

INDIAN STATISTICAL INSTITUTE

# On Adversarial Robustness of Deep Learning Systems

by

Akshay Chaturvedi

A thesis submitted in partial fulfillment for the  
degree of Doctor of Philosophy in Computer Science

Under the supervision of

Prof. Utpal Garain

Computer Vision and Pattern Recognition Unit

November 2021

# Declaration of Authorship

I, Akshay Chaturvedi, declare that this thesis titled, ‘On Adversarial Robustness of Deep Learning Systems’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.



26/11/2021

*“A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.”*

Douglas Adams

# Abstract

In the past decade, deep learning has been ubiquitous across diverse fields like natural language processing (NLP), computer vision, speech processing, etc. Despite achieving state-of-the-art performance, there are ongoing concerns regarding robustness and explainability of deep-learning systems. These concerns have further gained traction due to the presence of adversarial examples which make such systems behave in an undesirable fashion. To this end, this thesis explores several adversarial attacks and defenses for deep-learning based vision and NLP systems.

For vision/vision-and-language systems, the following two problems are studied in this thesis: (i) Robustness of visual question answering (VQA) systems: We study the robustness of VQA systems to adversarial background noise. The results show that, by adding minimal background noise, such systems can be easily fooled to predict an answer of the same as well as different category as the original answer. (ii) Task-agnostic adversarial attack for vision systems: We propose a task-agnostic adversarial attack named *Mimic and Fool* and show its effectiveness against vision systems designed for different tasks like image classification, image captioning and VQA. While the attack relies on the information loss that occurs in a convolutional neural network, we show that invertible architectures such as i-RevNet are also vulnerable to the proposed attack.

For NLP systems, the following three problems are studied in this thesis: (i) Invariance-based attack against neural machine translation (NMT) systems: We explore the robustness of NMT systems to non-sensical inputs obtained via an invariance-based attack. Unlike previous adversarial attacks against NMT system which make minimal changes to the source sentence in order to change the predicted translation, the invariance-based attack makes multiple changes in the source sentence with the goal of keeping the predicted translation unchanged. (ii) Defense against invariance-based attack: The non-sensical inputs obtained via the invariance-based attack do not have a ground truth translation. This makes standard adversarial training as a defense strategy infeasible. In this context, we explore several defense strategies to counteract the invariance-based attack. (iii) Robustness of multiple choice question-answering (MCQ) systems and intervention-based study: We explore the robustness of MCQ systems against the invariance-based attack. Furthermore, we also study the generalizability of MCQ systems to different types of interventions on the input paragraph.

# *Acknowledgement*

For the past two years, everyone's life has been deeply affected by the pandemic. During these difficult times, the role of others in one's life becomes even more apparent. In this regard, I wish to acknowledge the contribution of few people who have helped me so far. Firstly, I would like to thank my parents for obvious reasons. I would also like to thank my sister for her unconditional love and support. Special thanks to my friend, Mr. Johanan Wahlang, for introducing me to the field of machine learning, and eventually, natural language processing.

I am highly grateful to my supervisor, Prof. Utpal Garain. Research is rarely smooth sailing. His constant support and encouragement allowed me to learn, endure, and pursue research. Needless to say, this thesis would not have been possible without his guidance.

Special thanks to Dr. Niharika Gauraha and Dr. Buddhananda Banerjee for playing a pivotal role during the initial stages of my Ph.D. I especially would like to thank Dr. Masao Utiyama and Dr. Eiichiro Sumita for their kind support. Thanks to my friends; Mr. Uma Kant Sahoo, Dr. Abhisek Chakrabarty, Dr. Anabik Pal, Mr. Onkar Pandit, Mr. Arjun Das, Mr. Amit Yadav, Mr. Abijith KP, Mr. Joy Mahapatra, Mr. Soumen Kumar Koley, Mr. Shahansha Salim, Mr. Sourav Banerjee, Ms. Debleena Sarkar; at the institute with whom I had the pleasure to work and collaborate on some interesting problems. They have also been a constant source of support, not only academically but also otherwise. I would like to thank all the faculty members, research scholars, project-linked persons, and the office staff of the CVPR Unit for creating such a healthy work environment. Finally, I would like to thank everyone involved in ensuring the smooth functioning of this prestigious institute.

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Deep Learning: Background . . . . .	2
1.1.1 Multilayer Perceptron . . . . .	2
1.1.2 Recurrent Neural Network . . . . .	3
1.1.3 Long Short-Term Memory . . . . .	3
1.1.4 Convolutional Neural Network . . . . .	4
1.2 Adversarial Attack . . . . .	5
1.2.1 Origin . . . . .	5
1.2.2 Terminology . . . . .	6
1.3 Adversarial Attacks against Vision Systems . . . . .	6
1.3.1 Attacks against Image Classification systems . . . . .	7
1.3.2 Attacks against Other Vision Systems . . . . .	9
1.4 Adversarial Attacks against NLP Systems . . . . .	10
1.4.1 Challenges . . . . .	10
1.4.2 Previous Works . . . . .	11
1.5 Adversarial Defense . . . . .	12
1.5.1 Adversarial Training . . . . .	12
1.5.2 Challenges of Adversarial Training . . . . .	13
1.5.3 Other Approaches . . . . .	14
1.5.4 Defense and Attack: An Endless Cycle? . . . . .	15

1.6	Thesis Outline and Contributions . . . . .	16
1.7	Thesis Organization . . . . .	18
<b>2</b>	<b>Attacking VQA systems via Adversarial Background Noise</b>	<b>19</b>
2.1	Background . . . . .	20
2.1.1	VQA datasets . . . . .	20
2.1.2	VQA systems . . . . .	20
2.1.3	Adversarial Attack Against VQA systems . . . . .	22
2.2	Motivation . . . . .	22
2.3	Methodology . . . . .	23
2.3.1	Background Detection . . . . .	23
2.3.2	Targeted Adversarial Attack . . . . .	25
2.4	Implementation Details . . . . .	27
2.5	Datasets . . . . .	28
2.6	Results . . . . .	28
2.6.1	Success Rate . . . . .	30
2.6.2	Visualizing attention . . . . .	33
2.6.3	Transferability Results . . . . .	35
2.6.4	Mean/Median Filtering as Defense? . . . . .	36
2.7	Examples of the Attack . . . . .	37
2.8	Summary . . . . .	44
<b>3</b>	<b>Mimic and Fool: A Task-Agnostic Adversarial Attack</b>	<b>45</b>
3.1	Background . . . . .	46
3.1.1	Show and Tell . . . . .	46
3.1.2	Show, Attend and Tell . . . . .	46
3.1.3	Proposed Attack: Overview and Advantages . . . . .	47
3.2	Methodology . . . . .	48
3.2.1	Mimic and Fool . . . . .	48
3.2.2	One Image Many Outputs . . . . .	49
3.3	Implementation Details . . . . .	50
3.4	Results . . . . .	51
3.4.1	Results for Mimic and Fool . . . . .	52
3.4.2	Results for One Image Many Outputs . . . . .	54
3.4.3	Comparison with task specific attack . . . . .	57
3.4.4	OIMO for invertible architecture . . . . .	58
3.4.5	Quantitative study of Adversarial Noise . . . . .	59
3.5	Examples of the Attack . . . . .	61
3.5.1	Examples of Mimic and Fool . . . . .	61
3.5.2	Examples of One Image Many Outputs . . . . .	65
3.6	Summary . . . . .	68
<b>4</b>	<b>Exploring the Robustness of NMT systems to Non-sensical Inputs</b>	<b>70</b>
4.1	Background . . . . .	71

4.1.1	BLSTM-based encoder decoder with attention . . . . .	71
4.1.2	Transformer . . . . .	72
4.2	Motivation . . . . .	72
4.3	Methodology . . . . .	73
4.3.1	Vocabulary Pruning . . . . .	74
4.3.2	Position Indices Traversal . . . . .	74
4.3.3	Word Replacement . . . . .	75
4.3.4	Proposed method . . . . .	76
4.4	Implementation Details . . . . .	78
4.5	Evaluation Metrics . . . . .	80
4.5.1	Success rate . . . . .	80
4.5.2	BLEU-based metric . . . . .	81
4.6	Results . . . . .	82
4.6.1	Success rate . . . . .	83
4.6.2	BLEU-based metric . . . . .	85
4.6.3	A Comment on Types of Words Replaced . . . . .	89
4.6.4	Human evaluation . . . . .	90
4.6.5	Results on WMT Dataset . . . . .	91
4.7	Summary . . . . .	92
<b>5</b>	<b>Ignorance is Bliss: Exploring Defenses Against Invariance based Attacks on NMT systems</b>	<b>94</b>
5.1	Background . . . . .	95
5.1.1	Overview of the Proposed Method . . . . .	95
5.1.2	Bruteforce Attack . . . . .	96
5.1.3	Efficiency of Bruteforce Attack . . . . .	98
5.2	Defense Methodology . . . . .	100
5.2.1	Generating noisy samples . . . . .	101
5.2.2	Training Loss Function . . . . .	102
5.3	Implementation Details . . . . .	103
5.4	Evaluation Metrics . . . . .	103
5.5	Results . . . . .	104
5.5.1	Learn to Deal vs. Learn to Ignore . . . . .	104
5.5.2	BLEU score . . . . .	107
5.5.3	Random vs. Tackle-Bias . . . . .	107
5.6	Summary . . . . .	109
<b>6</b>	<b>Generalizability of Bruteforce Attack: A case-study on TQA and SciQ dataset</b>	<b>110</b>
6.1	Background . . . . .	111
6.1.1	Choosing the most relevant paragraph . . . . .	111
6.1.2	Neural Network Architecture . . . . .	112
6.1.3	Dealing with forbidden options . . . . .	114
6.1.4	Implementation Details . . . . .	115
6.1.5	Results . . . . .	115



---

6.2	Bruteforce Attack and Types of Intervention . . . . .	117
6.3	Results . . . . .	118
6.4	Summary . . . . .	120
<b>7</b>	<b>Conclusion</b>	<b>121</b>
	<b>Bibliography</b>	<b>126</b>

# List of Figures

1.1	Fast Gradient Sign Method (FGSM) on GoogLeNet [124] (Photo Courtesy: Goodfellow et al. [41]) . . . . .	5
2.1	Complementary images from VQA v2.0 (Photo Courtesy: Goyal et al. [43]) . . . . .	21
2.2	Example of the proposed attack. For the above question, both N2NMN and MAC network give the correct answer (“no”) when original image is given as input but incorrect answer (“yes”) when respective adversarial image is given as input. The noise is added only to the outside background of the image. . . . .	23
2.3	Images from SHAPES dataset. . . . .	23
2.4	Background Detection for CLEVR. Only the pixels outside the blue rectangle are modified in the proposed attack. . . . .	24
2.5	Background Detection for VQA v2.0. The pixels which are not inside any of the boxes are modified in the proposed attack. . . . .	24
2.6	Answer changes to <i>yes</i> for the adversarial image. For the adversarial image, a light silhouette of a triangle can be seen in top left and middle left. Such cases were considered <i>unsuccessful</i> . . . . .	30
2.7	Attention visualization for SHAPES. Note that the textual attention map remains same for the two images. . . . .	30
2.8	N2NMN predicts same category as the target answer. . . . .	32
2.9	Attention visualization for N2NMN on CLEVR. For both the adversarial images, the attack was successful i.e. the predicted answer was $A_{target}$ . . . . .	34
2.10	Attention visualization for MAC network on CLEVR. Note that the textual attention map remains same for all the images. For both the adversarial images, the attack was successful i.e. the predicted answer was $A_{target}$ . . . . .	34
2.11	Attention visualization for N2NMN on VQA v2.0. Note that the textual attention map remains same for all the images. For both the adversarial images, the attack was successful i.e. the predicted answer was $A_{target}$ . . . . .	35
2.12	Examples for N2NMN on SHAPES. . . . .	37
2.13	Examples for N2NMN on CLEVR <sub>same</sub> . . . . .	38
2.14	Examples for N2NMN on CLEVR <sub>diff</sub> . . . . .	39
2.15	Examples for MAC network on CLEVR <sub>same</sub> . . . . .	40
2.16	Examples for MAC network on CLEVR <sub>diff</sub> . . . . .	41

2.17	Examples for N2NMN on VQA <sub>same</sub> . . . . .	42
2.18	Examples for N2NMN on VQA <sub>diff</sub> . . . . .	43
3.1	Examples of Mimic and Fool. The first two rows show the original and adversarial images along with the predicted captions by Show and Tell and Show Attend and Tell respectively. The last row shows original and adversarial image for N2NMN (Q, P denote the question and the predicted answer respectively). . . . .	47
3.2	Example of <i>Mimic and Fool</i> for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer respectively. $P_{zero}$ denotes the predicted answer for zero image. . . . .	52
3.3	Examples of <i>Mimic and Fool</i> . For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics. . . . .	54
3.4	$I_{start}$ for <i>One Image Many Outputs</i> and the predicted captions. . . . .	54
3.5	Example of <i>One Image Many Outputs</i> for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer respectively. $P_{I_{start}}$ denotes the predicted answer for $I_{start}$ . . . . .	55
3.6	Examples of <i>One Image Many Outputs</i> . For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics. For adversarial images, ST and SAT denote Show and Tell and Show Attend and Tell respectively. . . . .	56
3.7	Both the images are classified as <i>ice bear</i> by bijective i-RevNet. . . . .	59
5.1	Histogram of $rank(w_{adv}   w_{org})$ for en-de Transformer. $w_{org}$ and $w_{adv}$ denote the original word and the replaced word during brute-force respectively. . . . .	99
6.1	Architecture of the proposed system. Attention layer attends on sentence embeddings $d_j$ 's using question-option tuple embeddings $h_i$ 's. Score Calculation layer calculates the cosine similarity between $m_i$ and $h_i$ which is passed through softmax to get the final probability distribution. . . . .	112

# List of Tables

2.1	Success rate (SR) of the proposed attack. For $\ \delta\ _2$ , the mean and standard deviation is calculated over the <i>successful</i> cases. bg-size denotes the <i>mean <math>\pm</math> std</i> of the percentage of an image detected as background using Section 2.3.1. . . . .	29
2.2	Success rate of Xu <i>et al.</i> [144]. For $\ \delta\ _2$ , the mean and standard deviation is calculated over the <i>successful</i> cases. . . . .	29
3.1	Success rate of <i>Mimic and Fool</i> . . . . .	52
3.2	BLEU and METEOR scores for <b>unsuccessful</b> cases. OIMO refers to <i>One Image Many Outputs</i> . B-1, B-2, B-3, B-4, and M represents BLEU-1, BLEU-2, BLEU-3, BLEU-4 and METEOR respectively. ST, and SAT represents Show and Tell, and Show Attend and Tell respectively. . . . .	53
3.3	Success rate of <i>One Image Many Outputs</i> . . . . .	55
3.4	Success rate and Time for task-specific methods. $n_q$ signifies the average number of questions per image. . . . .	57
3.5	Success rate of <i>One Image Many Outputs</i> for i-RevNet . . . . .	58
3.6	PSNR between $I_{adv}$ and $I_{start}$ for <i>One Image Many Outputs</i> (OIMO) and task-specific methods. . . . .	59
3.7	SSIM between $I_{adv}$ and $I_{org}$ for <i>Mimic and Fool</i> (MAF) and <i>One Image Many Outputs</i> (OIMO). . . . .	60
3.8	Examples of <i>Mimic and Fool</i> for N2NMN. Single adversarial image suffices for three image-question pairs. . . . .	61
3.9	Examples of Mimic and Fool for N2NMN. N2NMN predicts varied answers for the same question. . . . .	62
3.10	Examples for Show and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases. . . . .	63
3.11	Examples for Show Attend and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases. . . . .	64
3.12	Examples of <i>One Image Many Outputs</i> for N2NMN. Single adversarial image suffices for three image-question pairs. . . . .	65
3.13	Examples of <i>One Image Many Outputs</i> for N2NMN. N2NMN predicts varied answers for the same question. . . . .	66
3.14	Examples for Show and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases. . . . .	67
3.15	Examples for Show Attend and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases. . . . .	68

4.1	Example of the proposed attack. The English-German Transformer predicts the same translation for the two sentences even though multiple replacements are made. . . . .	73
4.2	Dataset Statistics . . . . .	79
4.3	BLEU score on the <i>test set</i> . . . . .	79
4.4	An example to showcase the prediction pipeline. Finally, $s_{fin}^{adv}$ is given as input to the NMT system. . . . .	80
4.5	Success Rate (in %) and number of replacements for different methods. <i>NOR</i> represents the mean/median of the normalized <b>Number Of Replacements</b> across all the sentences. The highest success rate is marked in bold. . . . .	84
4.6	Mean of char-F1 for different methods M. . . . .	85
4.7	BLEU scores for the original/adversarial sentence (src) and their respective translations by the four NMT systems. $l_1$ denotes the model under attack, $l_2$ denotes the other Transformer model. $l_1^{blstm}, l_2^{blstm}$ are the BLSTM counterparts of $l_1$ and $l_2$ . Similarly, $l_1^{moses}, l_2^{moses}$ are MOSES counterparts of $l_1$ and $l_2$ . The arrows in the table header denote whether lower/higher is better for an attack to be effective. . . . .	86
4.8	BLEU scores for the original/adversarial sentence (src) and their respective translation by the four NMT Systems. $l_1$ denotes the model under attack, $l_2$ denotes the other BLSTM model. $l_1^{trans}, l_2^{trans}$ are the Transformer counterparts of $l_1$ and $l_2$ . Similarly, $l_1^{moses}, l_2^{moses}$ are MOSES counterparts of $l_1$ and $l_2$ . The arrows in the table header denote whether lower/higher is better for an attack to be effective. . . . .	87
4.9	$e(M)$ for different methods M (lower values of $e(M)$ imply better attack efficiency). . . . .	88
4.10	Examples of <i>Min-Grad + Soft-Att</i> for BLSTM-based Encoder-Decoder with Attention. The NMT system predicts the same translation for src and adv-src. . . . .	88
4.11	Examples of <i>Min-Grad + Soft-Att</i> for Transformer. The NMT system predicts the same translation for src and adv-src. . . . .	89
4.12	Human evaluation: Mean and median of semantic similarity score for different NMT systems. . . . .	90
4.13	Success Rate (in %), number of replacements, and mean of char-F1 for different methods against Transformer trained on WMT 16 English-German. <i>NOR</i> represents the mean/median of the normalized <b>Number Of Replacements</b> across all the sentences. The highest success rate is marked in bold. . . . .	92
4.14	BLEU scores for the original/adversarial sentence (src) and their respective translation by the three NMT systems. $l_1$ denotes the Transformer trained on WMT 16 English-German, $l_2$ denotes the Transformer trained on WMT 14 English-French and $l_1^{wmt19}$ denotes the Transformer trained on WMT 19 English-German. . . . .	92

5.1	Example of bruteforce attack on English-German Transformer. The NMT system predicts the same translation ( <b>pred</b> ) for the clean source sentence ( <b>src</b> ) and the noisy sentence ( <b>adv-src</b> ). . . . .	96
5.2	Example of the two defense strategies. Learn to Deal strategy predicts a different translation for src and adv-src (the difference is shown in italics). Learn to Ignore strategy predicts “This sentence is not correct” in the target language (i.e., French in this case) for adv-src. . . . .	96
5.3	Success rate and mean, median of number of replacements (NOR) for bruteforce attack, and BLEU score on the test set. . . . .	98
5.4	BLEU scores for predicted translations of $s^{org}$ and $s^{adv}$ across NMT systems. $l_1$ denotes the Transformer under attack, $l_2$ denotes the Transformer for the other language pair, and $l_1^{blstm}, l_2^{blstm}$ denote respective BLSTM-based NMT systems. . . . .	100
5.5	Results for <i>learn to deal</i> (LTD) and <i>learn to ignore</i> (LTI) strategies for English-German. The lowest success rate, highest targeted translation (TT), and highest BLEU are marked in boldface. . . . .	105
5.6	Results for <i>learn to deal</i> (LTD) and <i>learn to ignore</i> (LTI) strategies for English-French. The lowest success rate, highest targeted translation (TT), and highest BLEU are marked in boldface. . . . .	106
5.7	Results for <i>learn to deal</i> (LTD) and <i>learn to ignore</i> (LTI) strategies on the modified bruteforce attack. . . . .	108
6.1	Accuracy for true-false and multiple choice questions on validation set of TQA dataset. . . . .	115
6.2	Accuracy of the QA systems on SciQ dataset. The first three accuracies are on validation set. The last accuracy is of $CNN_{2,3,4}$ on the test set. . . . .	116
6.3	Accuracy of different systems for true-false and multiple choice questions. Results marked with (*) are taken from Kembhavi et al. [64] and are on test set obtained using a different data split. Result of our proposed system is on publicly released validation and test set combined. . . . .	117
6.4	Example from SciQ validation set. We manually annotate the portion of paragraph responsible for the answer (shown in blue). . . . .	118
6.5	Success Rate of Bruteforce-Attack . . . . .	119
6.6	Transferability of Bruteforce-Attack. The adversarial example obtained for the Source QA system is given as input to the Target QA system. . . . .	119
6.7	Results for mask and option-specific interventions. Prediction count shows the number of times each of the option is predicted by the QA system. For <i>option-specific intervention</i> , the prediction count of the <i>desired option</i> is marked in bold. . . . .	120

# Abbreviations

<b>QA</b>	<b>Q</b> uestion <b>A</b> nswering
<b>VQA</b>	<b>V</b> isual <b>Q</b> uestion <b>A</b> nswering
<b>MLP</b>	<b>M</b> ulti <b>L</b> ayer <b>P</b> erceptron
<b>MT</b>	<b>M</b> achine <b>T</b> ranslation
<b>NMT</b>	<b>N</b> eural <b>M</b> achine <b>T</b> ranslation
<b>OIMO</b>	<b>O</b> ne <b>I</b> mage <b>M</b> any <b>O</b> utputs
<b>NMN</b>	<b>N</b> eural <b>M</b> odule <b>N</b> etwork
<b>N2NMN</b>	<b>E</b> nd- <b>t</b> o- <b>E</b> nd <b>M</b> odule <b>N</b> etwork
<b>MAC</b>	<b>M</b> emory, <b>A</b> ttention and <b>C</b> omposition
<b>BLEU</b>	<b>B</b> ilingual, <b>E</b> valuation Understudy
<b>METEOR</b>	<b>M</b> etric for <b>E</b> valuation of <b>T</b> ranslation with <b>E</b> xplicit <b>O</b> Rdering
<b>NLP</b>	<b>N</b> atural <b>L</b> anguage <b>P</b> rocessing
<b>LSTM</b>	<b>L</b> ong <b>S</b> hort <b>T</b> erm <b>M</b> emory
<b>BLSTM</b>	<b>B</b> idirectional <b>L</b> ong <b>S</b> hort <b>T</b> erm <b>M</b> emory
<b>RNN</b>	<b>R</b> eurrent <b>N</b> eural <b>N</b> etwork
<b>HOG</b>	<b>H</b> istogram of <b>O</b> riented <b>G</b> radient
<b>SIFT</b>	<b>S</b> cale <b>I</b> nvariant <b>F</b> eature <b>T</b> ransform
<b>SVM</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine
<b>RBF</b>	<b>R</b> adial <b>B</b> asis <b>F</b> unction
<b>CNN</b>	<b>C</b> onvolutional <b>N</b> eural <b>N</b> etwork
<b>ReLU</b>	<b>R</b> ectified <b>L</b> inear <b>U</b> nit
<b>BPTT</b>	<b>B</b> ackpropagation <b>T</b> hrough <b>T</b> ime

*Dedicated to my Parents*



# Chapter 1

## Introduction

*You don't want to cover a subject; You want to uncover it.*

Eleanor Duckworth

Deep learning has led to remarkable advancements in diverse fields such as computer vision, natural language processing (NLP), and speech processing amongst others. While the foundation for training deep learning systems was laid in 1980's [110], these systems gained popularity around 2012 after AlexNet [68], a convolutional neural network (CNN), achieved state-of-the-art results on ImageNet dataset [29]. Apart from AlexNet, another reason behind the popularity of deep learning in the past decade is the rapid improvement of graphics processing unit (GPU) which led to drastic reduction in training time. The advent of deep learning shifted the focus from feature engineering (such as HOG [27] and SIFT [80] in computer vision) to designing models which are end-to-end. End-to-end signifies that such models accept input in its raw form (e.g., pixel intensities of an image) in order to generate the desired output. Presently, deep learning systems have achieved impressive performance in varied tasks such as object detection [107], visual question answering [52], image captioning [2], and machine translation [128], etc.

Despite the impressive performance, deep learning systems are highly susceptible to adversarial attacks. Adversarial attacks, in the most general sense, can be defined as the process of fooling a machine learning system to behave in an undesirable fashion either by manipulating the decision boundary during training [91] or by generating malicious inputs during inference [41].

This thesis studies the adversarial robustness of several deep learning systems across computer vision and NLP. To do so, we design several adversarial attacks and defenses across vision and NLP tasks. The rest of this chapter is organized as follows. Section 1.1 provides a very basic background to deep learning. Section 1.2 discusses the origin and basic terminologies of adversarial attack. Section 1.3 discusses previous works on adversarial attacks against vision systems. Similarly, Section 1.4 discusses previous works on adversarial attacks against vision systems. Section 1.5 discusses previous works on adversarial defense. Section 1.6 discusses the outline and main contributions of the thesis. Finally, Section 1.7 discusses the organization of the rest of the thesis.

## 1.1 Deep Learning: Background

In this section, we provide a very brief background to some basic deep learning architectures. For an in-depth treatment of the subject, we refer the reader to Goodfellow et al. [40].

### 1.1.1 Multilayer Perceptron

Perceptron was introduced by Rosenblatt [108] as a binary classification system which can distinguish between the input signals from two different classes based on the learned weights of each input signal (i.e., stimuli). Multilayer perceptron (MLP) combines several perceptron units. A MLP consists of an input layer,  $L$  hidden layers and an output layer. The output of  $l^{th}$  layer is given by

$$o^l = f(W^l o^{l-1} + b^{l-1}) \quad (1.1)$$

where  $W^l$  is the weight matrix,  $o^{l-1}$  is the output of the  $(l-1)^{th}$  layer,  $b^{l-1}$  is the bias term of the  $(l-1)^{th}$  layer and  $f$  is a non-linear activation function. Some common non-linear activation functions are sigmoid function, hyperbolic tangent (i.e.,  $\tanh$ ) function, and Rectified Linear Unit (ReLU). The parameters of the MLP (i.e.,  $W^l$  and  $b^{l-1}$ ) are learned during training using the backpropagation algorithm [110].

## 1.1.2 Recurrent Neural Network

Multilayer perceptrons are ill-suited for tasks where either the input or output or both are sequential in nature. This is because of their inability to handle variable sequence length or larger sequences in the input/output. In natural language processing (NLP), there are several problems where the network needs to handle variable sequence length such as sentiment analysis, machine translation, part of speech (POS) tagging etc. To address this drawback, recurrent neural network (RNN) were designed [135]. A recurrent neural network consists of a feedback loop which allows it to handle variable sequence length. Mathematically, let  $x_t$  denote the input at time  $t$ , and  $h_{t-1}$  denote the output of the hidden layer at time  $t - 1$ , then the output of the RNN at time  $t$  (i.e.,  $y_t$ ) is given by

$$\begin{aligned}h_t &= f(Wx_t + Vh_{t-1} + b_h) \\y_t &= g(Uh_t + b_y)\end{aligned}\tag{1.2}$$

where  $U, V$ , and  $W$  are weight matrices;  $b_h, b_y$  are biases; and  $f, g$  are activation functions. All the parameters of a recurrent neural network are shared across time and are learned during training using the backpropagation through time (BPTT) algorithm [136].

## 1.1.3 Long Short-Term Memory

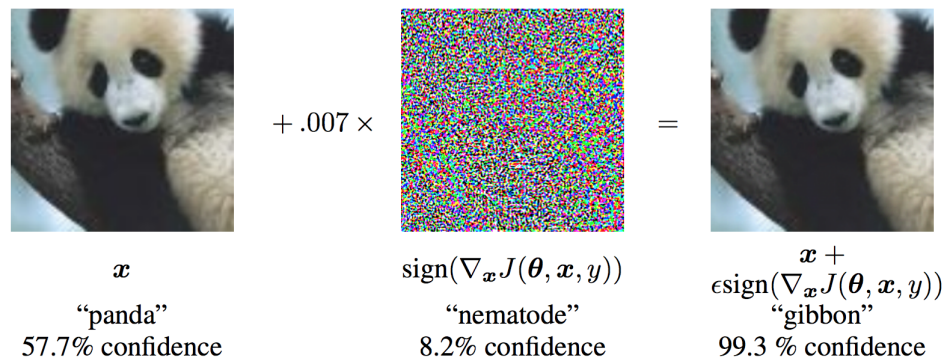
The BPTT algorithm in RNN leads to a learning problem. When the gradients are backpropagated through time, the gradients either explode due to the weight matrices having higher values or the gradients vanish due to the derivative of the activation function which typically lies between 0 and 1. The vanishing/exploding gradient problem leads to the inability of RNN to capture long-term dependencies [49]. Long-term dependency describes a scenario where the desired output is dependent on an input seen way back in time (e.g., in sentiment analysis, an article may have a positive sentiment due to a sentence present in the second-last paragraph). To remedy this issue, long short-term memory (LSTM) [50] was designed. A long short-term memory cell controls the flow of information at each time step using several gates. Mathematically, let  $x_t$  denote the input at time  $t$ ,  $c_{t-1}$  denote the cell state at time  $t - 1$ , and  $h_{t-1}$  denote the output of the LSTM cell at time  $t - 1$ , then the output of the LSTM cell at time  $t$  (i.e.,  $h_t$ ) is given by

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{1.3}$$

where  $W_f, W_i, W_c, W_o, U_f, U_i, U_c$ , and  $U_o$  are weight matrices;  $b_f, b_i, b_c$ , and  $b_o$  are biases;  $\sigma, \tanh$  denote the sigmoid and hyperbolic tangent function respectively; and  $\odot$  denotes the hadamard product. Similar to RNN, all the weight matrices and biases of LSTM are shared across time.  $f_t, i_t$  and  $o_t$  in Equation 1.3 denote the forget gate, input gate, and output gate respectively. These gates are responsible for controlling the flow of information inside the LSTM cell at a particular time step. Several variants of the LSTM cell have been proposed in the literature [44].

