-	0.53	0.5	-	0.60	0.61	-
Original text	0.74	0.88	1.0	0.52	0.75	1.0	0.68	1.00	1.0
Autoencoder	0.69	0.85	0.77	0.49	0.61	0.78	<i>0.26</i>	<i>0.12</i>	0.32
A <sup>4</sup> NT FBsem	<i>0.43</i>	<i>0.08</i>	0.40	<i>0.45</i>	<i>0.39</i>	0.53	<i>0.12</i>	<i>0.00</i>	0.28
A <sup>4</sup> NT CycML	<i>0.49</i>	<i>0.20</i>	0.57	<i>0.41</i>	<i>0.44</i>	<b>0.79</b>	<i>0.11</i>	<i>0.00</i>	<b>0.29</b>
A <sup>4</sup> NT CycML+Lang	<i>0.54</i>	<i>0.32</i>	<b>0.69</b>	<i>0.44</i>	<i>0.39</i>	<b>0.79</b>	<i>0.12</i>	<i>0.00</i>	<b>0.29</b>

Table 4.3: Performance of the style transfer anonymization in fooling the classifiers, across the three settings. F1 (lower is better) and Meteor (higher is better). F1-scores below chance levels are shown in italics.

document-level classification accuracies to the traditional stylometric methods (76.1% accuracy reported in Argamon *et al.* (2009), compared to 74.1% accuracy achieved by our model). In all three tasks, the performance on sentence-level is worse than on document-level classification. This is especially true in the gender classification task, where the sentence-level classification does only marginally better than the random chance (0.52 F1-score). This is expected since the stylistic differences between authors of different genders is not visible in strong in all the sentences. However there is usually a few sentences within a document where these stylistic differences are strong. Thus when we aggregate the classifier scores over a document, the classifier does much better and is well above the random chance (0.75 F1-score vs 0.50 chance-level). Document-level classification also generalizes better with less difference between training and validation set F1-scores in Table 4.2. Henceforth, we will use document-level F1-score as our primary metric when evaluating the effectiveness of A<sup>4</sup>NT networks.

#### 4.5.1.1 Quantifying privacy gains from A<sup>4</sup>NT

We evaluate the performance of our A<sup>4</sup>NT network using the attribute classifiers presented in the previous section. To obfuscate the authorship of a document, the A<sup>4</sup>NT network is run on every sentence in a document to transfer style from the true attribute to the target one. For example, to mask age of a teenager written document we use the A<sup>4</sup>NT network  $Z_{Teen-Adult}$ . Style-transferred document is input to the



attribute classifier of the source attribute and F1-score of this classifier is computed. This measures the privacy effectiveness of the style transfer. Meteor score is computed between the source sentence and the A<sup>4</sup>NT output, to measure the semantic similarity.

Table 4.3 shows these results in the three settings. On the small speech dataset all methods, including the autoencoder baseline described in Section 4.4.3, successfully fool the attribute classifier. They all obtain F1-scores below the chance-level, with our A<sup>4</sup>NT networks doing better. However the meteor scores of all models is significantly lower than in the blog dataset, indicating significant amount of semantic loss in the process of anonymization.

On the larger blog dataset, the autoencoder baseline fails to fool the attribute classifier, with only a small drop in F1-score of 0.03 (from 0.88 to 0.85) in case of age and 0.14 in case of gender (from 0.75 to 0.61). Our A<sup>4</sup>NT models however do much better, with all of them being able to drop the F1-score below the random chance.

The *FBsem* model using semantic encoder loss achieves the largest privacy gain, by decreasing the F1-scores from 0.88 to 0.08 in case of age and from 0.75 to 0.39 in case of gender. This model however suffers from poor meteor scores, indicating the sentences produced after the style transfer are no longer similar to the input.

The model using reconstruction likelihood to enforce semantic consistency, *CycML*, fares much better in meteor metric in both age and gender style transfer. It is still able to fool the classifier, albeit with smaller drops in F1-scores (still below random chance). Finally, with addition of the language smoothing loss (*CycML+Lang*), we see a further improvement in the meteor score in the blog-age setting, while the performance remains similar to *CycML* on blog-gender setting and the speech dataset. However, the language smoothing model *CycML+Lang* fares better in the user study discussed in Section 4.5.1.2 and also produces better qualitative samples as will be seen in Section 4.5.2.

**Generalization to other classifiers.** An important question to answer if A<sup>4</sup>NT is to be applied to protect the privacy of author attributes, is how well it performs against unseen NLP based adversaries? To test this we trained ten different attribute classifiers networks on the blog-age setting. These networks vary in architectures (LSTM, CNN and LSTM+CNN) and hyper-parameters (number of layers and number of units), but all of them achieve good performance in predicting the age attribute. The networks were chosen to reflect real-world architecture choices used for text classification. Results from evaluating the text generated by the A<sup>4</sup>NT networks using these “holdout” classifiers are shown in Table 4.4. The column “mean” shows the mean performance of the ten classifiers and “max” shows the score of best performing classifier

Holdout classifiers have good performance on the original text, achieving mean 0.85 document-level F1-score. Table 4.4 shows that all three A<sup>4</sup>NT networks generalize well and are able to drop the document F1-score of the holdout classifiers to the random chance level (0.54 for the blog-age setting). They perform slightly worse than on the seen LSTM classifier, but are able to significantly drop the performance of all the holdout classifiers (mean F1 score drops from 0.85 to 0.53 or below). This is a strong empirical evidence that the transformations applied by the A<sup>4</sup>NT networks are not specific to the classifier they are trained with, but can also generalize to other adversaries.

We conclude that the proposed A<sup>4</sup>NT networks are able to fool the attribute classifiers

on all three tested tasks and also show generalization ability to fool classifier architectures not seen during training.

Note that here we have considered dropping the classifier score below the random chance-level on average as the criteria for fooling the classifiers and protecting privacy as motivated in Section 4.4.2. However, in some scenarios it is desirable for the obfuscation model to fool the classifiers all the time and pushing adversary to achieving 0% classification accuracy. This guarantee would provide the user the freedom to use the obfuscation only when desired. Our A<sup>4</sup>NT model is not able to reach this level of 100% obfuscation. It remains an important open problem for the future work to achieve 100% obfuscation while preserving semantics, so that the user remains in full control.

**Comparison to prior work.** We also compare the performance of our model to a prior work on automatic anonymization (Karadzhov *et al.*, 2017). This work proposes to anonymize the writing style by computing average statistics on certain text features and applying pre-defined transformations to change the input text statistics towards the average. We refer to this model as *C-Avg* and test this model on our age obfuscation task using the official code from Karadzhov *et al.* (2017). The results are shown in Table 4.5. For fair comparison we compare the performance against a holdout classifier, not seen by our model as well. We can see that the *C-Avg* model (Karadzhov *et al.*, 2017) does not perform well on the age obfuscation task, managing to drop the F1-score only to 0.77 from 0.84, which is well above the random chance-level. Our A<sup>4</sup>NT model however drops the F1-score below the chance-level to 0.44. Our model does better in semantic similarity as well, achieving meteor score of 0.69 compared to 0.55 obtained by *C-Avg*. The poor performance of *C-Avg* model (Karadzhov *et al.*, 2017) on the age obfuscation task is due to the fact that *C-Avg* relies on hand designed transformations (e.g. substituting synonyms from a dictionary) which does not generalize well to the diverse writing styles found in the blog dataset. This highlights the advantage of the proposed approach to learn to perform obfuscation directly from the data.

**Different operating points.** Our A<sup>4</sup>NT model offers the ability to obtain multiple different style-transfer outputs by simply sampling from the models distribution. This is useful as different text samples might have different levels of semantic similarity and privacy effectiveness. Having multiple samples allows users to choose the level of semantic similarity vs privacy trade-off they prefer.

We illustrate this in Figure 4.6. Here five samples are obtained from each A<sup>4</sup>NT model for each sentence in the test set. By choosing the sentence with minimum, maximum or random meteor scores w.r.t the input text, we can obtain a trade-off between semantic similarity and privacy. We see that while the *FBsem* model offers limited variability, *CycML+LangLoss* offers a wide range of choices of operating points. All operating points of *CycML+LangLoss* achieve better meteor score than 0.5, which indicates this model preserves the semantic similarity well.

#### 4.5.1.2 Human judgments for semantic consistency

In machine translation and image captioning literature, it is well known that automatic semantic similarity evaluation metrics like meteor are only reliable to a certain extent.

Model	Seen Classifier F1-score	Holdout Classifiers	
		Mean F1	Max F1
Original text	0.88	0.85	0.87
Autoencoder	0.85	0.83	0.84
A <sup>4</sup> NT FBsem	0.08	0.19	0.31
A <sup>4</sup> NT CycML	0.20	0.41	0.58
A <sup>4</sup> NT CycML+Lang	0.32	0.53	0.62

Table 4.4: Evaluating the A<sup>4</sup>NT anonymization against previously unseen (holdout) classifiers, on blogdata (age). Document-level F1 score is used.

Model	Holdout Classifier F1-score	Meteor
Original text	0.84	1.0
<i>C-Avg</i> (Karadzhov <i>et al.</i> , 2017)	0.77	0.55
Ours	<b>0.44</b>	<b>0.69</b>

Table 4.5: Comparison of our A<sup>4</sup>NT model to prior work on automatic anonymization. We compare both privacy effectiveness against a classifier and semantic consistency (meteor metric).

Evaluation from human judges is still the gold-standard with which models can be reliably compared.

Accordingly, we conduct user studies to judge the semantic similarity preserved by our A<sup>4</sup>NT networks. The evaluations were conducted on a subset of 745 random sentences from the test split of the blog-age dataset. First, output from different A<sup>4</sup>NT models is obtained for the 745 test sentences. If any model generates identical sentences to the input, this model is ranked first automatically without human evaluation. Note that, in some cases, multiple models can achieve rank-1, when they all produce identical outputs. The cases without any identical sentences to the input are evaluated using human annotators on Amazon Mechanical Turk (AMT). An annotator is shown one input sentence and multiple style-transfer outputs and is asked to pick the output sentence which is closest in meaning to the input sentence. Three unique annotators are shown each test sample and majority voting is used to determine the model which ranks first. Cases with no majority from human evaluators are excluded.

The main goal of the study is to identify which of the three A<sup>4</sup>NT networks performs best in terms of semantic similarity according to human judges. We also compare the best of our three systems to the baseline model based on Google machine translation, discussed in Section 4.4.3.

For the machine translation baseline, we obtain style-transferred texts from four different language round-trips. We started with English  $\rightarrow$  German  $\rightarrow$  French  $\rightarrow$  English, and obtained three more versions with incrementally adding Spanish, Finnish and finally Armenian languages into the chain before the translation back to English.

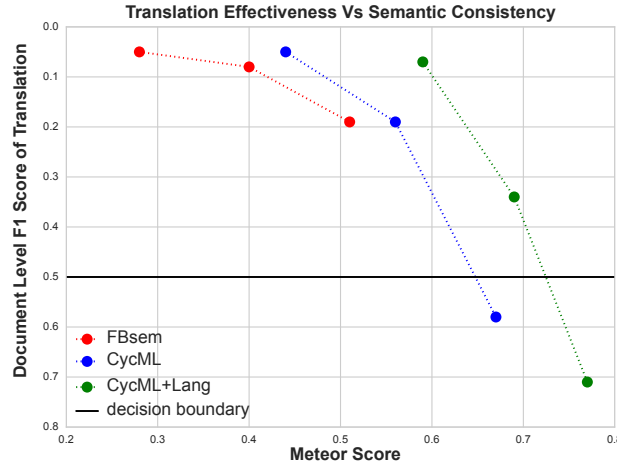


Figure 4.6: Operating points of A<sup>4</sup>NT models on test set.

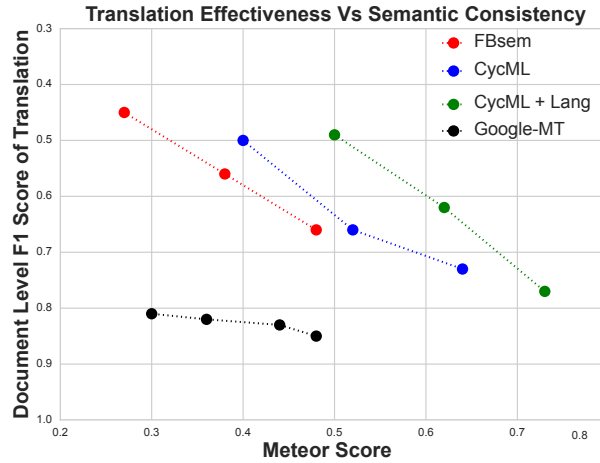


Figure 4.7: Privacy and semantic consistency of A<sup>4</sup>NT and the Google MT baseline on the human evaluation test set

To pick the operating points for the user study, we compare the performance of these four machine translation baselines and our three models on the human-evaluation test set in Figure 4.7. Note that here we show sentence-level F1 score on the y-axis as the human-evaluation test set is too small for document-level evaluation. We see that none of the Google machine translation baselines are able to fool the attribute classifiers. The model with 5-hop translation achieves best (lowest) F1-score of 0.81 which is only slightly less than the input data F1-score of 0.9. This model also achieves significantly worse meteor score than any of our A<sup>4</sup>NT models.

We conduct the user study comparing our style-transfer models on two operating points of 0.5 F1-score and 0.66 F1-scores, to obtain human judgments at two different levels of privacy effectiveness as shown in Table 4.6. We see that the model *CycML+Lang* outperforms the other two models at both operating points. *CycML+Lang* wins 50.74% of the time (ignoring ties) at operating point 0.5 and 57.87% of the time at operating

Operating Point	FBsem	CycML	CycML + Lang
0.66	32.02	39.75	<b>57.87</b>
0.5	15.03	31.68	<b>50.74</b>

Table 4.6: User study to judge semantic similarity. Three variants of our model are compared. Numbers show the % times the model ranked first. Can add to more than 100% as multiple models can have rank-1.

point 0.66. These results combined with quantitative evaluation discussed in Section 4.5.1 confirm that the cyclic ML loss combined with the language model loss gives the best trade-off between semantic similarity and privacy effectiveness.

Finally, we conduct the user study between the *CycML+Lang* model operating at 0.79 and the Google machine translation baseline with 3 hops. The operating point is chosen so that the two models are closest to each other in privacy effectiveness and meteor score. Results in Table 4.7 show that our model wins over the GoogleMT baseline by approximately 16% (59.46% vs 43.76% rank1) on semantic similarity as per human judges, while still having better privacy effectiveness. This is largely because our A<sup>4</sup>NT model learns not to change the input text if it is already ambiguous for the attribute classifier, and only makes changes when necessary. In contrast, changes made by GoogleMT round trip are not optimized towards maximizing privacy gain, and can change the input text even when no change is needed.

Apart from the relative evaluation between our model and the GoogleMT baseline, we additionally conduct separate a user study for both the models to assess the semantic similarity to the input sentence in an absolute scale. This study is conducted on the same human-evaluation test set containing 745 sentences and using the AMT platform as before. We show each human judge the input sentence and output form either of the models and ask them to rate the similarity to the input in a Likert scale from zero to five. We adopt the instruction used in SemEval task (Agirre *et al.*, 2016) to describe the different rating values to the user. Here zero rating corresponds to the worst case where the input and output sentences are not semantically related and five corresponds to the best case where they are equivalent in meaning. Each input-output pair is evaluated by three human judges and we report the mean score and standard deviation in Table 4.7. We see the same trend as in the relative evaluation and our model achieves better overall score of 4.51/5.0 compared to 4.16 obtained by the GoogleMT baseline. The score of the A<sup>4</sup>NT model lies between the ratings of 4.0 (sentences are equivalent with unimportant details differing) and 5.0 (sentences are equivalent). This shows that the A<sup>4</sup>NT model preserves the meaning of the input sentence on average, by making semantically equivalent changes to fool the authorship classifier.

#### 4.5.2 Qualitative analysis

In this section we analyze some qualitative examples of anonymized text produced by our A<sup>4</sup>NT model and try to identify the strengths and the weaknesses of this approach. Then we analyze the performance of the A<sup>4</sup>NT network on different levels of input

Comparison	A <sup>4</sup> NT CycML + Lang	GoogleMT
Operating point	0.79	0.85
Relative (% Rank 1)	<b>59.46</b>	43.76
Absolute (0-5)	<b>4.51±0.84</b>	4.16±0.89

Table 4.7: User study of our best model and the Google MT baseline.

difficulty. We use the attribute classifiers’ score as a proxy measure of the input text difficulty. If the text is confidently correctly classified (with classification score of 1.0) by the attribute classifier, then the A<sup>4</sup>NT network has to make significant changes to fool the classifier. If it is already misclassified, the style-transfer network should ideally not make any changes.

#### 4.5.2.1 Examples of style transfer for anonymization

Table 4.8 shows the results of our A<sup>4</sup>NT model *CycML+Lang* applied to some example sentences in the blog-age setting. Style transfer in both directions, teenager to adult and adult to teenager, is shown along with the corresponding source attribute classifier scores. The examples illustrate some of the common changes made by the model and are grouped into three categories for analysis (# column in Table 4.8).

**# 1. Using synonyms.** The A<sup>4</sup>NT network often uses synonyms to change the style to target attribute. This is seen in style transfers in both directions, teen to adult and adult to teen in category # 1 samples in Table 4.8. We can see the model replacing “yeh” with “ooh”, “would” with “will”, “...” with “,” and so on when going from teen to adult, and replacing “funnily enough” with “haha besides”, “work out” with “go out” and so on when changing from adult to teen. We can also see that the changes are not static, but depend on the context. For example “yeh” is replaced with “alas” in one instance and with “ooh” in another. These changes do not alter the meaning of the sentence too much, but fool the attribute classifiers thereby providing privacy to the author attribute.

**# 2. Replacing slang words.** When changing from teen to adult, A<sup>4</sup>NT often replaces the slang words or incorrectly spelled words with standard English words, as seen in category #2 in Table 4.8. For example, replacing “wad” (what) with “definitely”, “wadeva” with “perhaps” and “nuthing” with “ofcourse”. The opposite effect is seen when going from adult to teenager, with addition of “diz” (this) and replacing of “think” with “relized” (realized). These changes are learned entirely from the data, and would be very hard to encode explicitly in a rule-based system due to the variety in slangs and spelling mistakes.

**# 3. Semantic changes.** One failure mode of A<sup>4</sup>NT is when the input sentence has semantic content which is significantly more biased to the author’s class. These examples are shown in category #3 in Table 4.8. For example, when an adult author mentions his “wife”, the A<sup>4</sup>NT network replaces it with “crush”, altering the meaning of the input sentence. Some common entity pairs where this behavior is seen are with



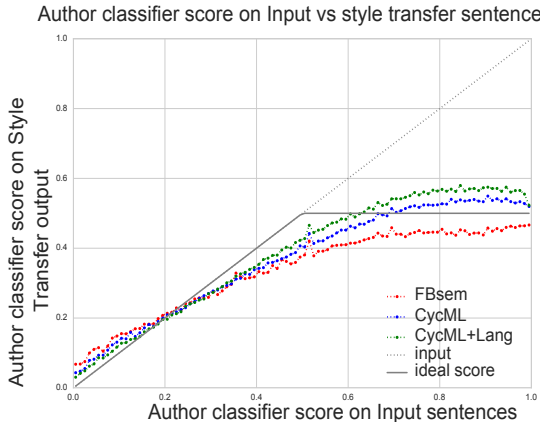


Figure 4.8: Output Privacy vs Privacy on Input.

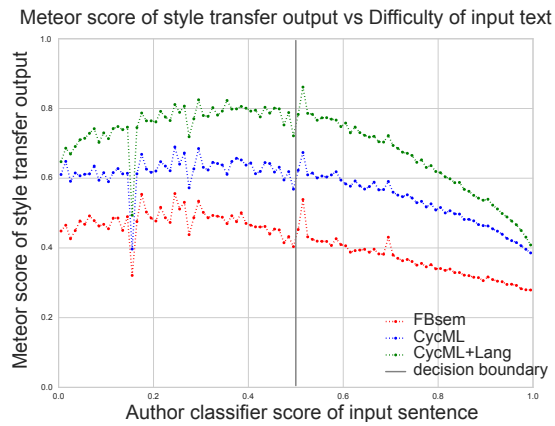


Figure 4.9: Meteor score plotted against input difficulty.

(*school*↔*work*), (*class*↔*office*), (*dad*↔*husband*), (*mum*↔*wife*), and so on. Arguably, in such cases, there is no obvious solution to mask the identity of the author without altering these obviously biased content words.

On the smaller speech dataset however, the changes made by the A<sup>4</sup>NT model alter the semantics of the sentences in some cases. Few example style transfers from Obama to Trump’s style are shown in Table 4.9. We see that A<sup>4</sup>NT inserts hyperbole (“better than anybody”, “horrible horrible”, “crooked”), references to “media” and “system”, all salient features of Trump’s style. We see that the style-transfer here is quite successful, sufficient to completely fool the identity classifier as was seen in Table 4.3. However, and somewhat expectedly, the semantics of the input sentence is generally lost. A possible cause is that the attribute classifier is too strong on this data, owing to the small dataset size and the highly distinctive styles of the two authors, and to fool them the A<sup>4</sup>NT network learns to make drastic changes to the input text.

#### 4.5.2.2 Performance across input difficulty

Figure 4.8 compares the attribute classifier score on the input sentence and the A<sup>4</sup>NT output. Ideally we want all the A<sup>4</sup>NT outputs to score below the decision boundary, while also not increasing the classifier score compared to input text. This “ideal score” is shown as grey solid line. We see that for the most part all three A<sup>4</sup>NT models are below or close to this ideal line. As the input text gets more difficult (increasing attribute classifier score), the *CycML* and *CycML+Lang* slightly cross above the ideal line, but still provide significant improvement over the input text (drop in classifier score of about  $\sim 0.45$ ).

Now, we analyze how much of input semantics is preserved with increasing difficulty. Figure 4.9 plots the meteor score of the A<sup>4</sup>NT output against the difficulty of the input text. We see that the meteor is high for sentences already across the decision boundary. These are easy cases, where the A<sup>4</sup>NT networks need not intervene. As the input gets more difficult, the meteor score of the A<sup>4</sup>NT output drops, as the network needs to do more changes to be able to fool the attribute classifier. The *CycML+Lang* model fares



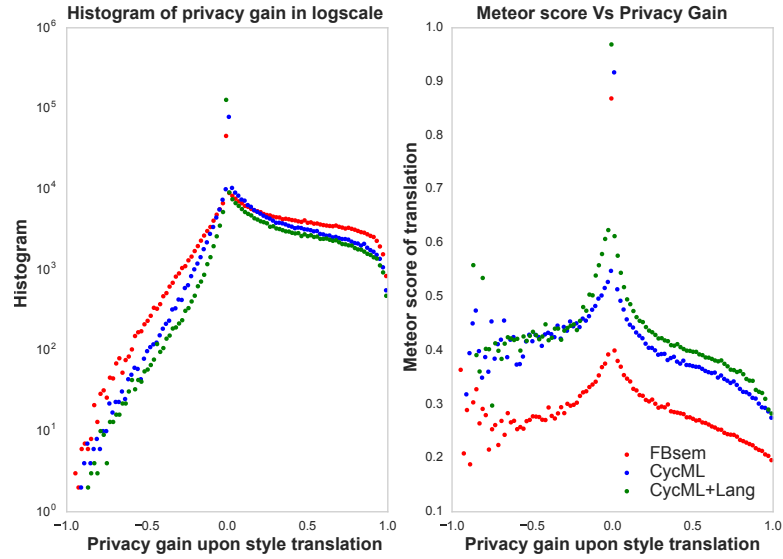


Figure 4.10: Histogram of privacy gain (left side) is shown alongside comparison of meteor score vs privacy gains.

better than the other two models, with consistently higher meteor across the difficulty spectrum.

Figure 4.10 shows the histogram of privacy gain across the test set. Privacy gain is the difference between the attribute classifier score on the input and the A<sup>4</sup>NT network output. We see that majority of transformations by the A<sup>4</sup>NT networks leads to positive privacy gains, with only a small fraction leading to negative privacy gains. This is promising given that this histogram is over all the 500k sentences in the test set. Meteor score plotted against privacy gain shown in Figure 4.10, again confirms that large privacy gains comes with a trade-off of loss in semantics.

#	Input: Teen	A(x)	Output: Adult	A(x)
1	and <u>yeh</u> ... it's raining lots now	0.97	and <u>ooh</u> ... it's raining lots now	0.23
1	<u>yeahh</u> ... i never let anyone really know how i'm feeling.	0.94	anyhow, i never let anyone really know how i'm feeling .	0.24
1	<u>yeh</u> , it's just goin ok here too!	0.95	alas, it's just goin ok here too!	0.30
1	<u>would</u> i go so far to say that i love her?	0.52	<u>will</u> i go so far to say that i love her?	0.36
2	<u>wad</u> a nice day.. spend almost the whole afternoon doing work!	0.99	<u>definitely</u> a nice day.. spend almost the whole afternoon doing work!	0.19
2	<u>wadeva</u> told u secrets <u>wad</u> did u do ?	0.98	<u>perhaps</u> told u secrets <u>why</u> did u do ?	0.49
2	i don't know <u>y</u> i even went into <u>dis</u> relationship	0.92	i don't know <u>why</u> i even went into <u>another</u> relationship .	0.33
2	i have <u>nuthing</u> else to say about this <u>horrid</u> day.	0.79	i have <u>ofcourse</u> else to say about this <u>accountable</u> day.	0.08
3	after school i <u>got</u> my hair cut so it looks nice again.	1.0	after <u>all</u> i <u>have</u> my hair cut so it looks nice again.	0.42
3	i had an interesting day at <u>skool</u> .	0.97	i had an interesting day at <u>wedding</u> .	0.05
#	Input: Adult	A(x)	Output: Teen	A(x)
1	<u>funnily</u> <u>enough</u> , i do n't care all that much.	0.58	<u>haha</u> <u>besides</u> , i do n't care all that much.	0.05
1	i <u>may</u> go to san francisco state, or i may go back.	0.54	i <u>shall</u> go to san francisco state, or i may go back.	0.09
1	i wonder if they 'll <u>work</u> out... hard to say.	0.52	i wonder if they 'll <u>go</u> out... hard to say.	0.39
2	one is to mix my exercise order a bit more.	0.97	one is to mix my <u>diz</u> exercise order a bit more.	0.08
2	ok, <u>think</u> i really will go to bed now.	0.79	ok, <u>relized</u> i really will go to bed now.	0.08
3	my first day going out to see <u>clients</u> after vacation.	0.98	my first day going out to see <u>some1</u> after vacation.	0.04
3	i'd tell my <u>wife</u> how much i love her every time i saw her.	0.96	i'd tell my <u>crush</u> how much i love her every time i saw <u>her</u> .	0.06
3	i <u>do</u> <u>believe</u> all you need is love.	0.58	i <u>dont</u> <u>think</u> all you need is love .	0.11

Table 4.8: Qualitative examples of anonymization through style transfer in the blog-age setting. Style transfer in both direction is shown along with the attribute classifier score of the source attribute.

Input: Obama	Output: Trump
we <u>can</u> do this because we are MISC.	we <u>will</u> do that because we are MISC.
we <u>can</u> do better than that.	we <u>will</u> do that better than <u>anybody</u> .
it's not about <u>reverend</u> PERSON.	it's not about <u>crooked</u> PERSON.
but i'm going to <u>need</u> your <u>help</u> .	but i'm going to <u>fight</u> <u>for</u> your <u>country</u> .
so that's my <u>vision</u> .	so that's my <u>opinion</u> .
their <u>situation</u> is getting worse.	their <u>media</u> is getting worse.
i'm <u>kind</u> of the <u>term</u> PERSON because i <u>do</u> care.	i'm <u>tired</u> of the <u>system</u> of PERSON <u>PERSON</u> because <u>they</u> <u>don't</u> care.
that's what <u>we</u> <u>need</u> to change.	that's what <u>she</u> <u>wanted</u> to change.
that's how our <u>democracy</u> <u>works</u> .	that's how our <u>horrible horrible</u> <u>trade deals</u> .

Table 4.9: Qualitative examples of style transfer on the speech dataset from Obama to Trump's style

## 4.6 CONCLUSIONS

We presented a novel fully automatic method for protecting privacy sensitive attributes of an author against NLP based attackers. Our solution consists of the A<sup>4</sup>NT network which learns to protect private attributes with novel adversarial training of a machine translation model. The A<sup>4</sup>NT network achieves this by learning to perform style-transfer without paired data.

A<sup>4</sup>NT offers a new data driven approach to authorship obfuscation. The flexibility of this end-to-end trainable model means it can adapt to new attack methods and datasets. Experiments on three different, attributes namely age, gender and identity, showed that the A<sup>4</sup>NT network is able to effectively fool the attribute classifiers in all three settings. We also show that the A<sup>4</sup>NT network performs well against multiple unseen classifier architectures. This strong empirical evidence suggests that the method is likely to be effective against previously unknown NLP adversaries.

We developed a novel solution to preserve the meaning of input text using likelihood of reconstruction. Semantic similarity (quantified by meteor score) of the A<sup>4</sup>NT network remains high for easier sentences, which do not contain obvious give-away words (school, work, husband etc.), but is lower on difficult sentences indicating the network effectively learns to identify and apply the right magnitude of change. The A<sup>4</sup>NT network can be operated at different points on the privacy-effectiveness and semantic-similarity trade-off curve, and thus offers flexibility to the user. The experiments on the political speech data show the limits to which style transfer based approaches can be used to hide attributes. On this challenging data with very distinct styles by the two authors, our method effectively fools the identity classifier but achieves this by altering the semantics of the input text.

## Part II

# Analyzing Model Robustness Through Image Manipulation

Despite rapid progress in computer vision benchmarks powered by deep neural networks, real-world deployment is hindered by sensitivity of these models to input distribution shifts. While many recent works have collected manually curated datasets to capture different distribution shifts (Barbu *et al.*, 2019; Hendrycks *et al.*, 2021, 2020), it is an expensive and slow process. Given the combinatorial nature of some attributes like co-occurring context objects, it might not even be feasible to collect these variations in a dataset. In this part of the thesis, we will explore developing generative models for controlled image editing and leveraging these models to create data variations like changes in context and appearance of objects and scenes. We show that this process can be used to efficiently create hard data at scale and analyze robustness of different computer vision systems.

In Chapter 5, we develop a generative model for removing objects from images, without requiring precise segmentation annotation. In Chapter 6, we leverage this object removal model to understand how much image classification and semantic segmentation networks rely on contextual evidence for making their predictions. Our analysis reveals that these networks exploit several spurious correlations in object co-occurrences to achieve good performance in i.i.d. setting, but break down when these correlations are broken in edited data. In Chapter 7, a similar methodology is used to analyze visual question answering models. Chapter 8 builds a generative model capable of editing appearance of an object, while keeping its pose and context intact. This synthesizer is then used to create challenging appearance variations targeting an object detector through adversarial optimization of the object appearance. Finally, in Chapter 9, we perform adversarial attack through a simulator to create adversarial weather configurations and study the performance of a semantic segmentation model under these variations.

## Contents

5.1	Introduction . . . . .	75
5.2	Learning to Remove Objects . . . . .	76
5.2.1	Two-staged editor architecture . . . . .	76
5.2.2	Mask priors . . . . .	77
5.2.3	Optimizing the in-painting network for removal . . . . .	78
5.3	Experimental Setup . . . . .	80
5.4	Results . . . . .	81
5.4.1	Qualitative results . . . . .	81
5.4.2	Quantitative evaluation of removal performance . . . . .	83
5.4.3	Ablation studies . . . . .	85
5.5	Conclusions . . . . .	86

RAPID progress has been seen in generative modeling of images, in particular synthesizing full images of faces or structured street scenes. In this chapter, we turn the focus towards more controlled editing of an input image and develop an automatic interaction-free object removal model. Similar to our A<sup>4</sup>NT model in Chapter 4, object removal model developed here learns to edit general scene images only using image-level labels and unpaired data in a GAN framework. We achieve this with two key contributions: a two-stage editor architecture consisting of a mask generator and image in-painter that cooperate to remove objects, and a novel GAN based prior for the mask generator that allows us to flexibly incorporate knowledge about object shapes. We experimentally show on two datasets that our method effectively removes a wide variety of objects using weak supervision only. The model developed here serves as a key tool for our work in Chapters 6 and 7, where we use image editing to study robustness of different computer vision systems.

## 5.1 INTRODUCTION

Automatic editing of scene-level images to add/remove objects and manipulate attributes of objects like color/shape etc. is a challenging problem with a wide variety of applications. Such an editor can be used for data augmentation (Shrivastava *et al.*, 2017), test case generation, automatic content filtering and visual privacy filtering (Orekondu *et al.*, 2018). To be scalable, the image manipulation should be free of human interaction and should learn to perform the editing without needing strong supervision. In this chapter, we investigate such an automatic interaction free image manipulation approach that involves editing an input image to remove target objects, while leaving the rest of the image intact.

The advent of powerful generative models like generative adversarial networks (GAN) has led to significant progress in various image manipulation tasks. Recent works have demonstrated altering facial attributes like hair color, orientation (Huang *et al.*, 2017a), gender (Lample *et al.*, 2017) and expressions (Choi *et al.*, 2018) and changing seasons in scenic photographs (Zhu *et al.*, 2017). An encouraging aspect of these works is that the image manipulation is learnt without ground truth supervision, but with using unpaired data from different attribute classes. While this progress is remarkable, it has been limited to single object centric images like faces or constrained images like street scenes from a single point of view (Wang *et al.*, 2018). In this work we move beyond these object-centric images and towards scene-level image editing on general images. We propose an automatic object removal model that takes an input image and a target class and edits the image to remove the target object class. It learns to perform this task with only image-level labels and without ground truth target images, i.e. using only unpaired images containing different object classes.