### 1.1.4 Convolutional Neural Network

Convolutional Neural Network (CNN) was introduced by Le Cun et al. [71]. CNN are specifically designed for processing images. Images, unlike text, are two-dimensional where nearby pixels are highly correlated. CNN typically consists of convolutional layers, pooling layers, and finally some fully connected layers. The convolutional layer contains several kernels of smaller spatial dimension than the original image. These kernels are responsible for finding localised pattern present in the image by making use of the convolution operation. The pooling layer (also known as subsampling layer) reduces the spatial dimension, thereby ensuring that the number of parameters in the fully connected layers are limited and that the kernels of the deeper convolutional layers have larger receptive fields. Due to this, the kernels of CNN work in a hierarchical fashion. While the kernels of the earlier convolutional layers are responsible for detecting edges, the kernels of the deeper convolutional layers detect more abstract patterns present in the image [149]. In the past decade, CNNs have been ubiquitous across variety of vision tasks [2, 45, 46, 107].



**Figure 1.1:** Fast Gradient Sign Method (FGSM) on GoogLeNet [124] (Photo Courtesy: Goodfellow et al. [41])

## 1.2 Adversarial Attack

Adversarial attacks can be broadly classified into two types: poisoning attacks, and evasion attacks [10]. Poisoning attacks take place during training, whereas evasion attacks take place during testing. In a poisoning attack, the adversary adds malignant inputs to the training data of the machine learning system. This allows the adversary to manipulate the decision boundary of the system. On the other hand, in an evasion attack, the adversary generates an input which *fools* a machine learning system to predict incorrectly or behave in an undesirable fashion. This input is referred to as an *adversarial example* and is usually generated by adding noise to the original/clean input. One such example is shown in Figure 1.1 where GoogLeNet [124] predicts an image of a *panda* incorrectly as a *gibbon* after an *imperceptible* noise is added to the original image [41]. Nguyen et al. [93] showed that images which are completely unrecognizable to humans are predicted as familiar objects with very high confidence by deep neural networks. This is an example of a machine learning system behaving in an undesirable fashion.

### 1.2.1 Origin

While the focus of this thesis is on adversarial attacks (evasion attack, to be more precise) and defenses for deep learning systems, the research in the field of adversarial machine learning originated long before the *deep learning era* [10, 59]. Wittel and Wu [137] proposed an evasion attack on statistical spam filters. Dalvi et al. [28] proposed an adversarial framework for training spam detection classifiers in light of the adversary. Soon after, Lowd and Meek [79] proposed *good*

*word attack* against statistical spam filters. A *good word attack* adds legitimate words (i.e., non-spam words) to spam emails allowing it to get past the statistical spam filters. Nelson et al. [91] explored poisoning attacks as well as defense for spam filters. Rubinstein et al. [109] proposed defenses against poisoning attacks for anomaly detectors. Šrندیć and Laskov [127] proposed a practical evasion attack against an online PDF malware detection service [119]. Biggio et al. [9] proposed an evasion attack against support vector machine (SVM) [25] and multi-layer perceptron for handwritten digit recognition [72] and PDF malware detection. Given the focus of this thesis, we will only discuss evasion attacks and defenses for deep learning systems from this point onwards.

## 1.2.2 Terminology

In this section, we introduce some terminologies related to adversarial attacks which will be used throughout this thesis. Adversarial attacks are typically categorised into two types: targeted, and non-targeted. In a *targeted attack*, the noise is added to the original input in order to ensure that the model makes a specific prediction. Whereas, in a *non-targeted attack* (also known as *untargeted attack*), the noise is added to the original input in order to ensure that the model makes an incorrect prediction. Adversarial attacks are also categorised on the basis of whether or not the adversary has access to the parameters and architecture of the model under attack. In this regard, in a *white-box attack*, the adversary has access to the architecture and the parameters of the model whereas, in a *black-box attack*, the adversary doesn't have access to the architecture and the parameters of the model. A *gray-box attack*, as the name suggests, is an adversarial attack where the adversary has *partial knowledge* about the architecture and the parameters of the model.

## 1.3 Adversarial Attacks against Vision Systems

In the initial years of research on this topic, the major focus was on designing attacks against image classifiers. Later, adversarial attacks were generalized against other vision systems as well as vision-and-language systems. Presently, there has been a plethora of work on this topic. In this section, we discuss some of these works. Section 1.3.1 discusses adversarial attacks against image classifiers.

Section 1.3.2 discusses adversarial attacks against other vision and vision-and-language systems.

### 1.3.1 Attacks against Image Classification systems

Adversarial attacks against deep learning based image classifiers was first introduced by Szegedy et al. [125]. Szegedy et al. [125] proposed a targeted attack where the adversarial examples were generated using box-constrained L-BFGS [75]. These examples have imperceptible noise and are also transferable across different models (i.e., the same adversarial example was able to fool multiple image classifiers). Soon after, Goodfellow et al. [41] proposed the first non-iterative (i.e., single-step) adversarial attack known as Fast Gradient Sign Method (FGSM). FGSM is a non-targeted attack which adds to the original image, a very small fraction of the sign of the gradient of the loss function (also known as cost function) with respect to the original image, in order to generate adversarial example. Similar to Szegedy et al. [125], the adversarial examples generated using FGSM have imperceptible noise. Figure 1.1 shows an example to FGSM attack. Kurakin et al. [70] and Madry et al. [82] proposed an iterative variant of the FGSM attack, known as projected gradient descent (PGD) attack.

Papernot et al. [99] proposed a targeted adversarial attack based on saliency maps known as Jacobian-based Saliency Map Attack (JSMA). In JSMA, the saliency maps consider the gradient of the models' output with respect to the original image. This allows the adversary to only modify the relevant pixels of the image in order to force the model to predict a target class. Papernot et al. [99] demonstrated the efficiency of their attack on the MNIST dataset [72]. Moosavi-Dezfooli et al. [89] proposed an iterative non-targeted attack known as DeepFool. At each iteration of DeepFool, the decision boundary of the non-linear classifier is approximated with a convex polyhedron and accordingly, the optimum perturbation required for misclassification is applied. Using this technique, DeepFool achieves a smaller perturbation than Szegedy et al. [125] and FGSM. Carlini and Wagner [16] proposed a targeted attack which further reduces the perturbation in comparison to DeepFool. The loss function for this attack includes the perturbation along with the difference between the maximum logit and the logit for the targeted class. Karmon et al. [63] proposed a targeted adversarial attack where the noise is only added to a very small region of the image. Moosavi-Dezfooli et al. [88]

proposed an image agnostic perturbation known as universal adversarial perturbation. This perturbation when added to any image leads to an adversarial image which is misclassified by the image classifier. Furthermore, Moosavi-Dezfooli et al. [88] also showed that the universal adversarial perturbation generalizes to other image classifiers as well.

The adversarial attacks, discussed so far, are white-box attacks. Adversarial attacks against image classifiers have also been studied in a more constrained setting. Su et al. [123] proposed one-pixel attack. The attack is based on differential evolution [122] and only needs access to the class probability scores and not the models' architecture and parameters. The attack succeeds in fooling image classifiers by modifying just a single pixel of the image. Similarly, Chen et al. [18] proposed a zeroth-order optimization based adversarial attack which only needs access to the class probability scores. Liu et al. [77] showed that while the non-targeted attacks are transferable to other image classifiers, the targeted attacks have low transferability across different architectures. They further proposed a targeted attack on ensemble of classifiers and showed that the adversarial examples, so obtained, have better transferability to the image classifier which is not part of the ensemble. Papernot et al. [98] proposed a black-box attack which is based on training a substitute classifier on a synthetic dataset. The synthetic dataset is created by passing images to the original classifier and using its predictions as ground truth. Then, a substitute classifier is trained on the synthetic data. This is followed by applying a white-box attack on the substitute classifier. Papernot et al. [98] showed that the adversarial examples, so obtained, are also successful in fooling the original classifier. Later, Papernot et al. [97] generalized this idea to support vector machines and decision trees. Brendel et al. [12] propose a black-box attack which does not rely on the idea of training a substitute classifier. Rather, the attack starts with an adversarial image with a large noise and tries to iteratively reduce the noise. Ilyas et al. [53] proposed an adversarial attack which only needs access to the value of the loss function of the classifier. The attack uses gradient priors for gradient estimation.

Apart from black box attacks, there also has been significant research on robustness of adversarial examples to image transformations. Kurakin et al. [69] printed adversarial images and then took its photo using mobile camera. This photo was then passed to the classifier to study whether the resultant photo is also adversarial. Kurakin et al. [69] showed that adversarial images obtained from non-iterated



attack are more robust to the above transformation. Eykholt et al. [34] proposed robust physical perturbation (RP<sub>2</sub>) to generate adversarial examples in the physical world which are robust to change in distance and angle of the camera. Athalye et al. [6] showed the existence of 3D adversarial objects which were obtained from 3D printing.

In this section, we see that there is a consistent effort in designing adversarial examples with imperceptible noise. While imperceptible noise does showcase the extent to which deep learning based image classifiers are fragile, from a robustness standpoint, the adversarial examples do not need to have *imperceptible noise* [7, 10, 38]. This point has also been argued by Biggio and Roli [10] and Gilmer et al. [38]. In fact, Gilmer et al. [38] designed semantics-preserving adversarial examples where the noise has a very large  $\ell_p$ -norm.

### 1.3.2 Attacks against Other Vision Systems

Xie et al. [140] proposed a white-box adversarial attack for semantic segmentation and object detection. The proposed adversarial attack is non-targeted, i.e., the attack tries to induce as many misclassifications as possible for both the tasks. While Xie et al. [140] designed adversarial examples in a digital setting, there has been a significant focus on designing adversarial attacks against object detectors in real-world setting. Chen et al. [19] proposed physical adversarial attack against Faster R-CNN, a state-of-the-art object detector. They studied both targeted and non-targeted variants of their attack on *stop-sign images*. The attack adds perceptible noise to the entire image and is able to generalize across multiple camera distances and angles. Soon after, Eykholt et al. [33] generalized the RP<sub>2</sub> algorithm [34] to design adversarial attacks against object detectors. They studied two different attack scenarios on *stop-sign images*, (i) disappearance attack and (ii) creation attack. Disappearance attack attempts to prevent the object detector to detect a particular object whereas the creation attack tries to make the object detector detect a non-existent object. They also proposed *sticker perturbation* where the noise is only added to the two rectangular strips placed above and below the stop sign. Zhao et al. [153] also proposed a white-box physical adversarial attack against object detectors which generalizes to wider camera angles than Chen et al. [19]. Adversarial attacks against object detectors have also been generalized to more challenging settings. Wei et al. [132] proposed adversarial attack for video

object detection and Jia et al. [56] studied adversarial attack against multiple object tracking.

Apart from adversarial attacks against vision systems, there also has been significant amount of research against vision-and-language systems. Xu et al. [144] proposed targeted adversarial attack against DenseCap [58] and visual question answering (VQA) systems. The goal of the adversarial attack against DenseCap is to keep the proposed regions unchanged while changing the caption of these regions to a target caption. For VQA, the goal of the attack is to change the prediction of the VQA system to a target prediction while limiting the amount of noise added to the image. Chen et al. [17] proposed a targeted adversarial attack, known as Show-and-Fool, for image captioning. They attacked Show and Tell, a neural image caption generator. They proposed two variants of the attack (i) targeted caption, and (ii) targeted keyword. In targeted caption method, the goal is to add noise to the image in order to generate a target caption, whereas in targeted keyword, the goal is to add noise in order to insert a target keyword in the predicted caption. Later, Xu et al. [145] also proposed a structural SVM-based [147] targeted adversarial attack for image captioning.

## 1.4 Adversarial Attacks against NLP Systems

In this section, we discuss adversarial attacks against natural language processing (NLP) systems. Section 1.4.1 discusses the challenges in designing attacks against NLP systems. In Section 1.4.2, we discuss some of the adversarial attacks against NLP systems in brief.

### 1.4.1 Challenges

Designing adversarial attacks against NLP systems is more challenging in comparison to adversarial attacks against vision systems. This is because textual inputs, unlike images, are discrete. Hence, the gradient of the loss function with respect to the input can not be used in a straightforward manner to generate adversarial text. Due to this reason, adversarial attacks against NLP systems are usually less potent than attacks against vision system. This was also observed by Cheng et al. [20] where the authors showed that sequence-to-sequence models used for machine

translation and text summarization are more robust to adversarial attack than image classifiers.

### 1.4.2 Previous Works

One of the earlier works on adversarial attack against NLP systems was by Papernot et al. [100] where the authors designed a white-box adversarial attack for sentiment classification. Similar to FGSM [41], the attack uses the sign of the gradient of the loss function to make multiple changes in the input sentence in order to flip the predicted sentiment. The attack chooses a new word for a particular position in the input sentence so that the sign of the difference of the embeddings of the new and the original word is closest to the sign of the gradient of the loss function with respect to the original word embedding. Liang et al. [74] proposed an adversarial attack against both character-level and word-level text classification systems. Ebrahimi et al. [32] proposed a non-iterative white box attack, known as HotFlip, against text classifiers. HotFlip uses the gradient of the loss with respect to one-hot encoded input to choose the optimum replacement.

Jia and Liang [55] proposed an adversarial attack, known as ADDSENT, against reading comprehension systems. The task of a reading comprehension system is to answer a question based on an input paragraph. Jia and Liang [55] showed that the prediction of the system changes when an adversarial sentence is added at the end of the input paragraph. This adversarial sentence is similar to the question but does not actually change the original answer. Wang and Bansal [131] improved ADDSENT by randomizing the placement of the adversarial sentence in the paragraph and dynamically generating fake answer options. Blohm et al. [11] studied several black-box and white-box attacks against both CNN-based and RNN-based reading comprehension systems. Feng et al. [36] showed that reading comprehension systems predict the same answer with high confidence even after multiple words have been removed from the question. They performed human evaluation to show that the reduced question is unanswerable.

Apart from reading comprehension systems, there has been significant amount of work on adversarial attacks against neural machine translation (NMT) systems. Belinkov and Bisk [8] showed that character-level NMT systems are vulnerable to synthetic and natural noises. Zhao et al. [154] generated adversarial

examples for NMT systems. These adversarial examples are similar to the original sentences and are generated with the goal of either dropping or introducing a keyword in the predicted translation. Ebrahimi et al. [31] showed the efficiency of the aforementioned HotFlip against NMT systems. Cheng et al. [22] showed that replacing words in the original source sentence by their synonyms leads to erroneous predicted translation by the NMT system. Cheng et al. [21] showed that the NMT systems predict different translations for semantically similar source sentences. Liu et al. [76] showed that the NMT systems are extremely sensitive to homophone noises. Cheng et al. [20] studied the robustness of NMT systems when only few words in the source sentence are changed. Zou et al. [156] showed that the predicted translation of the character-level NMT system can be significantly affected by perturbing few characters.

## 1.5 Adversarial Defense

In this section, we discuss some of the works on adversarial defense. For vision systems, similar to adversarial attack, the majority of the work has been on building robust image classifiers. Section 1.5.1 discusses adversarial training, which has been one of the most successful adversarial defense strategy in recent years [26]. In Section 1.5.2, we take a look at some of the challenges associated with adversarial training. Section 1.5.3 discusses some of the other adversarial defense strategies. Finally, in Section 1.5.4, we show that several adversarial defense strategies have been compromised by new and improved adversarial attacks leading to a constant *arms race* between the design of adversarial defense and attack.

### 1.5.1 Adversarial Training

Adversarial training signifies the use of adversarial examples for training a learning system. Adversarial training was formally introduced by Goodfellow et al. [41] where the authors modified the loss function to a linear combination of the standard loss and FGSM adversarial loss. They showed that minimizing the modified loss function leads to image classifiers which are more robust to FGSM attack. Later, Madry et al. [82] proposed adversarial training for image classifiers using a much stronger PGD adversary. They argued that PGD attack is the *universal* first-order adversary, i.e., PGD attack is the strongest adversarial attack which

solely relies on the information of the gradient. Hence, the usage of adversarial examples obtained via PGD attack is ideal for adversarial training. Unlike Goodfellow et al. [41] where linear combination of standard loss and adversarial loss was considered, Madry et al. [82] simply minimized the PGD adversarial loss. Zhang et al. [152] proposed an alternative framework of adversarial training, known as TRADES. The loss function in TRADES consists of two terms. The first term minimizes the standard loss whereas the second term minimizes the difference between the predictions of the original and adversarial examples. Madry et al. [82] and Zhang et al. [152] studied adversarial training for smaller datasets. Adversarial training was later scaled to ImageNet dataset as well [141]. Zhang and Wang [151] proposed an adversarial training framework for object detection.

Adversarial training has also been studied for NLP systems. Jia and Liang [55] and Wang and Bansal [131] studied adversarial training for reading comprehension systems. Belinkov and Bisk [8] and Ebrahimi et al. [31] studied black-box adversarial training for NMT systems. They showed that adversarial training leads to NMT systems which are more robust to character-level noises in the source sentence. Cheng et al. [21] studied adversarial training in order to make NMT systems robust to minor changes in the source sentence.

### 1.5.2 Challenges of Adversarial Training

Kurakin et al. [70] found that non-iterative adversarial training (such as training with FGSM adversarial examples) leads to *label leaking* effect. In *label leaking* effect, the image classifier learns to map the adversarial noise to the true label. In other words, the adversarial noise *leaks* the true label. Due to this, the image classifier overfits on the adversarial noise and achieves higher adversarial accuracy and lower natural accuracy. To remedy this effect, Kurakin et al. [70] suggests to perform non-iterative adversarial training where the *true label* is not used for generating adversarial examples. This effect is not found in iterative adversarial training.

Another main challenge of adversarial training is that it makes the system robust only to the specific type of noise used during training. This has been a common effect across vision and NLP systems. For example, Kurakin et al. [70] showed that image classifiers trained with non-iterative adversarial training are not robust to iterative adversarial attacks. Jia and Liang [55] showed that adversarial training of

reading comprehension systems makes the system robust to ADDSENT. However, a variant of ADDSENT is still able to fool the system. Similar observations were made by Ebrahimi et al. [31] for NMT systems trained for different character-level noises.

Lastly, a major challenge of adversarial training is that it is computationally expensive. For example, Xie et al. [141] used 128 Nvidia V100 GPUs for PGD adversarial training on ImageNet dataset. There have been some works which attempt to make adversarial training less expensive [115, 139, 150]. However, Andriushchenko and Flammarion [3] showed that these methods do not scale well to large  $\ell_\infty$  noises. They proposed FGSM adversarial training with gradient alignment to bridge the gap between FGSM adversarial training and PGD adversarial training. The gradient alignment tries to align the the gradient of loss with respect to the original input with the gradient of the loss with respect to randomly perturbed input. The gradient alignment step requires double backpropagation which increases the runtime in comparison to standard FGSM adversarial training.

### 1.5.3 Other Approaches

Papernot et al. [101] proposed *defensive distillation* for designing robust image classifiers. In defensive distillation, the classifier is *retrained* using softmax probabilities instead of the ground truth. The authors argue that training using these soft labels allows the classifier to generalize better around the neighborhood of the original data. While defensive distillation attempts to design robust classifiers, there also have been works which focus mainly on detecting adversarial examples [35, 84, 120, 143]. Metzen et al. [84] proposed augmenting the classifier with an adversarial detection subnetwork. However, they showed that it is possible to design adversarial attack which can fool both the classifier and detector. To remedy this issue, they proposed joint adversarial training of detector and classifier. Feinman et al. [35] sampled multiple model architectures obtained using dropout technique [121]. They showed that adversarial examples have higher uncertainty in the model output in comparison to original examples. Based on this insight, they used uncertainty estimates to detect adversarial examples. Xu et al. [143] proposed *feature squeezing* for detecting adversarial examples. They explore several feature squeezing methods such as bit depth reduction, median filtering, and image denoising. The main idea of their approach is that model's prediction

on adversarial example differs significantly before and after feature squeezing. Song et al. [120] proposed *PixelDefend* where log-likelihoods from PixelCNN [94, 112] is used for detecting adversarial examples. Furthermore, *PixelDefend* uses a greedy technique to *purify* the adversarial examples. The purified image is then fed to the image classifier. Akhtar et al. [1] proposed perturbation rectifying network (PRN) to defend against universal adversarial perturbation [88].

Apart from detecting adversarial examples, there also has been significant amount of work on certified defenses which provide theoretical guarantee regarding adversarial robustness for image classifiers [42, 73, 106, 118, 138]. Raghunathan et al. [106] used semidefinite programming to provide an upper bound on the worst-case loss for two-layer networks. They further minimize this upper bound to build robust image classifiers. Wong and Kolter [138] used outer approximation to provide an upper bound on the worst-case loss. Unlike Raghunathan et al. [106], their approach can be generalized to convolutional layers as well. Sinha et al. [118] proposed a robust surrogate loss obtained via Lagrangian relaxation and showed that, for imperceptible adversarial perturbation, the robust loss is easy to optimize. Lécuyer et al. [73] proposed PixelDP, which uses differential privacy to provides robustness guarantee for image classifiers. Gowal et al. [42] proposed interval bound propagation (IBP) which uses interval arithmetic to provide an upper bound on the maximum possible difference between pair of logits. The authors showed that IBP is computationally cheap and can be used to train robust classifiers on large datasets.

#### 1.5.4 Defense and Attack: An Endless Cycle?

Carlini and Wagner [16] proposed a targeted adversarial attack which is able to circumvent defensive distillation. The proposed attack achieved 100% success rate against image classifiers trained with defensive distillation. Carlini and Wagner [15] investigated the efficiency of 10 defense techniques which rely on *detecting* adversarial examples. They showed that, in a white-box setting, where the adversary has perfect knowledge of the defence and model's parameters, it is possible to design new loss functions to break all the 10 defense techniques. He et al. [48] proposed an adversarial attack to break defenses relying on feature squeezing. Athalye et al. [5] showed that multiple defense techniques such as *PixelDefend* rely on gradient masking [102]. Since majority of the adversarial attacks rely on the gradient

for designing adversarial examples, gradient masking allows these defense techniques to circumvent the attack. Hence, these defense techniques do not really result in robust image classifiers. To show this, Athalye et al. [5] proposed new adversarial attacks which succeed in circumventing these defenses. Along similar lines, Uesato et al. [126] showed that gradient-free adversarial attacks are able to bypass defenses which rely on gradient masking. Mosbach et al. [90] showed that adversarial logit pairing [62] provide *apparent robustness* by making the surface of the loss function harder to navigate. Furthermore, they also showed that it is possible to circumvent adversarial logit pairing by performing multiple random restarts of PGD attack. Croce and Hein [26] proposed a variant of PGD, known as Auto-PGD along with a new loss function which is invariant to shift and rescaling of logits. Furthermore, they showed that multiple defenses which were robust to PGD attack are vulnerable to Auto-PGD based attacks. So far, as a robust adversarial defense strategy, adversarial training has stood the test of time [26, 126]. As an example, Croce and Hein [26] showed that adversarially trained classifiers are robust to Auto-PGD based attack as well. In lieu of seemingly robust defenses being circumvented by new and improved attacks, Carlini et al. [14] proposed several guidelines for evaluating adversarial defenses in future.

## 1.6 Thesis Outline and Contributions

The goal of this thesis is to study the adversarial robustness of state-of-the-art deep learning systems. In this regard, this thesis explores evasion attacks across various vision and NLP tasks. For vision systems, as we have seen, there has been a plethora of work on studying adversarial robustness of image classifiers. However, this thesis mainly explores evasion attacks for other vision systems, specifically vision-and-language systems such as visual question answering (VQA), and image captioning. For NLP systems, this thesis mainly explores *invariance-based* evasion attacks against neural machine translation (NMT) systems and multiple-choice question answering systems. For NMT systems, the proposed *invariance-based* evasion attacks generate adversarial examples for which the ground truth is not available. This makes standard adversarial training *infeasible*. This thesis explores adversarial defense strategies in such a scenario. Finally, this thesis studies the generalizability of invariance-based attack to multiple choice QA systems and the



ability of such systems to handle different types of interventions on the input paragraph.

The main contributions of this thesis are as follows:

1. We explore the robustness of state-of-the-art VQA systems against an adversarial attack which only adds noise to the background of the image. We show that VQA systems can be fooled by adding minimal adversarial background noise. This holds true even for toy datasets where the VQA systems have very high accuracy and good-quality attention maps.
2. While the adversarial attacks designed so far are *task specific*, we propose a *task agnostic* adversarial attack, named Mimic and Fool. The proposed attack is designed for vision systems and only requires the knowledge of feature extractor in order to attack the system. We study the efficacy of this attack against VQA and image captioning systems. Furthermore, we propose a variant of this attack, named One Image Many Outputs (OIMO), which generates *natural looking* adversarial examples. We show that the proposed attack is able to attack invertible architectures as well.
3. Previous adversarial attacks against NMT systems make small changes to the source sentence in order to change the predicted translation. We take a different approach and propose an *invariance-based* adversarial attack which makes as many changes to the source sentence as possible with the goal of keeping the predicted translation unchanged. We also explore several evaluation metrics suitable to evaluate the proposed attack.
4. The proposed *invariance-based* adversarial attack generates adversarial examples for which there is no ground truth available. This makes the task of designing an adversarial defense harder in comparison to previous adversarial attacks against NMT systems where standard adversarial training was shown to be effective. In this regard, we explore several adversarial defense strategies for NMT systems to counteract such an attack.
5. We study the generalizability of the *invariance-based* adversarial attack to text-based multiple choice question answering systems. In this regard, we compare the adversarial robustness of CNN and LSTM-based multiple choice question answering systems. Furthermore, we also study the generalizability of these systems to two types of interventions on the input paragraph,

namely, mask intervention and option-specific intervention. The option-specific intervention ensures that the chosen option is the *correct* answer. The results show that CNN-based MCQ systems generalize better to such option-specific interventions in comparison to their LSTM counterpart.

## 1.7 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 studies the robustness of state-of-the-art VQA systems against adversarial background noise. Chapter 3 studies the *task agnostic* attack against vision systems. Chapter 4 studies *invariance-based* adversarial attack against state-of-the-art NMT systems. This chapter also discusses relevant metrics to evaluate the efficiency of the attack. Chapter 5 explores defense strategies to enhance robustness of NMT systems against *invariance-based* attacks. Chapter 6 studies the generalizability of such *invariance-based* attacks to text-based multiple choice question answering systems. This chapter also analyses the generalizability of such systems to interventions on the input paragraph. Finally, Chapter 7 discusses the findings of this thesis and scope of future works.

## Chapter 2

# Attacking VQA systems via Adversarial Background Noise

*Rarely do more than three or four variables really count. Everything else is noise.*

Martin J. Whitman

Given an image and a question about an image, the goal of a VQA system is to answer the question using the relevant information contained in the image. Previous adversarial attacks on VQA systems show that, for real-world datasets, minimal adversarial noise added to the entire image suffices to fool such systems [144]. In this chapter, we study whether VQA systems can be fooled by adding noise only to the background of the image, keeping the main image content unchanged. We study the vulnerability of VQA systems to adversarial background noise on real-world as well as toy datasets.

The rest of this chapter is organized as follows. Section 2.1 discusses about VQA datasets and systems which are used for experimentation. This section also discusses the adversarial attack proposed by Xu et al. [144] in detail. In light of Xu et al. [144], Section 2.2 discusses the motivation for the proposed adversarial attack. Section 2.3 describes the adversarial attack methodology. Section 2.4 provides the implementation details. Section 2.5 describes the datasets used for the proposed adversarial attack. Section 2.6 analyzes the results of the proposed adversarial attack. Section 2.7 shows several adversarial examples across VQA systems and datasets. Finally, Section 2.8 summarizes the chapter.

## 2.1 Background

In this section, we provide a background for the proposed adversarial attack. Section 2.1.1 discusses the two toy datasets and one real-world dataset used in this work. Section 2.1.2 describes the two state-of-the-art VQA systems. Section 2.1.3 discusses the adversarial attack by Xu et al. [144] in detail.

### 2.1.1 VQA datasets

In this chapter, we study the proposed adversarial attack on two toy VQA datasets, namely, SHAPES [51], and CLEVR [57]; and a real-world VQA dataset, i.e., VQA v2.0 [43]. The SHAPES dataset consists of yes/no questions. The images in SHAPES consist of several 2D objects (such as circle, triangle and squares) of different colors and sizes placed in a  $3 \times 3$  grid. CLEVR dataset contains images of 3D rendered objects (i.e. cubes, spheres, and cylinders) of different sizes and material. The question in CLEVR dataset belong to 6 different categories, namely, yes-no, color, shape, number, size, and material. VQA dataset [4] was the first large-scale real-world dataset containing  $\sim 200K$  real-world images and  $\sim 600K$  questions. Despite the wide diversity of questions and images, Kafle and Kanan [61] showed that a system which only takes question as input achieves  $\sim 50\%$  accuracy on VQA dataset. This is primarily due to the biases present in the dataset, such as, *tennis* being the most common answer for a question starting with “*What sport is*”. To remedy this issue, Goyal et al. [43] proposed VQA v2.0 dataset. This dataset attempts to reduce the bias present in VQA dataset by using *complementary images*. Concretely, given an image-question-answer triplet  $(I, Q, A)$ , VQA v2.0 dataset adds an additional triplet  $(I', Q, A')$  such that  $A'$  is different from  $A$ . In this case,  $I'$  is the complementary image to  $I$ . Figure 2.1 shows 4 complementary images along with the respective question and answer.

### 2.1.2 VQA systems

**End-to-End Module Network (N2NMN):** N2NMN [51] is based on the idea of differentiable modules where each module performs a specific task. N2NMN breaks down a question into a layout of modules (known as *module layout*) using a natural language parser. Since different module layout leads to different network



**Figure 2.1:** Complementary images from VQA v2.0 (Photo Courtesy: Goyal et al. [43])

architecture, N2NMN allows for an architectural design catered to a question. For example, for the question “How many hats are in the image?”, the module layout will look like  $count(find())$  where the  $find$  module will attend on the  $hats$  present in the image and the  $count$  module will count the  $hats$  using the attention output of  $find$  module. A possible drawback of N2NMN is that the set of modules might vary depending on the complexity of the dataset and thus, they need to be defined beforehand.

**MAC network:** MAC network [52] is a recurrent architecture based on the Memory, Attention and Composition (MAC) cell. Each MAC cell consists of two hidden states: memory and control. Memory stores the *intermediate results* and the control has the information about the *reasoning step*. Similar to an LSTM cell, MAC cell also consists of several units such as input unit, control unit, read unit, write unit and output unit. Each unit has its set of predefined operations either to attend on a relevant part of image/question or for aggregating information. Design of a general purpose reasoning cell allows MAC network to overcome the aforementioned drawback of N2NMN.

### 2.1.3 Adversarial Attack Against VQA systems

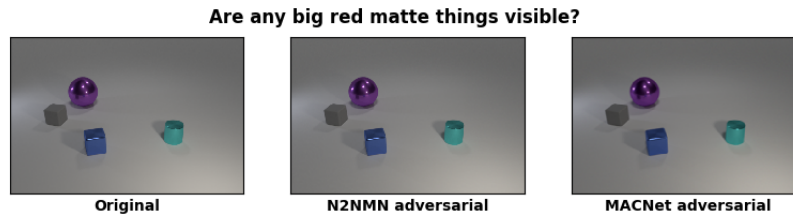
Xu et al. [144] proposed a targeted adversarial attack for VQA and image captioning. For VQA, their adversarial loss function is given as

$$L = -\log p_{target} + \lambda_1 \mathbb{1}(y_{curr} \neq y_{target})(\tau + \log p_{curr}) + \lambda_2 \text{ReLU}(d(I, I_{org}) - B + \epsilon) \quad (2.1)$$

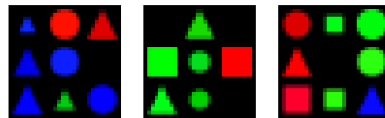
where  $p_{target}$  denotes probability of target class,  $\mathbb{1}(y_{curr} \neq y_{target})$  is an indicator function to check if the current predicted class,  $y_{curr}$ , is different from the target class,  $y_{target}$ ;  $p_{curr}$  denotes the probability of the current predicted class,  $d(I, I_{org})$  denotes the distance between the current image,  $I$ , and the original image,  $I_{org}$ ; and  $\lambda_1, \tau, \lambda_2, B, \epsilon$  are hyperparameters. Thus, their loss function consists of three parts. The first part tries to maximize the probability of the target class, the second part tries to minimize the probability of current predicted class if the current predicted class is different from target class and the third part ensures that the adversarial image lies within a fixed neighborhood of the original image. Their attack achieved 100% success rate for N2NMN and Multimodal Compact Bilinear pooling (MCB) [37] on Gold dataset (created using VQA validation set [4]). Furthermore, they showed that the success of an attack is more dependent on the target question-answer pair than the image.

## 2.2 Motivation

Since Xu et al. [144] already achieved 100% success rate, the fact that VQA systems are vulnerable to adversarial attack is established. The goal of the present work is to study the *extent of vulnerability* of current VQA systems. This motivates the idea of limiting the freedom of the adversary by only adding noise to the background of the image. Furthermore, unlike Xu et al. [144] who studied VQA systems trained on VQA dataset, we perform experiments on VQA v2.0 dataset which is a more balanced dataset, as discussed earlier. Apart from VQA v2.0, we also perform experiments on toy datasets: SHAPES, and CLEVR. The rationale behind this is that, unlike real-world dataset, the state-of-the-art VQA systems are already able to achieve impressive accuracy on the two toy datasets (N2NMN achieves 100% accuracy on SHAPES and MAC network achieves  $> 98\%$  accuracy on CLEVR) and have very good attention maps for a given image-question pair.



**Figure 2.2:** Example of the proposed attack. For the above question, both N2NMN and MAC network give the correct answer (“no”) when original image is given as input but incorrect answer (“yes”) when respective adversarial image is given as input. The noise is added only to the outside background of the image.



**Figure 2.3:** Images from SHAPES dataset.

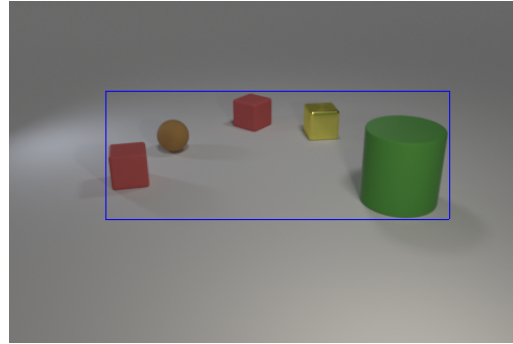
As a result, such toy datasets offer a nice way of understanding how adversarial examples work. For CLEVR images, we find a rectangular mask which segregates the main image content from the outside background and the adversary is only allowed to modify the outside background. Figure 2.2 shows an example of the proposed attack.

## 2.3 Methodology

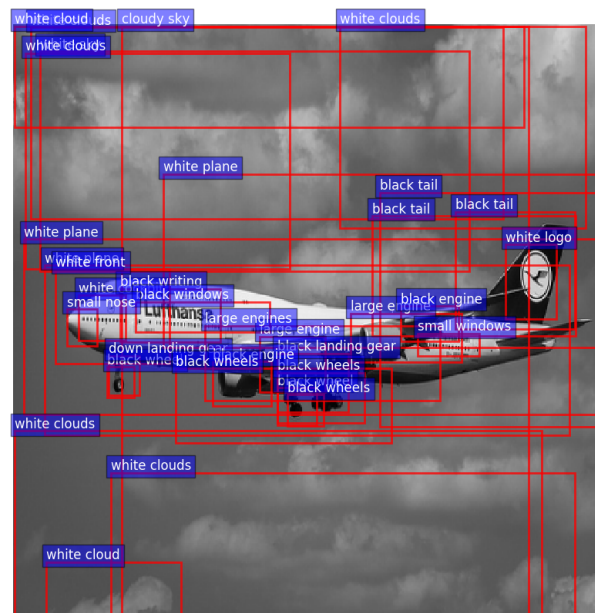
In this section, we explain our proposed method for generating adversarial examples. The method consists of two stages. In the first stage, we detect the background for an image, and in the second stage, we perform targeted adversarial attack on the given image-question pair by modifying just the background. In the following subsections, we describe the two stages in detail.

### 2.3.1 Background Detection

For SHAPES, the background is the set of *black pixels* present in an image (as evident from Figure 2.3). For CLEVR, we use canny edge detector [13] to detect edges of the objects present in the image. Then, we find the smallest rectangle such that all the detected edges lie inside it. Figure 2.4 gives an example of an image along with the rectangle. As we can see from Figure 2.4, the blue



**Figure 2.4:** Background Detection for CLEVR. Only the pixels outside the blue rectangle are modified in the proposed attack.



**Figure 2.5:** Background Detection for VQA v2.0. The pixels which are not inside any of the boxes are modified in the proposed attack.

rectangle segregates the main content of the image from the outside background. The proposed attack only modifies the pixel belonging to the outside background. Although in none of the cases did the rectangle cover the entire image, if such a case were to arise, one can always pad the image. For VQA v2.0, we detect the objects present in the image using Faster R-CNN, a state-of-the-art object detector [107]. The pixels which are not inside any of the detected boxes are considered as background. Figure 2.5 shows one such example.



### 2.3.2 Targeted Adversarial Attack

Let  $(I^{org}, Q, A)$  denote the original image-question-answer triplet where the image  $I^{org} \in [0, 255]^{h \times w \times 3}$ . Let  $A_{target}$  be the adversarial target for the image-question pair  $(I^{org}, Q)$ . Let  $\mathbb{A}$  denote the set of unique answers present in the dataset. Then, any VQA system can be considered as a function  $f_\theta : I \times Q \rightarrow \mathbb{P}$  where  $\mathbb{P}$  is the space of probability distributions over  $\mathbb{A}$  and  $\theta$  denotes the parameters of the model.

Given an image-question pair  $(I^{org}, Q)$ , our goal is to find an adversarial image  $I^{adv} \in [0, 255]^{h \times w \times 3}$  such that

$$\operatorname{argmax}(f_\theta(I^{adv}, Q)) = A_{target} \quad (2.2)$$

$$\text{and } \delta_{ijk} = 0 \text{ if } M_{ijk} = 0 \quad (2.3)$$

where  $\delta$  denotes the difference image  $I^{adv} - I^{org}$  and  $M \in \{0, 1\}^{h \times w \times 3}$  is a binary mask which is 1 for *background pixels* of  $I^{org}$  and 0 otherwise. Equation 2.3 ensures the *background constraint*, i.e., the adversarial noise is only added to the background pixels, detected in Section 2.3.1, of  $I^{org}$ .

For an image-question-answer triplet  $(I, Q, A_{target})$ , we use the standard cross-entropy loss function given by

$$L = -\log(f_\theta(I, Q)^T e_{target}) \quad (2.4)$$

where  $e_{target}$  is the one-hot encoded vector with value 1 for  $A_{target}$  and 0 otherwise. Thus,  $f_\theta(I, Q)^T e_{target}$  denotes the probability assigned by the model to  $A_{target}$ .

Before updating the image  $I$ , we mask the gradient  $\nabla_I L$  as follows

$$\nabla_I^{mask} L = c \cdot (M \odot \nabla_I L) \quad (2.5)$$

where  $\nabla_I^{mask} L$  is the *masked gradient* of loss function  $L$  with respect to image  $I$ ,  $\odot$  denotes element wise multiplication, and  $c \in \mathbb{R}$  is a hyperparameter. We use the *masked gradient*  $\nabla_I^{mask} L$  for updating the image  $I$ . We also use the truncating

**Algorithm 2.1:** Targeted Adversarial Attack

---

**Input:**  $I^{org}, Q, A_{target}, f_{\theta}, c_{init}$   
**Output:**  $I^{adv}$

$c_{list} \leftarrow [c_{init}]$   
 $last_{true}, last_{false}, success \leftarrow 0$   
 $I_{list}, norm_{list} \leftarrow []$

**for**  $i \leftarrow 1$  **to**  $max_{trials}$  **do**

$I \leftarrow I^{org}$   
 $c \leftarrow c_{list}[i]$

**for**  $j \leftarrow 1$  **to**  $max_{iters}$  **do**

Compute loss  $L$  for  $(I, Q, A_{target})$   
 Compute  $\nabla_I^{mask} L$   
 Update  $I$  using  $\nabla_I^{mask} L$  and truncate  
**if**  $I$  satisfies Equations 2.2 and 2.3 **then**

$success \leftarrow 1$   
 $\delta \leftarrow I - I^{org}$   
**append**  $I$  to  $I_{list}$  and  $\|\delta\|_2$  to  $norm_{list}$   
 $last_{true} \leftarrow i$   
**if**  $last_{false} = 0$  **then**  
 | **append**  $0.5c$  to  $c_{list}$   
**else**  
 | **append**  $0.5(c + c_{list}[last_{false}])$  to  $c_{list}$   
**break**

**else if**  $j = max_{iters}$  **then**

$last_{false} \leftarrow i$   
**if**  $last_{true} = 0$  **then**  
 | **append**  $2c$  to  $c_{list}$   
**else**  
 | **append**  $0.5(c + c_{list}[last_{true}])$  to  $c_{list}$

**end for**

**end for**

**if**  $success = 1$  **then**  
 |  $I^{adv} \leftarrow I_{list}[\text{argmin}(norm_{list})]$   
**else**  
 |  $I^{adv} \leftarrow I$

**return**  $I^{adv}$

---

function to ensure that  $I \in [0, 255]^{h \times w \times 3}$ . Note that the term  $M \odot \nabla_I L$  in Equation 2.5 ensures that only the background pixels of  $I^{org}$  are modified.

The hyperparameter  $c$  controls the amount of adversarial noise added at each iteration. Larger values of  $c$  allow for more noise to be added which can lead to faster convergence. To find an adversarial image with minimal adversarial noise,

we use a *binary search* strategy similar to Chen et al. [17]. For a particular  $c$ , if the attack is unsuccessful, we either increase the value of  $c$  by a factor of 2 if no previous  $c$  led to successful attack or take the average of the current  $c$  and the last  $c$  for which the attack was successful. Similarly, if the attack is successful, we either decrease the value of  $c$  by a factor of 2 if no previous  $c$  led to an unsuccessful attack or take the average of the current  $c$  and the last  $c$  for which the attack was unsuccessful. This process is repeated  $max_{trials}$  times. For each  $c$ , we run the proposed attack until either we find an adversarial image  $I^{adv}$  which satisfies Equations 2.2 and 2.3 or  $max_{iter}$  iterations are reached. Finally, amongst all the successful adversarial images, we choose the one whose difference image  $\delta$  has the smallest  $\ell_2$ -norm. If the attack is unsuccessful for all the  $max_{trials}$   $c$ 's, we return the last image (the image obtained using the largest value of  $c$ ) as the final image. Algorithm 2.1 summarizes the proposed attack. From the algorithm, one can see that the attack can be generalized for any number of categories.

## 2.4 Implementation Details

For SHAPES dataset, we train the N2NMN for 10,000 iterations using the original source code<sup>1</sup>. The trained model achieves 100% accuracy on train, validation and test set of SHAPES. For CLEVR dataset, we use the trained N2NMN provided with the source code and train the MAC network of length 4 for 25 epochs using the original source code<sup>2</sup>. N2NMN and MAC network achieve accuracy of 83.6% and 98.0% respectively on CLEVR validation set. For VQA v2.0 dataset, we use the trained N2NMN provided with the source code. To detect objects using the object detector, we use Faster R-CNN [2] trained on Visual Genome dataset [67]. For the proposed attack, in order to make the comparison across models and datasets fair, we use the same hyperparameters throughout. The value of the hyperparameters are as follows:  $c_{init} = 100$ ,  $max_{trials} = 5$ ,  $max_{iters} = 1,500$ . We implement the proposed attack in Tensorflow and the code is publicly available<sup>3</sup>. To implement the attack proposed by Xu et al. [144], we use the same value of hyperparameters as mentioned in their paper.

---

<sup>1</sup><https://github.com/ronghanghu/n2nmn>

<sup>2</sup><https://github.com/stanfordnlp/mac-network>

<sup>3</sup><https://github.com/akshay107/vqa-adv-background>

## 2.5 Datasets

The proposed attack is studied on five datasets:

1. **SHAPES**: We combine the validation and test sets of SHAPES to get 2048 image-question pairs. Since SHAPES is a yes/no dataset, we set  $A_{target}$  to *yes* when original answer is *no* and set  $A_{target}$  to *no* when original answer is *yes*.
2. **CLEVR<sub>same</sub>**: We choose 1000 image-question pairs from CLEVR validation set which were answered correctly by both MAC network and N2NMN. In CLEVR dataset, the answer can belong to six different categories: yes-no, color, shape, number, size and material. For CLEVR<sub>same</sub>, we randomly choose  $A_{target}$  (different from original answer) from same category as the original answer.
3. **CLEVR<sub>diff</sub>**: For this dataset, we use the same 1000 image-question pairs as CLEVR<sub>same</sub> but in this case, we randomly choose  $A_{target}$  from a different category than the original answer.
4. **VQA<sub>same</sub>**: We choose 500 image-question pairs from VQA v2.0 validation set which were correctly answered by N2NMN and only had one unique answer (since each image-question pair was answered by multiple annotators). In VQA v2.0 dataset, the answer can belong to four categories: yes-no, number, color and other. For VQA<sub>same</sub>, we randomly choose  $A_{target}$  (different from original answer) from same category as the original answer. We also ensure that the original answer and  $A_{target}$  are not similar since VQA v2.0 contains similar answers such as “black and white” and “black and gray”.
5. **VQA<sub>diff</sub>**: For this dataset, we use the same 500 image-question pairs as VQA<sub>same</sub> but in this case, we randomly choose  $A_{target}$  from a different category than the original answer.

## 2.6 Results

Table 2.1 summarizes the result of the proposed attack. In Table 2.1,  $\|\delta\|_2$  is normalized by the total number of pixels. This is done because the size of the

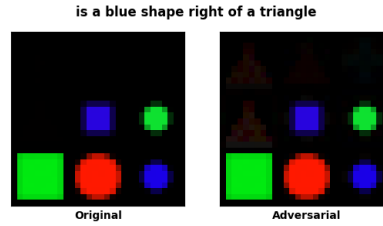
Dataset	Model	SR	$\ \delta\ _2$ ( <i>mean</i> $\pm$ <i>std</i> )	bg-size
SHAPES	N2NMN	68.9%	$0.37 \pm 0.33$	$65.0 \pm 7.6$
CLEVR <sub>same</sub>	N2NMN	100.0%	$1.9 \times 10^{-4} \pm 2.2 \times 10^{-4}$	$66.8 \pm 13.7$
	MAC network	100.0%	$1.2 \times 10^{-3} \pm 6.1 \times 10^{-4}$	
CLEVR <sub>diff</sub>	N2NMN	22.3%	$1.3 \times 10^{-3} \pm 1.8 \times 10^{-3}$	
	MAC network	73.9%	$8.6 \times 10^{-3} \pm 6.4 \times 10^{-3}$	
VQA <sub>same</sub>	N2NMN	88.8%	$1.9 \times 10^{-3} \pm 1.9 \times 10^{-3}$	$2.3 \pm 2.5$
VQA <sub>diff</sub>	N2NMN	56.4%	$4.4 \times 10^{-3} \pm 3.0 \times 10^{-3}$	

**Table 2.1:** Success rate (SR) of the proposed attack. For  $\|\delta\|_2$ , the mean and standard deviation is calculated over the *successful* cases. bg-size denotes the *mean*  $\pm$  *std* of the percentage of an image detected as background using Section 2.3.1.

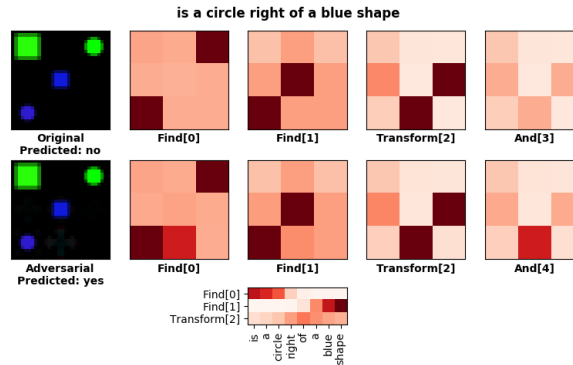
Dataset	Model	Success-rate	$\ \delta\ _2$ ( <i>mean</i> $\pm$ <i>std</i> )
SHAPES	N2NMN	74.2%	$0.32 \pm 0.06$
CLEVR <sub>same</sub>	N2NMN	100.0%	$1.2 \times 10^{-3} \pm 1.2 \times 10^{-3}$
	MAC network	100.0%	$1.5 \times 10^{-2} \pm 8.8 \times 10^{-3}$
CLEVR <sub>diff</sub>	N2NMN	21.7%	$2.2 \times 10^{-2} \pm 5.3 \times 10^{-3}$
	MAC network	100.0%	$1.8 \times 10^{-2} \pm 5.1 \times 10^{-3}$
VQA <sub>same</sub>	N2NMN	100.0%	$7.0 \times 10^{-3} \pm 3.8 \times 10^{-3}$
VQA <sub>diff</sub>	N2NMN	100.0%	$8.2 \times 10^{-3} \pm 2.8 \times 10^{-3}$

**Table 2.2:** Success rate of Xu *et al.* [144]. For  $\|\delta\|_2$ , the mean and standard deviation is calculated over the *successful* cases.

difference image depends on the dataset as well as the model. For example, images in SHAPES dataset are  $30 \times 30 \times 3$  and for CLEVR, N2NMN takes input image of size  $320 \times 480 \times 3$  whereas MAC network takes input image of size  $224 \times 224 \times 3$ . For VQA v2.0, N2NMN takes input image of size  $448 \times 448 \times 3$ . The bg-size in Table 2.1 denotes the background size relative to the original image, detected using the method described in Section 2.3.1. Table 2.2 summarizes the result of Xu *et al.* [144] on the five datasets. From Tables 2.1 and Table 2.2, we can see that, apart from SHAPES, the proposed attack adds significantly less noise than the attack in [144]. This is due to the binary search strategy employed in the proposed attack as opposed to the fixed learning rate used in [144]. In the following subsections, we analyze the result of the proposed attack on the five datasets in detail. We also visualize the attention maps for the original and adversarial images of N2NMN and MAC network. Furthermore, we study the *transferability* of the adversarial images for the two models.



**Figure 2.6:** Answer changes to *yes* for the adversarial image. For the adversarial image, a light silhouette of a triangle can be seen in top left and middle left. Such cases were considered *unsuccessful*.



**Figure 2.7:** Attention visualization for SHAPES. Note that the textual attention map remains same for the two images.

### 2.6.1 Success Rate

For SHAPES dataset, we test the proposed attack for N2NMN. Out of 2048 image-question pairs, the proposed attack is successful for 1431 pairs. However, for SHAPES dataset, the original answer might change for the adversarial image when the background is modified by the attack. In such a case, we can't consider the attack *successful*. To address this issue, we request 3 human annotators to answer 1431 adversarial image-question pairs as yes/no and discard those pairs where at least 2 annotators give a different answer than the original. There were 19 such pairs. Figure 2.6 shows one such pair. The final success rate, after discarding these cases, is 68.9%. Similarly, for Xu et al. [144], there were 27 such pairs which were discarded. As a matter of convenience, all the difference images are given in the supplementary material<sup>4</sup>.

It can be seen from Table 2.1 that  $\|\delta\|_2$  for SHAPES dataset is higher in comparison with the other two datasets. This can be because the SHAPES dataset has very little variability since it consists of only three shapes (circle, triangle and squares) which makes it easier for N2NMN to distinguish between these three shapes and

<sup>4</sup><https://www.isical.ac.in/~utpal/resources.php>

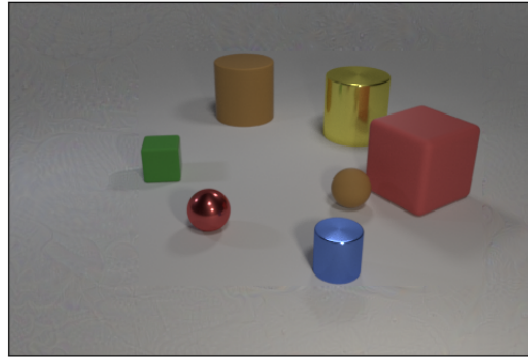
a random shape. The low success rate of Xu et al. [144], as shown in Table 2.2, further supports this rationale.

Figure 2.7 shows the attention maps for original and adversarial image-question pair. As we can see from the textual attention map,  $Find[0]$  is attending on “circle” and  $Find[1]$  is attending on “blue shape”. For the original image, both the find modules give the correct attention map. However for the adversarial image,  $Find[0]$  recognizes an *extra shape* in the bottom-middle as a circle which results in incorrect prediction by the network. The *extra shape* has four corners due to which the find module identifies it as a *circle*. This demonstrates the bias learned by the module during the training stage. Figure 2.7 is a typical example which demonstrates that studying adversarial attack in toy datasets offers better understanding than real world datasets.

The proposed attack achieves 100% success rate for both the models on  $CLEVR_{\text{same}}$  dataset. In the ideal scenario, a model with attention mechanism shouldn’t be fooled by background noise. The 100% success rate shows that the attention mechanisms used in the current state-of-the-art systems can be fooled even when the main content of the image is left untouched. Hence, one does not need to modify the entire image, as in Xu et al. [144], to fool state-of-the-art VQA systems. Furthermore, the amount of noise required to fool the system is imperceptible, as is evident from Table 2.1. From Table 2.1, we also see that  $\|\delta\|_2$  is an order of magnitude higher for MAC network than N2NMN which shows that the MAC network is more resilient to background noise than N2NMN. As expected, Xu et al. [144] also achieves 100% success rate for both the models.

The proposed attack achieves success rate of 22.3% and 73.9% for N2NMN and MAC network respectively on  $CLEVR_{\text{diff}}$  dataset. Note that Xu et al. [144] also achieves low success rate (21.7%) for N2NMN and 100.0% for MAC network. From Table 2.1, we can see that, for both the models,  $\|\delta\|_2$  is higher for  $CLEVR_{\text{diff}}$  dataset in comparison with  $CLEVR_{\text{same}}$  dataset. This is intuitive since one expects that more background noise will be needed for the model to predict an answer from different category than the same category. Since in  $CLEVR_{\text{diff}}$  dataset, the target answer doesn’t semantically match the question, the low success rate of N2NMN suggests that N2NMN is able to better capture the language-bias than MAC network. However, similar to  $CLEVR_{\text{same}}$  dataset, we see from Table 2.1 that the  $\|\delta\|_2$  is almost an order of magnitude higher for MAC network than N2NMN. This finding suggests that MAC network can better capture the language bias than

The big matte thing in front of the green rubber block is what color?



N2NMN prediction: cylinder  
Target: cube  
Actual: red

**Figure 2.8:** N2NMN predicts same category as the target answer.

N2NMN since more background noise is needed to make the MAC network predict an answer from a different category. In the following paragraph, we resolve this dilemma and state the explicit criterion under which a different targeted attack can be successful for N2NMN.

For CLEVR dataset, N2NMN has 15 modules. Out of these 15 modules, 7 are *answer modules*. By *answer modules*, we refer to those modules which occur at the end of the layout. These 7 modules are as follows: Exist, Count, EqualNum, MoreNum, LessNum, SameProperty and Describe. As mentioned before, CLEVR consists of answers from 6 different categories. However, apart from Describe module, all the other *answer modules* can only predict an answer belonging to a single category because during training they are only exposed to their respective category (e.g. Count module is only exposed to number category). On the other hand, Describe module can predict an answer belonging to one of four categories (color, shape, size and material). Hence, for N2NMN, a different category attack is possible if and only if both the original and the target answer belong to one of the four aforementioned categories. We find that out of 1000 image-question pairs in  $CLEVR_{diff}$ , 235 image-question pairs satisfy this criteria. Hence, effectively our proposed attack is successful for 223 out of 235 pairs (94.9%). In fact, our attack got the target category correct for the remaining 12 pairs as well. Figure 2.8 gives one such example. We find that the 217 successful cases of Xu et al. [144] also belong to the aforementioned 235 image-question pairs. For MAC network, we find that out of 261 unsuccessful pairs, our attack got the target category correct for only 1 pair.

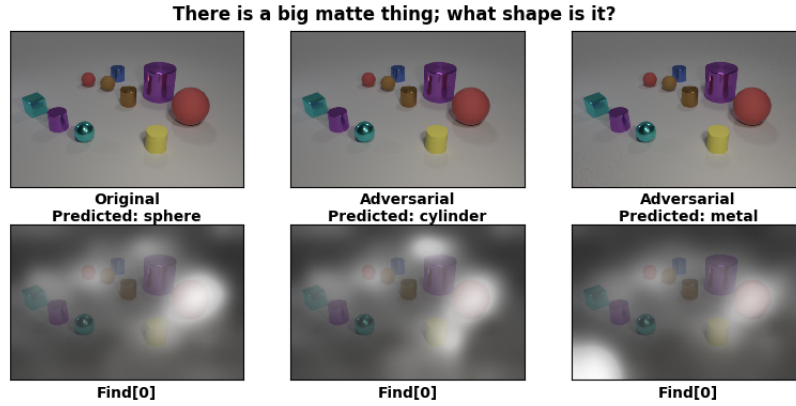


For VQA v2.0 datasets, as mentioned earlier, we detect the background using Faster-RCNN. As can be seen from Table 2.1, bg-size for VQA is very low. This is because the bounding boxes detected by Faster R-CNN often cover a very large portion of the image. Despite this fact, the proposed attack achieves a success rate of 88.8% on VQA<sub>same</sub> dataset. This shows that it is possible to fool state-of-the-art VQA systems by modifying very few pixels in the image. Xu et al. [144], which modifies the entire image, achieves a success rate of 100.0% on this dataset.

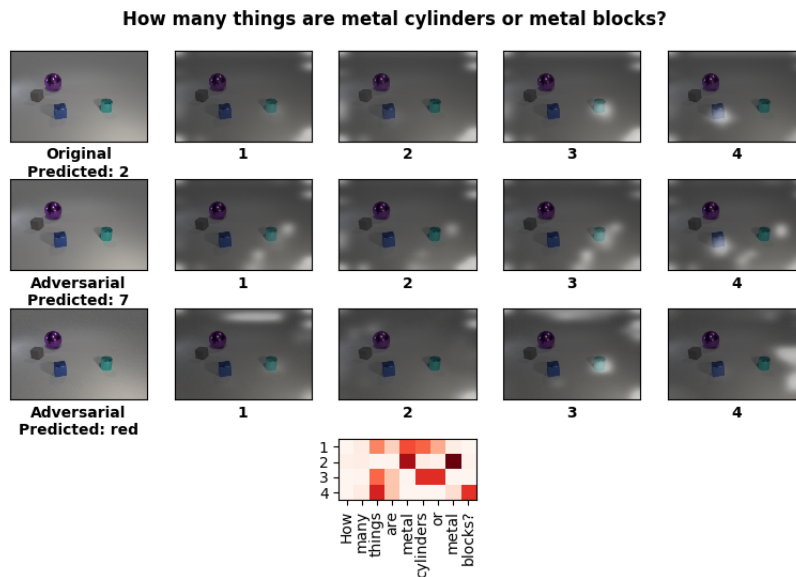
For VQA<sub>diff</sub> dataset, the proposed attack achieves a success rate of 56.4%. On the other hand, Xu et al. [144] achieve a success rate of 100.0%. This is different from the finding in CLEVR<sub>diff</sub> dataset where both the methods achieve a low success rate. The reason behind the high success rate of Xu et al. [144] on this dataset is that N2NMN uses only four modules for VQA v2.0 dataset, namely, Find, Transform, And, and Describe. Out of the four modules, only Describe module is an answer module. Because of this, it is possible to force N2NMN to predict from any desired category, which was not the case in CLEVR<sub>diff</sub>. This clearly shows that in order to make N2NMN robust to different category attack, multiple answer modules pertaining to separate categories are required. Note that Xu et al. [144] observed low success rate for N2NMN on *Non-sense dataset*. However, in their dataset, the question was not relevant to the image. In all the five datasets studied in this chapter, the question is always relevant to the image.