Our model learns to remove objects primarily by trying to fool object classifiers in a GAN framework. However, simply training a generator to re-synthesize the input image to fool object classifiers leads to degenerate solutions where the generator uses adversarial patterns to fool the classifiers. We address this problem with two key contributions. First we propose a two-stage architecture for our generator, consisting of a mask generator, and an image in-painter which cooperate to achieve removal. The mask generator learns to fool the object classifier by masking some pixels, while the in-painter learns to make the masked image look realistic. The second part of our solution is a GAN based framework to impose shape priors on the mask generator to encourage it to produce compact and coherent shapes. The flexible framework allows us to incorporate different shape priors, from randomly sampled rectangles to unpaired segmentation masks from a different dataset. Furthermore, we propose a novel locally supervised real/fake classifier to improve the performance of our in-painter for object removal. Our experiments show that our weakly supervised model achieves on par results with a baseline model using a fully supervised Mask-RCNN (He *et al.*, 2017) segmenter in a removal task on the COCO (Chen *et al.*, 2015) dataset.

An important use-case of our system would be in automatic content filtering, e.g. for privacy or parental control. This would involve automatic removal of objects and sensitive content from large databases or continuous streams of images. Content to be



removed in these scenarios are often personalized and beyond the usually studied object categories in computer vision. Thus a system which can learn to remove these objects from cheap image-level labels would be useful. We demonstrate the applicability of our object remover model to such content filtering task, by training it to automatically remove brand logos from images with only image level labels.

## 5.2 LEARNING TO REMOVE OBJECTS

We propose an end-to-end model which learns to find and remove objects automatically from images without any human interaction. It learns to perform this removal with only access to image-level labels without needing expensive ground-truth location information like bounding boxes or masks. Additionally, we do not have ground-truth target images showing the expected output image with the target object removed since it is infeasible to obtain such data in general.

We overcome the lack of ground-truth location and target image annotations by designing a generative adversarial framework (GAN) to train our model with only unpaired data. Here our editor model learns from weak supervision from three different classifiers. The model learns to locate and remove objects by trying to fool an object classifier. It learns to produce realistic output by trying to fool an adversarial real/fake classifier. Finally, it learns to produce realistic looking object masks by trying to fool a mask shape classifier. Let us examine these components in detail.

### 5.2.1 Two-staged editor architecture

Recent works [Lample \*et al.\* \(2017\)](#); [Choi \*et al.\* \(2018\)](#) on image manipulation utilize a generator network which takes the input image and synthesizes the output image to reflect the target attributes. While this approach works well for structured images of single faces, we found in own experiments that it does not scale well for removing objects from general scene images. In general scenes with multiple objects, it is difficult for the generator to remove only the desired object while re-synthesizing the rest of the image exactly. Instead, the generator finds the easier solution to fool the object classifier by producing adversarial patterns. This is also facilitated by the fact that the

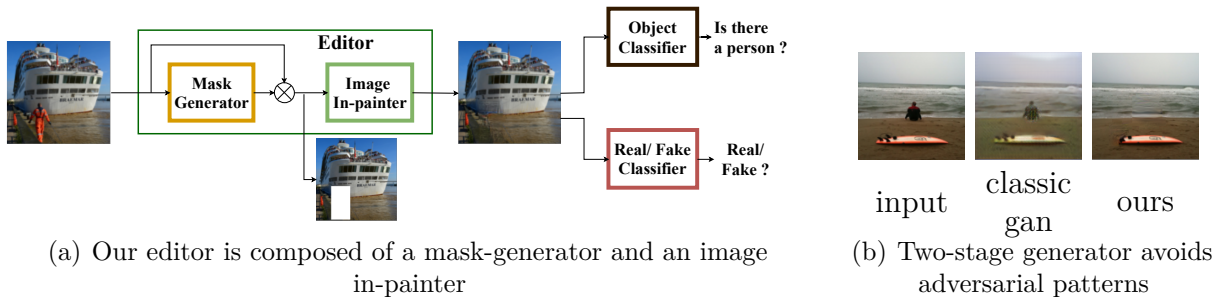


Figure 5.1: Illustrating (a) the proposed two-staged architecture and (b) the motivation for this approach

object classifier in crowded scenes has a much harder task than a classifier determining hair-colors in object centric images and thus is more susceptible to adversarial patterns. Figure 5.1(b) illustrates this observation, where a single stage generator from Choi *et al.* (2018) trying to remove the person, fools the classifier using adversarial noise. We can also see that the colors of the entire image have changed even when removing a single local object.

We propose a two-staged generator architecture shown in Figure 5.1(a) to address this issue. The first stage is a mask generator,  $G_M$ , which learns to locate the target object class,  $c_t$ , in the input image  $x$  and masks it out by generating a binary mask  $m = G_M(x, c_t)$ . The second stage is the in-painter,  $G_I$ , which takes the generated mask and the masked-out image as input and learns to in-paint to produce a realistic output. Given the inverted mask  $\widetilde{m} = 1 - m$ , final output image  $y$  is computed as

$$y = \widetilde{m} \cdot x + m \cdot G_I(\widetilde{m} \cdot x) \quad (5.1)$$

The mask generator is trained to fool the object classifier for the target class whereas the in-painter is trained to only fool the real/fake classifier by minimizing the loss functions shown below.

$$L_{cls}(G_M) = -\mathbb{E}_x [\log(1 - D_{cls}(y, c_t))] \quad (5.2)$$

$$L_{rf}(G_I) = -\mathbb{E}_x [D_{rf}(y)] \quad (5.3)$$

where  $D_{cls}(y, c_t)$  is the object classifier score for class  $c_t$  and  $D_{rf}$  is the real/fake classifier.

Here  $D_{rf}$  is adversarial, i.e. it is constantly updated to classify generated samples  $y$  as “fake”. The object classifier  $D_{cls}$  however is not adversarial, since it leads to the classifier using the context to predict the object class even when the whole object is removed. Instead, to make the  $D_{cls}$  robust to partially removed objects, we train it on images randomly masked with rectangles. The multiplicative configuration in (5.1) makes it easy for  $G_M$  to remove the objects by masking them out. Additionally, the in-painter also does not produce adversarial patterns as it is not optimized to fool the object classifier but only to make the output image realistic. The efficacy of this approach is illustrated in the image on the right on Figure 5.1(b), where our two-staged model is able to cleanly remove the person without affecting the rest of the image.

### 5.2.2 Mask priors

While the two-stage architecture avoids adversarial patterns and converge to desirable solutions, it is not sufficient. The mask generator can still produce noisy masks or converge to bad solutions like masking most of the image to fool the object classifier. A simple solution is to favor small sized masks. We do this by simply minimizing the exponential function of the mask size,  $\exp(\sum_{ij} m_{ij})$ . But this only penalizes large masks but not noisy or incoherent masks.

To avoid these degenerate solutions, we propose a novel mechanism to regularize the mask generator to produce masks close to a prior distribution. We do this by

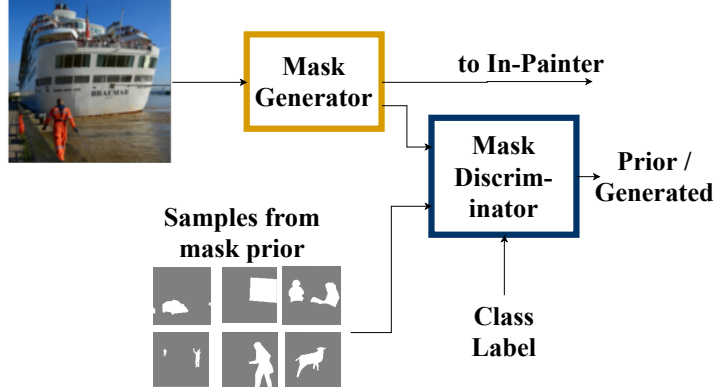


Figure 5.2: Imposing mask priors with a GAN framework

minimizing the Wasserstein distance between the generated mask distribution and the prior distribution  $P(m)$  using Wasserstein GAN (WGAN) (Arjovsky *et al.*, 2017) as shown in Figure 5.2. The WGAN framework allows flexibility while choosing the prior since we only need samples from the prior and not a parametric form for the prior.

The prior can be chosen with varying complexity depending on the amount of information available, including knowledge about shapes of different object classes. For example we can use unpaired segmentation masks from a different dataset as a shape prior to the generator. When this is not available, we can impose the prior that objects are usually continuous coherent shapes by using simple geometric shapes like randomly generated rectangles as the prior distribution.

Given a class specific prior mask distribution,  $P(m^p|c_t)$ , we setup a discriminator,  $D_M$  to assign high scores to samples from this prior distribution and the masks generated by  $G_M(x, c_t)$ . The mask generator is then additionally optimized to fool the discriminator  $D_M$ . The adversarial losses minimized by  $D_M$  and  $G_M$  are as below:

$$L(D_M) = \mathbb{E}_x [D_M(G_M(x, c_t), c_t)] - \mathbb{E}_{m^p \sim P(m^p|c_t)} [D_M(m^p, c_t)] \quad (5.4)$$

$$L_{\text{prior}}(G_M) = -\mathbb{E}_x [D_M(G_M(x, c_t), c_t)] \quad (5.5)$$

### 5.2.3 Optimizing the in-painting network for removal

The in-painter network  $G_I$  is tasked with synthesizing a plausible image patch to fill the region masked-out by  $G_M$ , to produce a realistic output image. Similar to prior works on in-painting (Yu *et al.*, 2018; Iizuka *et al.*, 2017; Liu *et al.*, 2018), we train  $G_I$  with self-supervision by trying to reconstruct random image patches and weak supervision from fooling an adversarial real/fake classifier. The reconstruction loss encourages  $G_I$  to keep consistency with the image while the adversarial loss encourages it to produce sharper images.

**Reconstruction losses.** To obtain self-supervision to the in-painter we mask random rectangular patches  $m^r$  from the input and ask  $G_I$  to reconstruct these patches. We minimize the  $L_1$  loss and the perceptual loss (Gatys *et al.*, 2015) between the in-painted

image and the input as follows:

$$L_{\text{recon}}(G_I) = \|G_I(\tilde{m}^r \cdot x) - x\|_1 + \sum_k \|\phi_k(G_I(\tilde{m}^r \cdot x)) - \phi_k(x)\|_1 \quad (5.6)$$

**Mask buffer.** The masks generated by  $G_M(x, c_t)$  can be of arbitrary shape and hence the in-painter should be able to fill in arbitrary holes in the image. We find that the in-painter trained only on random rectangular masks performs poorly on masks generated by  $G_M$ . However, we cannot simply train the in-painter with reconstruction loss in (5.6) on masks generated by  $G_M$ . Unlike random masks  $m^r$  which are unlikely to align exactly with an object, generated masks  $G_M(x, c_t)$  overlap the objects we intend to remove. Using reconstruction loss here would encourage the in-painter to regenerate this object. We overcome this by storing generated masks from previous batches in a *mask buffer* and randomly applying them on images from the current batch. These are not objects aligned anymore due to random pairing and we train the in-painter  $G_I$  with the reconstruction loss, allowing it to adapt to the changing mask distribution produced by the  $G_M(x, c_t)$ .

**Local real/fake loss.** In recent works on in-painting using adversarial loss [Yu et al. \(2018\)](#); [Iizuka et al. \(2017\)](#); [Liu et al. \(2018\)](#), in-painter is trained adversarially against a classifier  $D_{\text{rf}}$  which learns to predict global “real” and “fake” labels for input  $x$  and the generated images  $y$  respectively. A drawback with this formulation is that only a small percentage of pixels in the output  $y$  is comprised of truly “fake” pixels generated by the in-painter, as seen in Equation (5.1). This is a hard task for the classifier  $D_{\text{rf}}$  hard since it has to find the few pixels that contribute to the global “fake” label. We tackle this by providing local pixel-level real/fake labels on the image to  $D_{\text{rf}}$  instead of a global one. The pixel-level labels are available for free since the inverted mask  $\tilde{m}$  acts as the ground-truth “real” label for  $D_{\text{rf}}$ . Note that this is different from the patch GAN ([Isola et al., 2016](#)) where the classifier producing patch level real/fake predictions is still supervised with a global image-level real/fake label. We use the least-square GAN loss ([Mao et al., 2017](#)) to train the  $D_{\text{rf}}$ , since we found the WGAN loss to be unstable with local real/fake prediction. This is because,  $D_{\text{rf}}$  can minimize the WGAN loss with assigning very high/low scores to one patch, without bothering with the other parts of the image. However, least-squares GAN loss penalizes both very high and very low predictions, thereby giving equal importance to different image regions.

$$L(D_{\text{rf}}) = \frac{1}{\sum_{ij} \tilde{m}_{ij}} \sum_{ij} \tilde{m}_{ij} \cdot (D_{\text{rf}}(y)_{ij} - 1)^2 + \frac{1}{\sum_{ij} m_{ij}} \sum_{ij} m_{ij} \cdot (D_{\text{rf}}(y)_{ij} + 1)^2 \quad (5.7)$$

**Penalizing variations.** We also incorporate the style-loss ( $L_{\text{sty}}$ ) proposed in [Liu et al. \(2018\)](#) to better match the textures in the in-painting output with that of the input image and the total variation loss ( $L_{\text{tv}}$ ) since it helps produce smoother boundaries between the in-painted region and the original image.

The mask generator and the in-painter are optimized in alternate epochs using gradient descent. When the  $G_M$  is being optimized, parameters of  $G_I$  are held fixed and vice-versa when  $G_I$  is optimized. We found that optimizing both the models at

every step led to unstable training and many training instances converged to degenerate solutions. Alternate optimization avoids this while still allowing the mask generator and in-painter to co-adapt. The final loss function for  $G_M$  and  $G_I$  is given as:

$$L_{\text{total}}(G_M) = \lambda_c L_{\text{cls}} + \lambda_p L_{\text{prior}} + \lambda_{sz} \exp(\sum_{ij} m_{ij}) \quad (5.8)$$

$$L_{\text{total}}(G_I) = \lambda_{rf} L_{\text{rf}} + \lambda_r L_{\text{recon}} + \lambda_{tv} L_{\text{tv}} + \lambda_{sty} L_{\text{sty}} \quad (5.9)$$

### 5.3 EXPERIMENTAL SETUP

**Datasets.** Keeping with the goal of performing removal on general scene images, we train and test our model mainly on the COCO dataset (Chen *et al.*, 2015) since it contains significant diversity within object classes and in the contexts in which they appear. We test our proposed GAN framework to impose priors on the mask generator with two different priors namely rotated boxes and unpaired segmentation masks. We use the segmentation masks from Pascal-VOC 2012 dataset (Everingham *et al.*) (without the images) as the unpaired mask priors. To facilitate this we restrict our experiments on 20 classes shared between the COCO and Pascal datasets. To demonstrate that our editor model can generalize beyond objects and can learn to remove to different image entities, we test our model on the task of removing logos from natural images. We use the Flickr Logos dataset (Kalantidis *et al.*, 2011), which has a training set of 810 images containing 27 annotated logo classes and a test set of 270 images containing 5 images per class and 135 random images containing no logos.

**Evaluation metrics.** We evaluate our object removal for three aspects: *removal performance* to measure how effective is our model at removing target objects and *image quality assessment* to quantify how much of the original image is edited and finally *human evaluation* to judge removal.

- **Removal performance:** We quantify the removal performance by measuring the performance of an object classifier on the edited images using two metrics. *Removal success rate* measures the percentage of instances where the editor successfully fools the object classifier score below the decision boundary for the target object class. *False removal rate* measures the percentage of cases where the editor removes the wrong objects while trying to remove the target class. This is again measured by monitoring if the object classifier score drops below decision boundary for other classes.

- **Image quality assessment:** To be useful, our editor should remove the target object class while leaving the rest of the image intact. Thus, we quantify the usefulness by measuring similarity between the output and the input image using three metrics namely peak signal-to-noise ratio (pSNR), structural similarity index (ssim) (Wang *et al.*, 2004) and perceptual loss (Zhang *et al.*, 2018b). The first two are standard metrics used in image in-painting literature, whereas the perceptual loss (Zhang *et al.*, 2018b) was recently proposed as a learned metric to compare two images. We use the squeezeNet variant of this metric.

- **Human evaluation:** We conduct a study to obtain human judgments of removal performance. We show hundred randomly selected edited images to a human judge and asked if they see the target object class. To keep the number of annotations reasonable, we conduct the human evaluation only on the person class (largest class). Each image is shown to three separate judges and removal is considered successful when all three humans agree that they do not see the object class. The participants in the study were not aware of the project and were just asked to determine if they see a 'person' (either full body or clear body parts/ silhouettes) in the images shown. The outputs from different models were all shown in the same session to a human judge in a randomized order to prevent biasing the results against latter models. This human study evaluates the removal system holistically and helps verify that the removal performance measured by a classifier is similar to as perceived by the humans, and thus validating the automatic evaluation protocol.

**Baselines with additional supervision.** Since there is no prior work proposing a fully automatic object removal solution, we compare our model against removal using a stand-alone fully supervised segmentation model, Mask-RCNN (He *et al.*, 2017). We obtain segmentation mask predictions from Mask-RCNN and use our trained in-painter to achieve removal. Additionally we also compare our model to a weakly supervised segmentation method from Khoreva *et al.* (2017) (referred to as SDI), which learns to segment objects by using ground truth bounding boxes as supervision. Please note that both the above methods use stronger supervision in terms of object bounding boxes (Mask-RCNN and SDI) and object segmentation (Mask-RCNN) than our proposed method, which uses only image level labels.

## 5.4 RESULTS

We present qualitative and quantitative evaluations of our editor and comparisons to the Mask-RCNN based removal. Qualitative results show that our editor model works well across diverse scene types and object classes. Quantitative analysis shows that our weakly supervised model performs on par with the fully supervised Mask-RCNN in the removal task, in both automatic and human evaluation.

### 5.4.1 Qualitative results

Figure 5.3 shows the results of object removal performed by our model (last row) on the COCO dataset compared to the Mask-RCNN baseline. We see that our model works across diverse scene types, with single objects (columns 1-4) or multiple instances of the same object class (col. 5-6) and even for a fairly large object (last column). Figure 5.3 also highlight the problems with simply using masks from a segmentation model, Mask-RCNN, for removal. Mask-RCNN is trained to accurately segment the objects and thus the masks it produces very closely trace the object boundary, too closely for removal purposes. We can clearly see the silhouettes of objects in all the edited images on the second row. These results justify our claim that segmentation



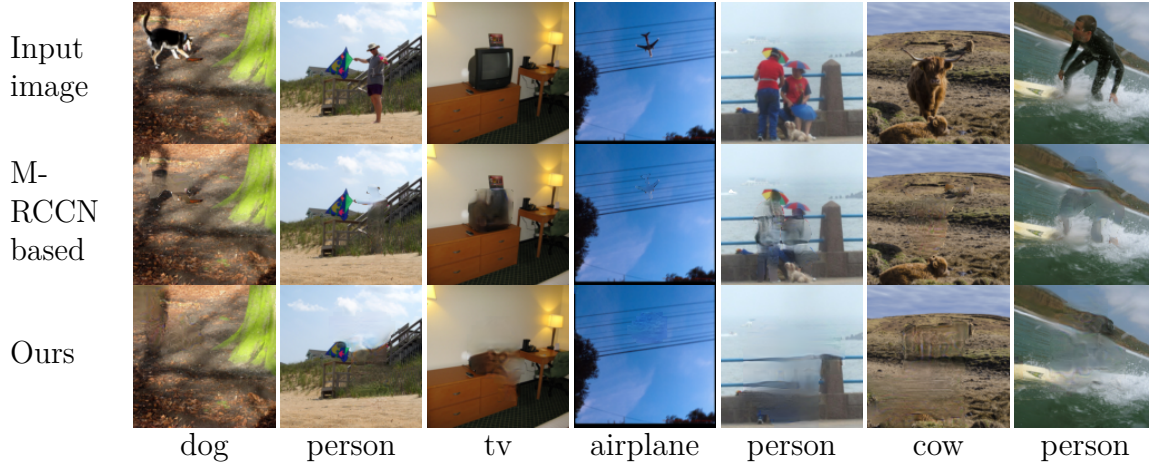


Figure 5.3: Qualitative examples of removal of different object classes

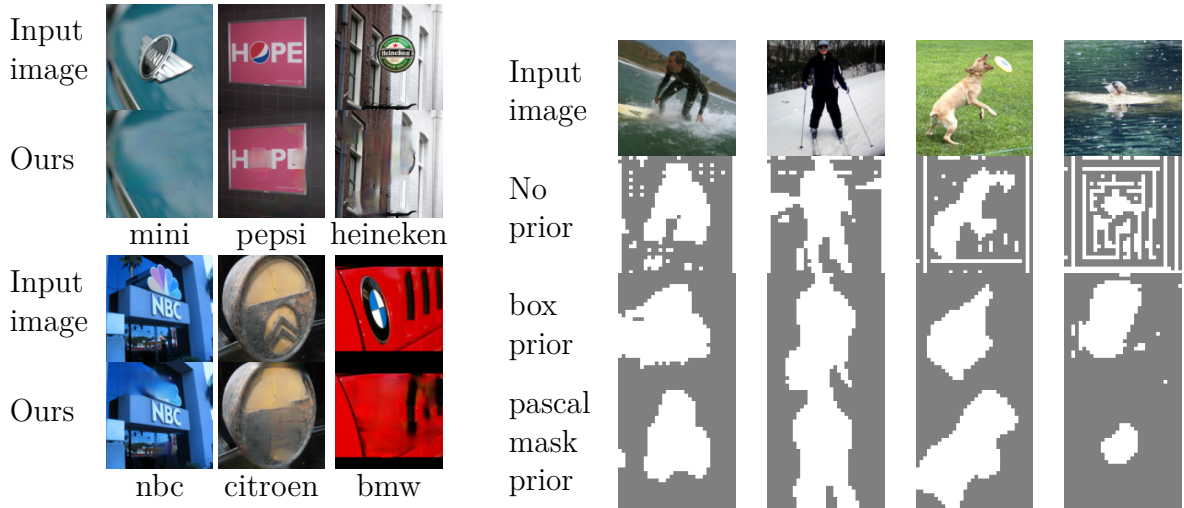


Figure 5.4: Results of logo removal    Figure 5.5: Effect of priors on generated masks

annotations are not needed to learn to remove objects and might not be the right annotations anyway.

Our model is not tied to notion of objectness and can be easily extended to remove other image entities. The flexible GAN based mask priors allow us to use random rectangular boxes as priors when object shapes are not available. To demonstrate this we apply our model to the task of removing brand logos automatically from images. The model is trained using image level labels and box prior. Qualitative examples in Figure 5.4 shows that our model works well for this task, despite the fairly small training set (800 images). It is able to find and remove logos in different contexts with only image level labels. The image on the bottom left shows a failure case where the model fails to realize that the text “NBC” belongs to the logo.

Figure 5.5 shows the masks generated by our model with different mask priors on the COCO dataset. These examples illustrate the importance of the proposed mask priors. The masks generated by the model using no prior (second row) are very noisy since the



model has no information about object shapes and is trying to infer everything from the image level classifier. Adding the box prior already makes the masks much cleaner and more accurate. We can note that the generated masks are “boxier” while not strictly rectangles. Finally using unpaired segmentation masks from the pascal dataset as shape priors makes the generated masks more accurate and the model is able to recover the object shapes better. This particularly helps in object with diverse shapes, for example people and dogs.

#### 5.4.2 Quantitative evaluation of removal performance

To quantify the removal performance we run an object classifier on the edited images and measure its performance. We use a separately trained classifier for this purpose, not the one used in our GAN training, to fairly compare our model and the Mask-RCNN based removal.

**Sanity of object classifier performance.** The classifier we use to evaluate our model achieves per-class average F1-score of 0.57, overall average F1-score of 0.67 and mAP of 0.58. This is close to the results achieved by recent published work on multi-label classification (Wang *et al.*, 2016a) on the COCO dataset, which achieves class average F1-score of 0.60, overall F1-score of 0.68 and mAP of 0.61. While these numbers are not directly comparable (different image resolution, different number of classes), it shows that our object classifier has good performance and can be relied upon. Furthermore, human evaluation shows similar results as our automatic evaluation.

**Effect of priors.** Table 5.1 compares the different versions of our model using different priors. The box prior uses randomly generated rectangles of different aspect ratios, area and rotations. The *Pascal* ( $n$ ) prior uses  $n$  randomly chosen unpaired segmentation masks for each class from the Pascal dataset. The table shows metrics measuring the removal performance, image quality and mask accuracy. The arrows  $\uparrow$  and  $\downarrow$  indicate if higher or lower is better for the corresponding metric. Comparing removal performance in Table 5.1 we see that while the model with no prior achieves very high removal rate (94%), but it does so with large masks (37 %) which causes low output image quality. As we add priors, the generated masks become smaller and compact. We also see that mIoU of the masks increase with stronger priors (0.22-0.23 for pascal prior), indicating they are more accurate. Smaller and more accurate masks also improve the image quality metrics and false removal rates which drop more than half from 36% to 16%. This is inline with the visual examples in Figure 5.5, where model without prior produces very noisy masks and quality of the masks improve with priors.

Another interesting observation from Table 5.1 is that using very few segmentation masks from pascal dataset leads to a drop in removal success rate, especially for the *person* class. This is because the *person* class has very diverse shapes due to varying poses and scales. Using only ten masks in the prior fails to capture this diversity and performs poorly (59%). As we increase the number of mask samples in the prior, removal performance jumps significantly to 81% on the person class. Considering these results, we note that the *pascal all* version offers the best trade-off between removal and image quality due to more accurate masks and we will use this model in comparison to

Prior	Removal Performance			Image quality metrics			Mask accuracy	
	removal success $\uparrow$		false $\downarrow$	percep. loss $\downarrow$	pSNR $\uparrow$	ssim $\uparrow$	mIoU $\uparrow$	% masked area $\downarrow$
	all	person	removal					
None	<b>94</b>	<b>96</b>	36	0.13	19.97	0.743	0.15	37.7
boxes	83	88	23	0.11	20.41	0.777	0.18	28.1
pascal (10)	67	59	17	<b>0.07</b>	<b>23.81</b>	<b>0.833</b>	<b>0.23</b>	<b>16.7</b>
pascal (100)	70	75	<b>16</b>	<b>0.07</b>	23.02	0.821	0.22	18.1
pascal (all)	73	81	<b>16</b>	0.08	22.64	0.803	0.22	20.2

Table 5.1: Quantifying the effect of using more accurate mask priors

Model	dilation	Removal Success	
		all	person
SDI: supervised with GT boxes	-	54	45
	7x7	64	65
Ours	-	73	<b>81</b>

Table 5.2: Comparison to weakly supervised semantic segmentation model, SDI (Khoreva *et al.*, 2017)

benchmarks.

**Benchmarking against GT and Mask-RCNN.** Table 5.3 compares the performance of our model against baselines using ground-truth (GT) masks and Mask-RCNN segmentation masks for removal. These benchmarks use the same in-painter as *our-pascal* model. We see that our model outperforms the fully supervised Mask-RCNN masks and even the GT masks in terms of removal (66%& 68% vs 73%). While surprising, this is explained by the same phenomenon we saw in qualitative results with Mask-RCNN in Figure 5.3. The GT and Mask-RCNN masks for segmentation are too close to the object boundaries and thus leave object silhouettes behind when used for removal. When we dilate the masks produced by Mask-RCNN before using for removal, the performance improves overall and is on par with our model (slightly better in all classes and a bit worse in the person class). The drawback of weak supervision is that masks are a bit larger which leads to bit higher false removal rate (16% ours compared to 10% Mask-RCNN dilated) and lower image quality metrics. However this is still a significant result, given that our model is trained without expensive ground truth segmentation annotation for each image, but instead uses only unpaired masks from a smaller dataset.

**Comparison to weakly supervised segmentation.** We compare to the weakly supervised SDI (Khoreva *et al.*, 2017) model in Table 5.2. We use the the output masks generated by SDI to mask the image and use the in-painter trained with our model to fill in the masked region. Simply using the masks from SDI without dilation results in poor removal performance with only 54% success overall and 45% success on the ‘person’ class. Upon dilation, the performance improves, but is still significantly worse than our model and Mask-RCNN.

Model	Supervision	Removal Performance			Image quality metrics		
		removal success $\uparrow$		false $\downarrow$	percep.	pSNR $\uparrow$	ssim $\uparrow$
		all	person	removal	loss $\downarrow$		
GT masks	-	66	72	5	<b>0.04</b>	<b>27.43</b>	<b>0.930</b>
Mask RCNN	Seg. masks &	68	73	6	0.05	25.59	0.900
Mask RCNN (dil. 7x7)	bound boxes	75	77	10	0.07	24.13	0.882
ours-pascal	image labels & unpaired masks	73	<b>81</b>	16	0.08	22.64	0.803

Table 5.3: Comparison to ground truth masks and Mask-RCNN baselines.

Additionally, SDI method starts from boxes generated by a fully supervised RCNN network and generates segmentation with weak supervision, whereas our model uses only image-level labels and hence is more generally applicable.

**Human evaluation.** We verify our automatic evaluation results using a user study to evaluate removal success as described in Section 5.3. The human judgements of removal performance follow the same trend seen in automatic evaluation, except that human judges penalize the silhouettes more severely. Our model clearly outperforms the baseline Mask-RCNN model without dilation by achieving 68% removal rate compared to only 30% achieved by Mask-RCNN. With dilated masks, Mask-RCNN performs similar to our model in terms of removal achieving 73% success rate.

#### 5.4.3 Ablation studies

**Joint optimization.** We conduct an experiment to test if jointly training the mask generator and the in-painter helps. We pre-train the in-painter using only random boxes and hold it fixed while training the mask generator. The results are shown in Table 5.5. Not surprisingly, the in-painting quality suffers with higher perceptual loss (0.10 vs 0.08) since it has not adapted to the masks being generated. More interestingly, the mask generator also degrades with a fixed in-painter, as seen by lower mIoU (0.19 vs 0.22) and lower removal success rate (0.68 vs 0.73). This result shows that it is important to train both the models jointly to allow them to adapt to each other for best performance.

**In-painting components.** Table 5.4.3 shows the ablation of the in-painter network components. We note that the proposed *mask-buffer*, which uses masks from previous batch to train the in-painter with reconstruction loss, significantly improves the results significantly in all three metrics. Using local loss improves the results in-terms of perceptual loss (0.10 vs 0.12) while being slightly worse in the other two metrics. However on examining the results visually in Figure 5.6, we see that the version with the global GAN loss produces smooth and blurry in-painting, whereas the version with local GAN loss produces sharper results with richer texture. While these blurry results do better in pixel-wise metrics like pSNR and ssim, they are easily seen by the human eye and are not suitable for removal. Finally addition of total variation and style loss



Input image    Global loss    Local loss

Figure 5.6: Comparing global and local GAN loss. Global loss smooth blurry results, while local one produce sharp, texture-rich images.

Mask buffer	GAN	TV+ Style	percep. loss ↓	pSNR ↑	ssim ↑
-	G	-	0.13	20.0	0.730
✓	G	-	0.12	<b>21.9</b>	<b>0.772</b>
✓	L	-	<b>0.10</b>	21.5	0.758
✓	L	✓	<b>0.10</b>	21.6	0.763

Table 5.4: Evaluating in-painting components

Joint training	Removal success ↑	mIou ↑	percep. loss ↓
-	0.68	0.19	0.10
✓	<b>0.73</b>	<b>0.22</b>	<b>0.08</b>

Table 5.5: Joint training helps improve both mask generation and in-painting

helps slightly improve the pSNR and ssim metrics.

## 5.5 CONCLUSIONS

We presented an automatic object removal model which learns to find and remove objects from general scene images. Our model learns to perform this task with only image level labels and unpaired data. Our two-stage editor model with a mask-generator and an in-painter network avoids degenerate solutions by complementing each other. We also developed a GAN based framework to impose different priors to the mask generator, which encourages it to generate clean compact masks to remove objects. Results show that our model achieves similar performance as a fully-supervised segmenter based removal, demonstrating the feasibility of weakly supervised solutions for the general scene-level editing task.

---

## Contents

6.1	Introduction . . . . .	88
6.2	Quantifying the Role of Context . . . . .	89
6.2.1	Object removal . . . . .	89
6.2.2	Measuring context dependency . . . . .	90
6.2.3	Data augmentation with object removal . . . . .	91
6.3	Experiments and Results . . . . .	93
6.3.1	Image level classification . . . . .	93
6.3.2	Semantic segmentation . . . . .	98
6.4	Conclusions . . . . .	101

---

IMPORTANCE of visual context in scene understanding tasks is well recognized in the computer vision community. However, to what extent the computer vision models are dependent on the context to make their predictions is unclear. A model overly relying on context will fail when encountering objects in different contexts than in training data and hence it is important to identify these dependencies before we can deploy the models in the real-world. In this chapter, utilizing the editor developed in Chapter 5, we propose a method to quantify the sensitivity of black-box vision models to visual context. We create context changes by removing selected objects from input images and measure the response of the target models, allowing us to quantify their robustness to these variations. We apply this methodology on two tasks, image classification and semantic segmentation, and discover undesirable dependency between objects and context, for example that “sidewalk” segmentation relies heavily on “cars” being present in the image. We propose an object removal based data augmentation solution to mitigate this dependency and increase the robustness of classification and segmentation models to contextual variations. Our experiments show that the proposed data augmentation helps these models improve the performance in out-of-context scenarios, while preserving the performance on regular data.

## 6.1 INTRODUCTION

Visual context of an object in an image is an important source of information for scene understanding tasks in both human and computer vision (Torralba *et al.*, 2010; Parikh *et al.*, 2012). Contextual cues such as presence of frequently co-occurring objects can help resolve ambiguities between visually similar classes and improve performance in various vision tasks including object detection (Mottaghi *et al.*, 2014; Bell *et al.*, 2016) and segmentation (Zhang *et al.*, 2018a). However, objects can also appear in previously unseen context or be absent from a very typical context. For example, we might find a keyboard on a desk without a monitor (object-without-context), or find a monitor without a keyboard (context-without-object). While humans can handle both these atypical scenarios gracefully, computer vision models often fail by ignoring the visual evidence for the object in object-without-context case or hallucinating objects which are not actually present in the image in context-without-object case. For example, in our experiments we find that *keyboard* is often not recognized without a nearby *monitor*, and semantic segmentation of roads suffers without *cars* (see Figure 6.1). While context can be an important cue, this kind of too heavy or even pathological dependency on contextual signals is undesirable, and it is important to systematically identify and ideally fix such cases. In this work, we analyze and quantify the effect contextual information on two tasks, multi-label classification and semantic segmentation.

Context generally refers to different kinds of information including co-occurring objects, scene type and lighting. For our analysis, we limit context to only the set of co-occurring objects in the image. While this might seem restrictive, we find in our analysis that image classification and segmentation models learn many interesting and undesirable dependencies between an object and other co-occurring objects (context) in the image. We use object removal as the main methodology to understand and quantify the role of context in downstream vision models. Specifically, we compare the output of the target models on the original input image and an edited version of this image with one object removed from it. If the model heavily uses the contextual relationship between removed object and the objects present in the image, removal will have an adverse effect on the model output. Measuring this helps us quantify the contextual dependencies learnt by the target model.

Ideally we want models which can utilize contextual cues when available, but are robust to variations in context and can detect and segment objects even when they appear out of context. However, machine learning based vision models are biased to the data seen frequently in training and tend to perform poorly on less frequent situations, for example the object-without-context and context-without-object scenarios. We address this by proposing a data augmentation scheme to expose the image classification and segmentation models to different contexts during training, and thus improving the robustness of the models to context. This is done by removing selected objects from images and training the models on the edited images to recognize and segment other objects in the image, even with contextual objects removed. Our experiments show that the classification and segmentation models trained with this data augmentation scheme are less sensitive to context changes and perform better on real out-of-context datasets,



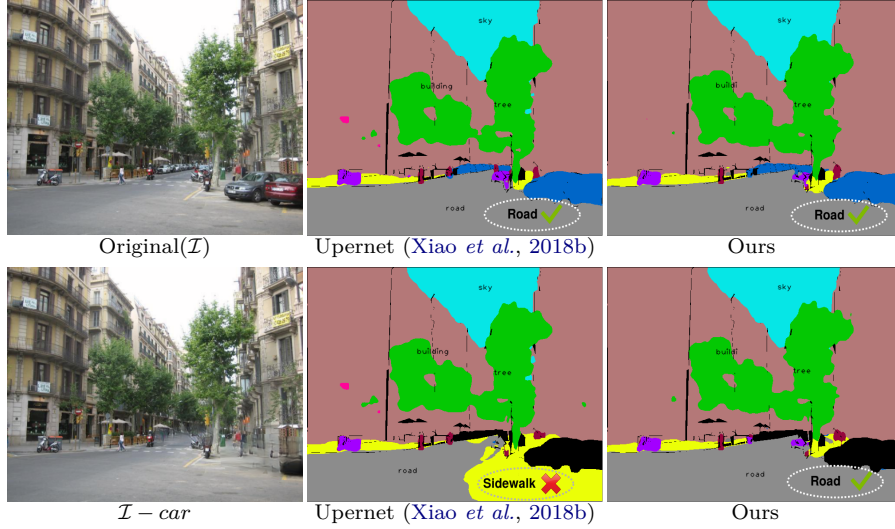


Figure 6.1: An example of the sensitivity of *road* and *sidewalk* segmentation to the context object *car*. Removing *car* from the image (second row) causes segmentation errors in the baseline model which hallucinates a sidewalk (yellow) when there is none. Our model trained with proposed data-augmentation is more robust to these context changes.

while preserving the baseline performance on the regular data splits.

To summarize, the main contributions of this paper are as follows: a) We propose an object removal based method to understand and quantify sensitivity of vision models to context, b) We apply this to analyze image classification and segmentation models and find some interesting and undesirable dependencies learnt by the models between classes and contextual objects and c) We propose a data augmentation scheme based on object removal to make the models more robust to contextual variation and show that it helps improve performance in out-of-context scenarios.

## 6.2 QUANTIFYING THE ROLE OF CONTEXT

We use object removal to quantify the contextual dependence of image classification and segmentation models, by designing metrics which measure the change in the model output between the original and the edited images with context objects removed. Now, we will discuss our removal model, present the robustness metrics and the data augmentation strategies to reduce the contextual dependence and improve performance in out-of-context setting.

### 6.2.1 Object removal

To create edited images with context objects removed, we need a fully automatic object removal model. For this, we utilize ground-truth object masks to remove the desired object and use an in-painting network to fill in the removed region. We co-opt the



in-painting network from the object removal model in Chapter 5, since this inpainter is directly optimized for removal, and can better handle irregular masks used in removal. The above removal method works well for medium sized objects, but struggles for large objects since then the in-painter needs to synthesize most of the image. Hence, we impose size restrictions on the objects we choose to remove to be less than 30% of the image. In the classification scenario on the COCO dataset, we consider all 80 object categories for removal. In the segmentation setting on the ADE20k dataset, we consider only the non-stuff categories (90 categories) for removal and measure the effects of removing these objects on the segmentation of all 140 categories. The stuff categories include objects like road, sky and field which are typically very large and hard to inpaint and hence are excluded from removal. An important point to note here is that the in-painter is not aware of the downstream models and is not optimized to fool/change their decisions. The effects of the in-painter are local and only around the removed object. Qualitative examples in Figures 6.2 and 6.3 show that the in-painting works well in the object removal setting.

### 6.2.2 Measuring context dependency

To understand the effect of contextual cues on image-classification and segmentation models, we test them on edited images where a context object has been removed. Precisely, given an original image  $I$  containing a set of objects  $C = \{c_1, \dots, c_n\}$ , we create a set of edited images  $\mathcal{I}_e = \{I - c_i | c_i \in C \text{ and removable}(c_i)\}$ . Then, we test the target model on  $I$  and  $\mathcal{I}_e$  and check if its output is consistent with the performed removal as described below.

**Image-level classification.** Given a trained classifier  $S_{c_i}$  for class  $c_i$ , we will now characterize how robust it is to changes in context of  $c_i$ . We first obtain classifier scores for the original image  $I$ , edited image  $I - c_i$  with object  $c_i$  removed and for the edited set  $\mathcal{I}_{\text{owc}} = \{I - c_j : c_j \in I, j \neq i\}$ , all of which contain the object  $c_i$  but have one context object removed. Ideally, if the classifier  $S_{c_i}$  is robust to context changes it should score all the images in  $\mathcal{I}_{\text{owc}}$  higher than the image  $I - c_i$ , since  $I - c_i$  does not contain the object  $c_i$  and the images in  $\mathcal{I}_{\text{owc}}$  do. Precisely, a classifier robust to context should satisfy the below in-equality:

$$S_{c_i}(I_{\text{owc}}) \geq S_{c_i}(I - c_i), \forall I_{\text{owc}} \in \mathcal{I}_{\text{owc}} \quad (6.1)$$

We can count the number of times this condition is violated to quantitatively measure the robustness of the classifier.

$$V^{\min}(c_i) = \frac{\sum_I \mathbb{1}[(\min_{I_{\text{owc}}} S_{c_i}(I_{\text{owc}})) < S_{c_i}(I - c_i)]}{\sum_I \mathbb{1}[c_i \in I]} \quad (6.2)$$

$$V^{\text{mean}}(c_i) = \frac{\sum_I \mathbb{1}[\mathbb{E}_{I_{\text{owc}}}[S_{c_i}(I_{\text{owc}})] < S_{c_i}(I - c_i)]}{\sum_I \mathbb{1}[c_i \in I]} \quad (6.3)$$

where  $\mathbb{1}$  is the indicator variable.  $V^{\min}(c_i)$  is a strict metric counting instances classifier scores  $I - c_i$  higher than any of the edited images, whereas  $V^{\text{mean}}(c_i)$  is a softer metric

counting instances where  $I - c_i$  is scored higher than the average score assigned to the edited images.

**Semantic segmentation.** To understand the role context plays in this pixel-level labeling task, we analyze the behaviour of a trained segmentation model by removing one object at a time from the original image. Specifically, we measure how the segmentation correctness of the rest of the image changes (as compared to segmentation of the original image) when we remove an object from the original image. Given a segmentation model  $P$ , we compute the intersection-over-union (IoU) for a class  $c_i$  (w.r.t. ground-truth) on the original image  $I$  and edited image  $I - c_j$ . If the IoU value changes more than threshold  $\alpha$ , we consider the segmentation prediction for class  $c_i$  to be affected by removal of  $c_j$ . Counting these violations we get,

$$AR(c_i, c_j) = \frac{\sum_I \mathbb{1} \left[ \left| \Delta \text{IoU}_{c_i c_j} \right| \geq \alpha \right]}{\sum_I \mathbb{1} [c_i, c_j \in I]} \quad (6.4)$$

where  $\Delta \text{IoU}_{c_i c_j}$  is the change in IoU of class  $c_i$  with removal of object  $c_j$  and  $\alpha$  is the change threshold. The matrix  $AR(c_i, c_j)$  represents the fraction of images where removing the object  $c_j$ , affects the segmentation of the object  $c_i$  with high values of  $AR(c_i, c_j)$  indicating that the segmentation model depends heavily on the presence of the context object  $c_j$  to segment  $c_i$ .

### 6.2.3 Data augmentation with object removal

We now present our data augmentation solution to reduce the sensitivity of classification and segmentation models to context distribution. The main idea is to expose these models to training images of object-without-context and context-without-object scenarios. This will help the models deal with the lack of contextual information and hence get more robust to context changes. For this, we perform object removal to create edited images with some objects removed and add these edited images to the training batch. Specific details of how to pick objects for removal and how to use them in training for the two tasks are discussed below.

**Classification.** We experiment with two strategies to use the edited images in the classifier training. In the first approach *Data-aug-rand*, a uniformly randomly sampled with uniform probability and the classifier is trained with simple binary cross-entropy loss using both original and edited images. Edited image is assigned the same labels as the same as the original image excluding the removed object class. In the second approach *Data-aug-const*, we explicitly optimize for robustness by including the inequality in (6.1) in the loss function. To do this, for randomly selected images in the training batch, we create the full edited image set  $\{I - c_i : c_i \in I\}$ . Then we can incorporate the robustness constraint as a hinge loss  $L_h$  with final loss being a weighted sum of the cross-entropy and the hinge losses.

$$\mathcal{L}_h(I) = \sum_{c_i \in I} \max \left[ 0, S_{c_i}(I - c_i) - \min_{c_j, j \neq i} S_{c_i}(I - c_j) \right] \quad (5)$$

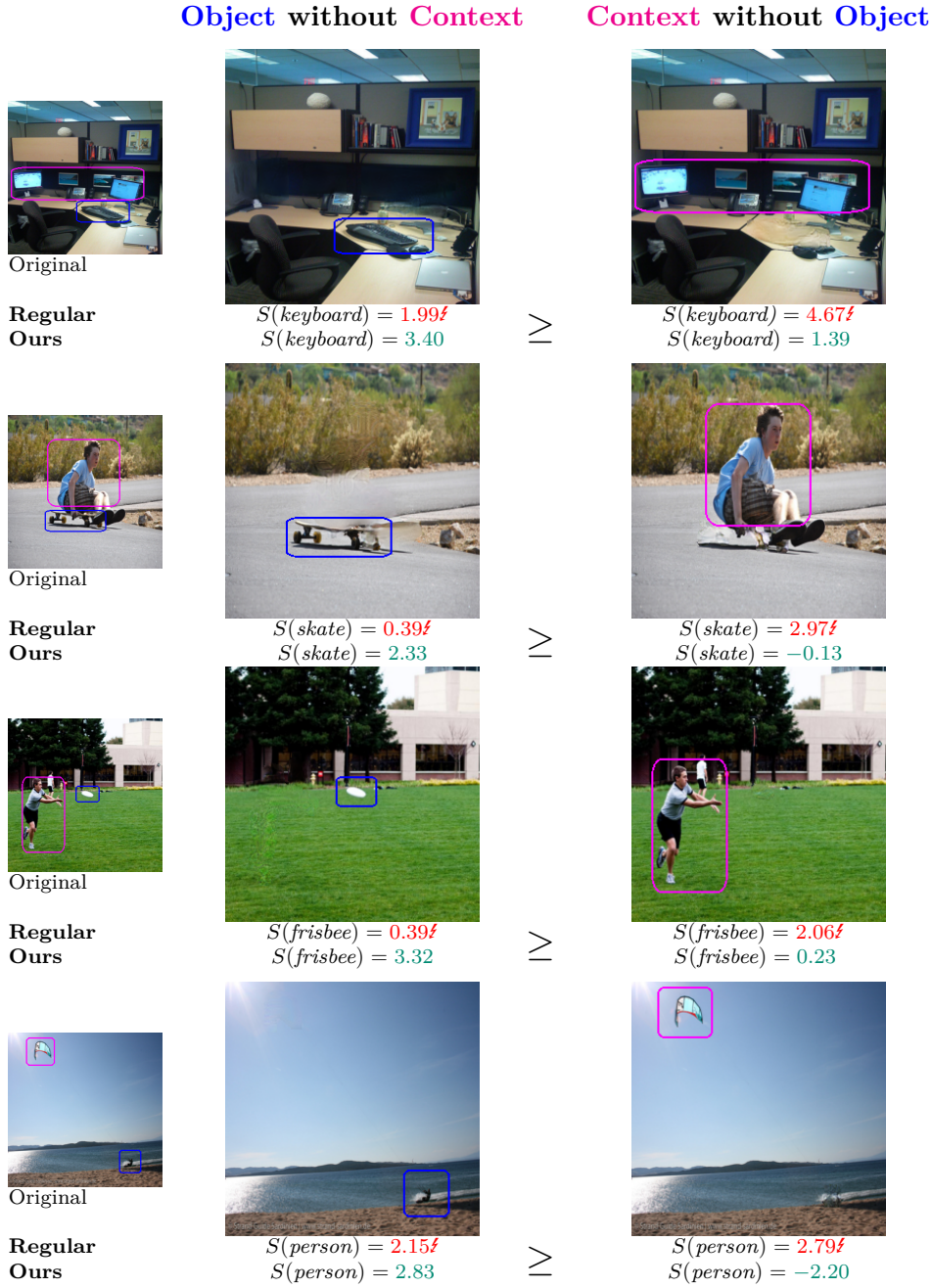


Figure 6.2: Context violations by image-level classifier. The primary object is marked with blue box and the context object is marked with magenta. The first column shows the original image, middle shows the image with only object and the third with only the context. We see that the baseline classifier depends heavily on the context and always scores the context only images (last column) higher than the image with only the primary object (middle column). The data augmented model does better and gets the ordering right.

**Segmentation.** We also perform data augmentation on the segmentation task by creating edited images with selected objects removed. The edited images can be used in training the segmentation model in two ways. First we can ignore the removed pixels and train the model to predict the original ground-truth labels on the rest of the image (*Ignore*). This helps the model learn that the labeling of a pixel should not be affected by the removal of a context object. Alternatively, we can explicitly tell the model that the removed object is not present by minimizing the likelihood assigned to the removed class at the edited pixels (*Negative loss*).

We explore three strategies to select objects to remove. The first strategy, *Random*, selects one random object to remove from the objects present in the image with uniform probability. However, sometimes the *Random* strategy can select very large object for removal, which can harm the quality of the edited image. To address this the *Sizebased* strategy selects objects based on their relative sizes in the image, assigning higher probability to smaller objects. The probability for picking an object is computed as  $p(c_i, I) \propto \left[ \frac{\sum_{c_j \in I} a(I, c_j)}{a(I, c_i)} \right]$  where  $a(I, c_i)$  is the area of the class  $c_i$  in image  $I$ . We also explore a hard negative mining based strategy, where we create harder training examples for the segmentation model by removing easy classes. This allows the model to focus on segmenting the harder classes while also becoming robust to context. Concretely, in *HardNegative* strategy we monitor the average cross-entropy loss  $l_{\text{avg}}(c_i)$  for an object class  $c_i$  and calculate the probability of removal of  $c_i$  as inversely proportional to  $l_{\text{avg}}(c_i)$ .

## 6.3 EXPERIMENTS AND RESULTS

This section presents the results of our analysis of how much the contextual information influences the performance of image classification and segmentation models. Using the robustness metrics defined in Section 6.2.2, we discover that the classification predictions on many well-performing classes are sensitive to context, and perform poorly on object-without-context and context-without-object images. Similar results are also found in the segmentation setting with the model depending heavily on context objects to correctly segment classes like *road*, *sidewalk*, *grass*. We also present results from our data-augmentation strategies, which help reduce this context dependence and improve robustness, without sacrificing performance.

### 6.3.1 Image level classification

#### 6.3.1.1 Experimental setup for classification

**Training data.** We run our classification experiments on the COCO dataset (Lin et al., 2014a), which contains 80 labeled object classes in their natural contexts. The dataset also has bounding box and segmentation annotation for each object. We use image-level labels to train the classifiers and use the object segmentation masks to test them with object removal.

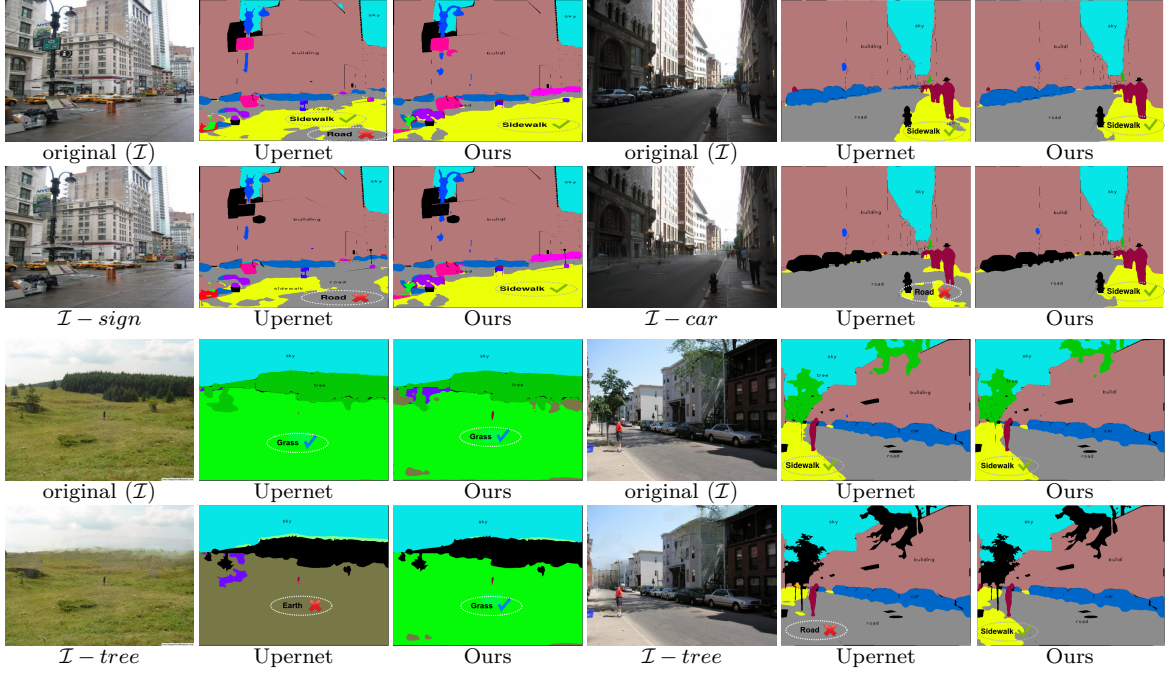


Figure 6.3: Examples of segmentation failures due to removal of a single context object. We see the segmentation of road, sidewalk and grass affected significantly when context objects like signboard, car and tree is removed (comparing odd and even rows). Model trained with proposed data-augmentation is more robust to these changes.

**Out-of-context testing.** Apart from testing the classifier models on regular COCO data we conduct additional experiments to quantify the performance in out-of-context scenarios with natural images. We divide the COCO images into two splits: the first split *Co-occur* with images having at least two objects in them and the second split *Single* with images containing a single object. The *Full* split is all images combining *Co-occur* and *Single*. The idea behind this splitting of the dataset is to separate out images where objects occur in their context (*Co-occur*) and images where object occur alone without the usual co-occurring context objects *Single*. Now we can train our models on the *Co-occur* split and test it on the *Single* split to measure, using only real images, how a classifier trained with only co-occurring objects performs when objects appear without the context seen in training. Additionally we also test our COCO trained models on the *UnRel* dataset (Peyre et al., 2017) which contains natural images with objects occurring in unusual contexts and relationships. We keep the classes which map to one of the 80 object classes in COCO, leaving 29 classes and 1071 images in the UnRel dataset.

**Baseline classifier.** The image-level classification model we test is based on the architecture proposed in Oquab et al. (2015). It consists of a Imagenet (Deng et al., 2009) pre-trained VGG-19 network for feature extraction network followed by two convolution layers, global max-pooling layer and a linear classification layer with sigmoid activations. The model is trained with binary cross-entropy loss. We train and test the model at single scale at 256x256 resolution, to simplify the analysis. Our classifier achieves similar mAP on real COCO data as reported in Oquab et al. (2015), with our



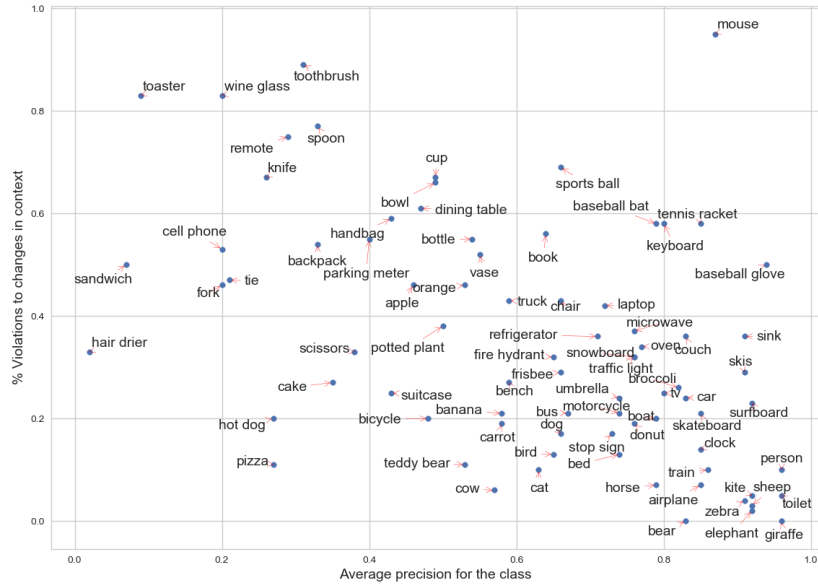


Figure 6.4: Comparing class-wise average precision to the % of violations in changes to context. Many well-performing categories (high mAP), have high percentage of violations, including mouse, tennis racket, keyboard, book, and sink.

Model	Training Data	COCO test set			Robustness Metrics		UnRel dataset $\uparrow$
		Full $\uparrow$	Co-occur $\uparrow$	Single $\uparrow$	$V^{\min}$ $\downarrow$	$V^{\text{mean}}$ $\downarrow$	
Baseline	Full (39k)	0.60	0.57	0.62	34%	24%	0.50
Data-aug-rand	Full (39k)	<b>0.61</b>	<b>0.58</b>	<b>0.65</b>	32%	22%	<b>0.54</b>
Data-aug-const	Full (39k)	0.60	<b>0.58</b>	0.63	<b>25%</b>	<b>14%</b>	0.52
Baseline	Co-occur (30k)	0.56	0.55	0.58	34%	24%	0.46
Data-aug-rand	Co-occur (30k)	<b>0.58</b>	<b>0.57</b>	<b>0.60</b>	31%	21%	0.49
Data-aug-const	Co-occur (30k)	<b>0.58</b>	<b>0.57</b>	<b>0.60</b>	<b>27%</b>	<b>15%</b>	<b>0.51</b>

Table 6.1: Effect of data augmentation on classification model

mAP slightly lower (0.600 vs 0.628 in [Oquab et al. \(2015\)](#)) due to single scale training and testing.

### 6.3.1.2 Analyzing classifier robustness to context

To measure the robustness of the trained classifier to context, we test it on real images and edited images and compute the robustness scores  $V^{\min}$  and  $V^{\text{mean}}$  as described in Section 6.2.2. Table 6.1 shows the robustness scores averaged over all classes computed on the COCO test along with the standard performance metric mean average precision (mAP) for the baseline classifier (first row). We can see that, despite achieving good mAP (0.6), the baseline classifier trained on full data performs poorly in-terms of robustness metrics. In about 34% of cases the model violates the context consistency requirement of (6.1). This means in 34% cases, the classifier scores images without the

Model	all (407 images)		with car (258)		without car (149)	
	Road	Sidewalk	Road	Sidewalk	Road	Sidewalk
Upernet	0.81	0.59	<b>0.86</b>	<b>0.67</b>	0.68	0.40
DataAug	<b>0.82</b>	<b>0.60</b>	<b>0.86</b>	0.65	<b>0.72</b>	<b>0.46</b>

Table 6.2: Comparing the performance of road and sidewalk segmentation on natural images with and without cars.

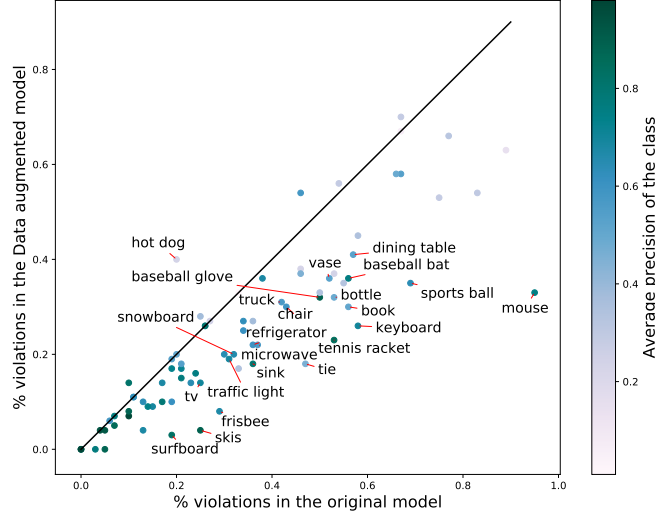


Figure 6.5: Comparing the % of violations in different classes with and without data augmentation. Points below the diagonal line show improvement with data-augmentation and the ones above degrade. The colors denote the average precision.

target object higher than an image where object is present but a context object has been removed. Comparing the per-class robustness score,  $V^{\min}(c_i)$  and the per-class average precision (AP) (see Figure 6.4 for visualization), we see that good performance in AP does not mean the classifier is robust to context. Many classes like mouse, keyboard, sink, tennis racket etc, which are performing well in AP ( $\geq 0.8$ ), but have poor robustness to changes in context ( $V_o^{\min} \geq 50\%$ ). In extreme case, the *mouse* classifier violates the consistency in more than 90% of cases, despite having very good AP (0.88). This indicates that the classifiers are relying too much on contextual evidence to detect the objects but perform poorly when tested on images where the context distribution is different from training.

We visualize the violations in Figure 6.2. In the first row we can see that the keyboard classifier scores the image with the keyboard removed higher (4.67) than the image with the keyboard but with the monitors removed (1.99). Similarly, we see the skateboard and the frisbee classifiers relying on person to hallucinate the respective objects. The violations shown in the first three rows of Figure 6.2 occur in objects with high co-occurrence dependence with other classes. However, context violations also occur in classes like *person* which appear in diverse contexts as seen in the last row of Figure 6.2. Here, the violation occurs in a difficult image where the *person* is small, but



a more distinct class with co-occurrence dependence on person is clearly visible (*kite*). The classifier uses the *kite* context to hallucinate that there is a *person*, even when the *person* has been removed.

### 6.3.1.3 Data augmentation to improve robustness

We train two variants of the data-augmented image classification models as described in Section 6.2.3. The first *Data-aug-rand* learns with standard cross-entropy loss on the edited images with a random object removed and the second *Data-aug-const* which is optimized directly for robustness using a set of edited images and hinge loss.

**Quantitative results.** We present the evaluation of the data-augmented and the baseline models in Table 6.1. On models trained with *Full* training data, the data-augmented model *Data-aug-rand* provides a small improvement in overall mAP on the COCO test set (0.61 vs 0.60). However measuring the performance on the two splits *Co-occur* and *Single* reveals that the improvement is significant on the *Single* split (0.65 vs 0.62), indicating that the data augmentation helps the classifier better deal with out of context objects. This is also seen when comparing the performance of the two models on the UnRel dataset, where *data-aug-rand* significantly improves over the baseline model (0.54 vs 0.50). This improved robustness of the data augmented classifier to context changes is also measured by our robustness metrics  $V^{\min}$  and  $V^{\text{mean}}$ . *Data-aug-rand* classifier makes overall 2% less violations under both worst-case ( $V^{\min}$ ) and average-case ( $V^{\text{mean}}$ ) context changes. Directly optimizing the robustness constraints allows the model *Data-aug-const* to significantly improve upon the baseline model in robustness metrics, while still obtaining improvement in the performance metrics. It exhibits much less worst-case (25% vs 34% for baseline) and average-case violations (14% vs 24% for baseline), while improving the performance in the UnRel dataset (0.52 mAP vs 0.50 for baseline). The benefit of optimizing for robustness is clearly seen when we constrain the training data to the *Co-occur* set, where the classifier never sees objects alone. Baseline model trained on the *Co-occur* set drops in performance on the *Single* (0.58 from 0.62 on when trained on *Full*) and the UnRel test sets (0.46 vs 0.50 with *Full*). However, with data augmentation and enforcing robustness constraints, we can recover some of this performance. On the *Single* test set *Data-aug-const* model trained on *Co-occur* set gets 0.58 mAP compared to 0.60 by baseline model trained on full data and even surpass it on the UnRel test set with 0.51 mAP. This shows that the data augmented model is able to overcome the contextual bias in the training set and perform well in unseen contexts.

When we compare the per-class robustness metrics between regular and data augmented models (data-aug-const), as shown in the Figure 6.5, we see that data-augmentation significantly reduces the worst case violations ( $V^{\min}$ ) on well-performing classes. For example,  $V^{\min}$  drops from 95% to less 36% for the mouse class and from 58% to 28% for the keyboard class. The effect of this increased robustness is seen in qualitative examples in Figure 6.2. In the first row, the baseline keyboard classifier gives too much weight to evidence from *monitor* and scores the image with only *monitor* higher than the image with only keyboard. However, the data augmented model correctly

orders the two images.

### 6.3.2 Semantic segmentation

So far, we have seen that multi-label classification models suffer from sensitivity to context, with classifiers often mixing up contextual and visual evidence. Next we will measure the context sensitivity of models in a more local and strongly supervised task of semantic segmentation.

#### 6.3.2.1 *Experimental setup for segmentation*

**Training and test data.** We conduct our semantic segmentation experiments primarily on the ADE20k dataset (Zhou *et al.*, 2017) containing 140 categories of labeled objects, in different settings. Some of the 140 classes are typical background classes like *sky*, *sea* and *wall* and are large and difficult to in-paint and are hence excluded from removal.

**Out-of-context testing.** Following the process in image-level classification, we also measure the performance of the segmentation models on real out-of-context data. This is done in two ways. First, we train the segmentation model in a restricted setting with only three classes *car*, *road* and *sidewalk*. Now, we can again make two splits of the training and testing images into the *Co-occur* split of images with at-least two objects (3317 images) and the *single* split with only a single object (1693 images). Then we train the segmentation models on *co-occur* split and test on *single* split to see how well it can perform segmentation without context. Additionally we also test the models trained with ADE20k data on the Pascal-context dataset (Mottaghi *et al.*, 2014) in order to measure the performance under a different context distribution. This is done by manually mapping the 59 labels in the pascal-context to ADE20k labels and restricting the segmentation model to produce only the mapped labels.

**Baseline segmentation model.** We use the recent UperNet (Xiao *et al.*, 2018b) model, with good results on the ADE20k, as our baseline segmentation model. We train the variant with the Resnet-50 encoder and a Upernet decoder with batch size of 6 images (maximum that fit in GPU) and with the default hyper-parameters suggested by the authors. This model achieves mean intersection-over-union (mIoU) of 0.377 and accuracy of 78.19% with single scale testing.

#### 6.3.2.2 *Context in semantic segmentation*

We analyze robustness of the segmentation models to context by removing objects and computing the matrix  $AR(c_i, c_j)$  presented in Section 6.2.2, which measures the % of images where removal of object  $c_j$  significantly affects segmentation of object  $c_i$ . The matrix  $AR(c_i, c_j)$  we obtain for the Upernet model in ADE20k dataset is a sparse matrix with sharp peaks. This indicates that the classes depend on specific context objects and are significantly affected by their removal. The sparsity also indicates that the effects on the segmentation are due the class being removed and not in-painting artifacts (otherwise the segmentation would be affected by all removal). Some of dependencies we discover in

Model	Removed pixels	mIoU	Acc
Upernet(Xiao <i>et al.</i> , 2018b)	-	0.377	78.31
DA (random)	Ignore	0.320	75.2
DA (sizebased)	Ignore	0.379	78.31
DA (hard negative)	Ignore	0.375	77.8
DA (sizebased)	Negative	0.377	78.25
DA (hard negative)	Negative	<b>0.385</b>	<b>78.47</b>

Table 6.3: Data augmentation results on ADE20k dataset.

$AR(c_i, c_j)$  are reasonable and harmless, for example between *pot* and *plant* ( $AR = 50\%$ ). Once you remove the *plant*, *pot* looks more like a *trash can* and the segmentation model often flips the label to *trash can*. However other dependencies are spurious and not desirable. For example, we notice that often the segmentation model uses presence of *car* to differentiate between *road* and *sidewalk*. Removing *car* affects the IoU of the *road* and *sidewalk* in 21% and 22% of cases respectively. This dependence is undesirable, and can be catastrophic in applications like self-driving cars.