## 2.6.2 Visualizing attention

Figure 2.9 and 2.10 show the attention maps for N2NMN and MAC network respectively for CLEVR. Similarly, Figure 2.11 shows the attention maps for N2NMN for VQA v2.0. In all the figures, we show the original image and successful adversarial images from same and different category attack along with their respective attention maps. In Figure 2.9, the find module is able to locate the *red sphere* for the original image. For the adversarial image for same category attack, while the find module is able to locate the *red sphere*, it also attends to the region at the top of the *purple cylinder*. Whereas, for the adversarial image from different category attack, the find module mostly attends at the bottom left corner of the image. Similarly in Figure 2.10, MAC cell is able to locate the *metal cylinder* and the *metal block* (time-step 3 and 4 respectively) for the original image. However, for the adversarial image from different category attack, it is unable to locate the



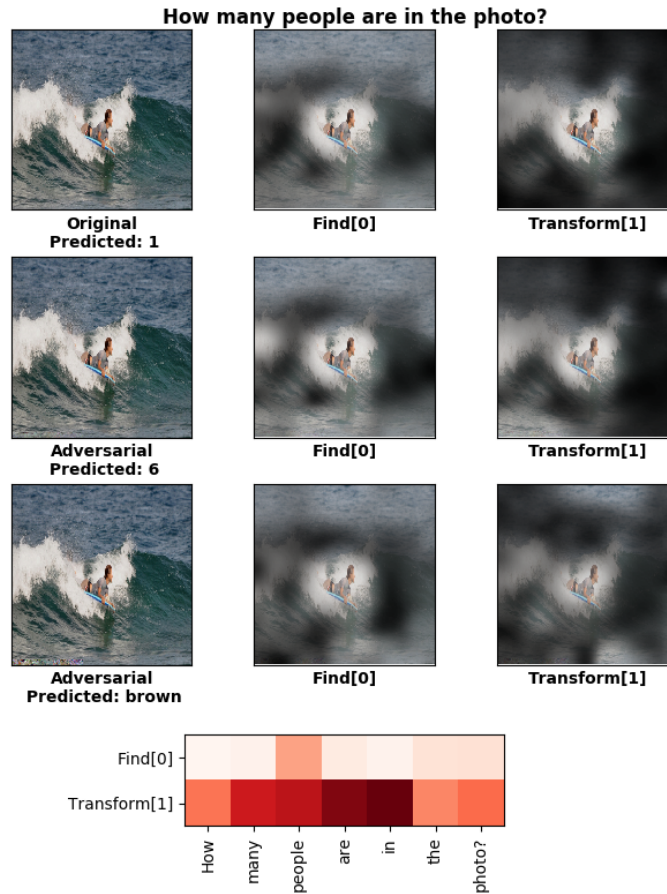
**Figure 2.9:** Attention visualization for N2NMN on CLEVR. For both the adversarial images, the attack was successful i.e. the predicted answer was  $A_{target}$ .



**Figure 2.10:** Attention visualization for MAC network on CLEVR. Note that the textual attention map remains same for all the images. For both the adversarial images, the attack was successful i.e. the predicted answer was  $A_{target}$ .

*metal block*. For the adversarial image from same category attack, while it is able to locate the two objects, the attention map is mostly outside the main content of the image for all the timesteps.

In Figure 2.11, for all the three images, the attention of Find module is not localised. For the original image, the attention of the Transform module is localised on the main object present in the image. For the adversarial image corresponding to the  $VQA_{same}$  dataset (i.e. the middle row), the attention of the Transform module is less localised and is also spread on the bottom-left of the image. Finally, the attention of the Transform module is least localised for the adversarial image



**Figure 2.11:** Attention visualization for N2NMN on VQA v2.0. Note that the textual attention map remains same for all the images. For both the adversarial images, the attack was successful i.e. the predicted answer was  $A_{target}$ .

corresponding to the  $VQA_{diff}$  dataset. The three attention maps corresponding to the Transform module do have significant attention on the main object present in the image. The same is true for the Find module in Figure 2.9. This gives an idea about how little does one need to *misguide* the attention in order to generate adversarial outputs.

### 2.6.3 Transferability Results

In this section, we study the transferability of adversarial examples between the two models. For this purpose, we use the final 1000 adversarial images returned by our proposed attack for both the CLEVR datasets and both the models. As mentioned earlier, the two models accept input images of different sizes, so we resize the adversarial images accordingly. We use two evaluation metrics: (i)

**Success-rate:** Percentage of image-question pairs for which the target model gave  $A_{target}$  as the answer. (ii) **Non-targeted Success-rate:** Percentage of image-question pairs for which the target model gave incorrect answer.

For CLEVR<sub>same</sub> dataset, the success-rate and non-targeted success-rate for N2NMN are 4.3% and 6.5% respectively. Whereas for MAC network, the non-targeted success-rate is only 0.1% . For CLEVR<sub>diff</sub> dataset,  $A_{target}$  was never predicted as the answer by the two models. The non-targeted success rate for N2NMN and MAC network on CLEVR<sub>diff</sub> dataset is 10.5% and 34.4% respectively. The high non-targeted success-rate for MAC network can be due to the low success rate of our proposed attack for N2NMN on CLEVR<sub>diff</sub> dataset because of which most of the adversarial image-question pairs have larger background noise (since they are unsuccessful image-question pairs). This hypothesis is supported by the fact that for MAC network, out of 344 successful image-question pairs, only 1 was successful image-question pairs for N2NMN. Whereas, for N2NMN, out of 105 image-question pairs, 72 were successful image-question pairs for MAC network. One possible explanation for the low transferability of adversarial examples amongst the two models could be that N2NMN uses VGG-16 features [117] whereas MAC network uses ResNet-101 features [46].

#### 2.6.4 Mean/Median Filtering as Defense?

In this section, we study whether simple pixel smoothing techniques like mean filtering, median filtering can act as a defense against the proposed attack. We observe that, for SHAPES and VQA v2.0 dataset, when such filters are added as a preprocessor, the accuracy of the trained N2NMN model on clean images reduces significantly ( $\sim 8\%$  for SHAPES and  $\sim 6\%$  for VQA v2.0). The significant drop in accuracy for SHAPES is primarily due to the small spatial size of the images. As a result, adding mean/median filter ( $3 \times 3$ ) results in hazy images. On the other hand, the drop in accuracy of the two VQA systems is significantly less ( $\sim 1\%$ ) for CLEVR. For CLEVR<sub>same</sub> and CLEVR<sub>diff</sub> datasets, when the adversarial images corresponding to Table 2.1 are fed to the integrated VQA pipeline (mean/median-filter + VQA system), the success rate reduces significantly by a factor of over 2. At this point, it is crucial to note that adding a mean/median filter to the VQA pipeline does not inherently make the entire VQA pipeline robust to adversarial attacks. Rather, the drop in success rate is due to *low transferability* and not

*enhanced robustness*. Athalye et al. [5] showed that the defense techniques which are based on image denoising give a *false sense* of security by *obfuscating* the gradients. For image classifiers which rely on image denoising, they further showed that if the adversary is aware of the preprocessing module (this is not a strong assumption given the fact that the adversary already knows the model architecture and its parameters) then the attack can be slightly modified in order to achieve very high success rate. To further establish this claim, we rerun our proposed attack, for  $\text{CLEVR}_{\text{same}}$  and  $\text{CLEVR}_{\text{diff}}$ , by backpropogating the gradient through the preprocessor (mean/median filter in this case) as well. The new attack achieves similar success rates as reported in Table 2.1 (100.0% for both the VQA systems on  $\text{CLEVR}_{\text{same}}$  and  $\sim 22.0\%$  for N2NMN,  $\sim 72.0\%$  for MAC network on  $\text{CLEVR}_{\text{diff}}$ ).

## 2.7 Examples of the Attack

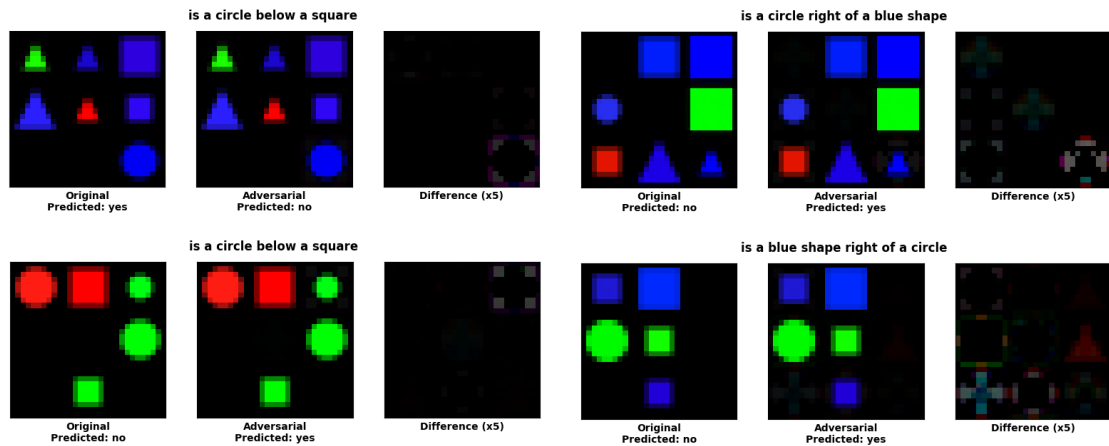


Figure 2.12: Examples for N2NMN on SHAPES.

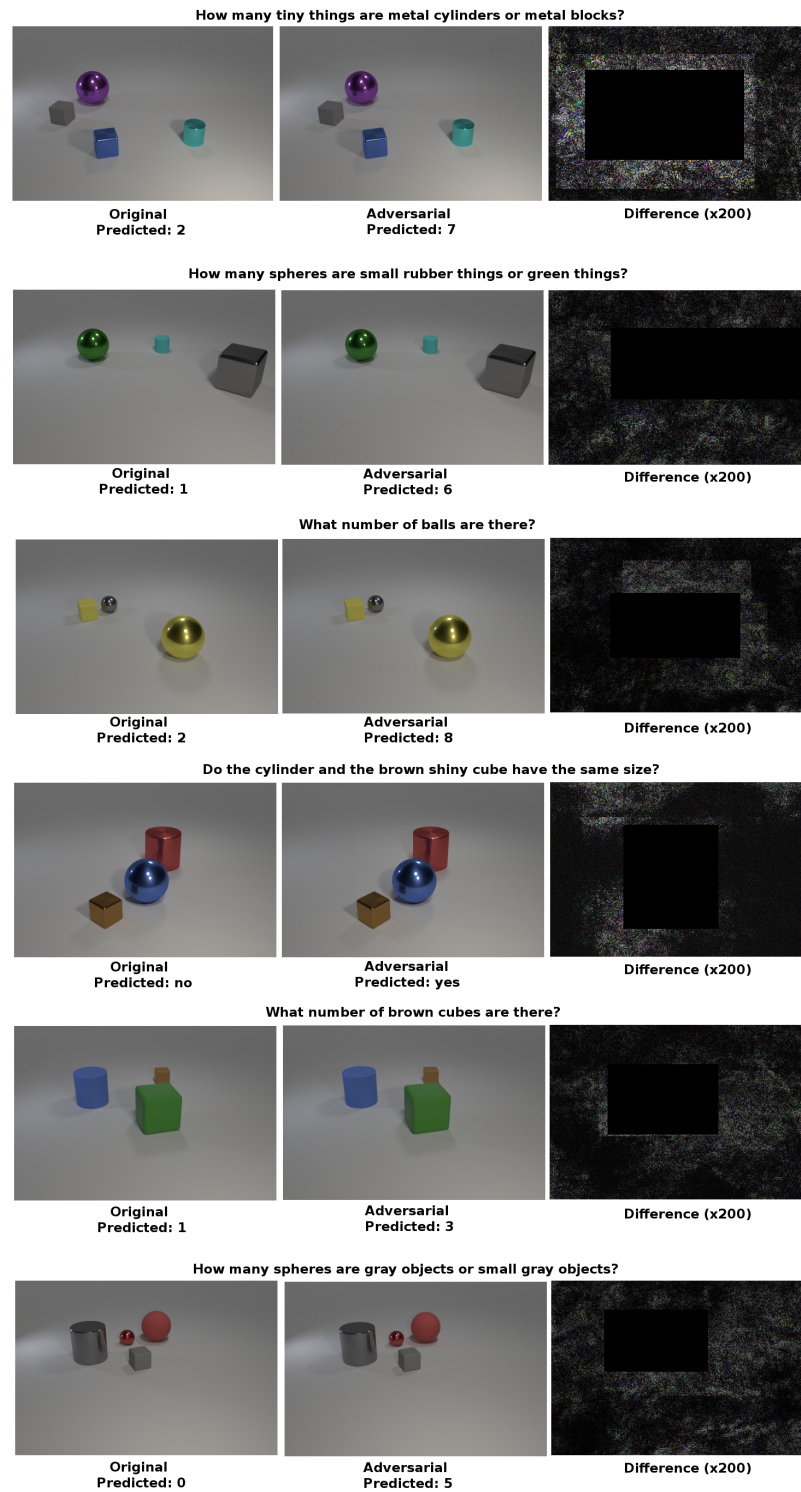


Figure 2.13: Examples for N2NMN on CLEVR<sub>same</sub>.

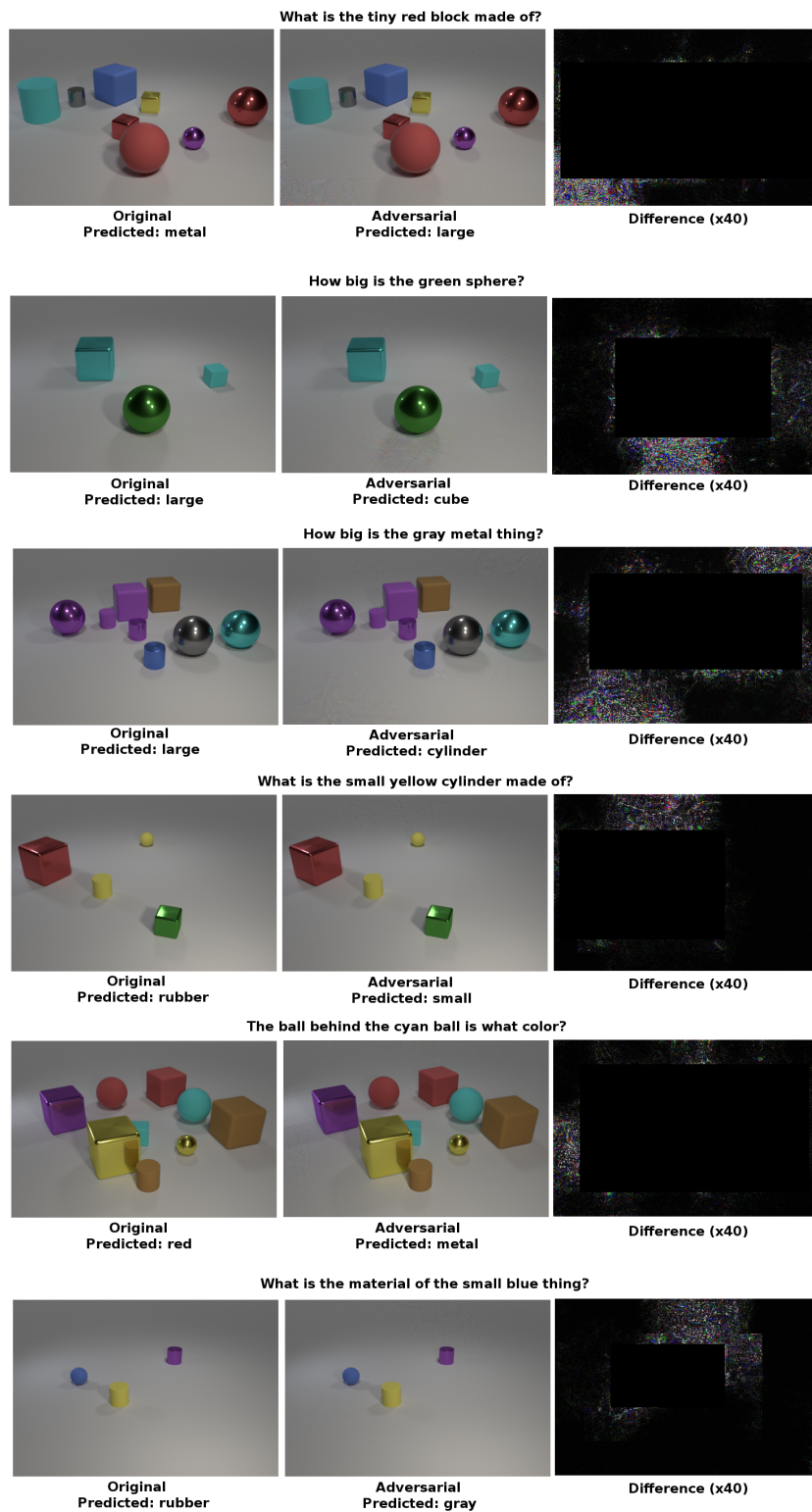


Figure 2.14: Examples for N2NMN on CLEVR<sub>diff</sub>.

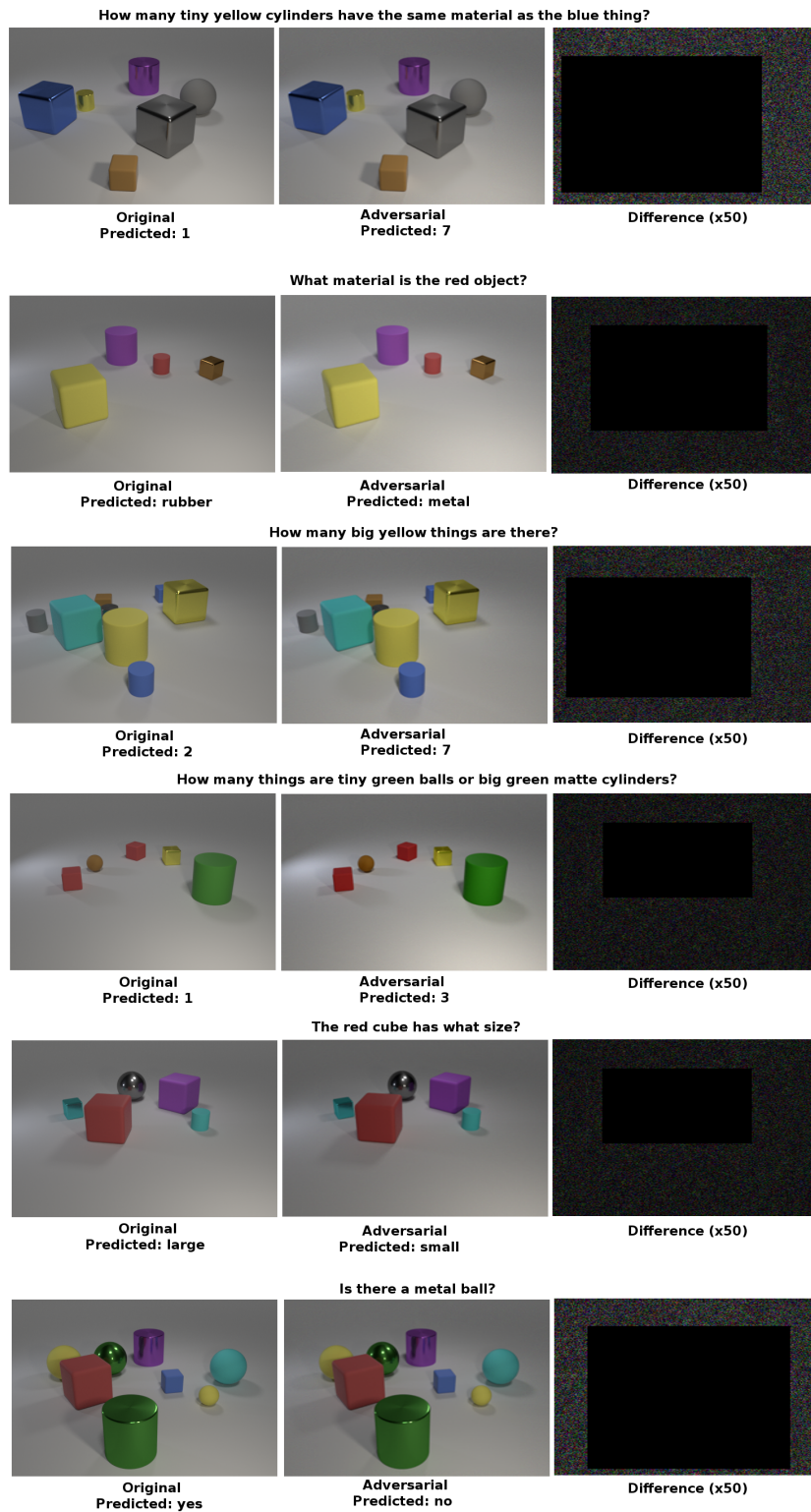


Figure 2.15: Examples for MAC network on CLEVR<sub>same</sub>.



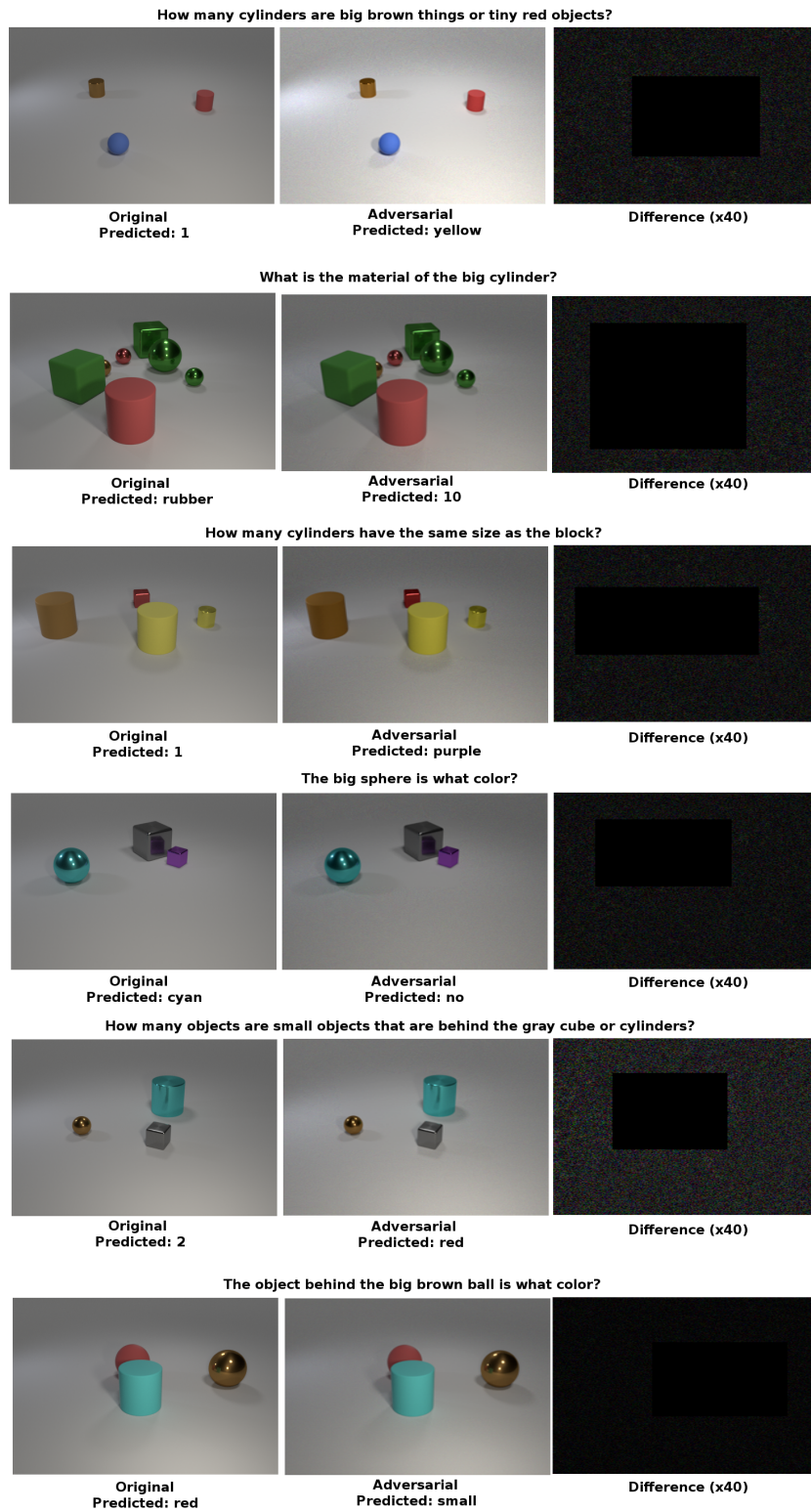


Figure 2.16: Examples for MAC network on CLEVR<sub>diff</sub>.

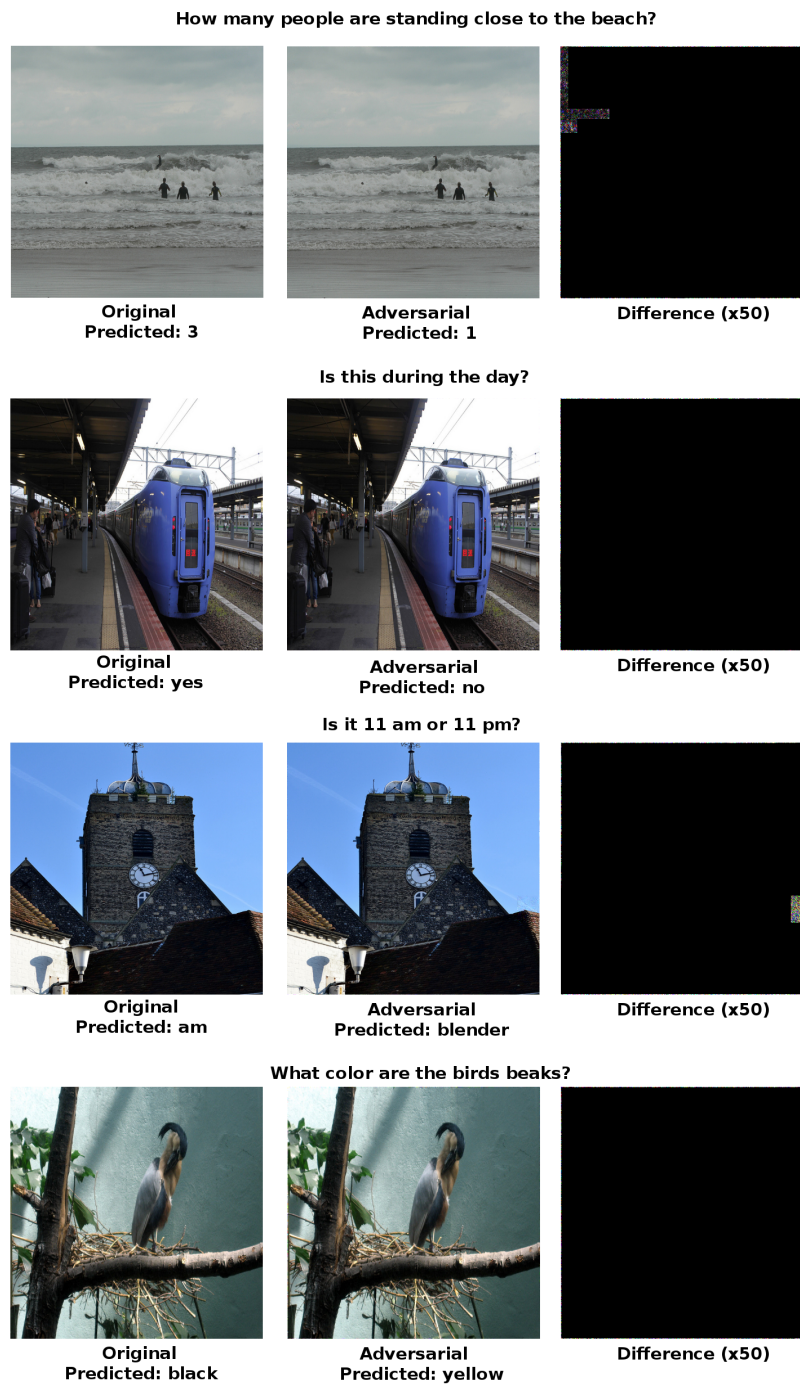


Figure 2.17: Examples for N2NMN on  $VQA_{\text{same}}$ .

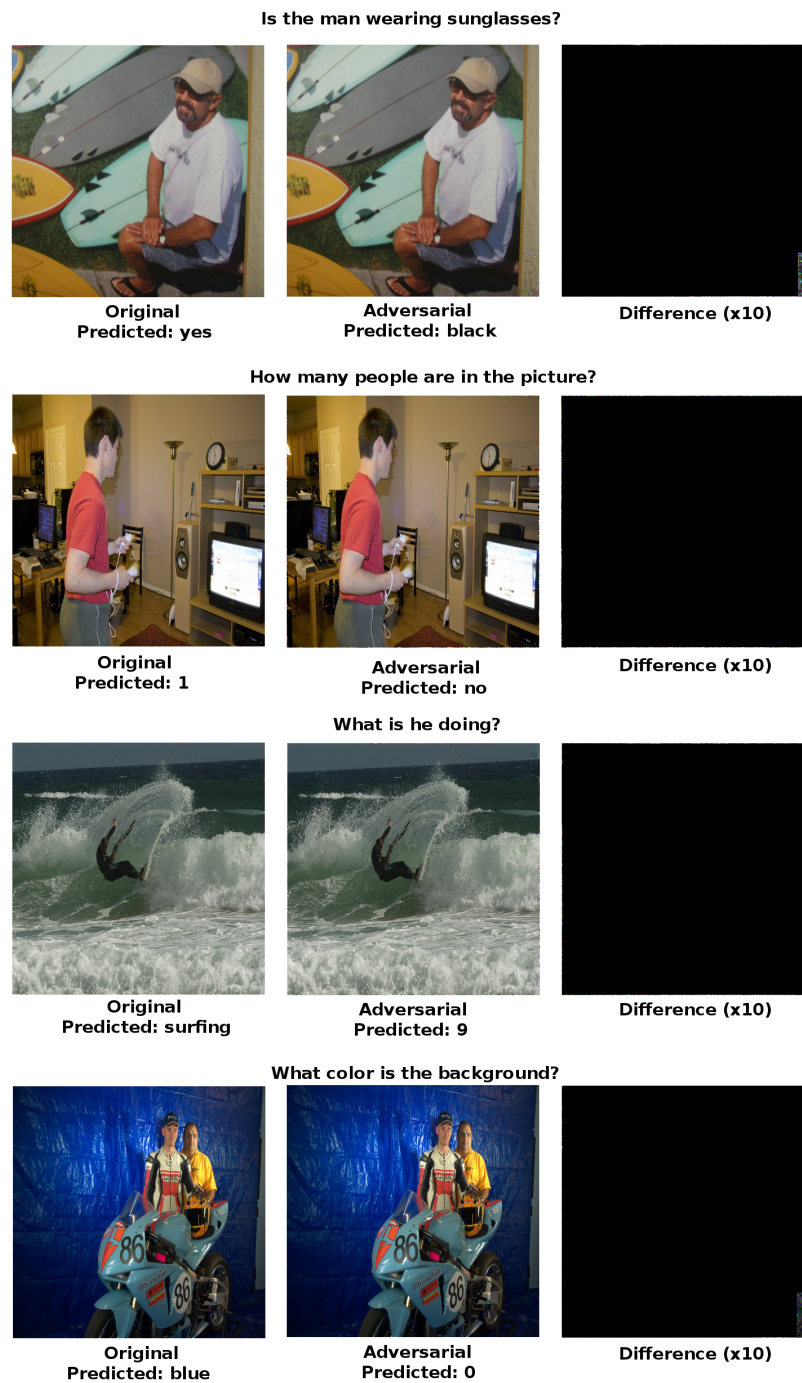


Figure 2.18: Examples for N2NMN on VQA<sub>diff</sub>.