We show qualitative examples where removal affects segmentation of Upernet model in Figure 6.3. The first two rows show the cases where removal of an object negatively impacts the segmentation of other objects. This include cases where removal of *street sign* and *car* severely affects segmentation of *road* and *sidewalk*, and a case where removal of *trees* affects segmentation of *grass*. We can see from these examples that while edit on the image is small and local, the effects of this removal on segmentation prediction is not local. Removal of a small objects can have drastic effects on segmentation in a far-away region.

### 6.3.2.3 Data augmentation for segmentation

Next we will look at the results of using data-augmentation for segmentation models. For this purpose we train the Upernet (Xiao *et al.*, 2018b) based data-augmented models on the ADE-20k dataset with on three different strategies for selecting the object to remove as discussed in Section 6.2.3.

**Quantitative results.** Table 6.3, shows the results comparing the data-augmented models with the baseline Upernet model. We can see that random sampling strategy, which worked well in image classification, fails here leading to drop in performance. This is because, many object categories in ADE20k dataset are large and difficult to remove like bed, sofa and mountain and random strategy suffers by picking these. Instead when we switch to size-based and hard-negative based sampling, we see that the performance improves and the the size-based sampling model achieves the best mIoU of the three models (0.379). Applying negative likelihood loss on the removed object class gets further improvement when combined with hard negative sampling. This model also improves upon the Upernet baseline (achieving 0.385 IoU vs 0.377 by Upernet), despite the fact that the removal based data-augmentation is designed to make the model more robust to contextual variations.

To understand how data-augmentation impacts sensitivity to context, Figure 6.6 visualizes the maximum sensitivity of a class to removal of other classes,  $\max_{c_j} AR(c_i, c_j)$  for different classes with and without data-augmentation. We see that for majority of classes robustness to context improves with data augmentation. For example *pillow* class is only affected 32% of the time with context changes, compared to 53% before data augmentation. Similarly, *road* and *sidewalk* classes are only affected 9% and 14% of the time respectively, compared to 21% and 22% before. This improved robustness translates into better generalization to real out-of-context data. We can see this in Table 6.2 where the performance of the *road* and *sidewalk* segmentation is measured on the validation set on images with and without cars. On the full set and on the split with cars, we see that the performance of the baseline Upernet and our augmented model (DA hard negative with negative loss) is equivalent. However, when we look at only images without car, the Upernet model performs significantly worse in both road (0.68 vs 0.72 for ours) and sidewalk (0.40 vs 0.46 for ours) segmentation. This quantitatively shows that the baseline model struggles to distinguish between *road* and *sidewalk* without *car* in the image, whereas our data augmentation is more robust and performs well even without context (car).

We also see the benefit of data augmentation in experiments on restricted *Co-occur* training set and on the Pascal-context dataset. Our data augmented model outperforms the Upernet model (both trained on the ADE20k dataset) when tested on the Pascal-context dataset in both mIoU and pixel accuracy. While the Upernet model achieves mIoU of 0.284 and pixel accuracy of 61.3% our data augmented model achieves 0.293 and 62.10% respectively, indicating that it is able to generalize better when tested on a dataset with different context distribution than one seen during training. Table 6.4 presents the experiments with the *Co-occur* training set in the three class setting. First we can see that when we switch from training on *Full* training data to *Co-occur* split (containing only images with atleast two objects), the performance of the Upernet greatly drops on the *Single* test split (from 0.67 to 0.52). This indicates that the model overfits to the context it sees, and is not able to segment objects when it seeing them out of context. However, with data-augmentation we generate images of objects without context, and can recover most of this performance loss (0.646). Surprisingly, data-augmented model trained on smaller *co-occur* data also outperforms the baseline trained with *Full* data when tested on the *co-occur* split.

Qualitative examples in Figure 6.3 also show the effect of increased robustness to context. While the baseline Upernet model is affected by context object removal causing drastic changes in predictions of other regions, our data augmented model is more stable. For example the removal of *signboard*, *car* or *tree* does not effect the segmentation of the *road* or *sidewalk* by our model.

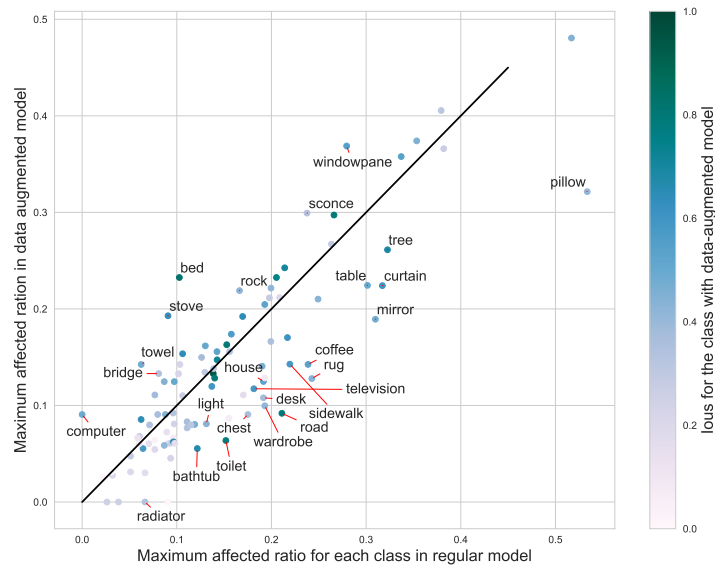


Figure 6.6: Comparing the context sensitivity of different classes with and without data augmentation with  $\max_{c_j} AR(c_i, c_j)$  metric. Points below the diagonal improve with data-augmentation. The color denotes the mIoU.

Model	Training Data	Full	Only Cooccur	Only Single
Upernet	Full (5k)	0.774	0.797	0.670
Data Aug	Full (5k)	0.742	0.754	<b>0.675</b>
Upernet	Co-occur (3.3k)	0.680	0.713	0.520
Data Aug	Co-occur (3.3k)	<b>0.82</b>	<b>0.86</b>	0.646

Table 6.4: Experiments in three class setting on ADE20k.

## 6.4 CONCLUSIONS

We have presented a methodology to analyze and quantify context sensitivity of image classification and segmentation models, based on editing images to remove objects and measuring the effect on the target model output. Our analysis shows that despite good performance in-terms of mAP, classifiers are, for certain classes like keyboard, mouse, skateboard, very sensitive to context objects and perform poorly when presented with images that are out of context. In semantic segmentation, our analysis shows similar dependency between classes. For example, we discover that the model depends on the presence of a car to segment roads and sidewalks and fails drastically when the car is not present in the image. We present a data augmentation scheme based on object removal to mitigate this and make the classification and segmentation models more robust to context changes. Our experiments show that the proposed data augmentation helps models generalize to out of context scenarios without losing performance in the i.i.d. settings, indicating that the data augmented models better balance contextual and visual information.



## Contents

7.1	Introduction . . . . .	104
7.2	Synthetic Dataset for Variances and Invariances in VQA . . . . .	105
7.2.1	InVariant VQA (IV-VQA) . . . . .	106
7.2.2	CoVariant VQA (CV-VQA) . . . . .	108
7.3	Experiments: Consistency Analysis . . . . .	108
7.4	Robustification by Data Augmentation . . . . .	113
7.5	Conclusions . . . . .	115



IN this chapter, we employ the similar analysis to Chapter 6, to visual question answering models. Applying object removal approach from Chapter 5 to create different versions of the input image, we test if visual question answering models are consistent in their response. While prior works have exposed brittleness of VQA models variations in the question language, this is the first effort to understand their robustness to visual variations. In contrast to Chapter 6, we also consider edits which alter the ground-truth answer in a predictable fashion, and test if the VQA model responds to the change accordingly. We perform our analysis on three diverse, state of the art VQA models and diverse question types with a particular focus on challenging counting questions. In addition, we show that models can be made significantly more robust against inconsistent predictions by augmenting the training set with our edited data.





## 7.1 INTRODUCTION

VQA allows interaction between images and language, with diverse applications such as interacting with chat bots to assisting visually impaired people. In these applications we expect a model to answer truthfully and based on the evidence in the image and the actual intention of the question. Unfortunately, this is not always the case even for state of the art methods. Instead of “sticking to the facts”, models frequently rely on spurious correlations and follow biases induced by data and/or model. For instance, recent works [Shah et al. \(2019\)](#); [Ray et al. \(2019\)](#) have shown that the VQA models are brittle to linguistic variations in questions/answers. Shah et al. in [Shah et al. \(2019\)](#) introduced VQA-Rephrasings dataset to expose the brittleness of the VQA models to linguistic variations and proposed cyclic consistency to improve their robustness. They show that if a model answers “Yes” to the question: “Is it safe to turn left?”, it answers “No” when the question is rephrased to “Can one safely turn left?”. Similarly Ray et al. in [Ray et al. \(2019\)](#) introduced ConVQA to quantitatively evaluate the consistency for VQA towards different generated entailed questions and proposed data augmentation module to make the models more consistent.

While previous works have studied linguistic modifications, our contribution is the first systematic study of automatic visual content manipulations at scale. Analogous to rephrasing questions for VQA, images can also be semantically edited to create different variants where the same question-answer (QA) pair holds. One sub-task of this broader semantic editing goal is object removal. One can remove objects in such a way that the answer remains invariant (wherein only objects irrelevant to the QA are removed) as shown in Figure 7.1 (top/middle). Alternately one could also make covariant edits where we remove the object mentioned in the QA and hence expect the

Q: Is this a kitchen?				
A: no		toilet removed; A: no		
				
	Baseline	Ours	Baseline	Ours
CL	no	no	yes	no
SAAA	no	no	no	no
SNMN	no	no	yes	no

---

Q: What color is the balloon?				
A: red		umbrellas removed; A: red		
				
	Baseline	Ours	Baseline	Ours
CL	pink	red	red	red
SAAA	pink	red	red	red
SNMN	pink	red	red	red

---



Q: How many zebras are there in the picture?				
A: 2		zebra removed A: 1		
				
	Baseline	Ours	Baseline	Ours
CL	2	2	2	1
SAAA	2	2	2	1
SNMN	2	2	2	1

Figure 7.1: VQA models change their predictions as they exploit spurious correlations rather than causal relations based on the evidence. Shown above are predictions of 3 VQA models on original and synthetic images from our proposed IV-VQA and CV-VQA datasets. ‘Ours’ denote the models robustified with our proposed data augmentation strategy.

answer to change in a predictable manner as shown in Figure 7.1 (bottom). We explore both invariant and covariant forms of editing and quantify how consistent models are under these edits.

We employ a GAN-based (Shetty *et al.*, 2018a) re-synthesis model to automatically remove objects. Our data generation technique helps us create exact complementary pairs of the image as shown in Figures 7.1, 7.2. We pick three recent models which represent different approaches to VQA to analyze robustness: a simple CNN+LSTM (CL) model, an attention-based model (SAAA Kazemi and Elqursh (2017)) and a compositional model (SNMN Hu *et al.* (2018)). We show that all the three models are brittle to semantic variations in the image, revealing the false correlation that the models exploit to predict the answer. Furthermore, we show that training data augmentation with our synthetic set can improve models robustness.

Our motivation to create this complementary dataset stems from the desire to study how accurate and consistent different VQA models are and to improve the models by the generated ‘complementary’ data (otherwise not available in the dataset). While data augmentation and cyclic consistency are making the VQA models more robust (Kafle *et al.*, 2017; Ray *et al.*, 2019; Shah *et al.*, 2019) towards the natural language part, we take a step forward to make the models consistent to semantic variations in the images. We summarize our main contributions as follows:

- We propose a novel approach to analyze and quantify issues of VQA models due to spurious correlation and biases of data and models. We use synthetic data to quantify these problems with a new metric that measures erroneous inconsistent predictions of the model.
- We contribute methodology and a synthetic dataset <sup>4</sup> that complements VQA datasets by systematic variations that are generated by our semantic manipulations. We complement this dataset by a human study that validates our approach and provides additional human annotations.
- We show how the above-mentioned issues can be reduced by a data augmentation strategy - similar to adversarial training. We present consistent results across a range of questions and three state of the art VQA methods and show improvements on synthetic as well as real data.
- While we investigate diverse question types, we pay particular attention to counting by creating an covariant edited set and show that our data augmentation technique can also improve counting robustness in this setting.

## 7.2 SYNTHETIC DATASET FOR VARIANCES AND INVARIANCES IN VQA

While robustness w.r.t linguistic variations (Shah *et al.*, 2019; Ray *et al.*, 2019) and changes in answer distributions (Agrawal *et al.*, 2018) have been studied, we explore how robust VQA models are to semantic changes in the images. For this, we create

---

<sup>4</sup><https://rakshithshetty.github.io/CausalVQA/>

a synthetic dataset by removing objects irrelevant and relevant to the QA pairs and propose consistency metrics to study the robustness. Our dataset is built upon existing VQAv2 (Goyal *et al.*, 2017) and MS-COCO (Lin *et al.*, 2014b) datasets. We target the 80 object categories present in the COCO dataset (Lin *et al.*, 2014b) and utilize a GAN-based (Shetty *et al.*, 2018a) re-synthesis technique to remove them. The first key step in creating this dataset is to select a candidate object for removal for each Image-Question-Answer (IQA) pair. Next, since we use an in-painter-based GAN, we need to ensure the removal of the object does not affect the quality of the image or QA in any manner. We introduce vocabulary mapping to take care of the former and area-overlapping criteria for the latter. We discuss these steps in detail to generate the edited set in irrelevant removal setting and later extend these to relevant object removal.

### 7.2.1 InVariant VQA (IV-VQA)

For the creation of this dataset, we select and remove the objects irrelevant to answering the question. Hence the model is expected to make the same predictions on the edited image. A change in the prediction would expose the spurious correlations that the model is relying on to answer the question. Some examples of the semantically edited images along with the original images can be seen in Figures 7.1, 7.2. For instance, in Figure 7.2 (top-right), for the question about the color of the surfboard, removing the person should not influence the model’s prediction. In order to generate the edited image, we first need to identify person as a potential candidate which in turn requires studying the objects present in the image and the ones mentioned in the QA. Since we use VQA v2 dataset (Goyal *et al.*, 2017), where all the images overlap with MS-COCO (Lin *et al.*, 2014b), we can access the ground-truth bounding box and segmentation annotations for each image. In total, there are 80 different object classes in MS-COCO which become our target categories for removal.

**Vocabulary mapping.** To decide if we can remove an object, we need to first map all the object referrals in question and answer onto the 80 COCO categories. These categories are often addressed in the QA space by many synonyms or a subset representative of that class. For example- people, person, woman, man, child, he, she, biker all refer to the category: ‘person’; bike, cycle are commonly used for the class ‘bicycle’. To avoid erroneous removals, we create an extensive list mapping nouns/pronouns/synonyms used in the QA vocabulary to the 80 COCO categories. Table 7.1 shows a part of the object mapping list. The full list can be found in code-release for the project <sup>5</sup>.

Let  $O_I$  represent the objects in the images (known via COCO segmentations),  $O_{QA}$  represent the objects in the question-answer (known after vocabulary mapping). Then our target object for removal,  $O_{target}$ , is given by  $O_I - \{O_I \cap O_{QA}\}$ . We assume that if the object is not mentioned in the QA, it is not relevant and hence can be safely removed.

**Area-Overlap threshold.** The next step is to make sure that the removal of  $O_{target}$  does not degrade the quality of the image or affect the other objects mentioned in the

---

<sup>5</sup><https://github.com/AgarwalVedika/CausalVQA>

COCO categories	Additional words mapped
person	man, woman, player, child, girl, boy people, lady, guy, kid, he etc
fire hydrant	hydrant, hydrate, hydra
wine glass	wine, glass, bottle, beverage, drink
donut	doughnut, dough, eating, food, fruit
chair	furniture, seat
...	...

Table 7.1: Example of vocabulary mapping from QA space to COCO categories. If any of these words (in the right column) occur in the QA, these words are mapped to the corresponding COCO category (in the left column).

QA. Since we use an in-painter based GAN (Shetty *et al.*, 2018a), we find that larger object removal is harder to in-paint leaving the images heavily distorted. In order to avoid such distorted images, we only remove the object if the area occupied by its largest instance is less than 10% of the image area. Furthermore, we also consider if the object being removed overlaps in any manner with the object that is mentioned in the QA. We quantitatively measure the overlap score as shown in Equation 7.1 where  $M_O$  denotes the dilated ground truth segmentation mask of all the instances of the object. We only remove the object if the overlap score is less than 10%.

$$\text{Overlap score}(O_{\text{target}}, O_{\text{QA}}) = \frac{(M_O)^{\text{target}} \cap (M_O)^{\text{QA}}}{(M_O)^{\text{QA}}} \quad (7.1)$$

**Uniform Ground-Truth.** Finally, we only aim to target those IQAs which have uniform ground-truth answers. In VQA v2 (Goyal *et al.*, 2017), all the questions have 10 answers, while it is good to capture diversity in open-ended question-answering, it also introduces ambiguity, especially in case of counting and binary question types. To avoid this ambiguity in our robustness evaluation, we build our edited set by only selecting to semantically manipulate those IQs which have a uniform ground truth answer.

Finally, we remove all the instances of the target object from the image for those IQAs which satisfy the above criteria using the inpainter GAN (Shetty *et al.*, 2018a). We call our edited set as IV-VQA as removal of objects does not lead to any change in answer, the answer is invariant to the semantic editing. Table 7.2 shows the number of edited IQAs in IV-VQA. While our algorithm involves both manually curated heuristics to select the objects to remove, and a learned in-painter-based GAN model to perform the removal, the whole pipeline is fully automatic. This allows us to apply it to the large-scale VQA dataset with 658k IQA triplets.

**Validation by Humans.** We get a subset (4.96k IQAs) of our dataset validated by three humans. The subset is selected based on an inconsistency analysis of 3 models covered in the next Section 7.3. Every annotator is shown the edited IQA and is asked to say if the answer shown is correct for the given image and question (yes/no/ambiguous). According to the study, 91% of the time all the users agree that our edited IQA holds.

#IQA	IV-VQA			CV-VQA		
	train	val	test	train	val	test
real	148013	7009	63219	18437	911	8042
realNE	42043	2152	18143	13035	648	5664
edit	256604	11668	108239	8555	398	3743

Table 7.2: IV-VQA and CV-VQA distribution. Real refers to VQA (Goyal *et al.*, 2017) IQAs with uniform answers, realNE refers to IQAs for which no edits are possible (after vocabulary mapping and area-overlap threshold), edit refers to the edited IQA. We split the VQA val into 90:10 ratio, where the former is used for testing purpose and latter for validation.

### 7.2.2 CoVariant VQA (CV-VQA)

An alternate way of editing images is to target the object in the question. Object-specific questions like counting, color, whether the object is present or not in the image are suitable for this type of editing. We choose counting questions where we generate complementary images with one instance of the object removed. If the model can count  $n$  instances of an object in the original image, it should also be able to count  $n - 1$  instances of the same object in the edited image. Next, we will describe how to generate this covariant data for counting.

First, we collect all the counting questions in the VQA set: selecting questions which contained words ‘many’ and ‘number of’ and which had numeric answers. Next, we focus on removing instances of the object which is to be counted in the question. Vocabulary mapping is used to identify the object mentioned in the question  $O_Q$ . Then only those images are retained where the number of the target object instances according to COCO segmentations match the IQA ground-truth answer  $A$  given by 10 human annotators.

For the generation of this set, we use the area threshold as 0.1, we only intend to remove the instance if it occupies less than 10% of the image. Furthermore for overlap, since we do not want the removed instance to interfere with the other instances of the object, two masks considered to measure the score are: (1). dilated mask of instance to be removed (2). dilated mask of all the other instances of the object. The object is only removed if the overlap score is zero.

We call our edited set as CV-VQA since removal of the object leads to a covariant change in answer. Table 7.2 shows the number of edited IQAs in VQA-CV. Figure 7.2 (bottom row) shows a few examples from our edited set. We only target one instance at a time.

## 7.3 EXPERIMENTS: CONSISTENCY ANALYSIS

The goal of creating edited datasets is to gauge how consistent are the models to semantic variations in the images. In IV-VQA, where we remove objects irrelevant to the QA from the image, we expect the models predictions to remain unchanged. In CV-VQA,




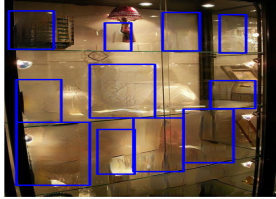


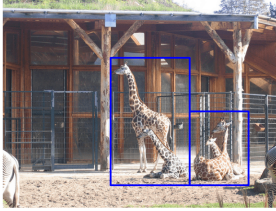

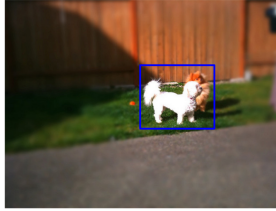
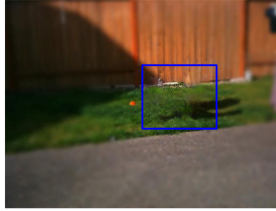



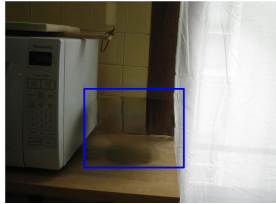
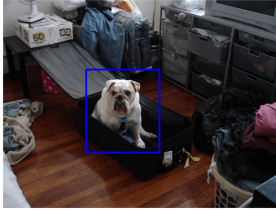
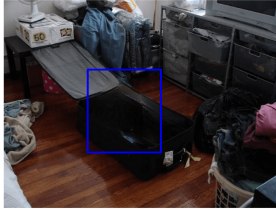
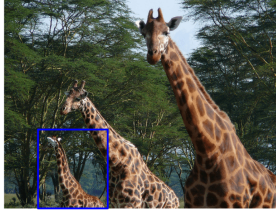
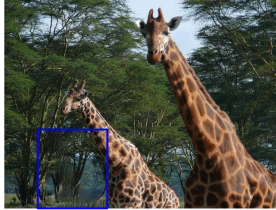
pos→neg			neg→pos		
Q: What are the shelves made of?			Q: What color is the surfboard?		
A: glass		<i>vases removed; A: glass</i>	A: white		<i>person removed; A: white</i>
					
CNN+LSTM	glass	wood	CNN+LSTM	yellow	white
SAAA	glass	metal	SAAA	white	white
SNMN	glass	metal	SNMN	yellow	white
Q: Are there zebras in the picture?			Q: Is there a cat?		
A: yes		<i>giraffes removed; A: yes</i>	A: no		<i>dogs removed; A: no</i>
					
CNN+LSTM	yes	no	CNN+LSTM	yes	no
SAAA	yes	no	SAAA	yes	no
SNMN	yes	no	SNMN	yes	no
Q: What sport is he playing?			Q: What room of a house is this?		
A: soccer		<i>sports-ball; A: soccer</i>	A: kitchen		<i>bowl; A: kitchen</i>
					
CNN+LSTM	soccer	tennis	CNN+LSTM	bathroom	kitchen
SAAA	soccer	tennis	SAAA	bathroom	kitchen
SNMN	soccer	tennis	SNMN	bathroom	kitchen
Q: How many dogs are there?			Q: How many giraffe are there?		
A: 1		<i>dog removed; A: 0</i>	A: 3		<i>giraffe removed; A: 2</i>
					
CNN+LSTM	1	2	CNN+LSTM	1	2
SAAA	1	1	SAAA	2	2
SNMN	1	1	SNMN	2	2

Figure 7.2: Existing VQA models exploit spurious correlations to predict the answer often looking at irrelevant objects. Shown above are the predictions for 3 different VQA models on original and edited images from our synthetic datasets IV-VQA and CV-VQA.

Trained by us		For comparison	
CNN+LSTM	53.32	d-LSTM Q + norm I (Lu <i>et al.</i> , 2015)	51.61
SAAA	61.14	SAN (Yang <i>et al.</i> , 2016)	52.02
		HieCoAttn (Lu <i>et al.</i> , 2016)	54.57
		MCB (Fukui <i>et al.</i> , 2016)	59.71
SNMN	58.34	NMN (Andreas <i>et al.</i> , 2016)	51.62

Table 7.3: Accuracy (in %) of different models when trained on VQA v2 train and tested on VQA v2 val.

where one of the instances to be counted is removed, we expect the predicted answer to reduce by one as well. Next, we briefly cover the models’ training and then study their performances both in terms of accuracy and consistency. We propose consistency metrics based on how often the models flip their answers and study the different type of flips qualitatively and quantitatively.

**VQA models and training.** For comparison and analysis, we select three models from the literature, each representing a different design paradigm: a simple CNN+LSTM (CL) model, an attention-based model (SAAA (Kazemi and Elqursh, 2017)) and a compositional model (SNMN (Hu *et al.*, 2018)). We use the official code for training the SNMN (Hu *et al.*, 2018) model, Hu (2018). SAAA (Kazemi and Elqursh, 2017) is trained using the code available online (Zhang, 2017). We modified this SAAA code in order to get CL model by removing the attention layers from the network. As we use the VQA v2 val split for consistency evaluation and testing, the models are trained using only the train split. Table 7.3 shows the accuracy scores on VQA v2 val set for models trained by us along with similar design philosophy models benchmarked in Agrawal *et al.* (2018) and Goyal *et al.* (2017). The models chosen by us exceed the performance of other models within the respective categories.

**Consistency.** The edited data is created to study the robustness of the models. Since we modify the images in controlled manner, we expect the models predictions to stay consistent. Robustness is quantified by measuring how often models change their predictions on the edited IQA from the prediction on original IQ. On IV-VQA, a predicted label is considered “flipped” if it differs from the prediction on the corresponding unedited image. On CV-VQA, if the answer on the edited samples is not one less than the prediction on original image, it is considered to be “flipped”.

We group the observed inconsistent behavior on edited data into three categories: 1. neg→pos 2. pos→neg 3. neg→neg. neg→pos flip means that answer predicted on the edit IQA was correct but the prediction on the corresponding real IQA was wrong. Other flips are defined analogously. In the neg→neg flip, answer predicted is wrong in both the cases. While all forms of label flipping show inconsistent behaviour, the pos→neg and neg→pos categories are particularly interesting. In these the answer predicted is correct before and afterward the edit, respectively. These metrics show that there is brittleness even while making correct predictions and indicate that models exploit spurious correlations while making their predictions.



	CL (%)	SAAA (%)	SNMN (%)
Accuracy orig	60.21	70.26	66.04
Predictions flipped	17.89	7.85	6.52
pos→neg	7.44	3.47	2.85
neg→pos	6.93	2.79	2.55
neg→neg	3.53	1.58	1.12

Table 7.4: Accuracy-flipping on real data/IV-VQA test set.

**Quantitative analysis.** Table 7.4 shows the accuracy along with the consistency numbers for all the 3 models on the IV-VQA test split. Consistency is measured across edited IV-VQA IQAs and corresponding real IQAs from VQA v2. Accuracy is reported on real data from VQA v2 (original IQAs with uniform answers). We follow this convention throughout our analysis. On the original data, we see that SAAA is the most accurate model (70.3%) as compared to SNMN (66%) and CL (60.2%). In terms of robustness towards the variations in the images, CL model is the least consistent with a 17.9% flipping on the edit set compared to the predictions on the corresponding original IQA. For SAAA, 7.85% flips, making SNMN the most robust model with 6.522% flips. SAAA and SNMN are much more stable than CL. A point noteworthy here is that SNMN turns out to be the most robust despite its accuracy being lesser than SAAA. This shows that higher accuracy does not necessarily mean we have the best model, further highlighting the need to study and improve the robustness of the models. Of particular interest are the pos→neg and neg→pos scores, which are close to 7% each for the CL model. For a neg→pos flip, the answer to change from an incorrect answer to one correct answer of the 3000 possible answers (size of answer vector). If the removed object was not used by the model, as it should be, and editing caused uniform perturbations to the model prediction, this event would be extremely rare ( $p(\text{neg} \rightarrow \text{pos}) = 1/3000 * 39.8 = 0.013\%$ ). However we see that this occurs much more frequently (6.9%), indicating that in these cases model was spuriously basing its predictions on the removed object and thus changed the answer when this object was removed.

In the CV-VQA setting, where we target counting and remove one instance of the object to be counted, we expect the models to maintain  $n/n-1$  consistency on real/edited IQA. As we see from Table 7.5, the accuracy on orig set is quite low for all the models reflecting the fact that counting is a hard problem for VQA models. SAAA (49.9%) is the most accurate model with SNMN at 47.9% and CL at 39.4%. In terms of robustness, we see that for all 3 models are inconsistent more than 75%, meaning for  $>75\%$  for the edited IQAs, if models could correctly count  $n$  objects in the original IQA, it wasn't able to count  $n-1$  instances of the same object in the edited IQA. These numbers further reflect that counting is a difficult task for VQA models and enforcing consistency on it seems to break all 3 models. In the next section, we discuss these flips with some visual examples.

**Qualitative analysis.** We visualize the predictions of the models on a few original and edited IQAs for all the 3 models in Figure 7.2. The left half shows examples of pos→neg

	CL (%)	SAAA (%)	SNMN (%)
Accuracy orig	39.38	49.9	47.948
Predictions flipped	81.41	78.44	78.92
pos→neg	28.69	31.66	32.35
neg→pos	20.57	25.38	23.51
neg→neg	32.14	21.4	23.06

Table 7.5: Accuracy-flipping on real data/CV-VQA test set.

and the right half shows the neg→pos flips. Existing VQA models often exploit false correlations to predict the answer. We study the different kinds of flips in detail here and see how they help reveal these spurious correlations.

**pos→neg.** VQA models more often rely on the contextual information/ background cues/ linguistic priors to predict the answer rather than the actual object in the question. For instance, removal of the glass vases from the shelves in Figure 7.2 (Top-left) from the image causes all 3 models to flip their answers negatively, perhaps models were looking at the wrong object (glass vases) to predict the material of the shelves that also happened to be glass. In absence of giraffes, models cannot seem to spot the occluded zebras anymore- hinting that maybe they are confusing zebras with giraffes. Removing the sports-ball from the field make all 3 models falsely change their predictions to tennis without considering the soccer field or the players. In the bottom-left, we also see that if models were spotting the one dog rightly in the original image, on it’s edited counterpart( with no dog anymore )- it fails to answer 0. Semantic edits impact the models negatively here exposing the spurious correlations being used by the models to predict the correct answer on the original image. These examples also show that accuracy should not be the only sole criterion to evaluate performance. A quick look at the Table 7.4 show that for IV-VQA, pos→neg flips comprise a major chunk (>40%) of all the total flips. For CV-VQA (refer Table 7.5) , these flips are 28-32% absolute- again reinforcing the fact that VQA models are far from learning to count properly.

**neg→pos.** Contrary to above, semantic editing here helps correct the predictions, meaning removal of the object causes the model to switch its wrong answer to one right answer by getting rid of the wrong correlations. For instance, removing the pink umbrella helps models predict correctly the color of the balloon Figure 7.1 (middle). In Figure 7.2 (second-right), removing the dogs leave no animals behind and hence models now can correctly spot the absence of cat- hinting that they were previously confusing cats and dogs. In absence of the bowl, models can identify the room as kitchen- shows that too much importance is given to the bowl (which is falsely correlated to bathroom) and not to the objects in the background such as microwave. Towards the bottom-right, we see that removing a giraffe helps all the 3 models now- it’s hard to say what is the exact reason for the behaviour but it indeed reflects upon the inconsistent behaviour of the models. From Table 7.4 we see that these flips also comprise a significant number of the total flips (>35%) for all the models. For CV-VQA (refer Table 7.5), these numbers are in range 20-25%, showing that counting is easier for these models when spurious

correlations are removed.

**neg→neg.** These flips where answers change show the inconsistent behavior of models as well but since both the answers are wrong- they are harder to interpret. But in the end goal of building robust models, we expect consistent behavior even when making incorrect predictions.

All these flips show that existing VQA models are brittle to semantic variations in images. While VQA models are getting steadily better in terms of accuracy, we also want our models to be robust to visual variations. We want VQA models to not just be accurate but use the right cues to answer correctly. Accuracy combined with consistency can help us understand the shortcomings of the models.

## 7.4 ROBUSTIFICATION BY DATA AUGMENTATION

In the previous section, we see that VQA models are brittle to semantic manipulations. While these flips expose the inconsistent behaviour, they also show the underlying scope of improvement for VQA models and can be used to make the models more robust. In order to leverage the variances brought in by the synthetic data, we finetune all the models using real and real+synthetic data. Our analysis shows that using synthetic data significantly reduces inconsistency across a variety of question types.

For fine-tuning experiments, we use a strict subset of IV-VQA with an overlap score of zero. The performance of all the baseline models on this strict subset remains similar to Table 7.4. For SNMN, the model trained using a learning rate of  $1e^{-3}$  is unstable while fine-tuning and hence we use a lower learning rate  $2.5e^{-4}$  to train the model and further finetune this model.

**InVariant VQA Augmentation.** In order to train and test different models, we aim at specific question types and see if we are able to boost the model’s performance on that question type. We select 4 question types based on how much they are affected from editing (i.e total number of flips/ total number of original IQA per question type) and if that question category has significant number of flipped labels in order to ensure we have enough edited IQAs for finetuning. Hence, we select the given 3 question categories and run our experiments on these splits: 1. ‘what color is the’ 2. ‘is there a’ 3. ‘is this a’ 4. ‘how many’. Additionally we focus on all the counting questions. All these specialized splits have around 6.3k-12.5k IQAs in the real train split with 10.8k-15.2k in edit train split.