## 2.8 Summary

This chapter proposed a targeted adversarial attack for VQA systems by only modifying the background pixels. We tested our method on two state-of-the-art models: MAC network and N2NMN and three datasets: SHAPES, CLEVR and VQA v2.0. Our proposed attack achieved impressive success rate for both the models. For CLEVR dataset, we showed that the current state-of-the-art models can be fooled simply by adding imperceptible noise to the background. The visualizations of the attention maps demonstrated how the attention mechanism can be *distracted* by such noise. Furthermore, we showed that, for successful adversarial examples, the norm of the difference image, i.e.  $\|\delta\|_2$ , is higher for MAC network than N2NMN for both the datasets, CLEVR<sub>same</sub> and CLEVR<sub>diff</sub>. We also explicitly stated the criterion under which a different category attack can be successful for N2NMN. The high success rate of Xu et al. [144] for N2NMN on VQA<sub>diff</sub> further shows the importance of category-specific answer module. Keeping this criterion in mind, a naive solution to secure N2NMN against different category attack is to replace Describe module with four separate modules (one for each category). However, this will not be a feasible solution for datasets with large number of categories.

## Chapter 3

# Mimic and Fool: A Task-Agnostic Adversarial Attack

*...mirages are things that aren't really there that you can see very clearly. "How do you see something that isn't there?"... "sometimes it's much simpler than seeing things that are"...*

Norton Juster

The adversarial attacks, proposed in the literature, are designed in a task-specific fashion. Such attacks use a task-specific adversarial loss function to generate adversarial examples. In this chapter, we propose a *task-agnostic* attack, named *Mimic and Fool*, for vision systems. The proposed attack is based on the fact that, for downstream computer vision tasks (e.g. image captioning, image segmentation etc.) the current deep learning systems use an image classifier (e.g. VGG16, ResNet50, Inception-v3 etc.) as a feature extractor. Thus, given a feature extractor, the proposed attack tries to find an adversarial image which can mimic the image feature of the original image; thereby ensuring that the two images give the same (or similar) output regardless of the task. This makes the proposed attack an *invariance-based* attack. Furthermore, since the proposed attack only requires information about the feature extractor of the model, it is a *gray-box attack*. In this chapter, we also propose a slight modification to the proposed attack to generate natural-looking adversarial images. Additionally, we also show the applicability of the proposed attack for invertible architecture.

The rest of this chapter is organized as follows. Section 3.1 discusses the two image captioning systems used for experimentation. This section also highlights the main idea and key advantages of Mimic and Fool over task-specific attacks. Section 3.2 describes the attack methodology. Section 3.3 provides the implementation details. Section 3.4 discusses the results of Mimic and Fool with regards to success rate, time, and applicability to invertible architecture. Section 3.5 provides some examples of the proposed attack. Finally, Section 3.6 summarizes the chapter.

## 3.1 Background

In this section, we discuss the architecture of two image captioning systems, namely, Show and Tell [129], and Show Attend and Tell [142] in Sections 3.1.1 and 3.1.2 respectively. In Section 3.1.3, we highlight the main idea of Mimic and Fool and its key advantages over task-specific attacks.

### 3.1.1 Show and Tell

Show and Tell [129] was the first deep neural network for image captioning. The captioning system uses a CNN in conjunction with LSTM. The image is embedded as a 2048-dimensional vector obtained via the global average pooling layer of Inception-v3. Hence, Inception-v3 acts as a feature extractor. The image embedding is then passed as input to LSTM in order to generate captions.

### 3.1.2 Show, Attend and Tell

Show, Attend and Tell [142] uses the idea of attention in order to attend on relevant portions of images while generating captions. The captioning system uses VGG-16 as a feature extractor. More precisely, the image is embedded as a  $14 \times 14 \times 512$  feature map. Hence, unlike Show and Tell, this image feature has spatial information about the image, i.e., each 512-dimensional vector in the feature map corresponds to a specific image portion. During decoding, at each time step, the attention layer predicts a probability distribution over the 512-dimensional vectors and outputs a *weighted* vector. The weighted vector is fed as input to the LSTM.



**Figure 3.1:** Examples of Mimic and Fool. The first two rows show the original and adversarial images along with the predicted captions by Show and Tell and Show Attend and Tell respectively. The last row shows original and adversarial image for N2NMN (Q, P denote the question and the predicted answer respectively).

### 3.1.3 Proposed Attack: Overview and Advantages

*Mimic and Fool* exploits the non-invertibility of CNN-based *feature extractors* to attack the downstream model. Given a model and its feature extractor, the proposed attack is based on the simple hypothesis that if two images are *indistinguishable* for the *feature extractor* then they will be *indistinguishable* for the model as well. Thus to attack any model, attacking its feature extractor suffices. Based on this insight, *Mimic and Fool* finds an adversarial image which can *mimic*

the feature of the original image thereby *fooling* the model. Figure 3.1 shows examples of *Mimic and Fool* on two captioning models: Show and Tell [129], Show Attend and Tell [142] and one VQA model: end-to-end neural module network (N2NMN) [51]. It is crucial to note that the goal of Mimic and Fool differs from traditional adversarial attacks [17, 41, 69, 144]. In traditional adversarial attacks, small amount of noise is added to the image in order to fool the model to generate a different output. Whereas, in Mimic and Fool, the goal is to generate an adversarial image which can fool the model to predict the same output as the original image. Hence, Mimic and Fool is an *invariance-based* attack. As we can see from Figure 3.1, the adversarial images obtained via *Mimic and Fool* are noisy images. In order to generate *natural-looking* adversarial images, we also propose a modified version of our attack, namely One Image Many Outputs (OIMO). In OIMO, we start with a fixed natural image and restrict the amount of noise that can be added to the image. Apart from task-agnosticity, *Mimic and Fool* offers other significant advantages: (i) *Mimic and Fool* is extremely fast and requires less computing resources since only the feature extractor needs to be loaded in the memory instead of the entire model. (ii) Due to the task-agnostic nature, we need to run the attack only at image-level which is a huge advantage in terms of time saved for tasks involving multiple modalities as input such as visual question answering. An adversarial attack designed specifically for VQA will run at image-question pair level.

## 3.2 Methodology

In this section, we describe the proposed attack, *Mimic and Fool*, and *One Image Many Outputs* (OIMO) which is able to generate *natural looking* adversarial images. Since both the attacks are task agnostic, we describe the attack in terms of the feature extractor instead of the model.

### 3.2.1 Mimic and Fool

Let  $f : \mathbb{R}^{m \times n \times 3} \rightarrow \mathbb{R}^d$  denote the feature extractor of the model. Hence,  $d$  will be  $14 \times 14 \times 1024$  if we extract *conv4* features from ResNet101 and  $d$  will be 2048 if we use output of average pooling layer of Inception-v3 as image feature. Let  $I_{org} \in [0, 255]^{m \times n \times 3}$  denote the original image. Given  $I_{org}$  and a feature extractor



$f$ , our goal is to find an adversarial image  $I_{adv} \in [0, 255]^{m \times n \times 3}$  which can *mimic* the image features of  $I_{org}$ . We model this task as a simple optimization problem given by

$$\min_I \frac{\|f(\text{trunc}(I)) - f(I_{org})\|_2^2}{d} \quad (3.1)$$

where  $\|\cdot\|_2$  denotes  $\ell_2$ -norm and *trunc* is truncating function which ensures that the intensity values lie in the range  $[0, 255]$ . Although  $I = I_{org}$  is a solution to the above optimization problem, it is highly unlikely that the algorithm will converge to this solution. This is because convolutional neural networks discard significant amount of spatial information as we go from lower to higher layers. Mahendran and Vedaldi [83] showed that the amount of invariance increases from lower to higher layer of AlexNet [68] and regularizers like *total variation* (TV) are needed to reconstruct the original image from higher layer features of AlexNet. We start with a *zero-image* and run the proposed attack for  $max_{iter}$  iterations and return the final truncated image  $\text{trunc}(I)$  as  $I_{adv}$ .

Some feature extractors such as Inception-v3 require the intensity values of the input image to be in the range  $[-1, 1]$ . In such a case, let  $I'_{org} \in [-1, 1]^{m \times n \times 3}$  be the scaled original image i.e.

$$I'_{org} = 2(I_{org}/255) - 1 \quad (3.2)$$

For this case, we modify the optimization problem defined in Equation 3.1 as follows

$$\min_I \frac{\|f(\tanh(I)) - f(I'_{org})\|_2^2}{d} \quad (3.3)$$

where *tanh* ensures that the input to feature extractor lies within the required range. We run the attack for  $max_{iter}$  iterations and rescale the final image  $\tanh(I)$  to get  $I_{adv}$  i.e.

$$I_{adv} = 255 \left( \frac{\tanh(I) + 1}{2} \right) \quad (3.4)$$

### 3.2.2 One Image Many Outputs

In *One Image Many Outputs* (OIMO), we start with an image  $I_{start} \in [0, 255]^{m \times n \times 3}$  instead of starting with *zero-image*. The image  $I_{start}$  is kept fixed throughout the

experiment. In OIMO, our goal is to modify  $I_{start}$  so as to *mimic* the feature of  $I_{org}$ . Equation 3.1 is modified as follows

$$\min_{\delta} \frac{\|f(\text{trunc}(I_{start} + \delta)) - f(I_{org})\|_2^2}{d} \quad (3.5)$$

Similar to Chen et al. [17], we modify the Equation 3.3 as follows

$$\min_{\delta} \frac{\|f(\tanh(I''_{start} + \delta)) - f(I'_{org})\|_2^2}{d} \quad (3.6)$$

where  $I''_{start} = \text{arctanh}(\lambda I'_{start})$ ,  $I'_{start} \in [-1, 1]^{m \times n \times 3}$  is the scaled starting image,  $\lambda$  is set to 0.9999 to ensure invertibility of  $\tanh$ ,  $\delta \in \mathbb{R}^{m \times n \times 3}$  is the learnable parameter. For this attack, we reduce the value of  $max_{iter}$  and initial learning rate to ensure that  $I_{adv}$  looks very similar to  $I_{start}$ .

Similar to *Mimic and Fool*, after running the attack for  $max_{iter}$  iterations,  $I_{adv}$  for Equation 3.5 is  $\text{trunc}(I_{start} + \delta)$ . For Equation 3.6,  $I_{adv}$  is given by the following equation

$$I_{adv} = 255 \left( \frac{\tanh(I''_{start} + \delta) + 1}{2} \right) \quad (3.7)$$

We name the proposed attack *One Image Many Outputs* since all the adversarial images look very similar to  $I_{start}$ .

### 3.3 Implementation Details

As stated earlier, we study the proposed attack for two image captioning models; Show and Tell, Show Attend and Tell and one VQA model, namely, N2NMN. We train the N2NMN model on VQA v2.0 dataset for 95K iterations with expert policy followed by 65K iterations in policy search after cloning stage using the original source code<sup>1</sup>. The trained N2NMN has 61.72% accuracy on VQAv2 test-dev set. For Show and Tell and Show Attend and Tell, we use already available trained models<sup>2,3</sup>.

<sup>1</sup><https://github.com/ronghanghu/n2nmn>

<sup>2</sup><https://github.com/KranthiGV/Pretrained-Show-and-Tell-model>

<sup>3</sup>[https://github.com/DeepRNN/image\\_captioning](https://github.com/DeepRNN/image_captioning)

Show and Tell uses 2048-dimensional feature from Inception-v3, Show Attend and Tell uses  $14 \times 14 \times 512$  feature map from VGG16, N2NMN uses output of *res5c* layer from ResNet-152 as image feature. The input images are of size  $299 \times 299 \times 3$ ,  $224 \times 224 \times 3$ ,  $448 \times 448 \times 3$  for Inception-v3, VGG16 and ResNet-152 respectively. The trained Show and Tell, Show Attend and Tell *fine-tune* their respective feature extractors whereas N2NMN does not use *fine-tuning*.

For *Mimic and Fool*, we set *max\_iter* to 1000, 1000 and 2000 for Inception-v3, VGG16 and ResNet-152 respectively. The initial learning rate is set to 0.025, 0.025 and 0.0125 for Inception-v3, VGG16 and ResNet-152 respectively. For *One Image Many Outputs*, we set *max\_iter* to 300, 500, 500 and set the initial learning rate to 0.0125, 0.0125, 0.00625 for Inception-v3, VGG16 and ResNet-152 respectively. We use Adam [65] as the optimizer and Keras [24] for implementing the proposed attacks. All experiments are done on a single 11 GB GeForce GTX 1080 Ti GPU. The code for *Mimic and Fool* is publicly available.<sup>4</sup>

### 3.4 Results

For studying the two proposed attacks, 1000 MSCOCO validation images are randomly selected. For the 1000 selected images, there are 5208 image-question pairs in VQA v2.0 dataset. For visual question answering, we discard those image-question pairs where the VQA model predicts the same answer for  $I_{start}$  and  $I_{org}$  (For Mimic and Fool,  $I_{start}$  is a zero-image). This is done to ensure that the VQA model predicts the same answer for  $I_{start}$  and  $I_{org}$  due to adversarial noise rather than language bias. The proposed attack is considered to be *successful* if the model gives the same output for the original and the adversarial image. Hence for image captioning, the two captions need to be exactly the same for the attack to be successful. In the following subsections, we analyze the behavior of the two proposed attacks on the three models: N2NMN, Show and Tell and Show Attend and Tell. We also study the effectiveness of the proposed method for an invertible architecture.

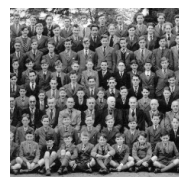
<sup>4</sup><https://github.com/akshay107/mimic-and-fool>

Task	Model	Feature Extractor	Success Rate	Average Time
Image Captioning	Show and Tell	Inception-v3	74.0 %	25.35 sec
	Show, Attend and Tell	VGG16	81.0 %	15.56 sec
VQA	N2NMN	ResNet-152	87.1 %	72.98 sec

**Table 3.1:** Success rate of *Mimic and Fool*

### 3.4.1 Results for Mimic and Fool

Table 3.1 shows the success rate of *Mimic and Fool* for the three models. Out of 5208 image question pairs, N2NMN predicts the same answer for  $I_{org}$  and zero-image for 1707 pairs. Out of the remaining 3501 pairs, *Mimic and Fool* is successful for 3049 image question pairs. This yields success rate of 87.1%. The high success rate shows that it is possible to *mimic* features extracted from a very deep network like ResNet-152 as well. Since *Mimic and Fool* is *task-agnostic*, we need to run the proposed attack at image level instead of image-question pair level. This is a huge advantage since it results in a drastic reduction in time. The advantage will be even more pronounced for any future tasks which have multiple modalities as input with image (or video) being one of the modalities. Figure 3.2 shows the predicted answer by N2NMN for different image-question pairs. From Figure 3.2, we can see that a single adversarial image suffices for three image-question pairs.



Original



Adversarial

**Q:** How many hands are in the picture?

**P:** 4

**P<sub>zero</sub>:** 1

**Q:** What type of place is this?

**P:** school

**P<sub>zero</sub>:** kitchen

**Q:** Is this a recent photo?

**P:** no

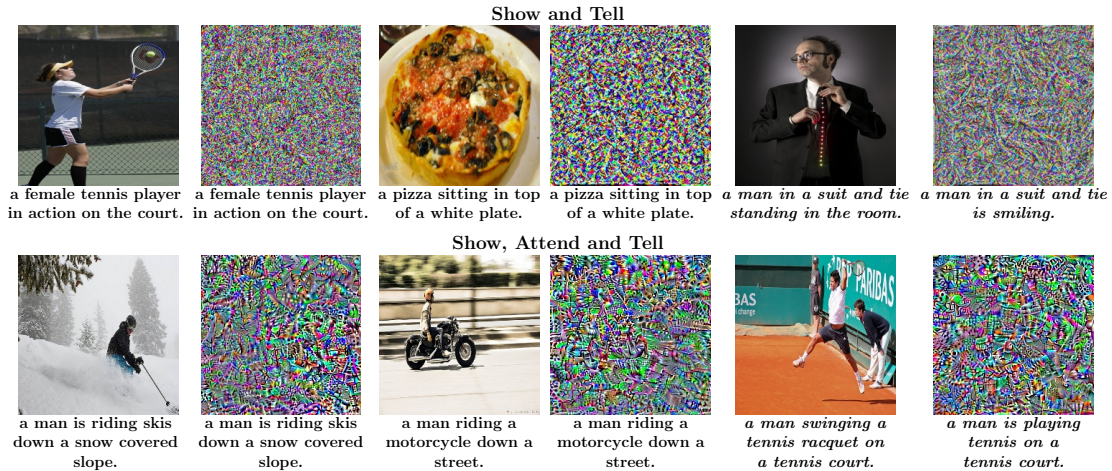
**P<sub>zero</sub>:** yes

**Figure 3.2:** Example of *Mimic and Fool* for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer respectively.  $P_{zero}$  denotes the predicted answer for zero image.

Model	Attack	B-1	B-2	B-3	B-4	M
Show and Tell	Show-and-Fool [17]	0.560	0.394	0.266	0.205	0.301
	Mimic and Fool	0.597	0.464	0.348	0.264	0.320
	OIMO	0.593	0.459	0.350	0.270	0.322
Show, Attend and Tell	EM [145]	0.765	0.650	0.529	0.423	0.425
	SSVM [145]	0.635	0.501	0.409	0.300	0.337
	Mimic and Fool	0.639	0.530	0.421	0.333	0.368
	OIMO	0.594	0.468	0.359	0.284	0.336

**Table 3.2:** BLEU and METEOR scores for **unsuccessful** cases. OIMO refers to *One Image Many Outputs*. B-1, B-2, B-3, B-4, and M represents BLEU-1, BLEU-2, BLEU-3, BLEU-4 and METEOR respectively. ST, and SAT represents Show and Tell, and Show Attend and Tell respectively.

As we can see from Table 3.1, *Mimic and Fool* is very fast. The attack only takes around 25 seconds for generating adversarial images for Show and Tell. The time taken for Show, Attend and Tell is even less since VGG16 is a shallower network. The proposed attack achieves success rate of 74.0% and 81.0% for Show and Tell and Show Attend and Tell respectively. This is especially encouraging result since generating exactly the same caption for an adversarial image is a very challenging task. This is because, as observed by Chen et al. [17], the number of possible captions are infinite which makes a captioning system harder to attack than an image classifier. Our results show that in order to generate the same caption, it suffices to attack just the encoder of the captioning model. This validates our initial hypothesis that in order to attack any model, attacking its feature extractor suffices. For the unsuccessful cases, the predicted captions for original and adversarial images are very similar. Figure 3.3 shows two successful and one unsuccessful examples of *Mimic and Fool* for Show and Tell and Show Attend and Tell. As we can see from Figure 3.3 that for the unsuccessful cases, the predicted captions for the original and adversarial images have a large amount of overlap. We also calculate the BLEU and METEOR score, using the pipeline provided by Sharma et al. [116], for unsuccessful adversarial cases as shown in Table 3.2. We use the predicted caption for the original image as reference while calculating these metrics.



**Figure 3.3:** Examples of *Mimic* and *Fool*. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics.

### 3.4.2 Results for One Image Many Outputs

The main idea behind *One Image Many Outputs* is to generate natural-looking adversarial images. We randomly choose an image from MSCOCO training set as the starting image. Figure 3.4 shows the starting image ( $I_{start}$ ) for *One Image Many Outputs* along with the predicted captions of Show And Tell and Show Attend and Tell. We use the same  $I_{start}$  for N2NMN. Similar to *Mimic* and *Fool*, we discard 1713 image-question pairs for which N2NMN predicts the same answer for  $I_{org}$  and  $I_{start}$ .



**Show and Tell:** a plastic container filled with lots of food.

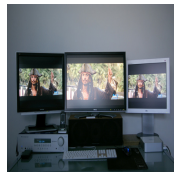
**Show, Attend and Tell:** a tray filled with different types of food.

**Figure 3.4:**  $I_{start}$  for *One Image Many Outputs* and the predicted captions.

In *One Image Many Outputs*, we reduce the value of  $max_{iter}$  and the initial learning rate to ensure that the adversarial image  $I_{adv}$  looks very similar to  $I_{start}$ . Reduction in  $max_{iter}$  results in even faster running time than *Mimic* and *Fool*. Table 3.3 shows the success rate of *One Image Many Outputs* for Show and Tell, Show Attend and Tell and N2NMN. As we can see from Table 3.1 and Table 3.3, the

Model	Success Rate	Time (in sec.)
Show and Tell	56.9 %	7.61
Show, Attend and Tell	50.3 %	7.78
N2NMN	72.8 %	36.50

**Table 3.3:** Success rate of *One Image Many Outputs*



Original

**Q:** Is there a thriller playing on the screen?

**P:** no

$P_{I_{start}}$ : yes

**Q:** Is this person sick?

**P:** no

$P_{I_{start}}$ : yes



Adversarial

**Q:** Is any one of these a TV?

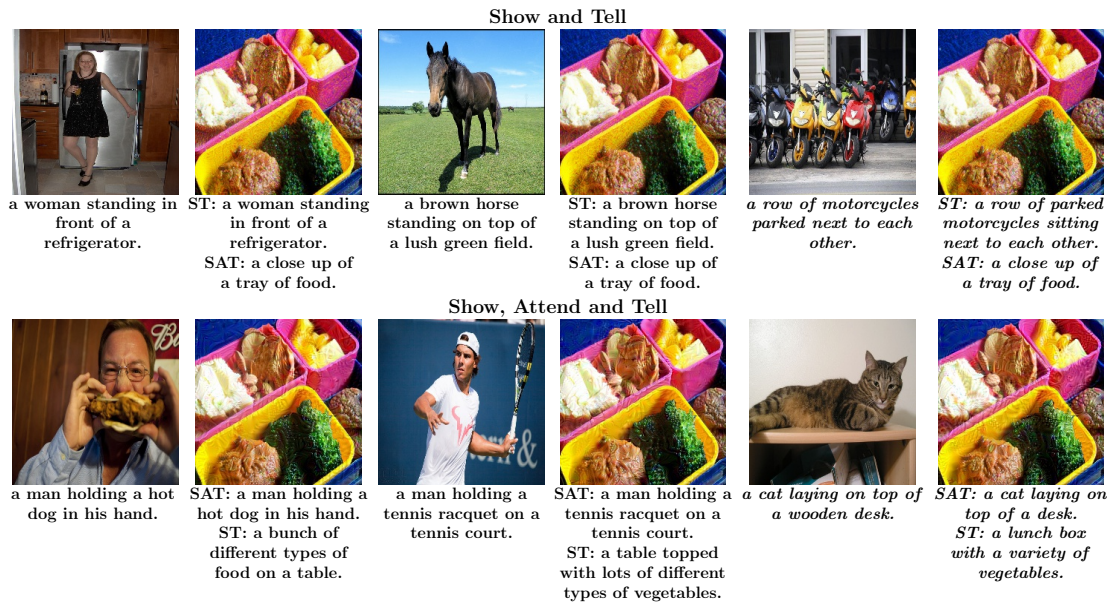
**P:** yes

$P_{I_{start}}$ : no

**Figure 3.5:** Example of *One Image Many Outputs* for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer respectively.  $P_{I_{start}}$  denotes the predicted answer for  $I_{start}$ .

success rate reduces for *One Image Many Outputs* in comparison to *Mimic and Fool*. This is intuitive since in *One Image Many Outputs*, the reduced value of  $max_{iter}$  and initial learning rate allows for less adversarial noise. Figure 3.5 shows an example of OIMO for N2NMN. Similar to *Mimic and Fool*, a single adversarial image suffices for multiple image-question pairs.

From Table 3.3, we can see that *One Image Many Outputs* takes under 8 seconds per image for both the captioning models. Considering this reduction and the fact that the attack is successful only when there is an exact match of captions, the success rate of *One Image Many Outputs* is impressive. Similar to *Mimic and Fool*, we find that for the unsuccessful cases of *One Image Many Outputs*, the captions predicted by the model for the adversarial and original images are very similar to each other. This shows that even when  $I_{adv}$  is very similar to  $I_{start}$ , it can still *mimic* features of an arbitrary image. This is further emphasized by the results in Table 3.2 which shows the BLEU and METEOR score for the unsuccessful cases

of *One Image Many Outputs*.

**Figure 3.6:** Examples of *One Image Many Outputs*. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics. For adversarial images, ST and SAT denote Show and Tell and Show Attend and Tell respectively.

Figure 3.6 shows two successful and one unsuccessful examples (shown in italics) of *One Image Many Outputs* for Show and Tell and Show Attend and Tell. For the adversarial images in Figure 3.6, ST and SAT denote Show and Tell and Show Attend and Tell respectively. As we can see from Figure 3.6, all the six adversarial images are very similar to the starting image,  $I_{start}$ . Also for the unsuccessful cases, the original and adversarial captions have a large amount of overlap and are semantically similar. In Figure 3.6, we see that for Show and Tell, the captions predicted by Show, Attend and Tell for the three adversarial images are the same. Similarly for Show, Attend and Tell, although the captions predicted by Show and Tell are different, they are semantically similar. Moreover, for both the captioning models, the predicted captions by the other captioning model are relevant captions for the starting image,  $I_{start}$ . In fact, we find that when the 1000 adversarial images for Show And Tell are given as input to Show, Attend and Tell, there are only 15 unique captions. All these 15 captions are relevant captions for  $I_{start}$ . Similarly, when the 1000 adversarial images for Show, Attend and Tell are given as input to Show and Tell, there are only 82 unique captions, most of which are relevant to  $I_{start}$ . We find that Show and Tell generates irrelevant captions for



Task	Model	Method	Success Rate	Time (in sec)
Image Captioning	Show and Tell	Show-and-Fool [17]	95.1 %	177.93
	Show, Attend and Tell	EM [145]	77.1 %	20.69
		SSVM [145]	82.1 %	18.73
VQA	N2NMN	Xu et al. [144]	100.0 %	$8.77 \times n_q$

**Table 3.4:** Success rate and Time for task-specific methods.  $n_q$  signifies the average number of questions per image.

$I_{start}$  only for 32 out of 1000 adversarial images. Since the two captioning models use different feature extractors, this result shows that the proposed attack is very dependent on the feature extractor. In other words, ensuring that the two images are *indistinguishable* for one feature extractor does not ensure that they will be *indistinguishable* for another feature extractor.

### 3.4.3 Comparison with task specific attack

In this section, we compare our proposed attack, OIMO with other task-specific attacks. For Show and Tell, we use Show-and-Fool [17]. For Show, Attend and Tell, we use EM and SSVM methods of Xu et al. [145]. For N2NMN, we use the VQA attack of Xu et al. [144]. For Show-and-Fool and EM and SSVM methods, we use the official implementation.<sup>56</sup> We implement the attack proposed by Xu et al. [144] using the default parameters mentioned in the paper. Similar to OIMO, we start with  $I_{start}$  and run the task specific attacks in order to generate adversarial outputs. Table 3.4 shows the success rate and time for different task-specific methods. Show-and-Fool achieves a success rate of 95.1% and takes 177.93 seconds per image. The EM and SSVM take less time for Show, Attend and Tell but have lower success rates. In contrast, OIMO takes around 8 seconds per image for both the captioning models. For unsuccessful cases, like OIMO, Show-and-Fool and EM and SSVM generate similar captions for original and adversarial images as evident from high BLEU and METEOR scores in Table 3.2. We find that for the adversarial images generated by Show-and-Fool, Show Attend and Tell generates only 11 unique captions, all of which are relevant captions for  $I_{start}$ . Chen et al. [17] study the transferability of Show-and-Fool between the captioning models, however

<sup>5</sup><https://github.com/IBM/Image-Captioning-Attack>

<sup>6</sup><https://github.com/wubaoyuan/adversarial-attack-to-caption>

in their study, the two captioning models use the same feature extractor. Similarly, we obtain only 3 and 5 unique captions from Show and Tell for adversarial images of EM and SSVM respectively. All these captions are relevant for  $I_{start}$ . Xu et al. [144] achieve 100.0% success rate. The attack takes 8.77 seconds for each image-question pair. The factor  $n_q$  in the time for Xu et al. [144] in Table 3.4 signifies the average number of questions per image, which can be arbitrarily large.

### 3.4.4 OIMO for invertible architecture

Recently, Jacobsen et al. [54] propose a deep invertible architecture, i-RevNet which learns a one-to-one mapping between image and its feature. These networks achieve impressive accuracy on ILSVRC-2012 [29]. For experimentation, we choose bijective i-RevNet which takes images of size  $224 \times 224 \times 3$  as input and the corresponding feature is of size  $3072 \times 7 \times 7$ . We use the pretrained i-RevNet provided in the official implementation<sup>7</sup> to test our proposed attack, *One Image Many Outputs*. We randomly choose 100 correctly classified images belonging to 41 different classes from the validation set of ILSVRC-2012. Furthermore, we choose a starting image,  $I_{start}$ , belonging to a different class. We also restrict the search space for adversarial images using the clipping function  $Clip_{I_{start}, \epsilon}$  (i.e. the adversarial noise is clipped to ensure that the adversarial image  $I_{adv}$  will lie in an  $\epsilon$   $\ell_\infty$ -neighborhood of  $I_{start}$ ). Starting with  $I_{start} \in [0, 255]^{224 \times 224 \times 3}$ , we run the proposed attack, OIMO, in order to *mimic* the feature for 100 images. Table 3.5 shows the success rate for different values of  $\epsilon$ . The high success rate shows that the proposed attack can be applied for invertible architecture like i-RevNet as well. This is because i-RevNet, despite being invertible, assigns similar features to dissimilar images. Figure 3.7 shows one such successful adversarial example.