For each question-type, we finetune all the models with corresponding real + IV-VQA IQAs for the particular question type. For a fair baseline, we also finetune all the models using just real data. Figure 7.3 (left) shows how different models, each specialized for a question type, behave when finetuned using real+synthetic data relative to finetuning using real data. The  $y$  axis denotes the reduction in flips and  $x$  axis represents the accuracy on the original set for. We observe that using synthetic data always reduces flipping as all the points lie above the  $y = 0$  axis. The amount of reduction differs for each question type and varies from model to model. For instance, CL model has the highest reduction in flips for question ‘is this a’ with no change in accuracy and while

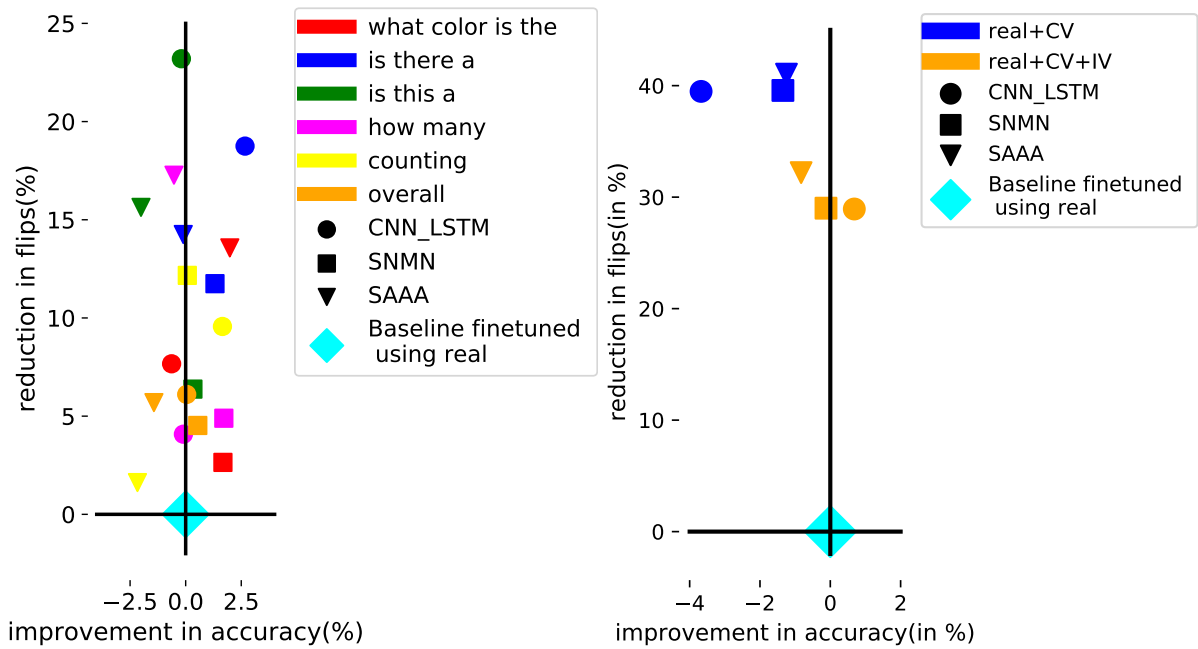


Figure 7.3: Accuracy-flipping results of finetuning experiments. Plots show relative performance of models finetuned using real+edit data w.r.t to using just real data.

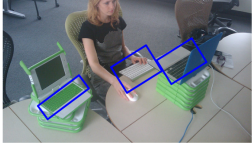
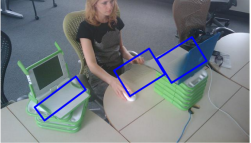





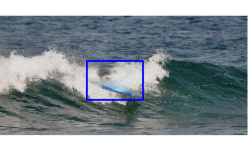
Q: What color is the mouse? A: white				Q:Is there a bowl on the table? A: no			
							
real	real+edit	real	real+edit	real	real+edit	real	real+edit
CL	white	white	white	white	white	no	no
SAAA	green	white	white	white	white	no	no
SNMN	green	white	white	white	white	no	no
Q: How many computer are there? A: 2				Q: How many people are in the water? A: 1			
							
real	real+edit	real	real+edit	real	real+edit	real	real+edit
CL	2	2	1	2	1	1	0
SAAA	1	2	2	2	1	1	0
SNMN	2	2	1	2	1	1	0

Figure 7.4: Visualizations from fine-tuning experiments using real/real+edit. Using real+edit makes models more consistent and in these examples- also accurate. Note: Striked-out objects are removed in the image below

question type ‘how many’ shows the least reduction. However for SAAA, ‘how many’ has the highest reduction with 2.5% drop in accuracy. For SNMN, counting has the highest reduction in flips. We also see that there are many points on the right side of  $x = 0$  axis showing that synthetic data also help improve accuracy on the test set. Figure 7.4 shows some of the examples for these specialized models. As we can see, finetuning the model with IV-VQA dataset helps in improving consistency and leads to more accurate predictions both on real as well as synthetic data.

Additionally, we also finetune all the baseline models with all the real data in VQA-v2 + IV-VQA data. Overall, we find that there is 5-6% relative improvement in flips for all 3 models: CL (17.15→16.1), SAAA (7.53→7.09), SNMN (8.09→7.72) with marginal improvement in accuracy% in case of CL (60.21 →60.24), 1% reduction in accuracy in case of SAAA (70.25→69.25) and 0.6% improvement in accuracy for SNMN (67.65→68.02).

**CoVariant VQA Augmentation.** For counting, we create our CV-VQA edit set by removing one instance of the object being counted and evaluate the models on both accuracy and consistency. Following the procedure above, we finetune all the models using real data, real+CV and real+CV+IV IQAs. We evaluate the  $n/n-1$  consistency for counting on CV-VQA for all the three models. The results are shown in Figure 7.3 (right). We see that using CV-VQA edit set reduces flipping by 40% for all 3 models with 1-4% drop in accuracy. Additionally we see that using CV-VQA + IV-VQA data reduce the flipping by 30%: CL (83.8→59.58), SAAA (77.74→52.71), SNMN (77.13→51.91) with comparable accuracy: CL (43.65→43.94), SAAA (50.87→50.45) and SMNM (50.67→50.61). Figure 7.4 (Bottom) shows that models when trained using synthetic data can show a more accurate and consistent behaviour.

## 7.5 CONCLUSIONS

We proposed a semantic editing based approach to study and quantify the robustness of VQA models to visual variations. Our analysis shows that the models are brittle to visual variations and reveals spurious correlations being exploited by the models to predict the correct answer. Next, we proposed a data augmentation based technique to improve models’ performance. Our trained models show significantly less flipping behaviour under invariant and covariant semantic edits, which we believe is an important step towards causal VQA models. By making our invariant and covariant VQA sets as well as evaluation and synthesis available to the community, we hope to support research in the direction towards causal VQA models.



## TESTING ROBUSTNESS TO APPEARANCE VARIATIONS WITH SEMANTIC ADVERSARIAL ATTACKS

---

### Contents

8.1	Introduction . . . . .	<b>118</b>
8.2	Synthesizing Semantic Adversarial Objects . . . . .	<b>119</b>
8.2.1	Synthesizer design . . . . .	120
8.2.2	Synthesizing semantic adversaries . . . . .	122
8.3	Experiments and Results . . . . .	<b>124</b>
8.3.1	Setup and datasets . . . . .	124
8.3.2	Semantic adversary for automated testing . . . . .	125
8.3.3	Semantic adversary for data augmentation . . . . .	128
8.4	Conclusions . . . . .	<b>130</b>

---

DEPARTING from object removal based analysis discussed in Chapters 6 and 7, in this chapter we analyze the robustness of object detectors to changes in visual appearance of objects. We achieve this by developing a differentiable object synthesizer network which can change an object’s appearance while retaining its pose. Using the synthesizer, we perform constrained adversarial optimization of an object’s appearance to produces rare/difficult versions of an object which fool the target object detector. Unlike pervious chapters, this testing process is targeted to a specific model, on account of the adversarial optimization. This enables our semantic adversary to efficiently create model specific hard examples – dropping the performance of the YoloV3 detector by more than 20 mAP points by changing the appearance of a single object and discovering failure modes of the model. The generated semantic adversarial data can also be used to robustify the detector through data augmentation, consistently improving its performance in both standard and out-of-dataset-distribution test sets, across three different datasets.



## 8.1 INTRODUCTION

Performance evaluation of computer vision systems is predominantly done by empirical evaluation on a fixed test set, often drawn from a similar distribution as the training data. However, due to limited sample size a fixed test set only captures a small portion of errors the model would make on diverse data seen during real-world deployment. This discrepancy manifests as poor out-of-dataset-distribution (OODD) generalization (Recht *et al.*, 2019, 2018; Hendrycks *et al.*, 2021), vulnerabilities to input noise (Hendrycks and Dietterich, 2019; Michaelis *et al.*, 2019) and adversarial perturbations (Goodfellow *et al.*, 2014). In this work, we propose automated testing through semantic adversarial editing which synthesizes difficult cases, targeted for a particular model, exposing its weaknesses. The error cases synthesized by our model often have atypical appearance and outside the distribution covered by fixed size datasets, however still within the true class boundary to human observers. Apart from its usefulness for testing, our semantic adversarial data can also be used to robustify the target model and improve its performance on OODD data.

To create reliable test data, we need to ensure that the generated sample is consistent with its label. At the same time, the created test data needs to be difficult, ideally capturing the different failure modes of the target model. Simply gathering more data is expensive and inefficient as the process is not targeted to the model. Our approach to meet both these criteria is to start from a real data point, and to make constrained semantic edits through a differentiable synthesizer model. The synthesis process is adversarially optimized to produce semantic changes which fool the target model. By only editing the appearances of individual objects with their pose and the scene held intact, we keep the changes minimal and realistic. An example is seen in Figure 8.1 where the appearance of “cow” is edited to change the detector prediction to “horse”.

Our key insight to constrain the semantic adversarial objects to be label-consistent is by limiting the range of synthesized appearance to be a combination of real ones. We first select a set of guiding templates by sampling instances of the same class from the real data. Then a new appearance is synthesized for the target object optimized to fool the detector, while staying within the convex hull spanned by the appearance of guiding instances. Since changing pose realistically is a much harder task, requiring reasoning over both object and the context, we keep it fixed. Our synthesizer network disentangles the object’s pose from its appearance thus allowing editing the appearance without affecting the pose.

Since our semantic adversarial object synthesis process is fully differentiable, we can mine new errors for a target model by directly optimizing the appearance to fool it. We demonstrate this by creating hard test data for the YoloV3 object detector (Redmon and Farhadi, 2018). The same mechanism can also be used to generate hard training data for the detector. The synthesized examples are hard positive examples, often lying close to detectors class boundaries. Our experiments on three dataset, COCO, BDD100k and Pascal, show that using the generated data to fine-tune the detector model improves the model performance and generalization to data distribution shift. To summarize, main contributions of our work are:

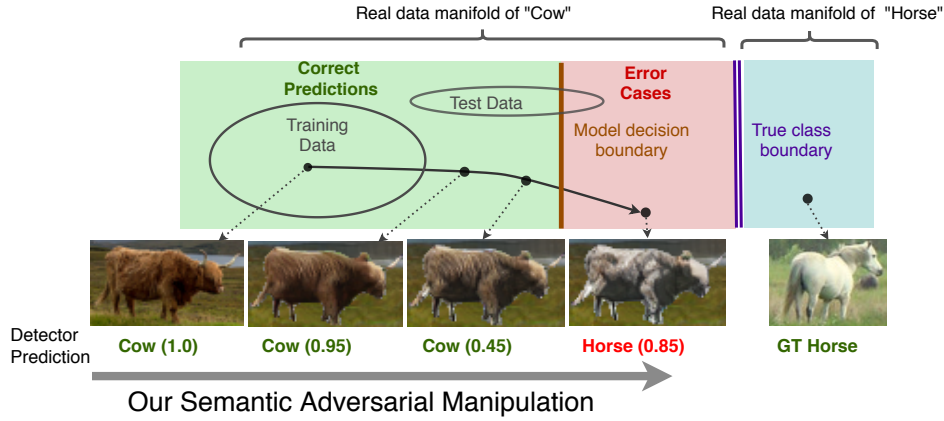


Figure 8.1: Standard testing paradigms only covers a small portion of errors models make in the real world due to sample size limitation. We propose semantic adversarial testing to find targeted failure cases through continuous optimization of object appearance to cross the model’s decision boundary, while remaining within the true class boundary.

- We propose the first method for automatized testing of computer vision models finding new error cases by synthesizing semantic adversarial examples.
- We design an object synthesizer network which disentangles object shape and appearance. This is achieved through a novel binary part segmentation bottleneck which scales better to the diverse object classes.
- We propose a novel mechanism to semantically change the object appearance to fool detectors, while keeping the appearance within the class boundaries as verified by a human study. Experiments show that our semantic adversary editing the appearance of a single object drops the detector performance by 20 mAP points and helps find new vulnerabilities of the model.
- Utility of our generated data is further shown by using it for training the YoloV3 detector. Experiments on three datasets show that the generated data helps improve the detector performance and generalization to OOD data.

## 8.2 SYNTHESIZING SEMANTIC ADVERSARIAL OBJECTS

Our main goal is to efficiently synthesize hard/error cases for an object detector from data manifold. We achieve this goal by starting from a real data point and adversarially editing its appearance through a synthesizer network to fool the target detector. This is a continuous optimization problem efficiently solvable through gradient descent. Additionally, we also need to make sure the synthesized sample is realistic and matches the original label. This is achieved first by only editing appearance of selected objects while retaining its pose, ensuring that the object instance fits well to the image context. Additionally, we constrain the space of appearances allowed during optimization to keep label consistency.

Our solution, shown in Figure 8.2, consists of two key contributions. First, we build an object synthesizer which disentangles object’s pose and appearance, thus allowing us

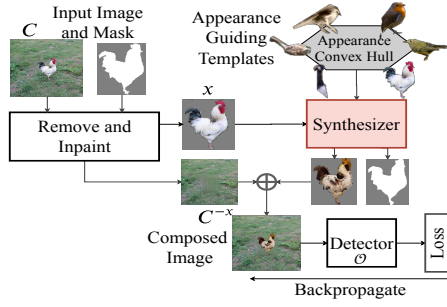


Figure 8.2: Our overall pipeline for creating the semantic adversaries

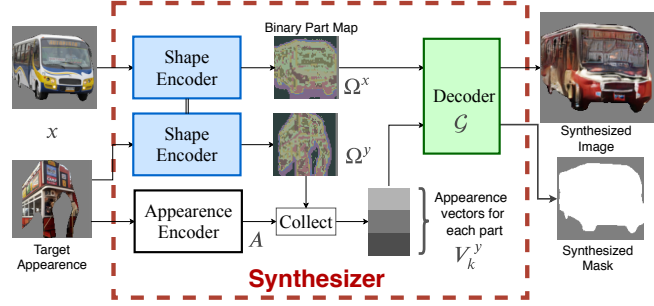


Figure 8.3: Synthesizer architecture to generate objects with disentangled appearance and pose latents

to generate various appearances for an object while keeping its original pose. This is enabled by a binary part segmentation bottleneck, which scales better to diverse object classes, a key requirement to scale to detection datasets like COCO. Second, we propose a novel optimization formulation wherein the latent appearance codes in the synthesizer are constrained to the convex hull of guiding templates. Under this constraint, the appearance is optimized to find the adversarial appearance for an object instance to fool the target detector.

### 8.2.1 Synthesizer design

To achieve disentanglement between pose and appearance we propose a modular architecture consisting of an appearance encoder producing latent codes representing the object appearance, a shape encoder producing a binary part segmentation of the object and a decoder which utilizes both the parts and the appearance vectors to synthesize the object. Note that the whole model is learned with only self-supervision, by learning to autoencode objects in the dataset. The overall architecture of synthesizer is shown in Figure 8.3. While this architecture is inspired by recent works [Lorenz et al. \(2019\)](#); [Jakab et al. \(2018\)](#), our solution differs in two crucial aspects, the type of bottleneck and the architecture of the decoder. To understand this difference, let us walk through the process of synthesizing an object given an instance  $x$  with the target shape and an instance  $y$  with target appearance.

**Shape Encoder.** A representation of the input object shape is first extracted by the shape encoder. This is a CNN with Unet ([Ronneberger et al., 2015](#)) structure which maps the input image into a  $K \times M \times N$  dimensional tensor  $Z$ , where  $K$  is the number of parts and  $M, N$  are the spatial dimensions.  $Z_{kij}$  represents the likelihood of the  $k^{\text{th}}$  part being present at locations  $ij$ . To disentangle shape and appearance, we need to restrict  $Z$  to only carry information about the spatial layout of the object instance. Prior works [Lorenz et al. \(2019\)](#); [Jakab et al. \(2018\)](#) do this by approximating  $Z$  with 2D Gaussians. While this works for classes like “person” whose parts fit well with gaussian shapes, we find that it does not work well on diverse object classes with complex sub-parts like “bicycle”, “bus”, and so on. Instead, we solve this by bottlenecking the information

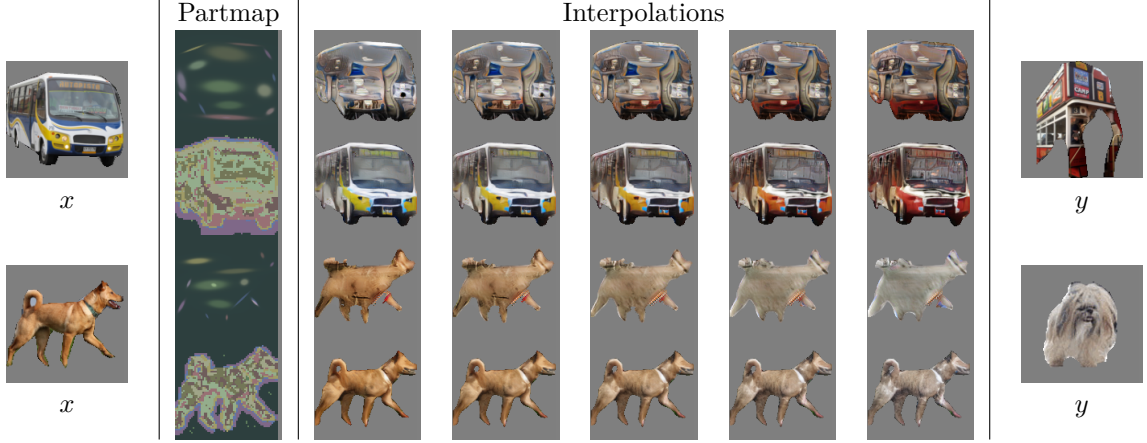


Figure 8.4: Appearance interpolations with Our (even rows) and the Gaussian bottleneck model (odd rows). The objects are generated using the shape code from  $x$  and by interpolating the appearance vectors from  $x$  and  $y$ .

in  $Z$  by converting it to a spatial probability distribution and sampling binary masks from it. Specifically, we obtain the part-probability distribution as  $P_{kij} = \text{softmax}_k[Z_{ij}]$  and sample binary part maps  $\Omega_{kij} = \text{gumbel\_softmax}_k[P_{ij}]$  from it. Here we use gumbel softmax approximation (Jang *et al.*, 2016; Maddison *et al.*, 2016) to sample from the multinomial distribution  $P_{kij}$  in-order to keep the sampling process differentiable.

**Appearance Encoder.** Object appearance is encoded with a CNN, which maps the input image to a tensor  $A$  of dimensions  $D \times M \times N$ . This spatial appearance map is reduced to  $K$  appearance codes  $\mathcal{V} = [V_1 \cdots V_k]$ , one for each part, by averaging  $A$  over the part activations  $V_k = \sum_{ij} P_{kij} A_{ij}$ .

**Decoder Network.** Now using the appearance vector  $\mathcal{V}^y$  extracted from image  $y$  and the binary part segmentation  $\Omega^x$  extracted from image  $x$ , the decoder network  $\mathcal{G}$  synthesizes the desired object and its segmentation mask. The appearance vectors  $\mathcal{V}_k^y$  are first projected onto their corresponding binary part activation map to reconstruct the spatial appearance map  $\tilde{A}^y = \mathcal{V}^y \Omega^x$ . Our decoder architecture, in contrast to Lorenz *et al.* (2019), utilizes spatially adaptive normalization layers (Park *et al.*, 2019) to input the appearance code at different resolutions to produce the four channel output (image + mask). We find that this helps better preserve the smaller appearance details in generated images as compared to inputting the appearance codes at the first layer.

**Training the Synthesizer.** We train the Synthesizer by learning to autoencode objects and to transfer appearance to other instances, similar to prior works Lorenz *et al.* (2019). Additionally we use an adversarial discriminator  $\mathcal{D}$ , to improve the sharpness of the generated images. When autoencoding, the model is trained end-to-end with  $l_1$  reconstruction loss for the image and cross-entropy loss for the segmentation mask. Paired training data for learning to transfer appearance is created in two ways. First, we apply simple affine transformations to object instances  $x$  to obtain  $T(x)$ , creating paired data. Now the appearance can be transferred by reconstructing  $x$  using shape  $\Omega^{T(x)}$  and appearance  $\mathcal{V}^x$  encodings and vice-versa. Secondly, the model is trained to transfer appearance to a random instance  $y$  of the same class by using the discriminator

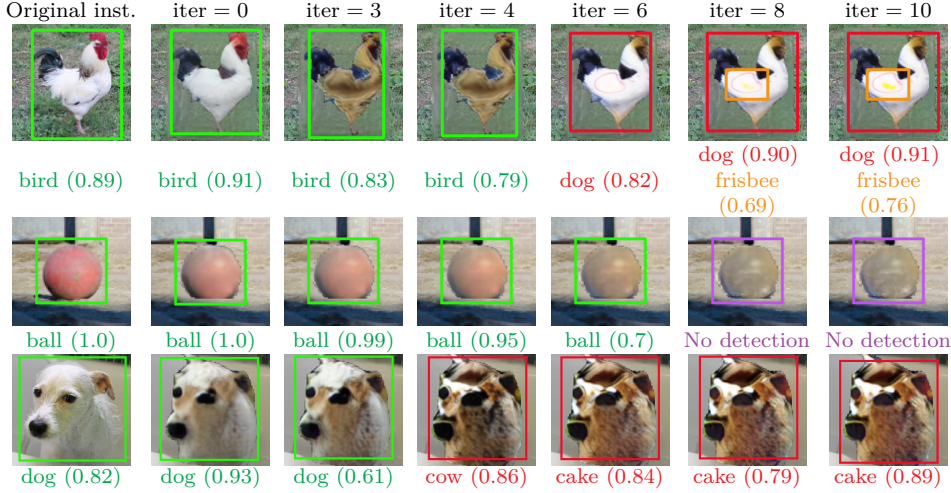


Figure 8.5: Intermediate steps when optimizing the appearance to fool the detector.

real/fake loss and cyclic reconstruction loss (Zhu *et al.*, 2017). Precisely, given shape code  $\Omega^y$  and  $\mathcal{V}^x$ , we generate a hybrid object  $xy$  and use the discriminator  $\mathcal{D}$  to evaluate realism and provide a training signal. We also re-encode  $xy$  to obtain appearance code  $\mathcal{V}^{xy}$ , and use it to reconstruct the original image as  $\tilde{x} = \mathcal{G}(\Omega^x, \mathcal{V}^{xy})$ . Apart from the reconstruction losses, we also impose additional constraints on the appearance and shape latent codes to provide intermediate supervision. For example,  $\Omega^{T(x)}$  should be same as  $T(\Omega^x)$  since an affine transformed input image should lead to an affine transformed part-map. Equations for these training losses are given below.

$$L_r = |x - \mathcal{G}(\Omega^x, \mathcal{V}^x)| + |T(x) - \mathcal{G}(\Omega^{T(x)}, \mathcal{V}^x)| + |x - \mathcal{G}(\Omega^x, \mathcal{V}^{xy})| \quad (8.1)$$

$$L_d = \mathcal{D}(\mathcal{G}(\Omega^x, \mathcal{V}^x)) + \mathcal{D}(\mathcal{G}(\Omega^{T(x)}, \mathcal{V}^x)) + \mathcal{D}(\mathcal{G}(\Omega^y, \mathcal{V}^x)) \quad (8.2)$$

$$L_a = \|\mathcal{V}^x - \mathcal{V}^{T(x)}\| + \|\mathcal{V}^x - \mathcal{V}^{xy}\| \quad (8.3)$$

$$L_p = -P^{T(x)} \log(T(P^x)) \quad (8.4)$$

Figure 8.4 compares the appearance transfer produced by our model trained on COCO dataset and a baseline model with identical structure, except using 2D Gaussians to bottleneck the shape encoding. We see big difference in quality of the generated images especially for objects like bus and dog. This performance gap can be understood by looking at the part representations extracted using the two methods also shown in Figure 8.4. We see that while Gaussian part maps are very crude approximations, our binary part maps captures detailed shape information, enabling better reconstruction and interpolation of appearance.

### 8.2.2 Synthesizing semantic adversaries

Now that we have a synthesizer which can effectively change appearance of a target object  $x$  using an appearance guiding template, let us leverage it to produce semantic



adversaries to fool an object detector. We start by extracting the shape ( $\Omega^x$ ) and appearance ( $\mathcal{V}^x$ ) representations for the target instance  $x$  occurring in image  $C$ , which we wish to edit to fool the detector  $\mathcal{O}$ . Instance  $x$  is removed from  $C$  using the ground-truth box and an object removal in-painter from Shetty *et al.* (2018a) to obtain canvas image  $C^{-x}$ . A new version of object  $x$  is synthesized as  $\mathcal{G}(\Omega^x, \mathcal{V}^x)$  and is pasted in place of the original to get the composed image. We denote this as  $C^{-x} + \mathcal{G}(\Omega^x, \mathcal{V}^x)$ . This process is illustrated in Figure 8.2.

A simple way to fool the detector would be to adversarially optimize the appearance vector  $\mathcal{V}^x$  until the object detector fails on the generated image  $\mathcal{G}(\Omega^x, \mathcal{V}^x)$ . However, in unconstrained optimization the appearance vectors often move into areas where synthesizer produces unrealistic images, which also fools the detector. We overcome this with a novel scheme which keeps the adversarially optimized  $\mathcal{V}^x$  from going far from the synthesizer’s input distribution. We first sample a set  $I = \{i_1, \dots, i_n\}$  of  $n$  guiding templates belonging to the same class and extract appearance codes for each of them  $\mathcal{V}^I = \{\mathcal{V}_k^{i_1}, \dots, \mathcal{V}_k^{i_n}\}$ . Now the appearance vector for the generated object is optimized to fool the detector while constraining it to remain within the convex hull spanned by  $\mathcal{V}^I$ .

$$\mathcal{V}_k^{adv} = \left\{ \sum_{j=1}^n \alpha_1^j \mathcal{V}_1^{i_j}, \dots, \sum_{j=1}^n \alpha_k^j \mathcal{V}_k^{i_j} \right\} \quad (8.5)$$

$$\max_{(a_1^1, \dots, a_k^n)} \mathcal{L}_{det} \left[ \mathcal{O} \left( C^{-x} + G \left( \Omega^x, \mathcal{V}_k^{adv} \right) \right) \right] \quad (8.6)$$

Here  $\alpha_k^j = \text{softmax}_n(a_k^j)$ , with  $\{a_k^1 \dots a_k^n\}$  being the interpolation co-efficients for part  $k$  and  $\mathcal{L}_{det}$  is the detector loss function which we maximize. There are total of  $n \times k$  interpolation coefficients which are optimized to find the adversary. Having independent part coefficients allows mixing and matching appearances from different templates for each part, and thus allowing richer appearances space to be explored through optimization. Further, since we only manipulate the latent appearance codes, the adversary cannot directly manipulate pixels to produce noisy patterns to fool the detector, but must instead rely on semantic changes. Detector loss is usually a sum of classification, objectness and box regression losses. We discard the box regression losses, as they are not directly affected by appearance and often leads to unstable behavior in optimization. Hence the detector loss becomes  $\mathcal{L}_{det} = \lambda L_{obj} + (1 - \lambda) L_{cls}$ , where  $\lambda \in [0, 1]$  is a co-efficient controlling how much the adversary focusses on causing missed detection versus misclassification. Spatial perturbations like position or scale of the object can be easily incorporated into our formulation by inserting a parametrized affine transformation matrix before pasting the object onto canvas image, allowing position and appearance to be jointly optimized to fool the detector.

Figure 8.5 depicts the adversarial appearance optimization steps to fool the YoloV3 detector. First row shows the synthesized bird changing from a reconstruction in the zeroth step to a different color by the fourth step, causing detector confidence to drop. More optimization leads to a bigger failure in the detector where the brown head and the yellow circle in the body of the synthesized bird causes the detector to see the object as a “dog” and a “frisbee”. The second row shows a case where the appearance of the “ball” is slowly changed to camouflage with the background and cause a missed detection. These

examples show that our method makes large semantic changes to the object appearance which fools the detector while looking plausible to human eye (empirically verified in Section 8.3.2).

## 8.3 EXPERIMENTS AND RESULTS

We evaluate our semantic adversary for two applications, as a diagnostic tool to find failure modes in the detector and as a hard data generation mechanism to improve the performance of these detectors. We measure the effectiveness of the semantic adversary in terms of the detector performance on generated adversarial test set. We verify label consistency of the semantic adversary by a human study where observers verify if the original class is preserved after adversarial editing. We also qualitatively examine the synthesized error cases and find different mechanisms which cause detector failures. Data augmentation experiments are run on three different datasets, COCO, VOC and BDD100k, and we measure the benefit of the generated adversarial data for improving model performance on both standard test, as well as generalization to out-of-dataset-distribution. First, we describe the experimental setup and datasets, followed by the analysis on effectiveness of the semantic adversary for diagnostics and data augmentation.

### 8.3.1 Setup and datasets

We conduct our data augmentation experiments on three datasets – COCO (Lin *et al.*, 2014a), PascalVOC (Everingham *et al.*)(VOC) and BDD100k (Yu *et al.*, 2020). COCO and VOC contains both indoor and outdoor images with common objects like person, car, table etc. While COCO has 120k training images with 80 classes, VOC is smaller with 20 classes and 14k training data (combining 2007+2012 splits). BDD100k is a large scale driving dataset with 100k street scenes captured from a car driving around major US cities, with annotation of objects like person, car, traffic light and so on. The object synthesizer and removal inpainter are both trained on the COCO dataset, due to availability of instance segmentation masks needed to extract the object patches. Since all the classes in VOC and 9/10 classes in BDD100k are part of COCO (except “rider” class), COCO trained model can be used to synthesize adversarial objects on these datasets. The synthesizer operates at  $128 \times 128$  resolution. The generated objects are scaled to match the target box.

We use the YoloV3 (Redmon and Farhadi, 2018) model as the target detector, as it is a popular single staged detector with fast runtime, making adversarial attack experiments run quicker. We train our baseline model from scratch using the implementation available in Ultralytics (2019), using all the standard data augmentation methods including color jittering and rotation. However, to keep the synthesis single resolution, all our detector models are trained on single fixed resolution ( $416 \times 416$  on COCO and VOC,  $704 \times 1248$  on BDD100k) as opposed to multi-scale training used in YoloV3, yielding a lower baseline performance. All the improvement reported from training on our synthesized data is



Optimize	n_obj	mAP ↓	Success rate by instance type ↑			Instance	Label correctness
			Edited	Co-occurring	Com-bined		
Real Data	0	81.2	-	-	-	Real	99%
Appear	1	62.4	74.99	58.62	61.10	Random label	11%
Pos + Appear	1	59.5	<b>77.69</b>	59.30	62.13	SemAdv (appearance)	93%
Pos + Appear	2	<b>46.5</b>	77.10	<b>61.54</b>	<b>65.82</b>		

Table 8.1: Overall and instance-level detector performance under semantic adversarial editing. *Co-occurring* refers to the other untouched objects in the image.

Table 8.2: Human study results on the label correctness of semantic adversarial editing.

in-complimentary to the standard augmentations. The evaluation is also performed at these fixed resolutions. The models on BDD100k and VOC are trained after initializing from a trained COCO model. This ensures that these models have already been exposed to the instances from the COCO dataset. When training on synthetic data, we start from the pre-trained model and fine-tune the last two layers in case of COCO and BDD100k and last three layers in case of Pascal. For fair comparison we also further fine-tune the pre-trained model using the exact same configuration, but only with real data to obtain the *Base-FT* model in all three datasets.

Apart from evaluating on i.i.d. test sets, we also measure the generalization to OOD data. This tests our hypothesis that semantic adversarial data improves the model robustness to OOD samples, since our adversarial data often contains atypical objects, from the tail of appearance distribution. To do this, we test the COCO trained model on the UnRel (Peyre *et al.*, 2017) and VOC test sets. The models are tested on the overlapping 29 classes in UnRel and all 20 classes in VOC. UnRel data contains objects in unusual relationships and contexts and will measure if the model generalizes to rare cases. Similarly VOC trained models are also tested on UnRel, for the overlapping 14 classes. The BDD100k models are tested on D2-City (Che *et al.*, 2019), with driving images from Chinese cities.

### 8.3.2 Semantic adversary for automated testing

To quantify the effectiveness of the semantic adversary, we create adversarial test sets using the COCO training images by optimizing the appearance of selected objects in each image to fool the detector. Objects are selected at random as long as they are not too small/large ( $\geq 32$  pixels and  $\leq 30\%$  of the image area). We do this with three variants of our approach. First only optimizes the appearance of one object instance. The second variant optimizes both the position and the appearance of the same object. In the third variant two random objects are chosen from each image and their position and appearance are adversarially optimized. Each of these test sets contain 37k images. Object detector is run on these three sets and performance is measured using mean average precision (mAP@0.5)

**Quantitative Analysis.** The results are reported in Table 8.1. We see that all the



Figure 8.6: Qualitative examples of the failure cases discovered by our semantic adversary. **Green** boxes are correct detections, **purple** boxes indicate missed detections and **red** boxes show the misclassified objects. Only relevant detections are marked.

semantic adversaries drop the performance of the model significantly, with mAP dropping from 81.2 on the corresponding real data to 62.4 with just optimizing the appearance. Optimizing the position and scale of the object along with appearance further degrades the detector performance, with mAP dropping to 59.5. When we adversarially modify two objects jointly, the detector performance drops again to 46.5 mAP, making it a 57% drop in detector performance. To understand this performance drop, we look at the effect on the detector’s confidence for each object instance. We consider it a success if the detector’s confidence drops after adding the semantic adversary. Table 8.1 presents the success rate on the edited as well as untouched objects in the same image. Firstly, we see that all three strategies drop the detector’s confidence on more than 74% of the semantically edited instances. Interestingly, about 60% of the untouched co-occurring instances are also negatively affected. This is often due to the contextual changes caused by the misclassification of the edited instances or minor occlusions produced by the edited instance. We note again that our semantic adversarial attacks are efficient, performed in just 10 steps of gradient descent. To put this in context, adversarial color jittering (Hosseini and Poovendran, 2018) takes about 200 trials to attack (success in 50% of cases) a simpler classification model on a smaller CIFAR-10 dataset.

We also compare the effectiveness of our semantic adversary to a standard  $L_\infty$  norm adversarial attack. For fair comparison we also restrict the  $L_\infty$  attacker to change pixels within the bounding box of a single object. The experiments show that our single object semantic adversarial attack (mAP=59.5) is roughly equivalent in strength to a  $L_\infty$  norm attack with  $\epsilon = 8/255$  (mAP=58.7).

**Human Study.** A natural question at this point is if the semantic adversarial samples are within the true class boundary. To answer this we turn to human observers. We conduct a study where a human judge is presented with an image and asked if the object highlighted with the box belongs to the specified class. If they consider the label correct, they are asked to also rate how typical the object appearance is from 1 to 5, with 1 corresponding to very unusual and 5 corresponding to very typical appearance. The study is conducted on a mix of 250 real and semantically edited instances each, with each instance rated by three independent observers. We also introduce additional 10% samples where labels are shuffled. This is done in-order to verify the work of the human annotators. Table 8.2 shows the label correctness results as judged by majority vote of three humans. As expected the label correctness is very high on real instances and is very low on label-shuffled instances. We also see that in 93% of cases, humans agree that semantic adversary preserves the label of the object instance. Performance drop on semantic adversary is small for human observers compared to the significant drop by object detectors seen before. The typicality rating provided by the human judges on real and semantically edited instances shown in Figure 8.7 helps understand this gap. While most of the real samples have a typicality rating of 4 or 5 (very typical), semantic adversary have lower rating between 2-3. These results show that the semantic adversary generates atypical examples which are still correctly detectable by humans, but are hard for our detectors. This is further supported by lower performance of the detector on less typical real samples (Accuracy > 70% on typicality rating = 5 and accuracy 35-40% on typicality rating between 1-3).

**Qualitative Analysis.** Examining the cases where the semantic adversary fools the detector reveals that the adversary causes missed detections and misclassification through four main mechanisms listed below and illustrated in Figure 8.6. All examples are from the strategy with editing single object and position.

- **Camouflaging** - Semantic adversary often causes missed detections by changing the appearance of the object to blend with the background. First row of Figure 8.6 illustrates a frisbee, a stop-sign and a hydrant being camouflaged.
- **Occlusion** - Second row shows cases where the semantic adversary causes missed detections by moving to partially occlude some co-occurring objects.
- **Appearance** - In many instances, the appearance of the object is altered to include small visual features which trigger misclassification by the detector. These can be seen in the third row in Figure 8.6. We see that "cow" is changed to "horse" based on color change and person is misclassified as a dog due to a small change in hue. These cases indicate that detectors often rely on false correlations of low-level textures or colors to certain classes, and often fail when these textures are altered, as also shown for classifiers in recent work (Geirhos *et al.*, 2019).
- **Contextual Appearance** - Last row shows examples where with a change in an object appearance, the contextual evidence overrides the visual features, causing misclassification. For example in the first image, a dog changed to white color is misclassified as a sheep as there are other sheep present nearby. Similarly, a falling person is mistaken as an airplane, and a surfboard as a boat.

We note that despite a few generation artifacts, with the COCO dataset being hard

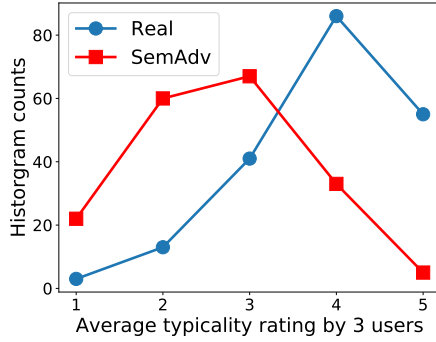


Figure 8.7: Comparing the typicality rating between real data and semantic adversary.

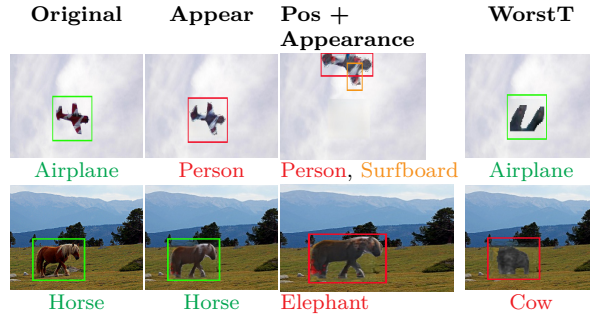


Figure 8.8: Comparing various adversarial strategies and the worst-case template baseline.

for current GAN models, these samples look plausible to human eye and we would not make the same predictions as the detector. This makes it a useful tool to explore the breaking points of a trained detector.

### 8.3.3 Semantic adversary for data augmentation

Apart from being a useful diagnostic tool, semantic adversaries can be used to generate training data. By targeting the detector, we can create tailored hard positives for the model, and thus get the most benefit when added to the model training set. We generate the training data with a similar process as in the previous section: first selecting an eligible object from each training image, adversarially optimizing its appearance and adding it back to the training set. The model is then fine-tuned with a combination of the original and the synthesized adversarial data for 50 epochs, and performance on the standard test sets and OOD data is measured. We now present this data augmentation results on the three datasets.

**COCO dataset.** Table 8.3 shows the data augmentation results on the COCO dataset. Comparing the *Baseline* and *Base+FT* models we see that the further fine-tuning the last two layers of the model improves the performance a bit on COCO and VOC test sets, while reducing a bit on UnRel (39.0 vs 38.8). Comparing this with the basic semantic adversary augmented model *SA-Rand-App*, which only edits object appearance, we see a bigger improvement on all three test sets. Table 8.3 also shows *Base+FreeAdv*, an adversarial baseline which allows for free manipulation of appearance vector under  $l_\infty$  constraint, without the convex hull constraint. Its poor performance compared to *SA-Rand-App* shows that the unconstrained attack does not work well as often the adversarial sample looks unrealistic. *SA-Rand-App* model generates the semantic adversaries using randomly sampled instances for guidance. We can further target the model weaknesses by sampling the templates from hard instances for the detector, i.e. setting the probability of picking an instance inversely proportional to the detectors confidence on it. The model trained this way, *SA-App*, further improves the performance a bit on COCO and significantly on UnRel (39.6 vs 39.2). Moreover, the *SA* model



which jointly optimizes position and appearance gets even better results, improving over *SA-App* in COCO and VOC test sets. Now, we compare our approach to a simpler baseline, where an object is replaced with the template which increases the detector loss the most. While this often fools the detector, it also places instances which do not fit with the image context, as seen in the examples in Figure 8.8. Thus, the *Base-WorstT* model using this data in training performs worse than *SA-App* on all test sets.

We can increase the benefit of semantic adversaries by editing more objects in the image, creating harder data. The *SA#2* model which edits appearance and position of two objects improves on COCO(+0.3 mAP) and UnRel(+0.4 mAP) test sets compared to *SA* editing single objects. Since our adversarial data is adaptive to the model, we can continue generating harder examples attacking the newly trained model. *SA#2x2* does this, further training the *SA#2* model using the adversarial data generated by attacking *SA#2*. This second iteration helps and *SA#2x2* still improves. By repeating this four times, we get the *SA#2x4* model which outperforms the baseline on all three test sets, COCO(+1.0 mAP), VOC(+1.3 mAP) and UnRel(+1.8 mAP). The gain is larger on ODD test sets, VOC and UnRel, indicating that training with semantic adversary improves the robustness of the model to input distribution changes. We also compare our approach to recent data augmentation approaches PSIS (Wang *et al.*, 2019a) and AutoAug (Cubuk *et al.*, 2019). For PSIS, we use the data provided by the authors (Wang, 2019) to fine-tune our baseline model same as before. While the PSIS data improves over the baseline, it falls short compared to our *SA#2x4* model in all test sets. Table 8.3 also shows that our approach is complimentary to AutoAug (Cubuk *et al.*, 2019), which applies augmentation policies on the entire image. While auto-augment improves the baseline performance, our *SA#2* model improves even more when combined with AutoAug.

**PascalVOC Dataset.** Results in Table 8.4 for data augmentation on VOC dataset show that, semantic adversarial data improves performance here as well. The *SAx5* model, which edits appearance and position of a single object, is better than the baseline on both VOC(+1 mAP) and the UnRel(+2.1mAP) test sets, again with bigger gains on the ODD data. *SA#2x5* which creates two adversarial objects underperforms *SAx5*, since VOC images often have only a single object, which causes the *SA#2x5* to add too many out-of-context objects in its generation.

**BDD100k Dataset.** On BDD100k (see Table 8.5), we found that adversary often caused drastic appearance changes when fooling the classifier. Since BDD100k has only 10 classes, the class boundaries are well separated and fooling the classifier needs large unrealistic appearance changes. Instead, optimizing to only reduce the objectness score (setting  $\lambda = 1$ ) leads to more realistic synthesis. This is seen when comparing *SA-App* models with  $\lambda = 0.5$  and  $\lambda = 1.0$ . The model with  $\lambda = 1.0$  performs much better on both the BDD and the ODD D2-city test sets, while also improving over the fine-tuned baseline. Additionally, we see in Table 8.5 that the *SA-App* performs better than *SA* which optimizes position, showing that it is better to edit objects in-place in structured scenes in BDD.

Model	obj	COCO	VOC	UnRel
Baseline	-	46.1	66.4	39.0
Base+FT	-	46.2	66.9	38.8
Base+FreeAdv	1	45.8	65.3	37.9
Base+WorstT	1	46.2	66.8	39.2
SA-Rand-App	1	46.5	67.1	39.2
SA-App	1	46.6	67.0	39.6
SA	1	46.7	67.3	39.4
SA#2	2	46.9	67.4	39.8
SA#2 x2	2	47.0	67.4	40.4
SA#2 x4	2	<b>47.1</b>	<b>67.7</b>	<b>40.8</b>
Base+AutoAug	-	47.0	67.6	40.4
SA#2+AutoAug	2	47.8	68.1	41.5
PSIS (Wang <i>et al.</i> , 2019a)	-	46.7	67.5	39.8

Table 8.3: Data augmentation results on COCO dataset. Metric used is mAP@0.5. AutoAug is the data augmentation proposed by Cubuk *et al.* (2019).

Model	obj	VOC	UnRel
Base+FT	0	74.0	42.9
Base+WorstT	1	73.7	43.4
SA x5	1	<b>75.0</b>	<b>45.0</b>
SA#2 x5	2	74.0	44.3

Table 8.4: Data augmentation results on VOC using semantic adversary.

Model	$\lambda$	BDD	D2City
Base+FT	-	50.7	34.7
SA-App	0.5	50.8	34.6
SA-App	1.0	<b>51.4</b>	<b>35.1</b>
SA	1.0	51.2	35.0

Table 8.5: Data augmentation results on BDD100k dataset.

## 8.4 CONCLUSIONS

We presented a method for automatic test case generation through semantic adversarial optimization of object appearances. Our approach can synthesize new OODD hard examples which cause failures in the target detector, while remaining realistic to human eye. Analysis of the synthesized data shows the different failure modes discovered by the process includes camouflaging, occlusions and appearance changes. Our adversarial data is also useful for data augmentation, consistently improving the detector on standard and OODD test sets, in three datasets. We hope that our work will facilitate future approaches to test models beyond finite datasets and hence develop more reliable performance metrics.

---

**Contents**

9.1	Introduction . . . . .	<b>132</b>
9.2	Adversarial Weather Optimization . . . . .	<b>133</b>
9.2.1	Testing with simulator in loop . . . . .	133
9.2.2	Adversarially optimizing weather . . . . .	134
9.3	Experiments and Results . . . . .	<b>136</b>
9.3.1	Experimental setup . . . . .	136
9.3.2	Quantitative results . . . . .	137
9.3.3	Qualitative analysis of the failure modes . . . . .	138
9.4	Conclusions . . . . .	<b>141</b>

---

So far in the thesis, we have developed methods and approaches to measure and improve robustness of different computer vision systems to semantic variations in the input. However, we have restricted ourselves to localized variations by focusing on individual objects. This is true for object removal based methods studied in Chapters 6 & 7 and object appearance editing based method developed in Chapter 8. We will now generalize this to study if we can optimize scene-level properties, which affect large portions of the image, to create hard examples to test vision systems. In this chapter, we propose an approach to create adversarial weather configuration for a scene, targeted to a specific model. In a simulation environment using CARLA (Dosovitskiy *et al.*, 2017), we adversarially optimize the weather settings to cause failures in semantic segmentation models. We compare different approaches to perform this optimization through a non-differentiable simulator. Our analysis demonstrates that models show significant drop in performance in this adversarial testing framework, compared to standard fixed test sets. This highlights the need to measure worst-case performance of models before deployment.



## 9.1 INTRODUCTION

Driving in adverse weathers is challenging even for human drivers. Control dynamics of the car change due to snow or rain. Perceiving the surrounding environment can also be difficult due to low visibility and changing appearance. In order to build autonomous driving systems which can safely navigate in adverse weather conditions, we need mechanisms to stress-test their robustness to weather variations. In this chapter, we study the robustness of perception models to visual changes caused by weather.

A key challenge in studying weather robustness of computer vision systems is the lack of large datasets with diverse weather annotations. Collecting, training/testing data in adversarial weather is a difficult and expensive process. Sakaridis *et al.* (2019) create a dataset to capture daylight changes, by driving through the same parts of Zurich during day and night time and obtain two versions of the scene. Of course this approach is difficult to emulate for other weather changes like fog, rain and snow, as the weather would need to cooperate with the kind of data samples we need. For this reason, we conduct our study in a simulated environment. We use the CARLA simulator proposed by Dosovitskiy *et al.* (2017) to create driving scenes and test the robustness of semantic segmentation models to weather variations.

Standard benchmarking protocol used in computer vision dictates that we create fixed size test sets from CARLA and test our models on these fixed sets. Following this we create test splits where weather is sampled from *clear* conditions or from difficult conditions like *foggy/rainy*. However, continuing in the spirit of Chapter 8, we compare these fixed test sets with an adaptive testing approach. Here we create worst-case weather configuration tailored to the model for each scene in the test set. This is done by adversarially optimizing the weather settings of the simulator to maximize segmentation errors of the target model, for each scene. This approach allows us to measure the worst-case performance of the model w.r.t weather.

Adversarially optimizing the weather parameters of the simulator has a critical technical challenge. The simulator is not differentiable and hence we cannot simply backpropagate the segmentation loss to optimize weather. We can get around this problem by using finite-differences to approximate the gradients. But, computing finite differences is computationally taxing. We can use fast approximators like simultaneous perturbations (SPSA) proposed by Spall (1992) Another option is to use gradient-free black-box optimizer (Cazenave *et al.*, 2019). We compare these approaches for their effectiveness in finding worst-case weather configurations.

In summary, our main contributions are:

- We investigate a method to stress-test robustness of semantic segmentation models to scene-level properties like weather. We use the CARLA simulator to create weather variants for a scene. Targeted worst-case weather configuration of the simulator are found by adversarially optimizing them to maximize the segmentation error.
- We compare different approaches to perform this optimization, despite the non-differentiability of the simulator. We find that sampling based gradient-free

optimization approach TBPSA (Cazenave *et al.*, 2019) works better than finite-difference based gradient approximators.

- We study the effect of the training data distribution on the robustness of the segmentation model. We find that while training on uniformly distributed data (w.r.t to the weather parameters) significantly improves the model robustness, these models still show a performance drop (10%) under semantic adversarial attack.

## 9.2 ADVERSARIAL WEATHER OPTIMIZATION

We use the CARLA simulator to study robustness of semantic segmentation models to weather variations in self-driving scenarios. By adversarially optimizing the weather parameters to fool the segmentation model we are able to measure the worst-case performance of the model w.r.t weather. In the following subsections we will describe the overall approach of testing models with a simulator in Section 9.2.1, and discuss methods to optimize the weather parameters despite the simulator being non-differentiable in Section 9.2.2.

### 9.2.1 Testing with simulator in loop

Overall flow of our testing approach is shown in Figure 9.1. We iterate over obtaining the image and ground-truth segmentation from the simulator, scoring the predicted segmentation by the target model and adversarially optimizing the weather configuration to increase the segmentation loss.

**Configuring the simulator.** To begin our testing, the first step is to configure the CARLA simulator. CARLA allows us to create driving scenes with camera mounted on the ego-vehicle, and other cars and pedestrians placed at desired locations. We start by placing the ego-vehicle in a randomly chosen drivable location on the map. We can then configure the weather, driving behaviors of the other cars and navigation policies of the pedestrians. However, since we are interested in measuring segmentation performance across weathers, we want to keep the rest of the scene static. For this, all car and pedestrian velocities are set to zero.

Once the scene is set, and we start obtaining camera and ground-truth segmentation readings from the simulator, we can optimize the weather parameters to fool the segmentation model. CARLA offers nine weather parameters affecting different aspects like sunlight, rain, fogginess and windiness. Full description of these parameters are shown in Table 9.1. All the parameters are continuous valued. At each time step  $t$ , we set the weather parameters  $W_t$  of the simulator  $C$  and feed the obtained image  $I_t = C(W_t)$  to the segmentation model  $S$ , to obtain predictions  $S(I_t)$ . The quality of the predicted segmentation is measured using the cross-entropy loss  $L_t$  between ground-truth  $G_t$  and  $S(I_t)$ . The adversary uses the current weather,  $W_t$ , and the loss to obtain the new weather parameters as:

$$W_{t+1} = \text{Adversary}[W_t, L_t] \quad (9.1)$$

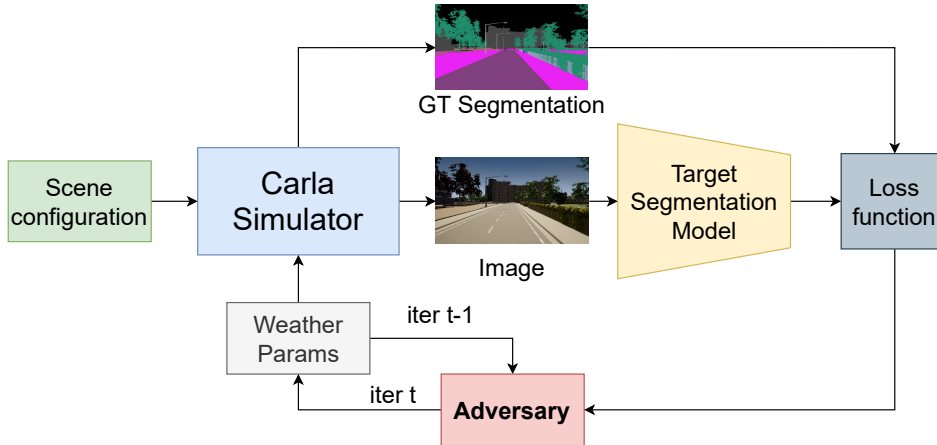


Figure 9.1: Overview of our pipeline to find worst-case weather configuration for a given scene.

Parameter	Function	Range
cloudiness	amount of sky covered by clouds	(0, 100)
precipitation	rain intensity	(0, 100)
precipitation deposits	amount of water deposited on the ground	(0, 100)
wind	wind strength, affects rain direction and trees	(0, 100)
fog density	concentration of fog	(0, 100)
fog distance	starting distance of the fog	(0, $\infty$ )
wetness	determines wetness of materials	(0, 100)
sun altitude	angle of the sun on vertical axis, affects day light	(-90, 90)
sun azimuth	angle of the sun on horizontal axis, affects light direction	(0, 360)

Table 9.1: Comparing efficiency of different attack mechanisms when attacking the model trained on uniform training data

### 9.2.2 Adversarially optimizing weather

If the simulator was differentiable we could use gradient descent to optimize the weather parameters by backpropagating the loss all the way. Then the update equation in 9.1 would simply be:

$$W_{t+1} = W_t + \alpha \frac{\partial L_t}{\partial W_t} \quad (9.2)$$

where  $\alpha$  is the step size. Since the simulator is non-differentiable, we explore three alternate approaches to perform this optimization. Finite differences and SPSA try to approximate the gradient by perturbing the weather vector and measuring the response. We also experiment with gradient-free optimization approaches, based on evolutionary algorithms (Arnold, 2012).

Apart from non-differentiability, another challenge for the optimization is the simulation noise. We observe that even with all the cars/pedestrians static and the weather held constant there are small changes in the image between time steps. This is caused

by the small movements in vegetation caused by wind, ripples in water puddles, and so on. While these are not large visual variations, it still causes fluctuations in the loss function between time-steps. In flat areas of the loss, this can mislead the optimization or cause oscillations. Hence the optimization methods need to account for this noise. Now let's look at these three optimization approaches in detail.

**Finite differences.** In this approach we approximate the derivative of the loss, by perturbing each weather parameter by a small amount and measuring its effect on the loss (Kiefer *et al.*, 1952). We use central differences where the parameter is perturbed in both directions (+ve and -ve) and the gradient is computed as:

$$H^j(i) = \begin{cases} h, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (9.3)$$

$$\frac{\partial L}{\partial w^j} = \frac{L(S(C(W + H^j))) - L(S(C(W - H^j)))}{2h} \quad (9.4)$$

where  $h$  is the probe step-size. Once we compute the gradient we can use Equation (9.2) to compute the next weather parameter to query. We can see from Equation (9.4) that finite differences approach needs two measurements for each parameter dimension, resulting in 18 queries to the simulator to compute gradients for the nine weather parameters. This is extremely slow and is a major drawback of this method.

**SPSA.** Simultaneous perturbation approach was proposed in Spall (1992) to overcome this limitation of finite differences. The key idea here is to take measurements in a randomly chosen vector direction by perturbing the whole parameter vector. This needs only two measurements per step, and is independent of the parameter dimension. As long as the random perturbation vector is zero mean and takes only non-zero values, this approach works well to approximate the gradients. It is common to use random vectors from the Rademacher distribution, where each dimension takes the values  $\Delta^j \in -1, 1$  with probability 0.5. Gradient approximated with spsa is given as

$$\frac{\partial L}{\partial W} = \frac{L(S(C(W + h\Delta))) - L(S(C(W - h\Delta)))}{2h\delta} \quad (9.5)$$

**Gradient free optimization.** An alternative to these gradient approximation based approaches are the gradient-free global optimization methods. Evolutionary algorithms are one such class of methods, which work by sampling a population of initial candidate solutions and evolving them over time to find the best candidates. Covariance matrix adaptation (CMA-ES) (Hansen and Ostermeier, 1996) does this by adapting the mean and variance of the sampling function to increase the likelihood of sampling the best performing candidates. In our work we use the TBPSA (Cazenave *et al.*, 2019; Hellwig and Beyer, 2016) algorithm implemented in the Nevergrad library (Rapin and Teytaud, 2018), which incorporates noise control strategies into CMA-ES algorithm.

### 9.3 EXPERIMENTS AND RESULTS

We proposed an approach to find the worst-case weather configuration for a model for each scene. It is a computationally intense process involving optimizing the weather parameters with a simulator in the loop. To understand the usefulness of this approach, we will now evaluate if there is a significant difference in the model performance on these worst-case samples compared to standard fixed size test sets. We compare the different optimization approaches discussed before on their effectiveness in decreasing the segmentation performance. We also study the effect of the training data distribution on the segmentation model robustness. We also qualitatively analyze the results and show interesting failure cases discovered by our approach. But first we will discuss the experimental setup.

#### 9.3.1 Experimental setup

We perform our studies on the segmentation model in [Zhu et al. \(2019\)](#), since it was one of the best performing models on cityscapes dataset with an open-source implementation. To avoid domain shift issues, we train this model from scratch on the CARLA simulator data first. We mimic the cityscapes dataset and create training set of 3000 street scenes by driving the ego-car in the simulator and periodically capturing images. The 3000 samples come from five different maps (Town01 to Town05) each contributing 600 samples. We also create a validation set of 500 samples for hyperparameter tuning and fixed size test sets of 500 samples. Adversarial testing is also carried out in the 500 scenes sampled from the five towns.

**Training and Test distributions.** To understand the effect of training data distribution on model robustness, we create different training sets with varying distributions of the weather as follows.

- *Clean*: In these samples the weather is held fixed to mimic a clear day. All weather parameters are set to zero, except sun altitude which is set to 60 degrees (mid-day light).
- *Gaussian*: While there have been attempts to collect datasets with different weather conditions (e.g. in BDD100k there are day/night and a few rainy/foggy images), they are never balanced as it is difficult to obtain images of extreme weather conditions. To mimic these imbalanced real-world datasets we create splits where we sample weather parameters from Gaussian distributions centered around clear day. Each weather parameter is sampled independently. We create splits with four different variances,  $\sigma = 5, 10, 20, 30$ . Note that we normalize all parameter ranges to 0-100 for this sampling.
- *Uniform*: This acts as the oracle case where all weather settings are uniformly sampled. While this is unrealistic compared to real-world data, it helps us understand the upper-limits of model performance.

Method	Mean worst-case mIoU ( $\downarrow$ )
Clear test set	85.55
Uniform test set	86.02
Finite differences	81.23
Random Sampling	77.49
SPSA	77.62
TBPSA	69.79

Table 9.2: Comparing efficiency of different attack mechanisms when attacking the model trained on uniform training data. Lower is better.

We train segmentation models on each of these training splits. For standard testing we create equivalent *Clean* and *Uniform* test sets to measure clear weather and adverse weather performance. Segmentation performance is measured using mean intersection-over-union (mIoU) averaged over each class.

### 9.3.2 Quantitative results

We now present the quantitative results of our experiments.

**Comparing optimization methods.** We compare the different optimization methods by attacking the segmentation model trained on *Uniform* training data. We chose this model, since it is the hardest model to attack. Table 9.2 reports the performance of the target model in terms of mIoU on the data created by different optimization methods. We can see the standard baseline performances on *Clean* and *Uniform* fixed size test sets in the first two rows. Model shows no drop in performance when tested on *Uniform* set compared to *Clean* set (86.02 vs 85.55) which seems to indicate that the model is robust to weather variations. However on adversarial weather configurations we start seeing performance drops. Finite differences approach drops performance by 5 points to 81.23. A random sampling baseline, where  $N$  different weather conditions are uniformly randomly sampled for each scene and the worst one is picked, does much better dropping performance by 9 points to 77.49. This indicates that finite differences approach is ineffective, as it is not able to handle the noisy optimization. The more efficient SPSA method does better than finite differences, but is still slightly worse than random sampling at 77.62 mIoU. Gradient-free approach TBPSA is much better at handling this noisy optimization and significantly drops the model performance to 69.79 mIoU. The gradient-free approaches have an advantage in this setting, where often SPSA gets stuck in local minimas. TBPSA ignores local gradients and can move quickly across the parameter space. This results in significantly improved performance compared to other methods. We will use TBPSA for rest of the experiments and analysis.

**Effect of training data.** So far we saw that while segmentation model trained on *Uniform* data shows good performance on standard *Uniform* test set, it shows significant performance drop on adversarial test set created by our attack using TBPSA. This also holds for other models trained on more realistic biased data. Table 9.3 shows the

Training Data	Clear day test	Uniform test set	Adversarial set
Only clear day	88.88	58.54	30.19
Gaussian ( $\sigma=5.0$ )	84.39	60.95	31.23
Gaussian ( $\sigma=10.0$ )	84.85	64.07	37.59
Gaussian ( $\sigma=20.0$ )	83.66	79.46	48.37
Gaussian ( $\sigma=30.0$ )	86.09	85.10	64.40
Uniform	85.55	86.02	69.79

Table 9.3: Effect of training data on the susceptibility of the segmentation models to adversarial weather. Mean IoU scores are reported for three different test sets. We use the TBPSA method to perform the attack. We can see that while training on more diverse data improves the model robustness, adversarially crated weather parameters (last column) still cause a significant drop in model performance compared to uniformly sampled ones (middle column).

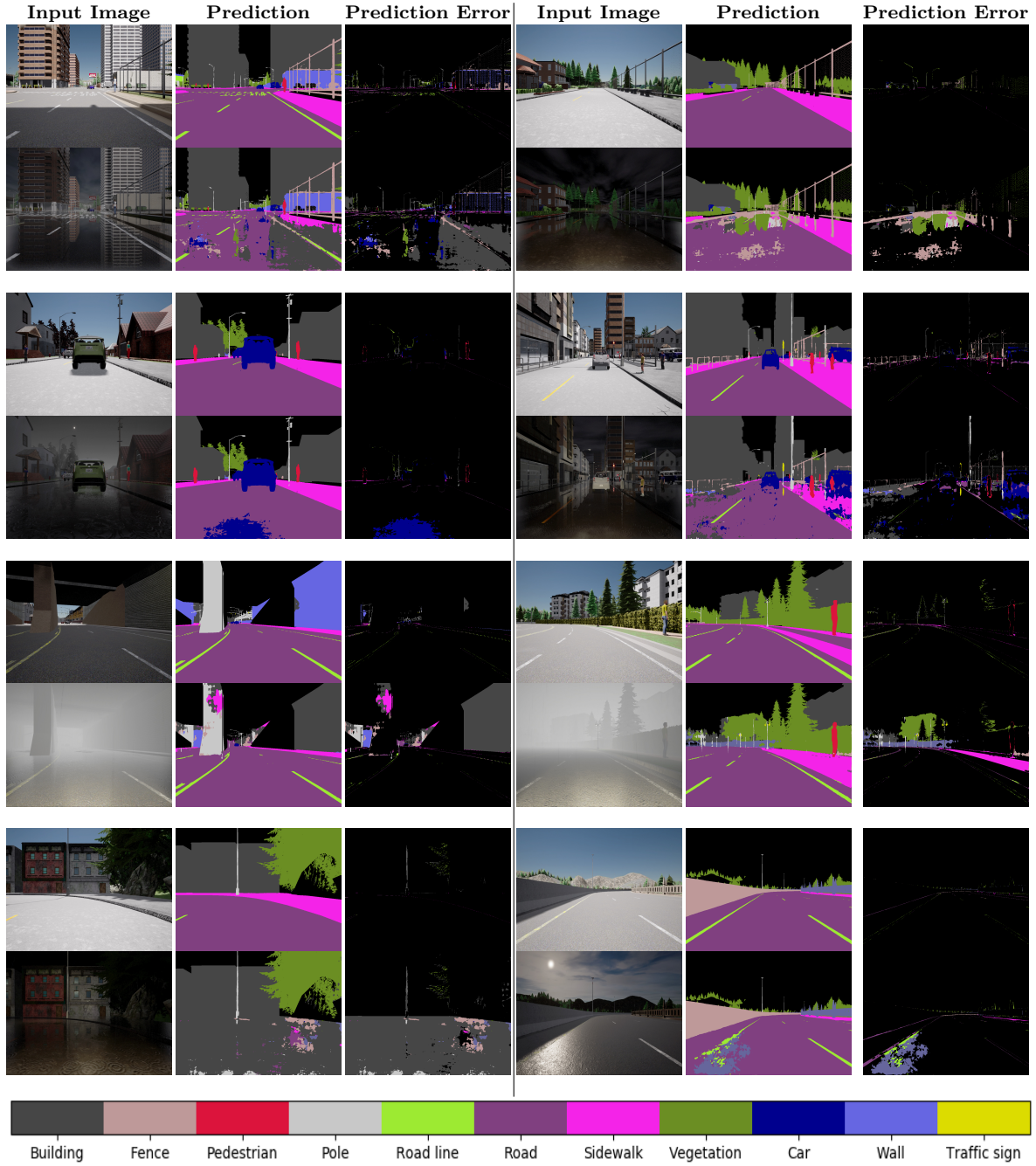
comparison of performance on *Clear*, *Uniform* and *Adversarial* test sets (using TBPSA) for models trained on different training distributions. We see that while all models have good performance on *Clear* test set, they improve significantly on *Uniform* test sets with increasing diversity of training data. However there is still a large gap in performance on the fixed uniform test sets compared to tailored adversarial test sets. The gap is larger for more biased models. For example *Gaussian* ( $\sigma=5.0$ ) model has a gap of nearly 30 mIoU points (60.95 vs 31.23). These results indicate that worst-case performance of the segmentation models is much worse than indicated by the *Uniform* test set. This highlights the need to identify and improve these worst-case failure modes before deploying these models for autonomous driving in the real-world.

### 9.3.3 Qualitative analysis of the failure modes

We will now look at qualitative cases of adverse weather found by our attack and understand the different types of failure modes discovered. Figure 9.2 shows examples of the adversarial weather created for the segmentation model trained on biased data (*Gaussian sigma* = 20). We see different mechanisms causing failures. In the first row we see that the reflections on the street caused by the rain-water puddling cause large segmentation failures. The model is unable to distinguish reflection from the real objects and erroneously segments them as real. In the second row, low light caused by night time cloudiness causes sporadic segmentation mistakes. Third row shows the cases where fog causes visibility issues leading to the model missing some background objects. Finally the last row shows that shiny surfaces caused by specular light and wetness causes model to hallucinate objects.