$\epsilon$	Success Rate
2	86.0 %
5	99.0 %
10	100.0 %

**Table 3.5:** Success rate of *One Image Many Outputs* for i-RevNet

<sup>7</sup><https://github.com/jhjacobson/pytorch-i-revnet>



**Figure 3.7:** Both the images are classified as *ice bear* by bijective i-RevNet.

### 3.4.5 Quantitative study of Adversarial Noise

Model	Attack	PSNR (mean $\pm$ std)
Show and Tell	Show-and-Fool [17]	$52.5 \pm 6.7$
	OIMO	$23.8 \pm 0.6$
Show, Attend and Tell	SSVM [145]	$42.1 \pm 1.2$
	EM [145]	$40.4 \pm 0.9$
	OIMO	$26.1 \pm 1.1$
N2NMN	Xu et al. [144]	$33.8 \pm 3.7$
	OIMO	$27.6 \pm 0.5$

**Table 3.6:** PSNR between  $I_{adv}$  and  $I_{start}$  for *One Image Many Outputs* (OIMO) and task-specific methods.

Table 3.6 shows the peak signal-to-noise ratio (PSNR) for OIMO and task-specific methods. The PSNR is calculated as follows

$$PSNR = 20 \log_{10} \left( \frac{255.0}{\sqrt{MSE}} \right) \quad (3.8)$$

where  $MSE = \frac{\|I_{adv} - I_{start}\|_2^2}{m \times n \times 3}$

where  $I_{adv}, I_{start} \in [0, 255]^{m \times n \times 3}$ . From Table 3.6, it is evident that the PSNR is low for OIMO in comparison with other task-specific methods. This is mainly because task-specific methods can exploit the deficiencies of encoder as well as the decoder and such attack methods can be stopped at the exact instant when an adversarial image leads to the desired output. Agnosticity, in any form, generally leads to more noise. As an example, *image-agnostic* universal adversarial perturbations (UAP) [88] are quasi-perceptible instead of being imperceptible. Table 3.7 shows




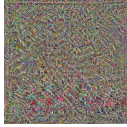
Model	Attack	SSIM (mean $\pm$ std)
Show and Tell	MAF	$1.8 \times 10^{-4} \pm 1.3 \times 10^{-3}$
	OIMO	$6.1 \times 10^{-4} \pm 2.9 \times 10^{-3}$
Show, Attend and Tell	MAF	$7.5 \times 10^{-4} \pm 2.7 \times 10^{-3}$
	OIMO	$6.8 \times 10^{-4} \pm 4.2 \times 10^{-3}$
N2NMN	MAF	$5.6 \times 10^{-4} \pm 1.5 \times 10^{-3}$
	OIMO	$4.5 \times 10^{-4} \pm 2.2 \times 10^{-3}$

**Table 3.7:** SSIM between  $I_{adv}$  and  $I_{org}$  for *Mimic and Fool* (MAF) and *One Image Many Outputs* (OIMO).

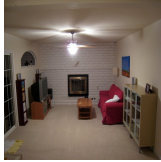

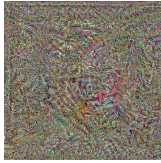
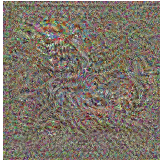


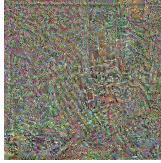
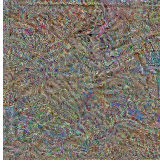


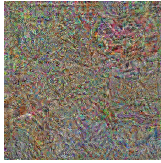
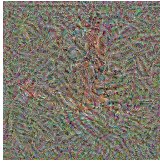
the SSIM [155] values between  $I_{adv}$  and  $I_{org}$  for the proposed methods. The *near-zero* values of SSIM clearly show that there is no resemblance between the original and adversarial image.

## 3.5 Examples of the Attack



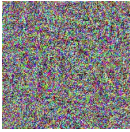
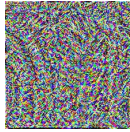


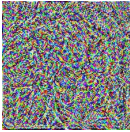
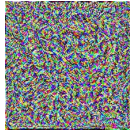

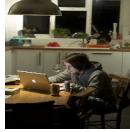
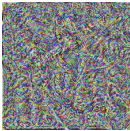
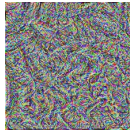
### 3.5.1 Examples of Mimic and Fool

Image	What is the brand of drink?	What does the phone say?	Is this a smartphone?
	pepsi	nokia	no
	pepsi	nokia	no
Image	What is the big red thing?	Are there clouds visible?	What city is this?
	fire truck	yes	london
	fire truck	yes	london



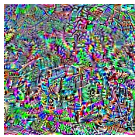
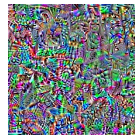

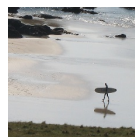

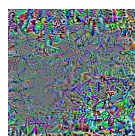


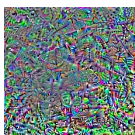

**Table 3.8:** Examples of *Mimic and Fool* for N2NMN. Single adversarial image suffices for three image-question pairs.

Image	What room is this?	Image	What is the man doing?
	living room		sitting
	living room		sitting
	bathroom		skateboarding
	bathroom		skateboarding
	bedroom		surfing
	bedroom		surfing

**Table 3.9:** Examples of Mimic and Fool for N2NMN. N2NMN predicts varied answers for the same question.

Image	Caption	Image	Caption
	a group of people on surfboards in the water.		a man riding a wave on top of a surfboard.
	<ol style="list-style-type: none"> <li>1. a group of people on surfboards in the water.</li> <li>2. a group of people on surfboards in the ocean.</li> <li>3. a group of people riding on top of surfboards.</li> </ol>		<ol style="list-style-type: none"> <li>1. a man riding a wave on top of a surfboard.</li> <li>2. a man riding a surfboard on top of a wave.</li> <li>3. a man on a surfboard riding a wave.</li> </ol>
Image	Caption	Image	Caption
	a man riding a skateboard up the side of a ramp.		a laptop computer sitting on top of a desk.
	<ol style="list-style-type: none"> <li>1. a man riding a skateboard up the side of a ramp.</li> <li>2. a man riding a skateboard down the side of a ramp.</li> <li>3. a man riding a skateboard up the side of a cement ramp.</li> </ol>		<ol style="list-style-type: none"> <li>1. a laptop computer sitting on top of a desk.</li> <li>2. a laptop computer sitting on top of a wooden desk.</li> <li>3. a desk with a laptop and a lamp.</li> </ol>
Image	Caption	Image	Caption
	a bus that is sitting in the street.		a woman sitting at a table with a laptop.
	<ol style="list-style-type: none"> <li>1. a bus that is driving down the street.</li> <li>2. a bus that is sitting on the side of the road.</li> <li>3. a bus that is parked on the side of the road.</li> </ol>		<ol style="list-style-type: none"> <li>1. a woman sitting in front of a laptop computer.</li> <li>2. a woman sitting at a table with a laptop.</li> <li>3. a woman sitting at a table with a laptop computer.</li> </ol>

**Table 3.10:** Examples for Show and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases.

Image	Caption	Image	Caption
	a red double decker bus driving down a street.		a group of people sitting in a room.
	<ol style="list-style-type: none"> <li>1. a red double decker bus driving down a street.</li> <li>2. a red double decker bus driving down the street.</li> <li>3. a red double decker bus traveling down a street.</li> </ol>		<ol style="list-style-type: none"> <li>1. a group of people sitting in a room.</li> <li>2. a group of people sitting on a couch.</li> <li>3. a group of people sitting on a bed.</li> </ol>
Image	Caption	Image	Caption
	a group of cows standing in a field.		a person on a beach with a surfboard.
	<ol style="list-style-type: none"> <li>1. a group of cows standing in a field.</li> <li>2. a group of cows standing next to each other.</li> <li>3. a group of cows standing next to each other in a field.</li> </ol>		<ol style="list-style-type: none"> <li>1. a person on a beach with a surfboard.</li> <li>2. a person on a beach holding a surfboard.</li> <li>3. a person walking on a beach with a surfboard.</li> </ol>
Image	Caption	Image	Caption
	a train traveling down the tracks near a forest.		a boat floating in a body of water.
	<ol style="list-style-type: none"> <li>1. a train traveling down the tracks near to a forest.</li> <li>2. a train traveling down the tracks near a forest.</li> <li>3. a train traveling down a train track next to a forest.</li> </ol>		<ol style="list-style-type: none"> <li>1. a boat floating in the water next to a lake.</li> <li>2. a boat floating in the water near a lake.</li> <li>3. a boat floating in the water next to a body of water.</li> </ol>



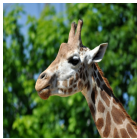



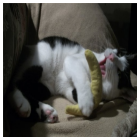

**Table 3.11:** Examples for Show Attend and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases.



### 3.5.2 Examples of One Image Many Outputs

Image	What number is on the bus?	How many busses are there?	What color is bus bumper?
	22	1	yellow
	22	1	yellow
Image	Is this a hotel room?	Which room is this?	What is on the nightstand?
	yes	bedroom	lamp
	yes	bedroom	lamp
Image	What are the people holding?	What are the people looking at?	How many headbands are pictured?
	tennis rackets	camera	2
	tennis rackets	camera	2










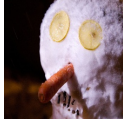


**Table 3.12:** Examples of *One Image Many Outputs* for N2NMN. Single adversarial image suffices for three image-question pairs.

Image	What kind of animal is this?	Image	What sport is being played?
	elephant		soccer
	elephant		soccer
	giraffe		skateboarding
	giraffe		skateboarding
	cat		tennis
	cat		tennis

**Table 3.13:** Examples of *One Image Many Outputs* for N2NMN. N2NMN predicts varied answers for the same question.

Image	Caption	Image	Caption
	a group of people standing next to each other.		a man riding a wave on top of a surfboard.
	<ol style="list-style-type: none"> <li>1. a group of people standing next to each other.</li> <li>2. a group of people standing next to each other eating food.</li> <li>3. a group of people standing next to each other eating pizza.</li> </ol>		<ol style="list-style-type: none"> <li>1. a man riding a wave on top of a surfboard.</li> <li>2. a person riding a surfboard on a wave.</li> <li>3. a man riding a surfboard on top of a river.</li> </ol>
Image	Caption	Image	Caption
	a herd of zebra standing on top of a lush green field.		a pile of luggage sitting next to each other.
	<ol style="list-style-type: none"> <li>1. a herd of zebra standing on top of a lush green field.</li> <li>2. a herd of zebra standing on top of a grass covered field.</li> <li>3. a herd of zebra standing next to each other on a field.</li> </ol>		<ol style="list-style-type: none"> <li>1. a pile of luggage sitting next to each other.</li> <li>2. a pile of luggage sitting on top of a wooden floor.</li> <li>3. a pile of luggage sitting on top of a floor.</li> </ol>
Image	Caption	Image	Caption
	a car driving down a road with a herd of cattle.		a man in a suit and tie standing on a street.
	<ol style="list-style-type: none"> <li>1. a car driving down a road next to a herd of cattle.</li> <li>2. a car driving down a road next to a herd of sheep.</li> <li>3. a car driving down a road next to a herd of animals.</li> </ol>		<ol style="list-style-type: none"> <li>1. a man in a suit and tie standing in front of a building.</li> <li>2. a man in a suit and tie standing in a street.</li> <li>3. man in a suit and tie with a hat on.</li> </ol>

**Table 3.14:** Examples for Show and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases.

Image	Caption	Image	Caption
	a couple of street signs on a pole.		a close up of a sandwich on a plate.
	1. a couple of street signs on a pole. 2. a couple of street signs on a street. 3. a couple of street signs hanging from a pole.		1. a close up of a sandwich on a plate. 2. a close up of a plate of food. 3. a close up of a plate of food on a table.
Image	Caption	Image	Caption
	a group of people standing on a beach.		a group of zebras standing next to each other.
	1. a group of people standing on a beach. 2. a group of people standing on top of a sandy beach. 3. a group of people standing in the sand.		1. a group of zebras standing next to each other. 2. three zebras standing next to each other in a zoo. 3. three zebras standing next to each other in a zoo enclosure.
Image	Caption	Image	Caption
	a person on a surfboard riding a wave.		a close up of a person holding a doughnut.
	1. a person riding a surfboard on top of a wave. 2. a person on a surfboard riding a wave. 3. a person riding a surfboard on a wave.		1. a close up of a person holding a hot dog. 2. a close up of a person holding a piece of food. 3. a close up of a person holding a piece of cake.

**Table 3.15:** Examples for Show Attend and Tell. The first two rows contain successful cases and the last row contains unsuccessful cases.

## 3.6 Summary

In this chapter, we proposed a task agnostic adversarial attack, *Mimic and Fool*. The proposed attack exploits the non-invertibility of CNN-based feature extractors and is based on the hypothesis that if two images are *indistinguishable* for the *feature extractor* then they will be *indistinguishable* for the model as well. The high success rate of *Mimic and Fool* for three models across two tasks validates this hypothesis. We also showed that the proposed attack works regardless of the depth of the feature extractor. Due to the task-agnostic nature, we need to run

---

the attack only at image-level which is a huge advantage in terms of time saved for tasks involving multiple modalities as input. We further proposed a variant of *Mimic and Fool*, named *One Image Many Outputs*, which generates *natural-looking* adversarial images. The results for this variant of the attack show that it is possible to *mimic* features of an arbitrary image by making minimal changes to a fixed image. This is an important insight into the nature of CNN-based feature extractors. We also demonstrated the applicability of the proposed attack for invertible architectures like i-RevNet.

## Chapter 4

# Exploring the Robustness of NMT systems to Non-sensical Inputs

*To do is to be.*

Plato

*To be is to do.*

Socrates

*Do be do be do.*

Frank Sinatra

Neural machine translation (NMT) systems have made remarkable gains leading to the state-of-the-art performance in the past few years [81, 128]. In this chapter, we explore the robustness of such systems by asking the following question “Is it possible for an NMT system to predict same translation even when multiple words in the source sentence have been replaced completely changing the meaning of the source?”. To this end, we propose an adversarial attack which uses a soft-attention based technique to make the aforementioned word replacements. Such an attack allows us to explore the ability of NMT systems to capture semantics of the source sentence. Similar to Mimic and Fool, the proposed attack is an invariance-based attack. However, unlike Mimic and Fool, the proposed attack is a white-box attack. In this chapter, we also propose an alternate BLEU-based metric and argue its benefits in comparison to standard metrics like success rate for

evaluating invariance-based attack against NMT systems. We study the robustness of NMT systems both in low-resource as well as high-resource setting. The results demonstrate that the NMT systems in high-resource setting are more robust to the proposed attack than in low-resource setting, despite achieving similar BLEU scores.

The rest of this chapter is organized as follows. Section 4.1 briefly discusses the two NMT systems considered in this chapter. Section 4.2 describes the motivation behind the proposed attack. Section 4.3 describes the proposed attack in detail. Section 4.4 provides the implementation details. Section 4.5 discusses several metrics for evaluating the efficiency of the proposed attack. Section 4.6 presents the results of the proposed attack in terms of these metrics. In this section, we present the result of the proposed attack both in low-resource and high-resource setting. Finally, Section 4.7 summarizes the chapter.

## 4.1 Background

In this section, we discuss the architecture of the two state-of-the-art NMT systems considered in this chapter, namely, BLSTM-based encoder decoder with attention [81] and Transformer [128].

### 4.1.1 BLSTM-based encoder decoder with attention

BLSTM-based encoder decoder with attention comprises of a bidirectional encoder LSTM and a unidirectional decoder LSTM. Let  $q_t$  denote the hidden state of the decoder LSTM at time  $t$  and  $h_{t'}$  denote the hidden state of bidirectional encoder LSTM at time  $t'$ . At each decoder time-step  $t$ , an attention mechanism constructs a context vector, denoted by  $c_t$ , to attend on particular words in the source sentence [81]. The context vector is constructed as follows

$$s_{t'}^t = q_t^T W^{attn} h_{t'} \quad (4.1)$$

$$\alpha_{t'}^t = \frac{\exp(s_{t'}^t)}{\sum_{t'} \exp(s_{t'}^t)} \quad (4.2)$$

$$c_t = \sum_{t'} \alpha_{t'}^t h_{t'} \quad (4.3)$$

where  $W^{attn}$  is a trainable parameter. Both  $c_t$  and  $q_t$  are then used to predict the target word at time-step  $t$ .

### 4.1.2 Transformer

A major drawback with LSTM-based architectures is that they are difficult to parallelize since such architectures require the output at previous time steps in order to compute the output at current time step (i.e., recurrence operation). This difficulty essentially leads to longer training time. To remedy this issue, Transformer [128] completely gets rid of recurrence and relies solely on attention. While recurrence operation in LSTM intrinsically captures position of a word in a sequence, Transformer makes use of sinusoidal-based positional embeddings for the same. Apart from positional embeddings, Transformer also makes use of multi-head attention. Multi-head attention comprises of several self-attention layers. The goal of self-attention layer is to incorporate context into individual word embeddings (i.e., context-aware embedding). To do so, self-attention layer associates three vectors (i.e., namely query, key and value) for each word in the sequence. Let  $Q, K, V$  denote the query, key and value matrices respectively. These matrices are of size  $l_{seq} \times d$  where  $l_{seq}$  denotes the sequence length and  $d$  is the dimension of the embeddings. Given the three matrices, self attention is given by

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (4.4)$$

For decoder, the multi-head attention is masked in order to ensure that Transformer only relies on previous target words in order to predict the next word. Furthermore, decoder also incorporates information from the encoder via another multi-head attention where the value and key comes from the encoder and the query comes from the decoder.

## 4.2 Motivation

Given a source sentence  $s = (s_1, s_2, \dots, s_n)$ , our goal is to replace multiple words  $s_i$ 's with new words  $s_i'$ 's while ensuring that the predicted translation remains unchanged. To achieve this, we propose a white-box soft-attention based adversarial



<b>src</b>	Not a single body should remain undiscovered or unidentified .
<b>adv-src</b>	unaware topic single body should remain unsubmitted covered Within ununclear fied surely
<b>pred</b>	Kein einziger Körper sollte unbehandelt oder geklärt bleiben .

**Table 4.1:** Example of the proposed attack. The English-German Transformer predicts the same translation for the two sentences even though multiple replacements are made.

attack. Table 4.1 shows an example of the proposed attack. This example shows an instance where the NMT system is invariant to multiple word replacements. Such word-level *invariances* captured by the model are *undesirable*. Ideally, we want the NMT system to be sensitive (i.e. not invariant) to multiple word replacements, especially when it leads to the change in semantics of the source sentence. Such examples showcase the inability of NMT system to capture semantics of the source sentence. From Table 4.1, it is also clear that the adversarial source sentence (i.e., adv-src) is non-sensical. In this regard, a question may arise “*Are there any practical implications if the NMT system behaves in an undesirable fashion to such non-sensical inputs?*”. We argue that there are two major practical implications. Firstly, it may lead to lack of trust of the end user on the NMT system if two semantically different sentences are assigned the same translation. This is in line with the work done by He and Glass [47] where a dialogue generator is expected to never output egregious sentences regardless of the *semantic correctness* of the input sentence. Secondly, such a behavior also poses real-world threats. Consider a scenario where an adversary, who is targeting an audience of the target language, publishes an article of non-sensical sentences in the source language. This article when accessed by the target language audience gets auto-translated to hate speech (or fake news). Such articles will be very difficult to prune out via automatic hate speech (or fake news) detector since they are non-sensical in the source language and are only translated to hate speech (or fake news) via the specific NMT system.

### 4.3 Methodology

In this section, we describe the proposed method in detail. In Section 4.3.1, we outline the vocabulary pruning method which is a pre-processing step of the

proposed method. Section 4.3.2 describes the proposed technique for position indices traversal. In Section 4.3.3, we describe the proposed technique for word replacement. Finally in Section 4.3.4, we combine the two techniques for doing multiple replacements over the source sentence.

### 4.3.1 Vocabulary Pruning

The NMT systems in the present work are subword-level and use a shared vocabulary for source and target languages. Let  $V_{shared}$  denote the shared vocabulary set. We use the training corpus in the source language pre byte-pair encoding to find the set of unique words in the source language. Let  $V_{unique}$  denote this set. We consider the set intersection  $V = V_{shared} \cap V_{unique}$ . Hence  $V$  denotes the set of *proper words* in the source language present in the vocabulary of NMT system. By proper words, we refer to those source words which are not further broken down into subwords after byte-pair encoding [113]. Let  $s^{org} = (s_1^{org}, s_2^{org}, \dots, s_n^{org})$  denote the original sentence in the source language. Given  $s^{org}$ , we remove the words present in the original sentence from the set  $V$ , i.e.,  $V_{prune} = V \setminus s^{org}$ . We use  $V_{prune}$  to select new words for replacement.

### 4.3.2 Position Indices Traversal

Let  $s = (s_1, s_2, \dots, s_n)$  denote a sentence in the source language and  $x$  denote the one-hot representation of the sentence  $s$  i.e.  $x = ((x_{11}, \dots, x_{1|V_{shared}|}), \dots, (x_{n1}, \dots, x_{n|V_{shared}|}))$  where  $x_{ij}$  is 1 if  $j^{th}$  word is present in  $i^{th}$  position and 0 otherwise. Let  $e = (e_1, e_2, \dots, e_n)$  denote the embedded version of input  $x$  where  $e_i$ 's are  $d$ -dimensional and  $t^{org} = (t_1^{org}, t_2^{org}, \dots, t_m^{org})$  denote the predicted translation of the NMT system for the original source sentence  $s^{org}$ . We consider the standard negative log likelihood loss  $L_{nll}$  given by

$$L_{nll} = - \sum_{i=1}^m \log(q(t_i^{org} | t_{<i}^{org}, x)) \quad (4.5)$$

where  $q(t_i^{org} | t_{<i}^{org}, x)$  denotes the probability assigned to the word  $t_i^{org}$  by the NMT system and  $x$  is one-hot representation of the source sentence  $s$ . Let  $ind_{vis} \subseteq$

$\{1, 2, \dots, n\}$  denote the set of position indices which have already been traversed. We choose the position for replacement,  $r$ , using the following equation

$$r = \operatorname{argmin}_{i \notin \text{ind}_{vis}} \|\nabla_{e_i} L_{nll}\|_2 \quad (4.6)$$

where  $\|\cdot\|_2$  is the  $\ell_2$ -norm,  $e_i$  is the  $i^{\text{th}}$  embedding and  $\nabla_{e_i} L_{nll}$  is the gradient of the loss function with respect to  $e_i$ . The rationale behind choosing the replacement position in this way is that the term  $\|\nabla_{e_i} L_{nll}\|_2$  tells us about the sensitivity of loss function with respect to the  $i^{\text{th}}$  embedding  $e_i$  and hence changing a word at a position which has the minimum  $\ell_2$ -norm should not have a large impact on the predicted translation. We refer to this technique as **Min-Grad**. We summarize the method in Algorithm 4.1.

---

**Algorithm 4.1:** Min-Grad

---

**Input:**  $s, t^{org}, \text{ind}_{vis}$

**Output:**  $r$

Get  $e, x$  from  $s$

Compute loss  $L_{nll}$  for  $(x, t^{org})$

$r = \operatorname{argmin}_{i \notin \text{ind}_{vis}} \|\nabla_{e_i} L_{nll}\|_2$

**return**  $r$

---

### 4.3.3 Word Replacement

Let  $r$  denote the position for word replacement. We replace  $(x_{r1}, x_{r2}, \dots, x_{r|V_{shared}|})$  with a probability distribution  $p$  i.e.  $p = (p_1, p_2, \dots, p_{|V_{shared}|})$  where  $p_i$  is set to 0 if the  $i^{\text{th}}$  word does not belong to  $V_{prune}$ . We set all the other  $p_i$ 's to be equal initially. Let  $x'$  denote the modified input. We modify the non-zero  $p_i$ 's using gradient descent in order to minimize  $L_{nll}$ . Note that only the non-zero  $p_i$ 's are modified. To modify the non-zero  $p_i$ 's, we update the underlying logits using gradient descent via Adam optimizer [65]. Hence, the modified  $p_i$ 's is obtained by applying the softmax function to the updated logits. In this way, we modify  $p_i$ 's until either  $\text{max}_{iter}$  iterations is reached or a particular word is assigned a probability greater than  $\text{max}_{prob}$  for  $n_{iter}$  consecutive iterations. The criteria of  $n_{iter}$  consecutive iterations is essential to ensure that the algorithm has found a stable solution. Finally, for the position  $r$ , we choose the  $j^{\text{th}}$  word where  $j = \operatorname{argmax}(p)$ . Since this technique picks a word using soft-attention over the vocabulary set  $V_{prune}$ , we refer to it as **Soft-Att**. We summarize the method in Algorithm 4.2.

**Algorithm 4.2:** Soft-Att

---

**Input:**  $s, t^{org}, r$   
**Output:**  $ind_{word}, loss$   
Initialize  $p, x'$  using  $s, r$   
 $count = \{\}$   
Initialize  $count$  to 0 for all word indices  
**for**  $j \leftarrow 1$  **to**  $max_{iter}$  **do**  
     $loss \leftarrow L_{nll}$  for  $(x', t^{org})$   
    Update  $p_i$ 's using gradient descent  
    Get  $x'$  from  $p$   
     $p_{max}, ind_{word} \leftarrow \max(p), \operatorname{argmax}(p)$   
    **for**  $ind$  in word indices **do**  
        **if**  $ind \neq ind_{word}$  **then**  
             $count[ind] = 0$   
    **end for**  
    **if**  $p_{max} > max_{prob}$  **then**  
         $count[ind_{word}] += 1$   
        **if**  $count[ind_{word}] == n_{iter}$  **then**  
            **break**  
    **else**  
         $count[ind_{word}] = 0$   
    **end for**  
**return**  $ind_{word}, loss$

---

### 4.3.4 Proposed method

In order to make multiple replacements over the original source sentence,  $s^{org}$ , we use the two methods (Min-Grad and Soft-Att) iteratively. We name the proposed method **Min-Grad + Soft-Att**.

The proposed method makes at most  $max_{sweep}$  sweeps over the source sentence. Within a particular sweep, we choose the position of replacement using Min-Grad method. This is followed by Soft-Att method to identify the new word to replace with, at the particular position. Note that Soft-Att always picks a word from the pruned vocabulary set,  $V_{prune}$ . Whether the replacement does take place depends on the *min loss criteria*. We initially set the min loss,  $l_{min}$ , to a very high value (i.e. 100). This ensures that at least one replacement always takes place. If in a previous sweep, a replacement has taken place at the position identified by the Min-Grad, then we compare the loss obtained from the Soft-Att method with the loss of the current sentence. If the loss obtained from the Soft-Att method is less than the loss of the current sentence, then the replacement is done and  $l_{min}$  is

**Algorithm 4.3:** Min-Grad + Soft-Att

---

```

Input:  $s^{org}, t^{org}$ 
Output:  $s^{adv}$ 
Get  $x$  from  $s^{org}$ 
 $l_{org} \leftarrow L_{nll}$  for  $(x, t^{org})$ 
 $n \leftarrow \text{len}(s^{org})$ 
 $s \leftarrow s^{org}$ 
 $l_{min} \leftarrow 100$ 
 $ind_{rep} \leftarrow []$ 
for  $j \leftarrow 1$  to  $max_{sweep}$  do
     $flag \leftarrow \text{False}$ 
     $ind_{vis} \leftarrow []$ 
    while  $\text{len}(ind_{vis}) \neq n$  do
        Get  $x$  from  $s$ 
         $l \leftarrow L_{nll}$  for  $(x, t^{org})$ 
         $r \leftarrow \text{Min-Grad}(s, t^{org}, ind_{vis})$ 
        append  $r$  to  $ind_{vis}$ 
         $ind_{word}, loss \leftarrow \text{Soft-Att}(s, t^{org}, r)$ 
        if  $r \in ind_{rep}$  and  $loss < l$  then
             $l_{min} \leftarrow \max(loss, l_{org})$ 
             $s[r] \leftarrow V_{shared}[ind_{word}]$ 
             $flag \leftarrow \text{True}$ 
        if  $r \notin ind_{rep}$  and  $loss < l_{min}$ 
            then
                append  $r$  to  $ind_{rep}$ 
                 $l_{min} \leftarrow \max(loss, l_{org})$ 
                 $s[r] \leftarrow V_{shared}[ind_{word}]$ 
                 $flag \leftarrow \text{True}$ 
        end if
    end while
    if not  $flag$  then
        break
    end if
end for
 $s^{adv} \leftarrow s$ 
return  $s^{adv}$ 

```

---

updated accordingly. The logic behind this step is to ensure that the new source sentence is better than the old one in terms of  $L_{nll}$ . Whereas, if no replacement has taken place so far at the position identified by the Min-Grad, then we compare the loss obtained from the Soft-Att method with  $l_{min}$ . If the loss obtained from the Soft-Att method is less than  $l_{min}$ , then the replacement is done and  $l_{min}$  is updated accordingly. We update  $l_{min}$  as  $l_{min} = \max(loss, l_{org})$  where  $loss, l_{org}$  are the loss obtained from the Soft-Att method and the original loss respectively. Capping

the min loss at original loss allows us to do more replacements while ensuring an optimal solution at the same time. We stop the algorithm if no replacement takes place in a particular sweep. For ease of understanding, we summarize the proposed method in Algorithm 4.3.

Apart from the proposed method, we also consider HotFlip-related baselines [32]. Overall, there are three baseline methods, namely, random + Soft-Att, Min-Grad + HotFlip and random + HotFlip. The *random* baselines refer to the method where traversal of position indices is done randomly instead of via Min-Grad and *HotFlip* baselines refer to the method where word replacement is done via HotFlip instead of Soft-Att. Given a position of replacement  $r$ , the HotFlip method computes  $\nabla_{x_r} L_{nll}$  where  $x_r$  is a one-hot encoded vector. Finally, it chooses the  $j^{th}$  word for replacement where  $j$  is given by

$$j = \underset{i|w_i \in V_{prune}}{\operatorname{argmin}} \nabla_{x_{r_i}} L_{nll} \quad (4.7)$$

where  $w_i$  denotes the  $i^{th}$  word. As before, whether this replacement does take place depends on *min loss criteria*.

Note that the other methods like [21, 22, 76] study robustness of NMT systems in a different framework and hence, these methods are not applicable for comparison with the method presented here. HotFlip being a general method for word/character replacement is relevant to our setting and hence, comparable to the proposed method.

## 4.4 Implementation Details

We perform experiments on three language pairs from TED talks dataset [105]. The two language pairs are (i) English-German (en-de), (ii) English-French (en-fr) and (iii) German-English (de-en). The dataset statistics for the language pairs are given in Table 4.2. We train BLSTM-based encoder-decoder with attention translation model using OpenNMT-py for the two language pairs. We use the standard implementation provided in the repository<sup>1</sup> for training. The model uses attention mechanism proposed by Luong et al. [81].

<sup>1</sup><https://github.com/OpenNMT/OpenNMT-py>

Language Pair	Training	Dev	Test
en-de	167,888	4,148	4,491
en-fr	192,304	4,320	4,866

**Table 4.2:** Dataset Statistics

Model	en-de	en-fr	de-en
BLSTM	26.33	39.32	33.97
Transformer	29.27	43.15	37.37

**Table 4.3:** BLEU score on the *test set*

We use the *Transformer base model* configuration [128] for both the language pairs. The model consists of 6 encoder-decoder layers. We closely follow the implementation provided by Sachan and Neubig [111] for training the Transformer models. Both the NMT systems, BLSTM-based encoder-decoder with attention and Transformer, use byte-pair encoding with 32,000 merge operations [113]. Also, both the NMT systems use beam search with beam width of 5 during prediction. Table 4.3 shows the BLEU score [103] for the trained NMT systems on the *test set* of TED dataset. The BLEU scores for Transformer are similar to the results reported by Sachan and Neubig [111]. As expected, Transformer achieves a higher BLEU score than BLSTM-based encoder-decoder with attention for the three language pairs.

To study the proposed attack, we randomly select 500 sentences from the *test set* of TED dataset. The values of the different hyperparameters are as follows:  $n_{sweep} = 5$ ,  $max_{iter} = 1000$ ,  $max_{prob} = 0.9$  and  $n_{iter} = 10$ . To update the  $p_i$ 's in Algorithm 4.2, we use Adam optimizer with learning rate of 1. In our experiments, we found that using a higher learning rate ensures faster convergence (mostly within 300 iterations). The size of the vocabulary set  $V$  (i.e. the set of *proper words* in the source language) for English-German, English-French, and German-English are 9,723, 11,699, and 11,284 respectively. The code for the proposed attack is publicly available.<sup>2</sup>

It is important to note that the final adversarial sentence used for evaluation, denoted as *adv-src*, is actually the decoded version of  $s^{adv}$ . Since byte-pair encoding (BPE) is a preprocessing module of the two NMT systems under consideration, *adv-src* will again be encoded using BPE to yield  $s_{fin}^{adv}$  which will be passed as an input to BLSTM/Transformer. It is possible for  $s_{fin}^{adv}$  to be different from  $s^{adv}$ .

<sup>2</sup><https://github.com/akshay107/nmt-attack>

<b>src</b>	Human contracted itself blind , malignant .
$s^{org}$	Human contra@@ cted itself blind , mal@@ ign@@ ant .
$s^{adv}$	Human contra@@ cted itself animals , mal@@ ign@@ den Please
<b>adv-src</b>	Human contracted itself animals , malignden Please
$s_{fin}^{adv}$	Human contra@@ cted itself animals , mal@@ ig@@ nden Please

**Table 4.4:** An example to showcase the prediction pipeline. Finally,  $s_{fin}^{adv}$  is given as input to the NMT system.

However, by replacing subwords with proper words, we ensure that the chances of the two sentences being different are low<sup>3</sup>. The rationale behind (re)encoding  $s^{adv}$  using BPE is that the goal of this work is to explore robustness of NMT systems and BPE is a crucial component of such systems, hence it is not judicious to bypass BPE by directly giving  $s^{adv}$  as input to BLSTM/Transformer. For ease of understanding, the entire pipeline is explained via an example sentence in Table 4.4. In this table, we can see that the two sentences,  $s^{adv}$  and  $s_{fin}^{adv}$ , are different.

## 4.5 Evaluation Metrics

As discussed earlier, the evaluation of the proposed invariance-based attack is more challenging than previous attacks mainly due to two distinct goals of the present attack: (i) To ensure that the predicted translation of the NMT system remains unchanged and (ii) To ensure the change in semantics of the original source sentence. In this section, we take a look at different metrics used to evaluate the efficiency of the attack. Furthermore, we propose a BLEU-based metric and discuss its advantages over standard metrics like success rate.

### 4.5.1 Success rate

For a particular NMT system, we define the *success rate* of a method as the percentage of adversarial sentences which were assigned the *same translation* as the original source sentence ( $s^{org}$ ) by the NMT system. This metric encapsulates the first goal of the adversarial attack while completely ignoring the second. Hence, along with success rate, we also consider another metric: number of replacements

<sup>3</sup>For example, for 500 adversarial sentences obtained via the proposed method for English-German, the two sentences are same in 484 cases.



(NOR). NOR is defined as the mean and the median of the number of replacements (normalized by the length of original sentence). It is more likely that the meaning of the sentence has changed if the NOR is higher. Hence, we can say that for two attacks with similar success rate, the one with higher NOR is better. However, higher NOR doesn't necessarily mean that the replacements are significant because of the following reason. Consider a scenario where for all the words in the sentence, very few characters are replaced. In such a case, number of replacements is as high as possible since all words have been replaced. But a human can easily disregard such character replacements as typos and can decipher the original sentence. Such a scenario is unlikely to occur for the proposed attack since it does not rely on character-level replacements. In order to ensure that this is the case, we consider another metric: char-F1. For each word  $w$  in the original sentence which is replaced by the word  $w'$  in the adversarial sentence, we calculate the char-F1. In this chapter, we report the mean of char-F1. Note that it is more judicious to consider char-F1 as a metric rather than fraction of characters replaced since the number of characters in the two words  $w$  and  $w'$  are different. A very high char-F1 shows that the two words  $w$  and  $w'$  have lots of common characters so a human might be able to figure out that it's a spelling mistake and consequently that the word  $w'$  is supposed to be the word  $w$ . However, a low char-F1 doesn't imply that the meaning of the original sentence has changed since the two words might be synonymous.

### 4.5.2 BLEU-based metric

While success rate is the most straightforward metric to measure the efficiency of an invariance based attack on an NMT system, it does have some disadvantages. As mentioned earlier, an attack method can achieve a higher success rate by doing fewer replacements. Hence, comparing the success rate, NOR and char-F1 simultaneously is a better approach. However, there are still few issues that we need to address (a) Although more number of replacements and lower char-F1 increase the chances of the meaning of the original sentence being changed, one can think of *pathological examples* where many replacements are made without significant change in the meaning, (b) It is possible that the original and adversarial sentences are assigned the same translation by the NMT system due to the property of the target language rather than a deficiency in the NMT system. As an example, if the target language does not have *gender markers* and *continuous tense*, then the

two sentences “He is playing guitar.” and “She plays guitar.” will have the same translation.

To address these issues, we propose a BLEU-based metric to evaluate efficiency of an invariance based attack. Consider 3 NMT systems: en-de Transformer, en-fr Transformer and en-de BLSTM. Let’s suppose that the proposed attack Min-Grad+Soft-Att is used to attack *en-de Transformer* resulting in pair of original/adversarial source sentences. To address issue (a), we can translate the original/adversarial source sentences to French using *en-fr Transformer*. If the meaning has not changed significantly, we can expect the BLEU score for the French translations to be high. To address issue (b), we can translate the original/adversarial source sentences to German using *en-de BLSTM* (since the target language is German). If the translations of Transformer were similar due to the property of target language, we can expect the BLEU score for the German translations by the BLSTM to be high as well.

To summarize, an *effective* invariance based attack is expected to give pair of original/adversarial source sentences whose corresponding translations by the model under attack have *high* BLEU scores and whose corresponding translations by the other NMT systems have *low* BLEU scores. In a general setting, let there be  $n$  NMT systems denoted by  $l_1, l_2, \dots, l_n$  where  $l_1$  is the NMT system under attack. Using the BLEU-based metric, we propose a composite score,  $e(M)$  to evaluate the efficiency of an attack method  $M$  as follows.

$$e(M) = \frac{b_{src} + (100 - b_{l_1}) + b_{l_2} + \dots + b_{l_n}}{n + 1} \quad (4.8)$$

where  $b_{src}, b_{l_i}$  denote the BLEU score for *src* and NMT system  $l_i$  respectively. For an attack method  $M$  to be more effective,  $e(M)$  should be lower.

## 4.6 Results

In this section, we discuss the results of the proposed method in comparison with the baseline methods. In Section 4.6.1, we look at the success rate and number of replacements of different methods across NMT systems. In Section 4.6.2, we evaluate the effectiveness of various method based on the proposed BLEU-based metric. Note that, we use BLSTM as a shorthand for BLSTM-based encoder-decoder with

attention in this section. We try to analyze the nature of replacements in successful adversarial examples and Section 4.6.3 presents our observations. In Section 4.6.4, we perform human evaluation to ensure that the adversarial sentences are semantically different than the original source sentences. Finally, in Section 4.6.5, we analyze the robustness of NMT systems on WMT dataset, i.e., a high-resource setting.

### 4.6.1 Success rate

Table 4.5 shows the success rate and the mean, median of the number of replacements (normalized by the length of original sentence) for different methods.

1: Comparing Min-Grad and random: As we can see from Table 4.5, for both Hotflip and Soft-Att, Min-Grad method gives significant improvement in success rate in comparison with random baseline across 5 of the 6 NMT systems. We also observe that the NOR for Min-Grad is comparable with random. This shows that the improvement in success rate is significant since otherwise, an attack method can achieve higher success rate by doing fewer replacements. For German-English BLSTM, Min-Grad + Soft-Att achieves a slightly lower success rate than random + Soft-Att while the NOR for the two methods is almost the same.

2: Comparing Soft-Att and HotFlip: From Table 4.5, across all the NMT systems, we can see that Soft-Att significantly outperforms HotFlip both in terms of success rate and NOR. In fact, random + Soft-Att outperforms Min-Grad + Soft-Att in terms of success rate and NOR across the 6 NMT systems.

3: Comparing BLSTM and Transformer: Table 4.5 shows that Transformer might be more robust to our proposed method than BLSTM since the proposed method has lower NOR in case of Transformer than BLSTM for the 3 language pairs. For English-German (en-de) and English-French (en-fr), the proposed method has higher success rate for BLSTM in comparison to Transformer. However, for German-English (de-en), we find that the proposed method has higher success rate for Transformers in comparison to BLSTM. It is to be noted that HotFlip has higher success rate and similar NOR in case for Transformer than BLSTM.

Another significant observation is that, for German-English (de-en), the NOR is significantly higher than English-German (en-de) and English-French (en-fr) for all the methods. Table 4.6 shows the mean of the char-F1 for different methods.

Model	Method	en-de		en-fr		de-en	
		Success Rate	NOR	Success Rate	NOR	Success Rate	NOR
BLSTM	random + HotFlip	25.4%	0.23, 0.21	28.2%	0.21, 0.18	31.4%	0.29, 0.27
	Min-Grad + HotFlip	31.8%	0.22, 0.19	40.2%	0.19, 0.17	44.4%	0.29, 0.27
	random + Soft-Att	61.2%	0.58, 0.62	64.6%	0.62, 0.67	<b>66.6%</b>	0.75, 0.81
	Min-Grad + Soft-Att	<b>67.8%</b>	0.58, 0.61	<b>70.8%</b>	0.61, 0.66	64.4%	0.75, 0.80
Transformer	random + HotFlip	35.0%	0.26, 0.24	40.6%	0.24, 0.21	50.8%	0.36, 0.34
	Min-Grad + HotFlip	45.0%	0.26, 0.24	44.0%	0.23, 0.21	56.8%	0.35, 0.33
	random + Soft-Att	50.2%	0.40, 0.39	59.0%	0.37, 0.35	64.6%	0.56, 0.59
	Min-Grad + Soft-Att	<b>61.6%</b>	0.41, 0.42	<b>64.8%</b>	0.36, 0.34	<b>71.4%</b>	0.58, 0.61

**Table 4.5:** Success Rate (in %) and number of replacements for different methods. *NOR* represents the mean/median of the normalized Number Of Replacements across all the sentences. The highest success rate is marked in bold.

Model	Method	char-F1		
		en-de	en-fr	de-en
BLSTM	random + HotFlip	0.16	0.18	0.16
	Min-Grad + HotFlip	0.13	0.14	0.15
	random + Soft-Att	0.26	0.29	0.29
	Min-Grad + Soft-Att	0.26	0.28	0.28
Transformer	random + HotFlip	0.21	0.22	0.22
	Min-Grad + HotFlip	0.19	0.19	0.20
	random + Soft-Att	0.23	0.23	0.25
	Min-Grad + Soft-Att	0.22	0.22	0.26

**Table 4.6:** Mean of char-F1 for different methods M.

Here, we can see that all the methods have low value of char-F1. This shows that the replacements made by all the methods are significant. Overall, as is evident from Table 4.5, our proposed method (Min-Grad + Soft-Att) achieves the highest success rate across 5 of the 6 NMT systems. For English-German BLSTM, the proposed method achieves slightly lower success rate than random+Soft-Att while having similar NOR. The difference in success rate is marginal in comparison with the other 5 NMT systems where the difference is in the range of 6 to 11 percent.

## 4.6.2 BLEU-based metric

Table 4.7 shows the BLEU scores for the original/adversarial sentence (src) and their respective translation by different MT systems. In Table 4.7,  $l_1$  denotes the *Transformer* model under attack (e.g. en-de),  $l_2$  denotes the other *Transformer* model (e.g. en-fr), and  $l_1^{blstm}$ ,  $l_2^{blstm}$  are the BLSTM counterparts of  $l_1$  and  $l_2$  and  $l_1^{moses}$ ,  $l_2^{moses}$  are the Moses [66] counterparts of  $l_1$  and  $l_2$ . Similarly, Table 4.8 shows the BLEU scores for the original/adversarial sentence (src) and their respective translation by different MT systems. In Table 4.8,  $l_1$  denotes the *BLSTM* model under attack,  $l_2$  denotes the other BLSTM model,  $l_1^{trans}$ ,  $l_2^{trans}$  are the *Transformer* counterparts of  $l_1$  and  $l_2$  and  $l_1^{moses}$ ,  $l_2^{moses}$  are the Moses counterparts of  $l_1$  and  $l_2$ . Note that for the language pair German-English (de-en),  $l_2$  denotes German-French (de-fr). For an attack to be effective, BLEU score for  $l_1$  should be high and the other six BLEU scores should be low. Note that the BLEU score for *src* is related with the number of replacements reported in Table 4.5. The two metrics are inversely related; more number of replacement implies lower BLEU score for *src*.

Transformer	Method	src ( $\downarrow$ )	$l_1$ ( $\uparrow$ )	$l_2$ ( $\downarrow$ )	$l_1^{blstm}$ ( $\downarrow$ )	$l_2^{blstm}$ ( $\downarrow$ )	$l_1^{moses}$ ( $\downarrow$ )	$l_2^{moses}$ ( $\downarrow$ )
<b>en-de</b>	random + HotFlip	51.04	80.49	47.53	36.42	43.66	42.72	48.51
	Min-Grad + HotFlip	53.23	83.13	49.15	36.51	44.76	43.86	50.06
	random + Soft-Att	32.01	84.79	<b>29.72</b>	<b>20.62</b>	27.85	25.83	<b>31.55</b>
	Min-Grad + Soft-Att	<b>31.17</b>	<b>88.55</b>	31.09	20.63	<b>27.43</b>	<b>25.11</b>	31.93
<b>en-fr</b>	random + HotFlip	55.51	85.18	40.35	52.00	36.18	56.41	46.64
	Min-Grad + HotFlip	57.92	88.40	41.98	54.39	37.68	57.90	48.80
	random + Soft-Att	<b>33.61</b>	89.77	<b>21.59</b>	<b>32.37</b>	<b>19.09</b>	<b>36.56</b>	<b>27.27</b>
	Min-Grad + Soft-Att	35.40	<b>91.99</b>	23.28	34.32	20.29	38.20	27.64
<b>de-en</b>	random + HotFlip	34.85	86.03	37.55	35.93	34.01	36.08	35.07
	Min-Grad + HotFlip	36.31	88.21	36.64	36.68	33.98	37.39	36.08
	random + Soft-Att	14.01	89.78	21.39	19.57	20.27	19.39	22.22
	Min-Grad + Soft-Att	<b>12.05</b>	<b>92.00</b>	<b>21.18</b>	<b>18.68</b>	<b>18.80</b>	<b>18.19</b>	<b>20.04</b>

**Table 4.7:** BLEU scores for the original/adversarial sentence (src) and their respective translations by the four NMT systems.  $l_1$  denotes the model under attack,  $l_2$  denotes the other Transformer model.  $l_1^{blstm}, l_2^{blstm}$  are the BLSTM counterparts of  $l_1$  and  $l_2$ . Similarly,  $l_1^{moses}, l_2^{moses}$  are MOSES counterparts of  $l_1$  and  $l_2$ . The arrows in the table header denote whether lower/higher is better for an attack to be effective.

BLSTM	Method	src ( $\downarrow$ )	$l_1$ ( $\uparrow$ )	$l_2$ ( $\downarrow$ )	$l_1^{trans}$ ( $\downarrow$ )	$l_2^{trans}$ ( $\downarrow$ )	$l_1^{moses}$ ( $\downarrow$ )	$l_2^{moses}$ ( $\downarrow$ )
<b>en-de</b>	random + HotFlip	57.09	71.35	48.84	43.90	49.42	47.50	53.98
	Min-Grad + HotFlip	59.28	75.55	50.38	45.96	52.26	49.83	57.05
	random + Soft-Att	<b>13.77</b>	87.14	<b>19.20</b>	<b>18.36</b>	<b>21.62</b>	<b>14.88</b>	<b>19.66</b>
	Min-Grad + Soft-Att	14.49	<b>89.86</b>	19.74	18.51	21.98	15.69	20.55
<b>en-fr</b>	random + HotFlip	60.87	79.62	39.28	58.60	41.73	59.39	50.81
	Min-Grad + HotFlip	63.97	84.87	41.16	61.80	44.94	62.99	53.30
	random + Soft-Att	12.99	92.44	10.62	28.34	12.12	<b>22.64</b>	<b>12.00</b>
	Min-Grad + Soft-Att	<b>12.66</b>	<b>93.87</b>	<b>9.95</b>	<b>27.21</b>	<b>11.92</b>	23.07	12.16
<b>de-en</b>	random + HotFlip	41.72	76.19	39.26	44.95	38.02	39.48	41.19
	Min-Grad + HotFlip	43.55	81.64	38.61	46.05	40.65	41.10	42.52
	random + Soft-Att	4.13	<b>89.62</b>	14.47	17.80	14.25	<b>12.94</b>	<b>13.83</b>
	Min-Grad + Soft-Att	<b>3.62</b>	88.76	<b>13.74</b>	<b>17.20</b>	<b>13.71</b>	13.29	14.35

**Table 4.8:** BLEU scores for the original/adversarial sentence (src) and their respective translation by the four NMT Systems.  $l_1$  denotes the model under attack,  $l_2$  denotes the other BLSTM model.  $l_1^{trans}, l_2^{trans}$  are the Transformer counterparts of  $l_1$  and  $l_2$ . Similarly,  $l_1^{moses}, l_2^{moses}$  are MOSES counterparts of  $l_1$  and  $l_2$ . The arrows in the table header denote whether lower/higher is better for an attack to be effective.

Model	Method(M)	e(M)		
		en-de	en-fr	de-en
BLSTM	random + HotFlip	47.05	47.29	38.35
	Min-Grad + HotFlip	48.46	49.04	38.70
	random + Soft-Att	<b>17.19</b>	15.18	12.54
	Min-Grad + Soft-Att	17.30	<b>14.73</b>	<b>12.45</b>
Transformer	random + HotFlip	41.34	43.13	32.50
	Min-Grad + HotFlip	42.06	44.32	32.70
	random + Soft-Att	26.11	<b>25.82</b>	17.87
	Min-Grad + Soft-Att	<b>25.54</b>	26.73	<b>16.70</b>

**Table 4.9:**  $e(M)$  for different methods M (lower values of  $e(M)$  imply better attack efficiency).

en-de	src	And because God loves her , I did get married .
	adv-src	plus because God loves them kilograms me been abused married .
	pred	Und weil Gott sie sie liebt , wurde ich verheiratet .
en-de	src	I want to know the people behind my dinner choices .
	adv-src	I want ordinarily know the humans behind my dinner flog arguments
	pred	Ich möchte die Menschen hinter meinen Abendessen kennen .
en-fr	src	I was clearly more nervous than he was .
	adv-src	adaptations was clearly more nervous label he was .
	pred	J'étais clairement plus nerveux qu'il était .
en-fr	src	A dome , one of these ten-foot domes .
	adv-src	An dome pale an of Those exes 3 foot domEvelyn tat
	pred	Un dôme , un de ces dômes de 3 mètres .

**Table 4.10:** Examples of *Min-Grad + Soft-Att* for BLSTM-based Encoder-Decoder with Attention. The NMT system predicts the same translation for src and adv-src.

From Tables 4.7 and 4.8, we can see that Soft-Att achieves a higher BLEU score for  $l_1$  in comparison with HotFlip for all the experimental settings. Moreover, the other four BLEU score are lower for Soft-Att than HotFlip. This result showcases the efficiency of the proposed method since it outperforms HotFlip in terms of success rate, number of replacements and BLEU scores. The fact that, for the proposed method, BLEU scores is low for other NMT systems also shows that the adversarial sentences are not transferable in nature. In other words, the pair of original/adversarial sentences are specific to the NMT system.

Table 4.9 shows  $e(M)$  values for different methods and across NMT systems. This table nicely summarizes the results presented in Tables 4.7 and 4.8. The  $e(M)$  values suggest that the state-of-the-art NMT systems are unable to capture the



<b>en-de</b>	<b>src</b>	Is it something about the light ?
	<b>adv-src</b>	Is Bald passage about the light ?
	<b>pred</b>	Geht es um das Licht ?
<b>en-de</b>	<b>src</b>	So the whole is literally more than the sum of its parts .
	<b>adv-src</b>	Small the whole is bucks more than number sum Von His parts rank
	<b>pred</b>	Das Ganze ist mehr als die Summe seiner Teile .
<b>en-fr</b>	<b>src</b>	They look like the stuff we walk around with .
	<b>adv-src</b>	Hudson look like the ping we walk fishes with .
	<b>pred</b>	Ils ressemblent à ce que nous marchons avec .
<b>en-fr</b>	<b>src</b>	There are many , many problems out there .
	<b>adv-src</b>	look numerous supported stays behold problems hundred there .
	<b>pred</b>	Il y a de nombreux problèmes .

**Table 4.11:** Examples of *Min-Grad + Soft-Att* for Transformer. The NMT system predicts the same translation for src and adv-src.

semantics of the adversarial examples generated by the the proposed method, Min-Grad+Soft-Att, in a low-resource setting. To further investigate whether the semantics of the original source sentences have changed via the proposed attack, we perform human evaluation, which is explained in Section 4.6.4.

### 4.6.3 A Comment on Types of Words Replaced

In order to understand what types of words are replaced to generate successful adversarial examples, we observe that there is no clear trend about the types of words replaced. Both highly frequent (stop words) and thematic words are getting replaced. The model under attack remains invariant to replacement of highly thematic words as well as frequent words by semantically very different words. Invariance is observed even in case of introduction of named-entities (NEs). While trying to understand if specific parts-of-speech (POS) are vulnerable, no clear tendency is noted. These observations are highlighted through the examples given in Tables 4.10 (for BLSTM-based encoder-decoder with attention model) and 4.11 (for Transformer based translation model) which are generated by *Min-Grad+Soft-Att* method.

Model	Score
en-de BLSTM	1.83, 1.0
en-fr BLSTM	1.86, 1.0
en-de Transformer	2.14, 1.0
en-fr Transformer	2.55, 2.0

**Table 4.12:** Human evaluation: Mean and median of semantic similarity score for different NMT systems.

#### 4.6.4 Human evaluation

To further ensure that the adversarial sentence (i.e.,  $s^{adv}$ ) is semantically different than the original sentence (i.e.,  $s^{org}$ ), we perform human evaluation. This experiment is done for 4 NMT systems, i.e., English-German BLSTM, English-French BLSTM, English-German Transformer, and English-French Transformer. For each NMT system, we randomly select 50 different pairs of original source sentence and adversarial sentence obtained via the proposed method, i.e., Min-Grad + Soft-Att. We select 4 participants for this task. All the 4 participants are NLP researchers. The task is carried out in two phases. In the first phase, the  $i^{th}$  participant is asked to make the adversarial sentences corresponding to the  $i^{th}$  NMT system grammatically correct, if possible. The participant is not provided with the original sentences in order to ensure fairness. In case an adversarial sentence is too noisy to make any sense, we ask the participant to leave the sentence as is. In the second phase, the other 3 participants are provided with the original sentences and the corresponding *manually cleaned* version of the adversarial sentences obtained from the  $i^{th}$  participant. For each pair of the original sentence and the corresponding *cleaned* version of the adversarial sentence, the participants are asked to quantify the semantic similarity by assigning an integral score ranging from 1 to 5. The score of 1 signifies that the two sentences are completely different semantically, while the score of 5 signifies that the two sentences are paraphrases of each other. In this way, each pair of  $(s^{org}, s^{adv})$  is assigned a semantic similarity score by three participants.

Out of the total 200 pairs, we find that the maximum and the minimum scores differ by more than 2 in 28 pairs. Out of these 28 pairs, we find that in 16 pairs, the scores of the two participants exactly match. This shows that the overall level of agreement between the participants is high. Table 4.12 shows the mean and the median of the semantic similarity score for the 4 NMT systems. From

Table 4.12, we can see that the mean and the median of the semantic similarity score are very low for all the 4 NMT systems. In fact, 3 NMT systems achieve the lowest median score possible. This shows that the adversarial sentences obtained via the proposed method are indeed semantically very different from the original sentences.

### 4.6.5 Results on WMT Dataset

In this section, we test our proposed attack against state-of-the-art NMT system trained on WMT dataset, i.e., a high resource setting. We use the Transformer model which is publicly available under the fairseq framework for experimentation [95, 96]. The model was trained on WMT 16 English-German dataset containing roughly 4.5 million sentences (i.e., one order of magnitude higher than TED dataset). It achieves a BLEU score of 29.30 on newstest2014. To study the proposed attack, we select 50 sentences from newstest2014. Table 4.13 shows the success rate, number of replacements and the mean of char-F1 for different attack methods. Similar to TED dataset, all the methods achieve low mean value of char-F1 for WMT dataset as well. Furthermore, as we can see from Table 4.13, *Min-Grad+Soft-Att* method achieves the best success rate and number of replacements. However, across all the methods, there is a significant drop in both the success rate and number of replacement in comparison to results obtained for TED dataset. Table 4.14 shows the BLEU scores for the original/adversarial sentence (src) and their respective translation by the three NMT systems. All the three NMT systems are publicly available under the fairseq framework. In Table 4.14,  $l_1$  denotes Transformer trained on WMT 16 English-German dataset (i.e., the model under attack),  $l_2$  denotes Transformer trained on WMT 14 English-French dataset [96],  $l_1^{wmt19}$  denotes Transformer trained on WMT 19 English-German dataset [92]. As we can see from Table 4.14, the BLEU scores for  $l_1$  are lower compared to the ones for TED dataset (i.e. Table 4.7). Furthermore, we observe that BLEU score for  $l_1^{wmt19}$  is on the higher side. This shows that replacements made during the attack are relying more on the properties of the target language. This further demonstrates the advantages of BLEU-based metric in comparison to success rate. Hence, we can conclude that NMT systems trained on larger dataset are significantly more robust to the proposed invariance-based attack in comparison to low-resource setting. From Table 4.3, we can see that the Transformer model achieves a similar BLEU score on en-de TED dataset (i.e., 29.27) as on the

Method	Success Rate	NOR	char-F1
random + HotFlip	16.0 %	0.19, 0.19	0.21
Min-Grad + HotFlip	14.0 %	0.21, 0.19	0.21
random + Soft-Att	24.0 %	0.43, 0.44	0.29
Min-Grad + Soft-Att	<b>34.0 %</b>	0.46, 0.48	0.27

**Table 4.13:** Success Rate (in %), number of replacements, and mean of char-F1 for different methods against Transformer trained on WMT 16 English-German. *NOR* represents the mean/median of the normalized Number Of Replacements across all the sentences. The highest success rate is marked in bold.

Method	src	$l_1$	$l_2$	$l_1^{wmt19}$
random + HotFlip	54.61	64.11	48.42	70.03
Min-Grad + HotFlip	52.54	73.08	52.92	65.53
random + Soft-Att	21.09	63.32	25.88	<b>42.34</b>
Min-Grad + Soft-Att	<b>18.47</b>	<b>64.79</b>	<b>22.66</b>	44.74

**Table 4.14:** BLEU scores for the original/adversarial sentence (src) and their respective translation by the three NMT systems.  $l_1$  denotes the Transformer trained on WMT 16 English-German,  $l_2$  denotes the Transformer trained on WMT 14 English-French and  $l_1^{wmt19}$  denotes the Transformer trained on WMT 19 English-German.

larger WMT dataset (i.e., 29.30). Despite this, the NMT system in low-resource setting is significantly less robust to the proposed attack. This shows that high BLEU score on the clean test set does not imply high robustness to adversarial attacks. This warrants the need of robustness analysis in conjunction with the BLEU score.

## 4.7 Summary

In this chapter, we showcased *undesirable invariances* captured by an NMT system. We define *undesirable invariances* as the scenario in which the predicted translation remains unchanged when multiple words in the source sentence are replaced changing the semantic of the input sentence. Three language pairs, namely, English-German (en-de), English-French (en-fr), and German-English (de-en) are considered to investigate the behaviour of two state-of-the-art NMT systems: BLSTM-based encoder-decoder with attention and Transformer. We break down the problem of replacing a word into two sub-problems: traversing position indices and replacing a word given a position. Two techniques, *Min-Grad*

---

and *Soft-Att* are proposed for the two sub-problems. The results show that the proposed techniques significantly outperform HotFlip and random related baselines. We also propose an alternate BLEU-based metric to evaluate an invariance based attack and argue the effectiveness of the proposed metric in comparison to success rate. Furthermore, we also perform human evaluation to show that the semantics of the original source sentence is drastically changed by the proposed method. This study is motivated to explore the robustness of NMT systems to nonsensical inputs. We study the robustness in low-resource setting ( $< 0.2$  million training samples) as well as high-resource setting ( $\sim 4.5$  million training samples). Our results demonstrate that the state-of-the-art NMT systems are significantly more robust in high-resource setting than low-resource setting. However, since most of the language pairs do not have a huge amount of training data, the lack of robustness of NMT systems to nonsensical inputs in low-resource setting is a concern.

## Chapter 5

# Ignorance is Bliss: Exploring Defenses Against Invariance based Attacks on NMT systems

*Where ignorance is bliss, 'tis folly to be wise.*

Thomas Gray

This chapter explores defense strategies against invariance-based attack on NMT systems. In the presence of gold translation for adversarial examples, standard adversarial training has been shown to improve robustness of NMT systems to the particular type of noise in consideration [21, 22, 31, 76]. However, as seen in Chapter 4, the adversarial examples obtained via invariance-based attack are nonsensical and do not have a *gold-translation*. The lack of gold translation makes tackling invariance-based attacks a challenging task. In this chapter, we propose two contrasting defense strategies for the same, namely, *learn to deal* and *learn to ignore*. Since the goal of this chapter is adversarial defense, we evaluate the defense strategies against *bruteforce attack* which is a stronger (although considerably slower) invariance-based attack than Min-Grad+Soft-Att.