Model trained with *Uniform* data is more robust to these kinds of failures as we saw in quantitative evaluation. However our attack still finds interesting errors made by the uniform model, as seen in Figure 9.3. We see that while the errors are smaller, the model still shows difficulty in distinguishing reflections (first row). In the second and third rows we see interesting failures where cloudiness and positioning of the sun





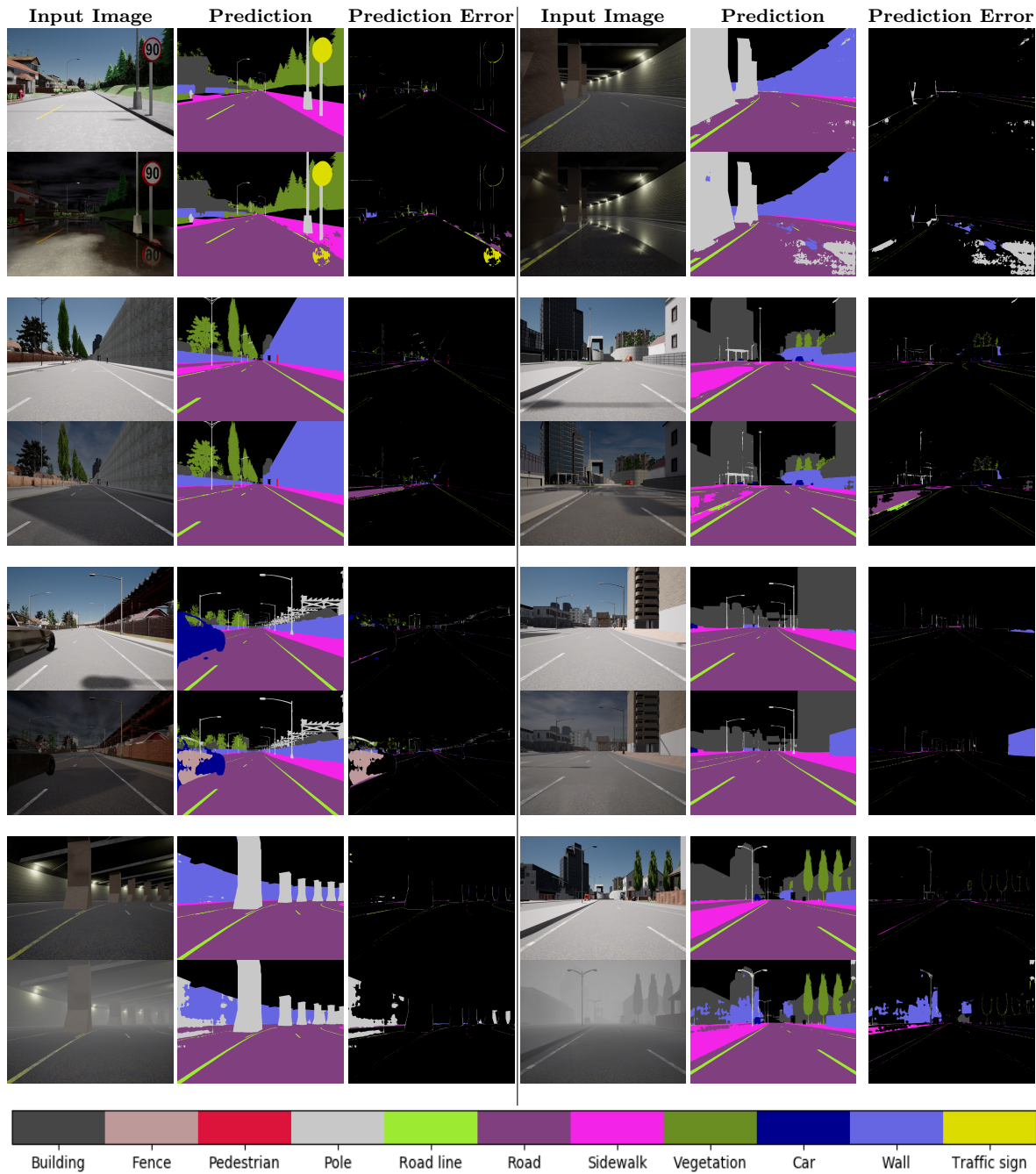


Figure 9.3: Examples of segmentation failures of the *Uniform* model caused by adverse weather found via our attack. Each pair of images show the original scene (top) and the scene in adversarial weather (bottom), along with the model prediction and errors in this prediction. We see that the model trained on *Uniform* data does better than the biased model. The errors are smaller. However, it shows failures with reflections (1st row), appearance change due to light (2nd row), shadows (3rd row) and fog (last row).

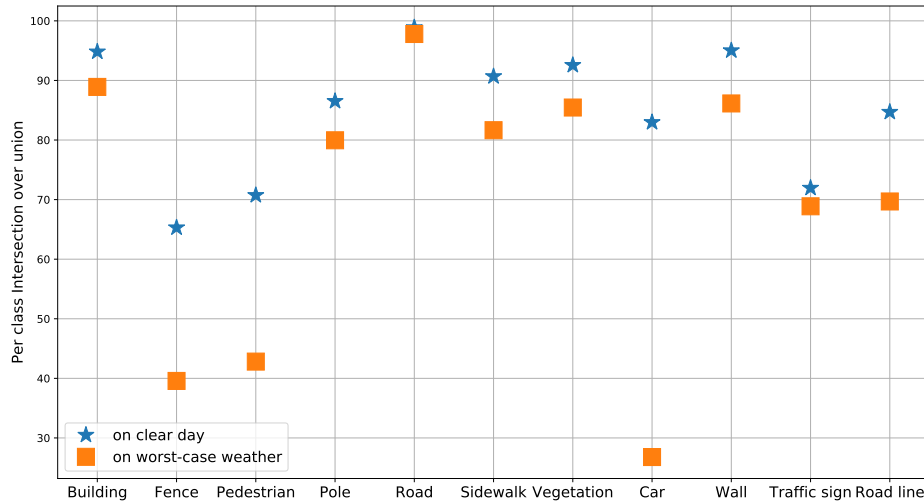


Figure 9.4: Class level performance drop caused by adversarial weathers on the model trained on *Uniform* data. Classes like car, pedestrian and fence are significantly affected.

causes shadows, leading the segmentation model to make mistakes. A car is seen as a fence (third row, first column) and a sidewalk is seen as drivable road (third row, second column). These kinds of mistakes would be truly dangerous in real world. We see that with adversarial attacks we can find the worst-case weather settings which induce errors, even on models trained on fully balanced data.

**Class-level effect.** Figure 9.4 shows the per class intersection over union on *Clear* weather as well as adversarial weather created by TBPSA attack. These results are for the model trained on *Uniform* data. Classes with biggest negative impact due to adversarial weather are pedestrian, fence and car. The model often misclassifies shiny surfaces caused by wetness as car, leading to a drop of more than 50 points in performance for this class. Pedestrian and fence classes are often affected by fog, and missed by the segmentation model. Lane markings (Road line) and wall classes also show more than a 10 point drop. In contrast the dominant road class is less affected, since it occupies a large area in most images.

## 9.4 CONCLUSIONS

In this chapter we proposed an approach to test the robustness of semantic segmentation models to adversarial weather conditions. We created targeted adverse weather configurations for a model for each scene, by adversarially optimizing the weather settings of the CARLA simulator. We show that gradient-free algorithms can successfully optimize the weather parameters through a non-differentiable simulator. In our experiments models show a significant drop in performance on the adverse weather settings created by our approach, compared to the performance on standard fixed test sets. This highlights the need to do worst-case testing before deploying models in the real world.



**Contents**


---

10.1 Key Contributions and Insights . . . . .	<b>143</b>
10.1.1 Automatic content manipulation . . . . .	143
10.1.2 Analyzing and improving robustness of computer vision systems	144
10.2 Future Perspectives . . . . .	<b>145</b>
10.2.1 Short-term research questions . . . . .	145
10.2.2 Long-term perspectives . . . . .	147

---

In this chapter we will summarize the key contributions made in the thesis as well as the insights derived from them. We will also discuss future research directions to conclude the thesis.

**10.1 KEY CONTRIBUTIONS AND INSIGHTS**

This thesis studied two concrete problems. First, we developed generative models to automatically and controllably edit text and image data. In both these domains, we developed adversarial training techniques to build content editing models using only unpaired datasets. Focusing on conditional editing enabled our generative models to operate on complex and unstructured datasets like COCO and ADE20k with crowded scenes. This was crucial in facilitating the second big focus area of the thesis – employing these generative models to measure and improve robustness of computer vision systems to semantic variations in the input image. By using the image editing models to create model-agnostic and model-specific test cases we were able to systematically analyze and improve robustness of various computer vision systems – including image classification, semantic segmentation, visual question answering and object detection systems – to context and appearance variations in the input. The following subsections will recap our key contributions in these two problem domains.

**10.1.1 Automatic content manipulation**

Lack of paired training data is the key challenge in learning content manipulation models. We overcome this challenge in the thesis by using adversarial training, combined with problem-specific constraints, to guide the generative model. We started our foray in this direction by building an adversarially trained image caption generator in Chapter 3. Despite having a discrete output space, we show that text generators can be trained in an adversarial framework by using Gumbel-Softmax approximation ([Jang \*et al.\*](#),

2016) to enable back-propagation. Key observation from this chapter is that adversarial training helps improve the diversity of generated captions, while maintaining same level of correctness.

In Chapter 4, we built a text style translation model A<sup>4</sup>NT, which obfuscates authorship of input text by mimicking a different style. We show that with adversarial training against attribute classifiers, combined with semantic consistency and language coherence losses, we can learn to perform text style transfer without paired data. In an application space without many automatic solutions, we show that our model learns to hide private attributes like age, gender and identity from the data, while mostly succeeding in retaining original meaning.

Switching to the image domain, we proposed an object removal model in Chapter 5 which learns from unpaired training data. Here, we show that we can avoid degenerate solutions with the right architectural constraints (two-staged model consisting of a mask generator with a binary output and an inpainter), and imposing a prior on the mask generator. We also show that this weakly-supervised (using image-level labels) model can achieve similar removal performance as a fully-supervised baseline (using segmentation labels), allowing us to operate on datasets without mask annotation. Part of this model also enabled our studies on context-sensitivity in Chapters 6 and 7.

Our final contribution in this direction is the object appearance editing model presented in Chapter 8. This model learns to disentangle the pose and appearance of an object, allowing it to edit its appearance while keeping the pose intact. We show that imposing appropriate bottlenecks at the output of two image encoders forces them to learn separate functions, one focusing on representing shape and the second on appearance. Here again adversarial losses, combined with cyclic reconstruction losses provide the training signal. This model is able to smoothly interpolate appearance of an object, which plays a key role in enabling semantic adversarial synthesis of corner-case object appearances to test the robustness of object detectors.

### 10.1.2 Analyzing and improving robustness of computer vision systems

The second important problem studied in this thesis is developing methods to test and improve robustness of computer vision systems to semantic variations in input. We propose a new approach in the thesis, and show that image editing models can be used to automatically synthesize new test cases and find failure modes of the target computer vision model. We studied two approaches to synthesize test cases – model-agnostic and model-specific. In model-agnostic approach we exhaustively synthesize all test-cases for a particular variation type and test the target model on them to find failure cases. This approach is taken in Chapters 6 and 7. In model-specific approach the synthesis is guided by the target model by adversarially optimizing the generator’s latent space to fool the target. This approach is taken in Chapters 8 and 9.

In Chapter 6, we measure the context-sensitivity of image classification and segmentation models by removing context objects. Using the edited images, we proposed simple metrics to quantify the robustness of classifiers and segmentation models. Our analysis show that, models rely too heavily on correlations between co-occurring objects to make



their predictions, causing failures when context changes. We show that our edited data is also useful beyond testing, and improves model generalization when used for data augmentation. We extended this analysis to VQA models in Chapter 7. We created two kinds of test sets, IV-VQA where edit is not expected to change the answer and CV-VQA where the answer changes in a controlled fashion. Analyzing three different VQA models using these test sets, we show that the modular SNMN model is more robust than other approaches, despite lower performance on standard i.i.d. test sets.

Our first model-specific test-case synthesis approach was presented in Chapter 8, where we developed and leveraged an object appearance editing model to synthesize hard corner cases. By adversarially optimizing the object appearance to fool the Yolov3 object detector, our test samples drop the performance of the detector by more than 20 mAP points. We show that by constraining the latent code during adversarial optimization to the convex-hull spanned by guiding instances, we can keep the generated appearances realistic to human eye. Qualitatively we show that this approach can automatically discover interesting failure modes including camouflaging, occlusion and confusing appearances.

In Chapter 9, we present a second model-specific testing approach, this time focusing on global appearance changes caused by weather. We use the CARLA simulator to create worst-case weather setting for a given scene, by adversarially optimizing the weather configuration to fool the segmentation model. We show that using gradient-free optimization technique we can effectively find worst-case weather settings for a scene, despite the simulator being a black box. Segmentation models show significant performance drop on these adversarially optimized weathers compared to standard fixed test sets. This highlights the usefulness of the model-specific testing approach in finding the limits of a model’s performance.

## 10.2 FUTURE PERSPECTIVES

Building robust computer vision models is an important and active research problem. This thesis took the first steps towards leveraging generative models to build automatic test-suites for measuring robustness of various computer vision systems. However, there is a long path forward to make generative-model driven testing more widely applicable. We will now discuss a few short-term follow up directions and longer term perspectives in the field.

### 10.2.1 Short-term research questions

**Improving authorship obfuscation.** In Chapter 4 we presented the A<sup>4</sup>NT model which obfuscates authorship by editing writing style while preserving content. While A<sup>4</sup>NT operated on sentence-level, we need obfuscated text to preserve coherence across the entire text. To achieve this the LSTM language generator used in our work should be upgraded to the latest transformers based language models which are better at modeling long-term relationships (Devlin *et al.*, 2019; Brown *et al.*, 2020). In some cases



obfuscation is not possible without changing the meaning, for example when an author talks about topics which are very predictive of their age. In such cases our A<sup>4</sup>NT model often alters the semantic content of the input. A better obfuscation model should instead detect these scenarios and warn the author that this content could compromise their privacy.

**Shape and geometric manipulations.** In this thesis we measured the robustness of computer vision models to three types of semantic variations – object removal (Chapter 6 and 7), appearance editing (Chapter 8) and weather simulation (Chapter 9). However, this only scratches the surface when it comes to possible semantic variations one could expect in natural scenes. Geometric and object shape variations are one such class to look at next. This includes 3D rotations, unusual poses for deformable objects like people and occlusions. To synthesize corner-case poses for an object, we need a generative model which can smoothly interpolate object poses, similar to our model in Chapter 8. This can be challenging due to space of valid poses often being discontinuous, for example for the person class. Synthesizing geometric variations can be especially useful to study the robustness of computer vision models like human pose detectors.

**Manipulations at the scene-graph level.** Object relationships is another dimension to consider when creating semantic variants of the image. This includes for example editing the image of *a man riding a bike* to a more unusual relationship like *a man pushing a bike*. We can generalize these manipulations as a task of first automatically editing image scene-graphs (Krishna *et al.*, 2017), and then generating images based on the new graph. Dhano *et al.* (2020) take a step in this direction, by building an interactive system where the user manipulates the scene-graph corresponding the input image and the model synthesizes the corresponding image. However to enable automated testing in this space, the scene-graph editing will need to be done by a generative model operating in graph domain. Generating objects in unusual relationships could be useful in testing scene-level computer vision systems like image captioning and visual question answering.

**Better data augmentation strategies.** A shortcoming of the work presented in this thesis is the simple data augmentation approach we took to utilize the synthetic hard examples generated by models. The edited semantic variants are not i.i.d. samples to the original training data, but have a very specific relationship to the original samples they are created from. Simply adding these samples back to the training set as independent samples might not make full use of them. Loss functions which explicitly exploit the relationship between the original and edited samples to further regularize the target model might be a good avenue to explore. Minimizing worst-case loss over semantic image variants could be a good candidate for this, given the success of this approach in Sagawa *et al.* (2020) to improve worst-group generalization. Condition variance penalties from Heinze-Deml and Meinshausen (2020) is also a suitable option since semantic editing creates exactly the two instance variants needed by this loss function.

**Robustness Certification in Semantic Space.** In the field of adversarial robustness, certified defenses have recently gained popularity (Raghunathan *et al.*, 2018; Sinha *et al.*, 2018; Wong and Kolter, 2018; Cohen *et al.*, 2019). These methods guarantee the neural network to be robust against any adversarial perturbation under a specific strength (in

terms of  $l_p$  norm). The guarantees are achieved by adversarial training (Sinha *et al.*, 2018) or by minimizing the worst-case loss over all reachable outputs under  $l_p$  bounded input perturbations. However these certificates only apply for norm-bounded adversary and can be broken with semantic perturbations (Ghiasi *et al.*, 2019). An interesting future work is to certify robustness to semantic variations, like the ones explored in this thesis. The certification could guarantee robustness to a particular type of editing or to a particular generative model. Mohapatra *et al.* (2020) take a first step in this direction, by certifying robustness to semantic perturbations like translations, 2D rotations, hue, saturation and contrast. It would be interesting to generalize this to plausible semantic variations reachable by a generative model.

### 10.2.2 Long-term perspectives

While the previous section has looked at immediate follow up directions related to the thesis, we will now present a longer-term speculative view on the path to build robust computer vision models. Looking forward, we believe there are two major directions of research – first on how to efficiently test and find failure modes of computer vision systems and second on how to learn better models to correct these failure modes.

**Content generation as a framework for mining hard examples.** This thesis mainly focused on using 2D image-domain generative models to synthesize test-cases. While these work well for creating local variations like object removal or changing appearance of one object, altering more complex relationships involving multiple objects in the scene is not possible with current generative models. For example, finding the worst case positions of objects in the scene which causes a detector failure. To generate such variations, the model should also respect physical constraints.

3D models and simulation would allow us to model such complex relations and scene level effects, while keeping physical constraints satisfied. However, traditional simulation scenarios are painstakingly hand-designed to match real-world data (e.g. the maps in the CARLA simulator). Procedural content generation (PCG) used in video games offers an alternative where things like game levels, maps etc. are programmatically generated on the fly from simple primitives. This can be done using search-based algorithms (Togelius *et al.*, 2011) or sampling from a learned model. Recently, procedural generation techniques have been also used to generate training data for machine learning models. Wang *et al.* (2019b, 2020) use evolution strategies to continuously generate new more difficult environments to train their reinforcement learning agent. In computer vision, domain randomization is used to improve generalization from simulation to real world data (Tobin *et al.*, 2017; Tremblay *et al.*, 2018). See Risi and Togelius (2020) for comprehensive overview on the current state and avenues for the use of PCG in machine learning.

A big challenge with using PCG to synthesize corner cases is still the combinatorial search space of all the simulation configuration. Even with a few objects, number of things one can change explodes and it becomes infeasible to brute force search for failure cases. Moreover, most of these variations are harmless and do not affect the model being tested too much. Simulators with differentiable rendering (Kato *et al.*, 2020) might

offer a way forward, by allowing us to directly optimize the simulation configuration to find the failure modes of the model being tested. Early works such as [Liu et al. \(2019\)](#); [Venkatesh et al. \(2020\)](#) explore this approach for simple single object settings. Extending this to full complex scenes and synthesizing worst-case scene configurations to find failure modes of computer vision systems is an exciting line of future work.

**Causal computer vision models and data generation.** Creating data variations is a useful mechanism to teach computer vision models to be invariant and hence robust to certain changes. Standard data augmentation does this for simple variations like translations, color jittering and rotations. This thesis showed that data augmentation with semantic variations can help improve generalization. However this process is still adhoc, needing careful balancing of the number of real and augmented samples to not lose i.i.d. performance.

Causality offers a principled framework to learn robust models and is a very promising research direction in the long-term [Schölkopf \(2019\)](#). One can think of images as a result of causal generative process containing many independent factors (location, objects, time of day, camera parameters) ([Lopez-Paz et al., 2017](#)). Only some of these factors are causal to the label (e.g. label cow is only caused by the object cow), while other could be spuriously correlated to the label in the current dataset (e.g. grass background is correlated with the label cow). Causal learning algorithms aim to learn models which discover the causal features from data and only rely on them to make their predictions. This also makes the model robust to variations in the spuriously correlated factors, like changes in context. Some of these algorithms ([Arjovsky et al., 2019](#); [Peters et al., 2016](#); [Heinze-Deml and Meinshausen, 2020](#)) assume access to a set of different training datasets where the causal relation is unchanged, but the strength of the spurious correlations vary. For example, two datasets one with cows often in pastures and other with cows often in the city. This kind of data is hard to manually curate for every factor of variation. Generative model based image editing might offer a scalable approach to create these data by intervening on non-causal factors. For example, we can edit backgrounds to create balanced sub-groups of data with every background. Recent work by [Sauer and Geiger \(2021\)](#) takes a step in this direction. They model three independent factors (object shape, texture and background) using a causal GAN, and use this to learn an ensemble of three classifiers which only rely on one of these factors each. These classifiers trained on counterfactual data show better out-of-distribution generalization. Extending this to general scenes, where causal structure is not known apriori is still a hard challenge.

## LIST OF FIGURES

---

1.1	Illustrative example of unusual appearance . . . . .	2
1.2	Overview of visual variations covered in the thesis . . . . .	7
3.1	Diversity across images in image captioning . . . . .	30
3.2	Diversity across multiple captions for each image . . . . .	30
3.3	Caption generator model . . . . .	32
3.4	Architecture of the discriminator network . . . . .	34
3.5	N-gram distribution comparison . . . . .	40
3.6	Qualitative results . . . . .	41
3.7	Vocabulary size as a function of word counts. . . . .	42
4.1	GAN framework to train our A <sup>4</sup> NT network . . . . .	49
4.2	Block diagram of the attribute classifier network . . . . .	50
4.3	Block diagram of the A <sup>4</sup> NT network . . . . .	52
4.4	Cyclic semantic loss . . . . .	54
4.5	Semantic consistency using cyclic reconstruction . . . . .	55
4.6	Operating points of A <sup>4</sup> NT models on test set. . . . .	65
4.7	Privacy and semantic consistency of A <sup>4</sup> NT and baselines . . . . .	65
4.8	Output Privacy vs Privacy on Input. . . . .	68
4.9	Meteor score plotted against input difficulty. . . . .	68
4.10	Histogram of privacy gain . . . . .	69
5.1	Overview of object removal model . . . . .	76
	(a) Our editor is composed of a mask-generator and an image in-painter	76
	(b) Two-stage generator avoids adversarial patterns . . . . .	76
5.2	Imposing mask priors with a GAN framework . . . . .	78
5.3	Qualitative examples of removal of different object classes . . . . .	82
5.4	Results of logo removal . . . . .	82
5.5	Effect of priors on generated masks . . . . .	82
5.6	Comparing global and local GAN loss . . . . .	86
6.1	Example of context sensitivity in semantic segmentation models . . . . .	89
6.2	Context violations by image-level classifier . . . . .	92
6.3	Qualitative examples of segmentation failures due to object removal . . . . .	94
6.4	Comparing per-class average precision to robustness metrics . . . . .	95
6.5	Per-class violations in segmentation models . . . . .	96
6.6	Context sensitivity of across classes . . . . .	101
7.1	VQA models exploiting spurious correlations . . . . .	104
7.2	Qualitative analysis of VQA spurious correlations . . . . .	109
7.3	Effect of fine-tuning on VQA robustness . . . . .	114
7.4	Qualitative results from fine-tuning . . . . .	114
8.1	Illustrating the idea of automated adversarial testing . . . . .	119
8.2	Our overall pipeline for creating the semantic adversaries . . . . .	120
8.3	Synthesizer network architecture . . . . .	120

8.4	Appearance interpolation comparison . . . . .	121
8.5	Intermediate steps when optimizing the appearance to fool the detector. .	122
8.6	Failure cases discovered by our semantic adversary . . . . .	126
8.7	Comparing the typicality rating between real data and semantic adversary.	128
8.8	Comparing adversarial strategies and template baselines . . . . .	128
9.1	Overview of the semantic adversarial weather synthesis . . . . .	134
9.2	Segmentation failures of biased model on adverse weather . . . . .	139
9.3	Segmentation failures of uniform model on adverse weather . . . . .	140
9.4	Class level effect of adversarial weathers . . . . .	141

## LIST OF TABLES

---

Tab. 3.1	Comparing performance of baseline and adversarial models . . . . .	38
Tab. 3.2	Category-level Spice metric comparison . . . . .	38
Tab. 3.3	Human evaluation of caption correctness . . . . .	39
Tab. 3.4	Diversity metrics . . . . .	39
Tab. 3.5	Ablation study . . . . .	43
Tab. 4.1	Comparing statistics of the two datasets. . . . .	58
Tab. 4.2	F1-scores of the attribute classifiers . . . . .	61
Tab. 4.3	Performance of style transfer anonymization . . . . .	61
Tab. 4.4	A <sup>4</sup> NT anonymization against unseen (holdout) classifiers . . . . .	64
Tab. 4.5	Comparing A <sup>4</sup> NT model to prior work . . . . .	64
Tab. 4.6	User study to judge semantic similarity . . . . .	66
Tab. 4.7	User study of A <sup>4</sup> NT and the Google MT baseline . . . . .	67
Tab. 4.8	Qualitative examples of style transfer . . . . .	70
Tab. 4.9	Style transfer on the speech dataset . . . . .	71
Tab. 5.1	Quantifying the effect of using more accurate mask priors . . . . .	84
Tab. 5.2	Comparison to weakly supervised segmentation . . . . .	84
Tab. 5.3	Comparison to ground truth masks and Mask-RCNN baselines. . . . .	85
Tab. 5.4	Evaluating in-painting components . . . . .	86
Tab. 5.5	Joint training helps improve both mask generation and in-painting . . . . .	86
Tab. 6.1	Effect of data augmentation on classification model . . . . .	95
Tab. 6.2	Segmentation dependence on cars on real data . . . . .	96
Tab. 6.3	Data augmentation results on ADE20k dataset. . . . .	99
Tab. 6.4	Experiments in three class setting on ADE20k. . . . .	101
Tab. 7.1	Vocabulary mapping from QA space to COCO . . . . .	107
Tab. 7.2	IV-VQA and CV-VQA data statistics . . . . .	108
Tab. 7.3	Performance of VQA baselines . . . . .	110
Tab. 7.4	Accuracy-flipping on real data/IV-VQA test set. . . . .	111
Tab. 7.5	Accuracy-flipping on real data/CV-VQA test set. . . . .	112
Tab. 8.1	Detector performance under semantic adversarial editing . . . . .	125
Tab. 8.2	Human study of label correctness . . . . .	125
Tab. 8.3	Data augmentation results on COCO . . . . .	130
Tab. 8.4	Data augmentation results on VOC . . . . .	130
Tab. 8.5	Data augmentation results on BDD100k dataset. . . . .	130
Tab. 9.1	Comparison of attack mechanisms . . . . .	134
Tab. 9.2	Comparison of attack mechanisms . . . . .	137
Tab. 9.3	Effect of training data on robustness to weather . . . . .	138





## BIBLIOGRAPHY

---

- A. Abbasi and H. Chen (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace, *ACM Transactions on Information Systems (TOIS)*. 22
- S. Afroz, M. Brennan, and R. Greenstadt (2012). Detecting hoaxes, frauds, and deception in writing style online, in *Security and Privacy (SP), 2012 IEEE Symposium on 2012*. 46
- S. Afroz, A. C. Islam, A. Stolerma, R. Greenstadt, and D. McCoy (2014). Doppelgänger finder: Taking stylometry to the underground, in *Security and Privacy (SP), 2014 IEEE Symposium on 2014*. 22
- V. Agarwal, R. Shetty, and M. Fritz (2020). Towards Causal VQA: Revealing and Reducing Spurious Correlations by Invariant and Covariant Semantic Editing, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020*. 11
- E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation, in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) 2016*. 66
- A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi (2018). Don’t Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 21, 25, 105, 110
- R. Alaifari, G. S. Alberti, and T. Gauksson (2019). ADef: an iterative algorithm to construct adversarial deformations, *Proceedings of the International Conference on Learning Representations (ICLR)*. 18
- M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen (2019). Strike (With) a Pose: Neural Networks Are Easily Fooled by Strange Poses of Familiar Objects, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 18
- P. Anderson, B. Fernando, M. Johnson, and S. Gould (2016). SPICE: Semantic Propositional Image Caption Evaluation, in *Proceedings of the European Conference on Computer Vision (ECCV) 2016*. 37
- J. Andreas and D. Klein (2016). Reasoning about pragmatics with neural listeners and speakers, in *emnlp 2016*. 33

- J. Andreas, M. Rohrbach, T. Darrell, and D. Klein (2016). Neural module networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 21, 110
- S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh (2015). VQA: Visual Question Answering, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2015*. 21
- S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler (2009). Automatically profiling the author of an anonymous text, *Communications of the ACM*. 22, 46, 61
- M. Arjovsky and L. Bottou (2017). Towards principled methods for training generative adversarial networks, in *Proceedings of the International Conference on Learning Representations (ICLR) 2017*. 14, 36
- M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz (2019). Invariant risk minimization, *arXiv:1907.02893*. 24, 148
- M. Arjovsky, S. Chintala, and L. Bottou (2017). Wasserstein Generative Adversarial Networks, in *Proceedings of the International Conference on Machine Learning (ICML) 2017*. 78
- D. V. Arnold (2012). *Noisy optimization with evolution strategies*, Springer Science & Business Media. 134
- O. Ashual and L. Wolf (2019). Specifying object attributes and relations in interactive scene generation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2019*. 6
- A. Azulay and Y. Weiss (2019). Why do deep convolutional networks generalize so poorly to small image transformations?, *Journal of Machine Learning Research (JMLR)*. 18
- D. Bagnall (2015). Author identification using multi-headed recurrent neural networks, *arXiv preprint arXiv:1506.04891*. 22, 49, 50
- D. Bahdanau, K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate, *Proceedings of the International Conference on Learning Representations (ICLR)*. 23, 48, 51
- A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz (2019). Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models, in *Advances in Neural Information Processing Systems (NeurIPS) 2019*. 18, 73
- E. Barenholtz (2014). Quantifying the role of context in visual object recognition, *Visual Cognition*, vol. 22(1). 18

- D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba (2017). Network Dissection: Quantifying Interpretability of Deep Visual Representations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 19
- S. Beery, G. Van Horn, and P. Perona (2018). Recognition in terra incognita, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 8
- S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick (2016). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 18, 88
- S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan (2010). A theory of learning from different domains, *Machine learning*, vol. 79(1-2), pp. 151–175. 8, 17
- S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer (2015). Scheduled sampling for sequence prediction with recurrent neural networks, in *Advances in Neural Information Processing Systems (NeurIPS) 2015*. 20
- L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. v. d. Oord (2020). Are we done with ImageNet?, *arXiv preprint arXiv:2006.07159*. 2
- S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning (2015). A large annotated corpus for learning natural language inference, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 55
- M. Brennan, S. Afroz, and R. Greenstadt (2012). Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity, *ACM Transactions on Information and System Security (TISSEC)*. 46, 47
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.* (2020). Language models are few-shot learners, *arXiv preprint arXiv:2005.14165*. 1, 145
- A. Caliskan and R. Greenstadt (2012). Translate Once, Translate Twice, Translate Thrice and Attribute: Identifying Authors and Machine Translation Tools in Translated Text, in *2012 IEEE Sixth International Conference on Semantic Computing 2012*. 46
- A. Caliskan-Islam, R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi, and R. Greenstadt (2015). De-anonymizing programmers via code stylometry, in *USENIX Security Symposium 2015*. 22
- N. Carlini and D. Wagner (2017). Towards evaluating the robustness of neural networks, in *Security and Privacy (SP), 2017 IEEE Symposium on 2017*. 8, 17
- D. Castro, R. Ortega, and R. Muñoz (2017). Author Masking by Sentence Transformation, in *Working notes of Conference and Labs of the Evaluation (CLEF) 2017*. 46, 47

- T. Cazenave, M. Oquab, J. Rapin, and O. Teytaud (2019). Parallel Noisy Optimization in Front of Simulators: Optimism, Pessimism, Repetitions Population Control, in *proceedings of CEC 2019 workshops 2019*. 132, 133, 135
- C. Chan, S. Ginosar, T. Zhou, and A. A. Efros (2019). Everybody dance now, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2019*. 1
- Z. Che, G. Li, T. Li, B. Jiang, X. Shi, X. Zhang, Y. Lu, G. Wu, Y. Liu, and J. Ye (2019). D2-City: A Large-Scale Dashcam Video Dataset of Diverse Traffic Scenarios, *arXiv preprint arXiv:1904.01975*. 125
- L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40. 19
- Q. Chen and V. Koltun (2017). Photographic Image Synthesis With Cascaded Refinement Networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 6
- X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick (2015). Microsoft COCO Captions: Data Collection and Evaluation Server, *arXiv preprint arxiv:1504.00325*. 6, 36, 75, 80
- Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo (2018). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 6, 14, 15, 75, 76, 77
- COCO (2017). *Microsoft COCO Image Captioning Challenge*, <https://competitions.codalab.org/competitions/3221#results>. 29
- J. Cohen, E. Rosenfeld, and Z. Kolter (2019). Certified adversarial robustness via randomized smoothing, in *International Conference on Machine Learning 2019*. 146
- A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*. 55
- M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 3
- A. Criminisi, P. Pérez, and K. Toyama (2004). Region filling and object removal by exemplar-based image inpainting, *IEEE Transactions on image processing*. 16

- E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le (2019). Autoaugment: Learning augmentation strategies from data, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2019*. 25, 26, 129, 130
- B. Dai, S. Fidler, R. Urtasun, and D. Lin (2017). Towards diverse and natural image descriptions via a conditional gan, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 14, 15, 20, 33
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. 2, 17, 21, 94
- M. Denkowski and A. Lavie (2014a). Meteor Universal: Language Specific Translation Evaluation for Any Target Language, *ACL 2014*. 29, 37
- M. Denkowski and A. Lavie (2014b). Meteor Universal: Language Specific Translation Evaluation for Any Target Language, in *Proceedings of the Ninth Workshop on Statistical Machine Translation 2014*. 59
- E. L. Denton, S. Chintala, R. Fergus, *et al.* (2015). Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks, in *Advances in Neural Information Processing Systems (NeurIPS) 2015*. 14, 29
- C. Desai, D. Ramanan, and C. C. Fowlkes (2011). Discriminative models for multi-class object layout, *International Journal of Computer Vision (IJCV)*, vol. 95(1). 18
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) 2019*. 145
- J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell (2015). Language Models for Image Captioning: The Quirks and What Works, in *ACL 2015*. 19
- T. DeVries and G. W. Taylor (2017). Improved regularization of convolutional neural networks with cutout, *arXiv preprint arXiv:1708.04552*. 25
- H. Dhamo, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht (2020). Semantic Image Manipulation Using Scene Graphs, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020*. 15, 146
- J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell (2015). Long-term recurrent convolutional networks for visual recognition and description, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 19, 31

- J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell (2016). Long-term Recurrent Convolutional Networks for Visual Recognition and Description, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 33
- J. Dong and T. Lin (2019). Marginan: Adversarial training in semi-supervised learning, *Advances in neural information processing systems*. 5
- A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun (2017). CARLA: An Open Urban Driving Simulator, in *Proceedings of the 1st Annual Conference on Robot Learning 2017*. 9, 131, 132
- B. Dumont, S. Maggio, and P. Montalvo (2018). Robustness of rotation-equivariant networks to adversarial perturbations, *arXiv preprint arXiv:1802.06627*. 18
- T. Durand, N. Thome, and M. Cord (2016). Weldon: Weakly supervised learning of deep convolutional neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 18
- N. Dvornik, J. Mairal, and C. Schmid (2018). Modeling Visual Context Is Key to Augmenting Object Detection Datasets, in *Computer Vision – ECCV 2018 2018*. 25
- D. Dwibedi, I. Misra, and M. Hebert (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 25
- D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia (2017). Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description, in *Proceedings of the Second Conference on Machine Translation 2017*. 59
- L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry (2019). Exploring the Landscape of Spatial Robustness, in *Proceedings of the International Conference on Machine Learning (ICML) 2019*. 18
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>. 80, 124
- A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth (2010). Every Picture Tells a Story: Generating Sentences from Images, in *Proceedings of the European Conference on Computer Vision (ECCV) 2010*. 19
- W. Fedus, I. Goodfellow, and A. Dai (2018). MaskGAN: Better Text Generation via Filling in the \_\_\_\_\_, *Proceedings of the International Conference on Learning Representations (ICLR)*. 15



- J. R. Finkel, T. Grenager, and C. Manning (2005). Incorporating non-local information into information extraction systems by gibbs sampling, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) 2005*. 57
- Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan (2018). Style Transfer in Text: Exploration and Evaluation, in *Proceedings of the Conference on Artificial Intelligence (AAAI) 2018*. 15
- A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach (2016). Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, in *EMNLP 2016*. 110
- Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). Domain-adversarial training of neural networks, *Journal of Machine Learning Research (JMLR)*. 5
- H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu (2015). Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering, in *Advances in Neural Information Processing Systems (NeurIPS) 2015*. 21
- L. Gatys, A. Ecker, and M. Bethge (2015). A Neural Algorithm of Artistic Style, *Nature Communications*. 78
- R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann (2020). Shortcut learning in deep neural networks, *Nature Machine Intelligence*. 17
- R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness., in *Proceedings of the International Conference on Learning Representations (ICLR) 2019*. 25, 127
- A. Ghiasi, A. Shafahi, and T. Goldstein (2019). Breaking Certified Defenses: Semantic Adversarial Examples with Spoofed Robustness Certificates, in *International Conference on Learning Representations 2019*. 147
- T. Gokhale, P. Banerjee, C. Baral, and Y. Yang (2020). MUTANT: A Training Paradigm for Out-of-Distribution Generalization in Visual Question Answering, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*. 21, 25
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets, in *Advances in Neural Information Processing Systems (NeurIPS) 2014*. 6, 13, 14, 27, 29, 31, 47, 53, 118
- I. Goodfellow, J. Shlens, and C. Szegedy (2015). Explaining and Harnessing Adversarial Examples, in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*. 17



- Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh (2017). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 21, 106, 107, 108, 110
- H. Grassegger and M. Krogerus (2017). *The Data That Turned the World Upside Down*, [https://motherboard.vice.com/en\\_us/article/mg9vvv/how-our-likes-helped-trump-win](https://motherboard.vice.com/en_us/article/mg9vvv/how-our-likes-helped-trump-win). 46
- A. Hamdi and B. Ghanem (2019). Towards Analyzing Semantic Robustness of Deep Neural Networks, *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 18
- N. Hansen and A. Ostermeier (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, in *Proceedings of IEEE international conference on evolutionary computation 1996*. 135
- J. Hays and A. A. Efros (2007). Scene completion using millions of photographs, in *ACM Transactions on Graphics (TOG) 2007*. 16
- K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017). Mask R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 6, 16, 75, 81
- K. He, X. Zhang, S. Ren, and J. Sun (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2015*. 17
- K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 21, 36, 42
- C. Heinze-Deml and N. Meinshausen (2020). Conditional variance penalties and domain shift robustness, *Machine Learning*. 24, 146, 148
- M. Hellwig and H.-G. Beyer (2016). Evolution under strong noise: A self-adaptive evolution strategy can reach the lower performance bound-the pccmsa-es, in *International Conference on Parallel Problem Solving from Nature 2016*. 135
- L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell (2016). Generating Visual Explanations, in *Proceedings of the European Conference on Computer Vision (ECCV) 2016*. 33
- L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach (2018). Women also snowboard: Overcoming bias in captioning models, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 19
- J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, and V. Fischer (2017). Universal adversarial perturbations against semantic image segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 17

- D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, *et al.* (2020). The many faces of robustness: A critical analysis of out-of-distribution generalization, *arXiv preprint arXiv:2006.16241*. 25, 73
- D. Hendrycks and T. Dietterich (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, in *Proceedings of the International Conference on Learning Representations (ICLR) 2019*. 8, 17, 25, 118
- D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan (2019). AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty, in *Proceedings of the International Conference on Learning Representations (ICLR) 2019*. 25
- D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song (2021). Natural Adversarial Examples, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2021*. 18, 73, 118
- G. E. Hinton, T. J. Sejnowski, *et al.* (1986). Learning and relearning in Boltzmann machines, *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1(282-317). 13
- T. Hinz, S. Heinrich, and S. Wermter (2020). Semantic Object Accuracy for Generative Text-to-Image Synthesis, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 6
- S. Hochreiter and J. Schmidhuber (1997). Long short-term memory, *Neural computation*. 21, 50
- H. Hosseini and R. Poovendran (2018). Semantic adversarial examples, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops) 2018*. 18, 126
- R. Hu (2018). Official code release for Explainable Neural Computation via Stack Neural Module Networks, <https://github.com/ronghanghu/snm>. 110
- R. Hu, J. Andreas, T. Darrell, and K. Saenko (2018). Explainable Neural Computation via Stack Neural Module Networks, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 21, 105, 110
- R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko (2017). Learning to Reason: End-to-End Module Networks for Visual Question Answering, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 21
- R. Huang, S. Zhang, T. Li, and R. He (2017a). Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 75

- X. Huang and S. Belongie (2017). Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 1
- X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie (2017b). Stacked generative adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 15
- D. A. Hudson and C. D. Manning (2018). Compositional Attention Networks for Machine Reasoning, in *Proceedings of the International Conference on Learning Representations (ICLR) 2018*. 21
- D. A. Hudson and C. D. Manning (2019). GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 21
- F. Huszar (2015). How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary?, *arXiv preprint arXiv:1511.05101*. 20
- S. Iizuka, E. Simo-Serra, and H. Ishikawa (2017). Globally and locally consistent image completion, *ACM Transactions on Graphics (TOG)*. 78, 79
- K. Ikeda, G. Hattori, C. Ono, H. Asoh, and T. Higashino (2013). Twitter User Profiling Based on Text and Community Mining for Market Analysis, *Knowledge-Based Systems*. 46
- P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros (2016). Image-to-image translation with conditional adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 15, 43, 79
- M. Jain, J. C. van Gemert, and C. G. Snoek (2015). What do 15,000 object categories tell us about classifying and localizing actions?, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 18
- T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi (2018). Unsupervised learning of object landmarks through conditional image generation, in *Advances in Neural Information Processing Systems (NeurIPS) 2018*. 16, 120
- E. Jang, S. Gu, and B. Poole (2016). Categorical Reparameterization with Gumbel-Softmax, *Proceedings of the International Conference on Learning Representations (ICLR)*. 5, 15, 20, 27, 29, 33, 53, 121, 143
- R. Jia and P. Liang (2017). Adversarial Examples for Evaluating Reading Comprehension Systems, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing 2017*. 23
- J. Johnson, A. Gupta, and L. Fei-Fei (2018). Image Generation from Scene Graphs, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 6, 15

- P. Juola (2013). *How a Computer Program Helped Show J.K. Rowling write A Cuckoo's Calling*, <https://goo.gl/mkZai1>. 46
- P. Juola *et al.* (2008). Authorship attribution, *Foundations and Trends® in Information Retrieval*. 46
- G. Kacmarcik and M. Gamon (2006). Obfuscating document stylometry to preserve author anonymity, in *Proceedings of the COLING/ACL on Main conference poster sessions 2006*. 22
- K. Kafle and C. Kanan (2017). Visual question answering: Datasets, algorithms, and future challenges, in *Computer Vision and Image Understanding (CVIU) 2017*. 21
- K. Kafle, M. Yousefhussien, and C. Kanan (2017). Data Augmentation for Visual Question Answering, in *Proceedings of the International Conference on Natural Language Generation (INLG) 2017*. 24, 105
- Y. Kalantidis, L. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis (2011). Scalable Triangulation-based Logo Recognition, in *Proceedings of ACM International Conference on Multimedia Retrieval (ICMR) 2011*. 80
- G. Karadzhov, T. Mihaylova, Y. Kiprova, G. Georgiev, I. Koychev, and P. Nakov (2017). The Case for Being Average: A Mediocrity Approach to Style Masking and Author Obfuscation, in *International Conference of the Cross-Language Evaluation Forum for European Languages 2017*. 23, 51, 63, 64
- A. Karpathy and L. Fei-Fei (2015). Deep visual-semantic alignments for generating image descriptions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 19, 36
- T. Karras, T. Aila, S. Laine, and J. Lehtinen (2018). Progressive growing of gans for improved quality, stability, and variation, *Proceedings of the International Conference on Learning Representations (ICLR)*. 14, 15
- T. Karras, S. Laine, and T. Aila (2019). A style-based generator architecture for generative adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 1, 6, 14, 15, 27
- T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila (2020). Analyzing and improving the image quality of stylegan, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 14
- H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon (2020). Differentiable rendering: A survey, *arXiv preprint arXiv:2006.12057*. 147
- V. Kazemi and A. Elqursh (2017). Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering, in *ArXiv 2017*. 21, 105, 110

- Y. Keswani, H. Trivedi, P. Mehta, and P. Majumder (2016). Author Masking through Translation., in *Working notes of Conference and Labs of the Evaluation (CLEF) 2016*. 23, 46, 59
- A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele (2017). Simple Does It: Weakly Supervised Instance and Semantic Segmentation., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 81, 84
- J. Kiefer, J. Wolfowitz, *et al.* (1952). Stochastic estimation of the maximum of a regression function, *The Annals of Mathematical Statistics*. 135
- D. P. Kingma and M. Welling (2013). Auto-encoding variational bayes, *Proceedings of the International Conference on Learning Representations (ICLR)*. 13
- R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler (2015). Skip-thought vectors, in *Advances in Neural Information Processing Systems (NeurIPS) 2015*. 55
- M. Klingemann (2018). *Memories of Passersby I*, <http://quasimondo.com/>. 1, 14
- R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.* (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations, *International Journal of Computer Vision (IJCV)*. 146
- A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems (NeurIPS)*. 24
- G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg (2013). Babytalk: Understanding and generating simple image descriptions, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 19
- M. J. Kusner and J. M. Hernández-Lobato (2016). GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution, *arXiv preprint arXiv:1611.04051*. 15
- G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, *et al.* (2017). Fader networks: Manipulating images by sliding attributes, in *Advances in Neural Information Processing Systems (NeurIPS) 2017*. 75, 76
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, pp. 2278–2324. 21, 25
- J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan (2016). A Diversity-Promoting Objective Function for Neural Conversation Models, in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2016*. 20, 29

- J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky (2017). Adversarial Learning for Neural Dialogue Generation, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 14, 20, 33, 36
- Y.-J. Li, C.-S. Lin, Y.-B. Lin, and Y.-C. F. Wang (2019). Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2019*. 16
- Z. Li, X. Jiang, L. Shang, and H. Li (2018). Paraphrase Generation with Deep Reinforcement Learning, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*. 59
- B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi (2018). Deep Text Classification Can be Fooled, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2018*. 17, 23
- Z. Liang, W. Jiang, H. Hu, and J. Zhu (2020). Learning to Contrast the Counterfactual Samples for Robust Visual Question Answering, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*. 25
- T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014a). Microsoft COCO: Common objects in context, in *Proceedings of the European Conference on Computer Vision (ECCV) 2014*. 3, 17, 93, 124
- T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014b). Microsoft COCO: Common Objects in Context, in *Proceedings of the European Conference on Computer Vision (ECCV) 2014*. 106
- G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro (2018). Image Inpainting for Irregular Holes Using Partial Convolutions, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 78, 79
- H.-T. D. Liu, M. Tao, C.-L. Li, D. Nowrouzezahrai, and A. Jacobson (2019). Beyond Pixel Norm-Balls: Parametric Adversaries using an Analytically Differentiable Renderer, in *Proceedings of the International Conference on Learning Representations (ICLR) 2019*. 18, 148
- S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy (2017). Improved image captioning via policy gradient optimization of SPIDeR, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 20
- J. Long, E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 18
- D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou (2017). Discovering Causal Signals in Images, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 148



- D. Lorenz, L. Bereska, T. Milbich, and B. Ommer (2019). Unsupervised Part-Based Disentangling of Object Shape and Appearance, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 16, 120, 121
- J. Lu, X. Lin, D. Batra, and D. Parikh (2015). *Deeper LSTM and normalized CNN Visual Question Answering model*, [https://github.com/VT-vision-lab/VQA\\_LSTM\\_CNN](https://github.com/VT-vision-lab/VQA_LSTM_CNN). 21, 110
- J. Lu, C. Xiong, D. Parikh, and R. Socher (2017). Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 38
- J. Lu, J. Yang, D. Batra, and D. Parikh (2016). Hierarchical Question-image Co-attention for Visual Question Answering, in *Advances in Neural Information Processing Systems (NeurIPS) 2016*. 21, 110
- P. Luc, C. Couprie, S. Chintala, and J. Verbeek (2016). Semantic Segmentation using Adversarial Networks, in *Advances in Neural Information Processing Systems Workshops(NeurIPS-W) 2016*. 14, 29
- L. Ma, Z. Lu, and H. Li (2016). Learning to Answer Questions from Image Using Convolutional Neural Network, in *Proceedings of the Conference on Artificial Intelligence (AAAI) 2016*. 21
- X. Ma and E. Hovy (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) 2016*. 51
- C. J. Maddison, A. Mnih, and Y. W. Teh (2016). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables, *Proceedings of the International Conference on Learning Representations (ICLR)*. 15, 29, 33, 121
- A. Makazhanov, D. Rafiei, and M. Waqar (2014). Predicting political preference of Twitter users, *Social Network Analysis and Mining*. 46
- M. Malinowski and M. Fritz (2014). A multi-world approach to question answering about real-world scenes based on uncertain input, in *Advances in Neural Information Processing Systems (NeurIPS) 2014*. 21
- M. Malinowski, M. Rohrbach, and M. Fritz (2015). Ask Your Neurons: A Neural-based Approach to Answering Questions about Images, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2015*. 21
- V. Manjunatha, N. Saini, and L. S. Davis (2019). Explicit Bias Discovery in Visual Question Answering Models, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 21



- X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley (2017). Least squares generative adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 79
- M. Marszalek and C. Schmid (2007). Semantic hierarchies for visual object recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2007*. 18
- M. Mathieu, C. Couprie, and Y. LeCun (2016). Deep multi-scale video prediction beyond mean square error, *Proceedings of the International Conference on Learning Representations (ICLR)*. 43
- A. W. McDonald, S. Afroz, A. Caliskan, A. Stolerma, and R. Greenstadt (2012). Use Fewer Instances of the Letter "i": Toward Writing Style Anonymization., in *Privacy Enhancing Technologies 2012*. 22, 46, 47, 51
- C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel (2019). Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming, *arXiv preprint arXiv:1907.07484*. 17, 118
- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed Representations of Words and Phrases and their Compositionality, in *Advances in Neural Information Processing Systems (NeurIPS) 2013*. 50
- S. S. Mirkamali and P. Nagabhushan (2015). Object removal by depth-wise image inpainting, *Signal, Image and Video Processing*. 16
- M. Mirza and S. Osindero (2014). Conditional Generative Adversarial Nets, *arXiv preprint arXiv:1411.1784*. 14, 15
- J. Mohapatra, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel (2020). Towards Verifying Robustness of Neural Networks Against A Family of Semantic Perturbations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020*. 147
- A. A. Morgan-Lopez, A. E. Kim, R. F. Chew, and P. Ruddle (2017). Predicting age groups of Twitter users based on language and metadata features, *PLOS ONE*. 46, 48
- F. Mosteller and D. L. Wallace (1963). Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers, *Journal of the American Statistical Association*. 22
- R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille (2014). The role of context for object detection and semantic segmentation in the wild, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. 88, 98

- K. Muandet, D. Balduzzi, and B. Schölkopf (2013). Domain generalization via invariant feature representation, in *Proceedings of the International Conference on Machine Learning (ICML) 2013*. 8
- A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song (2012). On the feasibility of internet-scale author identification, in *Security and Privacy (SP), 2012 IEEE Symposium on 2012*. 22, 46
- K. Nazeri, E. Ng, and M. Ebrahimi (2018). Image colorization using generative adversarial networks, in *International conference on articulated motion and deformable objects 2018*. 1, 14
- E. Ntavelis, A. Romero, I. Kastanis, L. Van Gool, and R. Timofte (2020). SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing Objects, in *Proceedings of the European Conference on Computer Vision (ECCV) 2020*. 15
- B. M. Oh, M. Chen, J. Dorsey, and F. Durand (2001). Image-based modeling and photo editing, in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques 2001*. 4
- M. Oquab, L. Bottou, I. Laptev, and J. Sivic (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 18, 94, 95
- T. Orekondy, M. Fritz, and B. Schiele (2018). Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 4, 75
- R. Overdorf and R. Greenstadt (2016). Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution, *Proceedings on Privacy Enhancing Technologies*. 46
- D. Parikh, C. L. Zitnick, and T. Chen (2012). Exploring tiny images: The roles of appearance and contextual information for machine and human object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34(10). 18, 88
- T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu (2019). Semantic Image Synthesis with Spatially-Adaptive Normalization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 121
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library, in *Advances in Neural Information Processing Systems (NeurIPS) 2019*. 56

- J. Pennington, R. Socher, and C. Manning (2014). Glove: Global vectors for word representation, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014*. 50
- J. Peters, P. Bühlmann, and N. Meinshausen (2016). Causal inference by using invariant prediction: identification and confidence intervals, *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. 148
- V. Petsiuk, A. Das, and K. Saenko (2018). RISE: Randomized Input Sampling for Explanation of Black-box Models, in *Proceedings of the British Machine Vision Conference (BMVC) 2018*. 19
- J. Peyre, I. Laptev, C. Schmid, and J. Sivic (2017). Weakly-supervised learning of visual relations, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 94, 125
- A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie (2007). Objects in context, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2007*. 18
- A. Radford, L. Metz, and S. Chintala (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, *Proceedings of the International Conference on Learning Representations (ICLR)*. 14, 29
- A. Raghunathan, J. Steinhardt, and P. Liang (2018). Certified Defenses against Adversarial Examples, in *Proceedings of the International Conference on Learning Representations (ICLR) 2018*. 146
- M. Ranzato, S. Chopra, M. Auli, and W. Zaremba (2016). Sequence level training with recurrent neural networks, *Proceedings of the International Conference on Learning Representations (ICLR)*. 20
- J. Rapin and O. Teytaud (2018). *Nevergrad - A gradient-free optimization platform*, <https://GitHub.com/FacebookResearch/Nevergrad>. 135
- A. Ray, K. Sikka, A. Divakaran, S. Lee, and G. Burachas (2019). Sunny and Dark Outside?! Improving Answer Consistency in VQA through Entailed Question Generation, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019*. 21, 24, 104, 105
- B. Recht, R. Roelofs, L. Schmidt, and V. Shankar (2018). Do CIFAR-10 classifiers generalize to CIFAR-10?, *arXiv preprint arXiv:1806.00451*. 17, 118
- B. Recht, R. Roelofs, L. Schmidt, and V. Shankar (2019). Do ImageNet Classifiers Generalize to ImageNet?, in *Proceedings of the International Conference on Machine Learning (ICML) 2019*. 17, 118

- J. Redmon and A. Farhadi (2018). Yolo3: An incremental improvement, *arXiv preprint arXiv:1804.02767*. 118, 124
- S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee (2016). Generative Adversarial Text to Image Synthesis, in *Proceedings of the International Conference on Machine Learning (ICML) 2016*. 15
- S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards real-time object detection with region proposal networks, in *Advances in Neural Information Processing Systems (NeurIPS) 2015*. 36
- S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel (2016). Self-critical Sequence Training for Image Captioning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 20
- D. Rezende and S. Mohamed (2015). Variational Inference with Normalizing Flows, in *Proceedings of the International Conference on Machine Learning (ICML) 2015*. 13
- M. T. Ribeiro, S. Singh, and C. Guestrin (2016). Why should i trust you?: Explaining the predictions of any classifier, in *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining 2016*. 19
- S. Risi and J. Togelius (2020). Increasing generality in machine learning through procedural content generation, *Nature Machine Intelligence*. 147
- M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele (2013). Translating Video Content to Natural Language Descriptions, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2013*. 19
- O. Ronneberger, P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation, in *International Conference on Medical image computing and computer-assisted intervention 2015*. 120
- A. Rosenfeld, R. Zemel, and J. K. Tsotsos (2018). The elephant in the room, *arXiv preprint arXiv:1808.03305*. 19
- S. Ruder, P. Ghaffari, and J. G. Breslin (2016). Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution, *arXiv preprint arXiv:1609.06686*. 22, 46, 49, 50
- S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang (2020). Distributionally Robust Neural Networks, in *Proceedings of the International Conference on Learning Representations (ICLR) 2020*. 24, 146
- C. Sakaridis, D. Dai, and L. V. Gool (2019). Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2019*. 132

- T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen (2016). Improved techniques for training gans, in *Advances in Neural Information Processing Systems (NeurIPS) 2016*. 34, 35
- S. Samanta and S. Mehta (2017). Towards Crafting Text Adversarial Samples, *arXiv preprint arXiv:1707.02812*. 23
- A. Sauer and A. Geiger (2021). Counterfactual Generative Networks, in *Proceedings of the International Conference on Learning Representations (ICLR) 2021*. 148
- J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker (2006). Effects of Age and Gender on Blogging., in *AAAI spring symposium: Computational approaches to analyzing weblogs 2006*. 48, 57
- B. Schölkopf (2019). Causality for machine learning, *arXiv preprint arXiv:1911.10500*. 148
- M. Shah, X. Chen, M. Rohrbach, and D. Parikh (2019). Cycle-Consistency for Robust Visual Question Answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 21, 24, 104, 105
- V. Shankar, A. Dave, R. Roelofs, D. Ramanan, B. Recht, and L. Schmidt (2019). A systematic framework for natural perturbations from videos, *Proceedings of the International Conference on Machine Learning Workshops (ICML Workshop)*. 18
- T. Shen, T. Lei, R. Barzilay, and T. Jaakkola (2017). Style Transfer from Non-Parallel Text by Cross-Alignment, in *Advances in Neural Information Processing Systems (NeurIPS) 2017*. 15
- R. Shetty, M. Fritz, and B. Schiele (2018a). Adversarial Scene Editing: Automatic Object Removal from Weak Supervision, in *Advances in Neural Information Processing Systems (NeurIPS) 2018*. 10, 105, 106, 107, 123
- R. Shetty, M. Fritz, and B. Schiele (2020). Towards Automated Testing and Robustification by Semantic Adversarial Data Generation, in *Proceedings of the European Conference on Computer Vision (ECCV) 2020*. 11
- R. Shetty, H. R-Tavakoli, and J. Laaksonen (2016). Exploiting Scene Context for Image Captioning, in *ACMMM Vision and Language Integration Meets Multimedia Fusion Workshop 2016*. 31, 32, 36
- R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele (2017). Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 10, 53
- R. Shetty, B. Schiele, and M. Fritz (2018b). A<sup>4</sup>NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation, in *USENIX Security '18 2018*. 10

- R. Shetty, B. Schiele, and M. Fritz (2019). Not Using the Car to See the Sidewalk—Quantifying and Controlling the Effects of Context in Classification and Segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 11
- A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb (2017). Learning from simulated and unsupervised images through adversarial training, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 5, 75
- A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe (2019). Animating Arbitrary Objects via Deep Motion Transfer, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 16
- P. Simard, D. Steinkraus, and J. Platt (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. 2003*. 25
- K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*. 36, 42
- A. Sinha, H. Namkoong, and J. Duchi (2018). Certifying Some Distributional Robustness with Principled Adversarial Training, in *Proceedings of the International Conference on Learning Representations (ICLR) 2018*. 146, 147
- Y. Song, R. Shu, N. Kushman, and S. Ermon (2018). Constructing unrestricted adversarial examples with generative models, in *Advances in Neural Information Processing Systems (NeurIPS) 2018*. 18
- J. C. Spall (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE transactions on automatic control*. 132, 135
- J. T. Springenberg (2016). Unsupervised and semi-supervised learning with categorical generative adversarial networks, *Proceedings of the International Conference on Learning Representations (ICLR)*. 15
- E. Stamatatos (2009). A survey of modern authorship attribution methods, *Journal of the Association for Information Science and Technology*. 22, 46
- D. Stutz, M. Hein, and B. Schiele (2019). Disentangling adversarial robustness and generalization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 18
- I. Sutskever, O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks, in *Advances in Neural Information Processing Systems (NeurIPS) 2014*. 23, 47, 48, 51



- R. S. Sutton and A. G. Barto (1998). *Reinforcement learning: An introduction*, vol. 1, MIT press Cambridge. 33
- C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2014). Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR) 2014*. 17
- R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt (2020). Measuring Robustness to Natural Distribution Shifts in Image Classification, in *Advances in Neural Information Processing Systems (NeurIPS) 2020*. 25
- D. Teney, E. Abbasnedjad, and A. v. d. Hengel (2020). Learning what makes a difference from counterfactual examples and gradient supervision, in *Proceedings of the European Conference on Computer Vision (ECCV) 2020*. 25
- T. Tieleman and G. Hinton (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural networks for machine learning*. 56
- J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel (2017). Domain randomization for transferring deep neural networks from simulation to the real world, in *IEEE/RSJ international conference on intelligent robots and systems (IROS) 2017*. 147
- J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne (2011). Search-based procedural content generation: A taxonomy and survey, *IEEE Transactions on Computational Intelligence and AI in Games*. 147
- A. Torralba, K. P. Murphy, and W. T. Freeman (2010). Using the forest to see the trees: exploiting context for visual object detection and localization, *Communications of the ACM*, vol. 53(3). 18, 88
- J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield (2018). Training Deep Networks With Synthetic Data: Bridging the Reality Gap by Domain Randomization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops) 2018*. 147
- S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, and V. Chari (2019). Learning to generate synthetic data via compositing, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 25
- Ultralytics (2019). *PyTorch Implementation of YoloV3*, <https://github.com/ultralytics/yolov3>. 124
- D. Ulyanov, A. Vedaldi, and V. Lempitsky (2016). Instance normalization: The missing ingredient for fast stylization, *arXiv preprint arXiv:1607.08022*. 1



- L. G. Valiant (1984). A theory of the learnable, *Communications of the ACM*, vol. 27(11), pp. 1134–1142. 17
- A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.* (2016). Conditional image generation with pixelcnn decoders, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, pp. 4790–4798. 13
- A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu (2016). Pixel recurrent neural networks, in *Proceedings of the International Conference on Machine Learning (ICML) 2016*. 13
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is All you Need, in *Advances in Neural Information Processing Systems (NeurIPS) 2017*. 1
- R. Vedantam, C. Lawrence Zitnick, and D. Parikh (2015). CIDEr: Consensus-Based Image Description Evaluation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 29
- R. Venkatesh, E. Wong, and J. Z. Kolter (2020). Semantic Adversarial Robustness with Differentiable Ray-Tracing, in *Advances in Neural Information Processing Systems Workshops(NeurIPS-W) 2020*. 148
- A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra (2016). Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, *arXiv preprint arXiv:1610.02424*. 20, 29
- O. Vinyals, A. Toshev, S. Bengio, and D. Erhan (2015). Show and tell: A neural image caption generator, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. 19, 31
- H. Wang (2019). *Implentation of Data Augmentation for Object Detection via Progressive and Selective Instance-Switching*, <https://github.com/Hwang64/PSIS>. 129
- H. Wang, Q. Wang, F. Yang, W. Zhang, and W. Zuo (2019a). Data Augmentation for Object Detection via Progressive and Selective Instance-Switching, *arXiv preprint arXiv:1906.00358*. 25, 26, 129, 130
- J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu (2016a). Cnn-rnn: A unified framework for multi-label image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 83
- M. Wang and W. Deng (2018). Deep visual domain adaptation: A survey, *Neurocomputing*. 8, 17
- Q. Wang and A. B. Chan (2019). Describing like humans: on diversity in image captioning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 20

- R. Wang, J. Lehman, J. Clune, and K. O. Stanley (2019b). Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions, *arXiv preprint arXiv:1901.01753*. 147
- R. Wang, J. Lehman, A. Rawal, J. Zhi, Y. Li, J. Clune, and K. Stanley (2020). Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions, in *Proceedings of the International Conference on Machine Learning (ICML) 2020*. 147
- T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro (2018). High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 75
- X. Wang, A. Shrivastava, and A. Gupta (2017). A-fast-rcnn: Hard positive generation via adversary for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 25
- Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004). Image quality assessment: from error visibility to structural similarity, *IEEE transactions on image processing*. 80
- Z. Wang, F. Wu, W. Lu, J. Xiao, X. Li, Z. Zhang, and Y. Zhuang (2016b). Diverse Image Captioning via GroupTalk, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2016*. 20
- R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen (2017). Sequence-to-Sequence Models Can Directly Transcribe Foreign Speech, in *Interspeech 2017*. 51
- R. J. Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine learning*. 5, 15, 20, 33
- E. Wong and Z. Kolter (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope, in *Proceedings of the International Conference on Machine Learning (ICML) 2018*. 146
- J. T. Woolley and G. Peters (1999). *The American presidency project*. 57
- Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. Hengel (2016a). Visual Question Answering: A Survey of Methods and Datasets, in *CVIU 2016*. 21
- Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.* (2016b). Google’s neural machine translation system: Bridging the gap between human and machine translation, *arXiv preprint arXiv:1609.08144*. 23, 48
- C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song (2018a). Spatially Transformed Adversarial Examples, in *Proceedings of the International Conference on Learning Representations (ICLR) 2018*. 18

- K. Xiao, L. Engstrom, A. Ilyas, and A. Madry (2021). Noise or signal: The role of image backgrounds in object recognition, in *Proceedings of the International Conference on Learning Representations (ICLR) 2021*. 19
- T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun (2018b). Unified Perceptual Parsing for Scene Understanding, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 89, 98, 99
- C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille (2017). Adversarial examples for semantic segmentation and object detection, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 17
- T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He (2018). AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 15
- W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry (2012). Paraphrasing for style, *Proceedings of COLING 2012*. 47
- Z. Yang, X. He, J. Gao, L. Deng, and A. Smola (2016). Stacked Attention Networks for Image Question Answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 21, 110
- T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei (2017). Boosting image captioning with attributes, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2017*. 38
- Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo (2016). Image captioning with semantic attention, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 38
- F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell (2020). BDD100K: A diverse driving dataset for heterogeneous multitask learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020*. 124
- J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang (2018). Generative Image Inpainting with Contextual Attention, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 78, 79
- L. Yu, W. Zhang, J. Wang, and Y. Yu (2016). SeqGAN: sequence generative adversarial nets with policy gradient, *Proceedings of the Conference on Artificial Intelligence (AAAI)*. 5, 14, 20, 33
- M. D. Zeiler and R. Fergus (2014). Visualizing and understanding convolutional networks, in *Proceedings of the European Conference on Computer Vision (ECCV) 2014*. 19

- O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. F. Dominguez (2018). Wilddash-creating hazard-aware benchmarks, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 8
- H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal (2018a). Context Encoding for Semantic Segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. 18, 19, 88
- P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh (2016). Yin and Yang: Balancing and Answering Binary Visual Questions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*. 21
- R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang (2018b). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, *arXiv preprint arXiv:1801.03924*. 80
- Y. Zhang (2017). *Re-implementation of Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering in PyTorch*, <https://github.com/Cyanogenoid/pytorch-vqa>. 110
- H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia (2017a). Pyramid scene parsing network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 19
- J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang (2017b). Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*. 19
- B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba (2017). Scene Parsing through ADE20K Dataset, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. 6, 98
- J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 15, 54, 75, 122
- Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro (2019). Improving semantic segmentation via video propagation and label relaxation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. 136