The rest of this chapter is organized as follows. Section 5.1 discusses the main idea behind the two defense strategies. This section also describes the bruteforce attack in detail and evaluate its efficiency against Transformer. Section 5.2 describes the proposed defense strategies in detail. Section 5.3 describes the implementation details. Section 5.4 introduces some metrics for evaluating the efficiency of the

proposed defense. Section 5.5 analyzes the results of the two defense strategies. Finally, Section 5.6 summarizes the chapter.

## 5.1 Background

This section is organized as follows. Section 5.1.1 discusses the main idea behind the two proposed defense strategies. Section 5.1.2 describes the bruteforce attack in detail. Finally, Section 5.1.3 presents the results of bruteforce attack against Transformer on TED dataset [105].

### 5.1.1 Overview of the Proposed Method

Table 5.1 shows an adversarial example obtained via bruteforce attack. From this example, it is clear that not only do we not have gold translation for such noisy adversarial sentences (i.e., adv-src), but generating gold translation for such sentences is a tedious task as well. In order to design a defense strategy for invariance-based attacks in such a scenario, it is crucial to decide how an NMT system should behave for adversarial examples like the one shown in Table 5.1. In other words, a desirable behaviour for NMT system in such a scenario needs to be explicitly stated. In this chapter, we consider the following two behaviours of NMT system desirable, (i) NMT system predicts a *different translation* for such adversarial examples, and (ii) NMT system predicts a *dummy sentence* (such as “This sentence is not correct.” in the target language) whenever it is fed an invariance-based adversarial example. This leads to two contrasting defense strategies, namely *learn to deal* and *learn to ignore*. In *learn to deal*, NMT system learns not to predict the same translation. This essentially teaches the translation system *what not to do*. In *learn to ignore*, NMT system learns to output a dummy sentence for any invariance-based adversarial example. This teaches the translation system *what to do*. The motivation behind *learn to ignore* strategy is aligned with the ongoing effort in the machine learning community where along with improving the classification accuracy, one effort is to teach a machine to learn to say “I don’t know” for an input completely unknown to it (i.e., out-of-distribution detection) since it is better to show ignorance rather than giving a wrong prediction for an out-of-distribution input [39, 87, 130]. Table 5.2 shows example of the two defense strategies.

<b>src</b>	Well right now , as you can see , the results can be somewhat comical .
<b>adv-src</b>	mortality 2030 present prior like yourself able sees agents lay outcomes could rowed fairly weird An .
<b>pred</b>	Nun , im Moment , wie Sie sehen können , können die Ergebnisse ziemlich komisch sein .

**Table 5.1:** Example of bruteforce attack on English-German Transformer. The NMT system predicts the same translation (**pred**) for the clean source sentence (**src**) and the noisy sentence (**adv-src**).

<b>Learn to Deal</b>	
<b>src</b>	Keyloggers silently sit on your computer , hidden from view , and they record everything you type .
<b>adv-src</b>	Keyvoyage es ships don sitting ' yours computational , hidden by standpoint ; plus them playground everything yours manner coin
<b>pred</b>	Keyschiffe sitzen auf Ihrem Computer , versteckt von Sicht , und sie zeichnen alles auf , was <i>Sie tippen</i> .
<b>adv-pred</b>	Keyschiffe sitzen auf Ihrem Computer , versteckt von Sicht ; und sie zeichnen alles auf , was <i>Ihre Münze angeht</i> .

<b>Learn to Ignore</b>	
<b>src</b>	For community though , we start at the very beginning .
<b>adv-src</b>	To community Yet ; us begin in dark total starting '
<b>pred</b>	Pour la communauté pourtant , nous commençons au tout début .
<b>adv-pred</b>	Cette phrase n'est pas correcte .

**Table 5.2:** Example of the two defense strategies. Learn to Deal strategy predicts a different translation for src and adv-src (the difference is shown in italics). Learn to Ignore strategy predicts “This sentence is not correct” in the target language (i.e., French in this case) for adv-src.

### 5.1.2 Bruteforce Attack

Similar to Chapter 4, we perform vocabulary pruning to obtain the set  $V_{prune}$ . Let  $t^{pred}$  denote the predicted translation for the source sentence,  $s^{org}$ , by the NMT system. Bruteforce attack changes multiple words in  $s^{org}$  with words in  $V_{prune}$  with the goal of keeping the predicted translation unchanged. To achieve this, the attack uses the standard negative log-likelihood loss function,  $L_{nll}$ , for the tuple  $(s, t^{pred})$  given by

$$L_{nll} = -\frac{1}{m} \sum_{i=1}^m \log(p(t_i^{pred} | t_{<i}^{pred}, s)) \quad (5.1)$$

where  $p(t_i^{pred} | t_{<i}^{pred}, s)$  denotes the probability assigned to the word  $t_i^{pred}$  by the NMT system and  $m$  is the length of  $t^{pred}$  and  $s$  is a source sentence. Algorithm 5.1



**Algorithm 5.1:** Bruteforce Attack

---

**Input:**  $s^{org}, t^{pred}$   
**Output:**  $s^{adv}$

$l_{org} \leftarrow L_{null}$  for  $(s^{org}, t^{pred})$   
 $n \leftarrow \text{len}(s^{org})$   
 $s \leftarrow s^{org}$   
 $l_{global} \leftarrow 100$   
**for**  $j \leftarrow 1$  **to**  $max_{sweep}$  **do**  
     $flag \leftarrow \mathbf{False}$   
    Sample  $\pi(n)$  from  $Sym(n)$   
    **for**  $r$  **in**  $\pi(n)$  **do**  
         $s^{temp} \leftarrow s$   
         $l^r \leftarrow [ ]$   
        **for**  $w$  **in**  $V_{prune}$  **do**  
             $s^{temp}[r] \leftarrow w$   
             $l_w^r \leftarrow L_{null}$  for  $(s^{temp}, t^{pred})$   
            **append**  $l_w^r$  to  $l^r$   
        **end for**  
         $l_{min}^r \leftarrow \min(l^r)$   
         $ind_{word} \leftarrow \text{argmin}(l^r)$   
         $w_r \leftarrow V_{prune}[ind_{word}]$   
        **if**  $s[r] \neq w_r$  **and**  $l_{min}^r < l_{global}$   
            **then**  
                 $l_{global} \leftarrow \max(l_{min}^r, l_{org})$   
                 $s[r] \leftarrow w_r$   
                 $flag \leftarrow \mathbf{True}$   
            **end if**  
        **end if**  
    **end for**  
    **if not**  $flag$  **then**  
        **break**  
    **end if**  
**end for**  
 $s^{adv} \leftarrow s$   
**return**  $s^{adv}$

---

summarizes the bruteforce attack. In the algorithm,  $\pi(n)$  denotes a permutation over  $n$  positions in  $s^{org}$ . In other words,  $\pi(n) \in Sym(n)$  where  $Sym(n)$  denotes the symmetric group on the set  $\{1, 2, \dots, n\}$ . The attack makes at most  $max_{sweep}$  sweeps over the source sentence. Within a sweep, the attack traverses all the positions in a random order. For a particular position  $r$ , it looks for a word  $w_r \in V_{prune}$  which will result in the minimum negative log-likelihood loss,  $l_{min}^r$ , assuming that  $w_r$  has been inserted at position  $r$ . Replacement is actually done if  $l_{min}^r$  is also less than the current global loss,  $l_{global}$ . At the start of the attack,  $l_{global}$

	Success rate	NOR	BLEU
<b>en-de</b>	84.0%	0.81, 0.84	29.22
<b>en-fr</b>	84.6%	0.82, 0.86	42.59

**Table 5.3:** Success rate and mean, median of number of replacements (NOR) for bruteforce attack, and BLEU score on the test set.

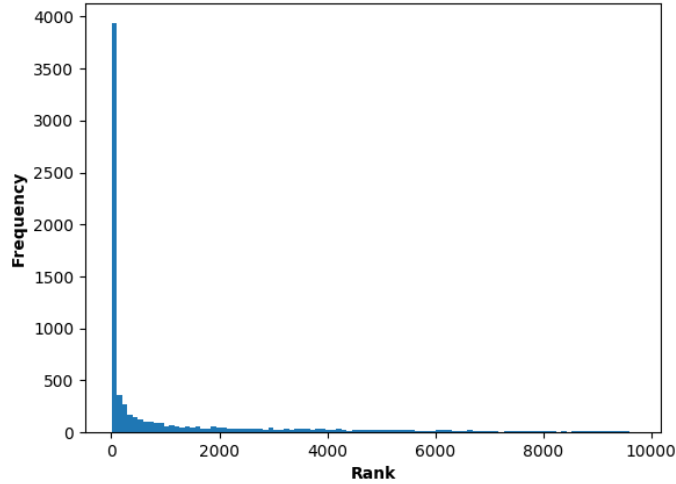
is set to a very high value to ensure that at least one replacement always takes place. If a replacement does take place,  $l_{global}$  is updated as  $\max(l_{min}^r, l_{org})$  where  $l_{org}$  is the original loss, i.e.,  $L_{nll}$  for the tuple  $(s^{org}, t^{pred})$ . Clipping  $l_{global}$  at  $l_{org}$  allows for more number of replacements. The attack is stopped if no replacement takes place within a sweep since continuing the attack will not result in any more replacements.

### 5.1.3 Efficiency of Bruteforce Attack

We evaluate the efficiency of bruteforce attack on 500 randomly chosen sentences from the *test set* of TED talks dataset [105]. The experiments are conducted for two language pairs, English-German (en-de) and English-French (en-fr). The average lengths of the 500 chosen sentences are 18.45, 18.40 for English-German and English-French respectively. The *Transformer base model* configuration consisting of 6 encoder-decoder layers is used for both the language pairs. The implementation provided by Sachan and Neubig [111] is followed. Byte pair encoding with 32,000 merge operations is used. During prediction, beam width is set to 5. For bruteforce attack,  $max_{sweep}$  is set to 5 for all the experiments. Note that  $|V|$  is 9,723 and 11,699 for en-de and en-fr respectively.

Table 5.3 shows the success rate and the mean, median of the number of replacements of bruteforce attack, and BLEU score for the two NMT systems.<sup>1</sup> The success rate is defined as the percentage of adversarial sentences, i.e.,  $s^{adv}$  obtained from bruteforce attack, which are assigned the same translation as  $s^{org}$  by the NMT system. We also report the BLEU score for the two systems on the test set. As we can see from Table 5.3, bruteforce attack achieves a high success rate for both the systems and replaces a large fraction of words ( $> 80\%$ ) per sentence

<sup>1</sup>Note that BLEU score for en-de and en-fr in Table 5.3 differs from Table 4.3 due to the difference in computing infrastructure.



**Figure 5.1:** Histogram of  $\text{rank}(w_{adv} | w_{org})$  for en-de Transformer.  $w_{org}$  and  $w_{adv}$  denote the original word and the replaced word during bruteforce respectively.

on an average. In contrast, for larger datasets like WMT dataset, we observe that the bruteforce attack achieves a relatively low success rate ( $\sim 36\%$ ).

To study the nature of replacements made by bruteforce attack, we take a look at the embedding layer of the NMT system. This layer embeds each word<sup>2</sup> to a vector. Let  $e(w)$  denote the embedding of a word  $w$  and  $w_{org}$  denote the original word. Consider the set,  $\text{sim}_{w_{org}}^V$ , given by

$$\text{sim}_{w_{org}}^V = [\text{sim}(e(w_{org}), e(w)) | w \in V] \quad (5.2)$$

where  $\text{sim}(\cdot, \cdot)$  denotes cosine similarity. Let  $w_{adv}$  denote the word which replaced  $w_{org}$  during bruteforce attack. We analyse the rank of  $\text{sim}(e(w_{org}), e(w_{adv}))$  in the set  $\text{sim}_{w_{org}}^V$ . Lower the rank, higher the cosine similarity. We denote this rank by  $\text{rank}(w_{adv} | w_{org})$ . Figure 5.1 shows the histogram of  $\text{rank}(w_{adv} | w_{org})$  for en-de Transformer. We observe a similar pattern for en-fr Transformer as well. From the figure, it is evident that the bruteforce attack heavily relies on *low-rank replacements*.

Since most of the replacements that occur during the bruteforce attack are *low-rank*, a natural question arises: “Is bruteforce attack really an attack in the true sense? In other words, are the translations of  $s^{org}$  and  $s^{adv}$  supposed to be same?”. We argue that this is not the case. This is mainly due to two reasons: (i) The

<sup>2</sup>In reality, embedding layer embeds each subword to a vector since the NMT system is subword-level. We use the term *word* for sake of simplicity.

	$l_1$	$l_2$	$l_1^{blstm}$	$l_2^{blstm}$
<b>en-de</b>	95.10	12.90	11.15	7.88
<b>en-fr</b>	96.97	4.47	15.27	4.52

**Table 5.4:** BLEU scores for predicted translations of  $s^{org}$  and  $s^{adv}$  across NMT systems.  $l_1$  denotes the Transformer under attack,  $l_2$  denotes the Transformer for the other language pair, and  $l_1^{blstm}$ ,  $l_2^{blstm}$  denote respective BLSTM-based NMT systems.

bruteforce attack replaces a large fraction of words as shown in Table 5.3. More the number of replacement, lesser the chance that the semantics of the original sentence is preserved. (ii) Since the Transformer used in the present study is subword-level, anytime a subword is replaced, it gets replaced by a *proper word* in  $V$ . Such replacements definitely change the semantics of the sentence.

To further quantify the argument, similar to Chapter 4, we also report the BLEU score between the predicted translations of  $s^{org}$  and  $s^{adv}$  by the two Transformer models as well as their BLSTM-based encoder-decoder with attention counterparts [81]. Table 5.4 shows these results. In Table 5.4,  $l_1$  denotes the Transformer under attack,  $l_2$  denotes the Transformer for the other language pair (e.g., if en-de Transformer is under attack then  $l_2$  denotes en-fr Transformer), and  $l_1^{blstm}$ ,  $l_2^{blstm}$  denote respective BLSTM-based encoder-decoder with attention systems. In Table 5.4, the low BLEU scores for the NMT systems not under attack (i.e.,  $l_2, l_1^{blstm}, l_2^{blstm}$ ) shows that the two sentences,  $s^{org}$  and  $s^{adv}$  are not semantically similar.

## 5.2 Defense Methodology

Since bruteforce attack is extremely slow (the time statistics are given in §5.3), it is not feasible to incorporate the adversarial examples from the attack during training stage. Hence, the insight that bruteforce attack relies heavily on *low-rank replacements* is crucial to design NMT systems which can withstand the attack. Since the search space of noisy sentences is exponentially large, this insight allows us to efficiently generate noisy samples which have a commonality with adversarial examples obtained from bruteforce attack. Essentially, the aforementioned insight makes the task of generating *good-quality* noisy samples tractable.

### 5.2.1 Generating noisy samples

Let  $w$  be a word in the original (i.e., clean) source sentence,  $s^{org}$ . We define top- $k$  neighbors of word  $w$ , denoted as  $N_k(w)$ , as the following set

$$N_k(w) = [\text{rank}(w' | w) \leq k \mid w' \in V_{prune}] \quad (5.3)$$

It is crucial to note that  $N_k(w)$  is not a fixed set and is dependent on the weights of the embedding layer. As the weight of the embedding layer changes during training,  $N_k(w)$  will also change. To overcome this obstacle, we train the NMT system only on the clean data for  $ep_{freeze}$  epochs (a hyperparameter). Training exclusively on the clean data allows the NMT system to learn *good-quality* embeddings. After  $ep_{freeze}$  epochs, we freeze the embedding layer and train the NMT system on both clean and noisy data. The noisy samples are generated by replacing  $w$ 's by  $w'$ 's where  $w' \in N_k(w)$ . We refer to the tuple  $(w, w')$  as a *pair*.

Let  $s^{org} = (w_1, w_2, \dots, w_n)$  be a clean source sentence. To generate a noisy sentence,  $s^{noisy}$ , from  $s^{org}$ , we explore two different methods as follows.

**Random:** We randomly select  $\lceil frac \times n \rceil$  positions where  $\lceil \cdot \rceil$  is the ceiling function. For a selected position  $r$ , we randomly choose a word  $w'_r$  where  $w'_r \in N_k(w_r)$  to replace  $w_r$ .

**Tackle-Bias:** Selecting positions randomly is intrinsically biased towards words with high unigram count, i.e., the words that occur frequently in the training corpus are more likely to get selected. To tackle this bias, we assign a probability,  $p_r$ , to each position  $r$  in  $s^{org}$  as follows

$$\begin{aligned} c(w_r) &= \sum_{w \in V} \text{count}(w_r, w) \\ p_r &= \frac{\exp(-\mu c(w_r))}{\sum_{i=1}^n \exp(-\mu c(w_i))} \end{aligned} \quad (5.4)$$

where  $c(w_r)$  denotes the running count of the number of times the word  $w_r$  has been replaced so far. Note that the probability  $p_r$  is inversely proportional to  $c(w_r)$ . We sample  $\lceil frac \times n \rceil$  positions without replacement from the probability distribution. For a selected position  $r$ , we again define a similar probability distribution where probability of a pair  $p(w_r, w)$  is inversely proportional to its running count. Finally, we sample a pair  $(w_r, w'_r)$  where  $w'_r \in N_k(w_r)$  from the probability distribution and replace  $w_r$  with  $w'_r$ .

## 5.2.2 Training Loss Function

In this section, we describe the two strategies, *learn to deal* and *learn to ignore*, in detail. For both the strategies, as mentioned earlier, the NMT system is trained exclusively on clean data for  $ep_{freeze}$  epochs. After this point, the embedding layer is frozen and the system is trained on both clean and noisy data. Note that the noisy data is generated for every training iteration using the method described in §5.2.1. Hence, for a particular  $s^{org}$ , multiple noisy sentences, i.e.,  $s^{noisy}$  are generated.

**a) Learn to Deal:** In *learn to deal*, our goal is to teach NMT system not to predict the same translation for clean source sentence and its noisy counterpart. To this end, after  $ep_{freeze}$  epochs, the loss function,  $L_{tot}$ , for this strategy is given as

$$L_{tot} = L_{nll}(s^{org}, y^{tgt}) - \lambda L_{nll}(s^{noisy}, y^{pred}) \quad (5.5)$$

where  $s^{org}$ ,  $s^{noisy}$  is the original and noisy sentences respectively,  $y^{tgt}$  denotes the *ground truth* translation for  $s^{org}$ ,  $y^{pred}$  is the predicted translation of  $s^{org}$  by the NMT system with its current set of parameters, and  $\lambda$  is a hyperparameter. In order to efficiently train the NMT system, we use greedy sampling instead of beam search to obtain  $y^{pred}$ . Note that  $y^{pred}$  is a *dynamic ground truth* in the sense that it will change continuously as the parameters of NMT system are updated. In this strategy, we truncate the noisy loss,  $L_{nll}(s^{noisy}, y^{pred})$ , if its value exceeds a certain value, denoted as *clip*. In such a case, the second term of equation 5.5 does not contribute to the back-propagation computation. During training, we minimize the loss function,  $L_{tot}$ . This essentially minimizes the clean loss and maximizes the noisy loss simultaneously.

**b) Learn to Ignore:** In *learn to ignore*, our goal is to teach NMT system to predict a dummy sentence in the target language for a noisy source sentence. To this end, after  $ep_{freeze}$  epochs, the loss function,  $L_{tot}$ , for this strategy is given as

$$L_{tot} = L_{nll}(s^{org}, y^{tgt}) + \lambda L_{nll}(s^{noisy}, y^{dmy}) \quad (5.6)$$

where  $y^{dmy}$  denotes the dummy sentence. For this work,  $y^{dmy}$  is set to “*Cette phrase n’est pas correct.*” for English-French and “*Dieser Satz ist nicht korrekt.*” for English-German. Both the dummy sentences translate to “*This sentence is not*

*correct.*” in English. During training, we minimize the loss function,  $L_{tot}$ . This essentially minimizes both the clean and noisy losses simultaneously.

### 5.3 Implementation Details

To evaluate the effectiveness of the proposed defense strategies, we use the same set of 500 sentences as mentioned in §5.1.3. The values of the hyperparameters, pertaining to §5.2, are as follows:  $ep_{freeze} = 7$ ,  $\mu = 0.01$ ,  $\lambda = 0.01$ ,  $clip = 15$ , and  $k = 20$ . We set  $k$  to be 20 since a significantly large percentage of replacements that occur during the bruteforce attack on the two NMT systems are within top-20 (40.0% for en-de Transformer and 42.1% for en-fr Transformer). We observe that lowering the value of  $ep_{freeze}$  leads to drop in BLEU score whereas increasing it further reduces the effectiveness of the defense. We set  $clip$  to 15 since a high value of noisy loss signifies *bad-quality noisy samples*. Training is done using single GPU and bruteforce attack is done using 8 GPUs.

The GPU specification is 32 GB Tesla V100-SXM2. We train the NMT systems for a total of 40 epochs. After  $ep_{freeze}$  epochs, BLEU score is calculated on a small subset of validation data every 1,000 training iterations. Finally, the NMT system having the best BLEU score is chosen. Training the NMT system exclusively on the clean data takes approximately 8 hours whereas training using the proposed defense strategies takes around 12 hours. The bruteforce attack on 500 sentences takes approximately 52 hours. The code for the proposed attack is publicly available.<sup>3</sup>

### 5.4 Evaluation Metrics

This section describes two additional metrics used to evaluate the effectiveness of defense strategies.

**Coverage:** We defined the notion of a *pair* in § 5.2.1. During training, a pair is sampled according to the sampling strategy (Random/Tackle-bias). Coverage signifies the percentage of *pairs* that were covered during training. We hypothesize

<sup>3</sup><https://github.com/akshay107/nmt-defense>

that higher the coverage, more effective the defense strategy.

**Targeted Translation (TT):** This metric is for *learn to ignore* strategy. It signifies the percentage of cases for which the NMT system predicts the dummy sentence,  $y^{dummy}$ , for adversarial sentences obtained from bruteforce attack. Higher the TT, more effective the *learn to ignore* strategy.

## 5.5 Results

Tables 5.5 and 5.6 show the success rate, BLEU score on the TED test set, the mean and the median of the number of replacements (NOR) and the other metrics discussed in §5.4 for English-German (en-de) and English-French (en-fr) respectively. The term *Original* in the two tables refer to the Transformer model trained only on clean data (i.e., same as Table 5.3). Overall, it is evident from both the tables that the two strategies are successful in reducing the success rate and number of replacements of the bruteforce attack. In the following subsections, we draw comparison between the two defense strategies and analyse BLEU scores for the NMT systems trained with the two defense strategies. We also compare the two noise sampling approaches.

### 5.5.1 Learn to Deal vs. Learn to Ignore

From Tables 5.5 and 5.6, it can be seen that both the strategies, *learn to deal* and *learn to ignore*, are successful in reducing the success rate and number of replacements for bruteforce attack for both the language pairs. In terms of comparison between the two strategies, *learn to ignore* strategy comfortably outperforms *learn to deal* strategy. *Learn to deal* strategy, under optimum settings, reduces the success rate from 84.0% to 62.2% for English-German and from 84.6% to 73.8% for English-French. On the other hand, *Learn to ignore* strategy, under optimum settings, reduces the success rate from 84.0% to 27.2% for English-German and from 84.6% to 37.0% for English-French. Hence, it is indeed easier to teach NMT systems *what to do* rather than *what not to do*. The targeted translation (TT) for *learn to ignore* strategy is also significantly high. This shows that NMT systems have the ability to predict the dummy sentence,  $y^{dummy}$ , for noisy sentences obtained



<i>frac</i>	<i>Noise</i>	Strategy	Success Rate (in %)	NOR	Coverage (in %)	TT (in %)	BLEU
0.2	Random	LTD	66.4	0.76, 0.80	72.8	-	29.37
		LTI	28.2	0.73, 0.75	86.5	70.2	30.16
	Tackle-Bias	LTD	65.0	0.78, 0.83	88.1	-	29.62
		LTI	35.4	0.76, 0.79	95.2	64.0	28.67
0.3	Random	LTD	65.6	0.75, 0.79	71.0	-	29.48
		LTI	<b>27.2</b>	0.72, 0.75	86.9	<b>70.8</b>	29.86
	Tackle-Bias	LTD	62.2	0.75, 0.8	88.0	-	29.28
		LTI	36.6	0.70, 0.73	95.4	61.2	29.30
0.4	Random	LTD	65.2	0.74, 0.79	70.1	-	28.75
		LTI	31.6	0.68, 0.71	91.8	64.4	30.09
	Tackle-Bias	LTD	68.8	0.78, 0.83	89.3	-	29.55
		LTI	32.0	0.69, 0.73	94.2	65.0	30.15
0.5	Random	LTD	68.4	0.76, 0.81	66.2	-	29.69
		LTI	40.4	0.71, 0.75	87.8	51.0	<b>30.23</b>
	Tackle-Bias	LTD	66.6	0.74, 0.79	86.8	-	29.38
		LTI	27.8	0.70, 0.75	92.5	68.0	30.07
<b>Original</b>			84.0	0.81, 0.84	-	-	29.22

**Table 5.5:** Results for *learn to deal* (LTD) and *learn to ignore* (LTI) strategies for English-German. The lowest success rate, highest targeted translation (TT), and highest BLEU are marked in boldface.

<i>frac</i>	Noise	Strategy	Success Rate (in %)	NOR	Coverage (in %)	TT (in %)	BLEU
0.2	Random	LTD	78.4	0.74, 0.77	63.0	-	43.10
		LTI	<b>37.0</b>	0.70, 0.73	86.3	<b>61.6</b>	43.34
	Tackle-Bias	LTD	78.8	0.79, 0.83	90.0	-	42.75
		LTI	51.8	0.74, 0.77	94.9	46.8	42.10
0.3	Random	LTD	76.0	0.75, 0.79	61.2	-	43.11
		LTI	38.2	0.69, 0.73	86.5	58.0	43.29
	Tackle-Bias	LTD	73.8	0.74, 0.78	89.8	-	42.45
		LTI	47.6	0.71, 0.73	94.9	50.6	43.17
0.4	Random	LTD	74.6	0.72, 0.76	64.4	-	43.28
		LTI	38.4	0.68, 0.71	89.4	58.4	<b>43.64</b>
	Tackle-Bias	LTD	74.6	0.73, 0.77	88.2	-	42.82
		LTI	49.2	0.68, 0.72	95.0	47.0	43.14
0.5	Random	LTD	79.2	0.76, 0.80	56.6	-	43.45
		LTI	45.8	0.69, 0.71	92.7	50.0	43.04
	Tackle-Bias	LTD	76.0	0.75, 0.77	84.8	-	43.19
		LTI	40.8	0.69, 0.72	93.5	54.2	43.29
<b>Original</b>			84.6	0.82, 0.86	-	-	42.59

**Table 5.6:** Results for *learn to deal* (LTD) and *learn to ignore* (LTI) strategies for English-French. The lowest success rate, highest targeted translation (TT), and highest BLEU are marked in boldface.

from bruteforce attack. Based on this finding, we can conclude that *learn to ignore* is a better strategy than *learn to deal*.

### 5.5.2 BLEU score

From Tables 5.5 and 5.6, we can see that both the defense strategies also improve the BLEU score on the TED test set in comparison with the original Transformer model. *Learn to deal* strategy improves the BLEU score from 29.22 to 29.69 for English-German and from 42.59 to 43.45 for English-French. Similarly, *learn to ignore* strategy improves the BLEU score from 29.22 to 30.23 for English-German and from 42.59 to 43.64 for English-French. With regards to BLEU score also, we can see that *learn to ignore* performs better than *learn to deal*. The high BLEU score along with significantly high TT for *learn to ignore* shows that the NMT systems under this strategy can distinguish between clean and noisy sentences.

### 5.5.3 Random vs. Tackle-Bias

From Tables 5.5 and 5.6, we can see that Tackle-Bias has significantly larger coverage than random for different values of *frac* and across defense strategies. Contrary to our hypothesis mentioned in §5.4, larger coverage achieved by Tackle-Bias approach does not result in more robust NMT systems. To further analyse the importance of coverage with regards to success rate, we modify the bruteforce attack to ensure that  $rank(w_{adv} | w_{org}) > k$ . This constraint ensures that the pair  $(w_{org}, w_{adv})$  was not encountered while training the NMT systems using the proposed defense strategies. We refer to such pairs as unseen pairs. Table 5.7 shows the performance of the modified bruteforce attack for the different English-German and English-French NMT systems. As expected, the results clearly show that the modified bruteforce attack is less potent than the original bruteforce attack as reflected by the lower values of NOR.

Furthermore, from Table 5.7, we observe a drop in the success rate and number of replacements (NOR) for the models trained with the two defense strategies in comparison with the original models. This shows that the NMT systems trained with the two defense strategies are able to generalize to unseen pairs as well. The high values of targeted translation (TT) for *learn to ignore* strategy further attest to this generalization ability. The ability to generalize is crucial since it shows

<i>frac</i>	<i>Noise</i>	Strategy	Success Rate (in %)		NOR		TT (in %)	
			en-de	en-fr	en-de	en-fr	en-de	en-fr
0.2	Random	LTD	68.8	78.2	0.57, 0.60	0.55, 0.55	—	—
		LTI	31.4	36.6	0.54, 0.55	0.52, 0.52	63.0	50.6
	Tackle-Bias	LTD	66.0	78.6	0.57, 0.58	0.60, 0.62	—	—
		LTI	33.6	47.6	0.58, 0.59	0.57, 0.57	66.4	61.2
0.3	Random	LTD	66.0	72.2	0.56, 0.58	0.56, 0.57	—	—
		LTI	26.4	35.8	0.53, 0.53	0.51, 0.50	62.4	55.8
	Tackle-Bias	LTD	64.6	74.4	0.56, 0.59	0.54, 0.56	—	—
		LTI	35.4	42.8	0.52, 0.52	0.53, 0.53	70.6	58.6
0.4	Random	LTD	65.2	73.4	0.55, 0.57	0.53, 0.54	—	—
		LTI	31.6	38.4	0.50, 0.50	0.50, 0.50	66.6	47.6
	Tackle-Bias	LTD	66.2	76.4	0.57, 0.58	0.54, 0.55	—	—
		LTI	29.4	47.0	0.52, 0.53	0.52, 0.52	63.4	57.0
0.5	Random	LTD	69.0	78.6	0.56, 0.57	0.57, 0.58	—	—
		LTI	43.0	45.8	0.51, 0.50	0.51, 0.50	63.4	54.4
	Tackle-Bias	LTD	68.6	76.2	0.55, 0.55	0.55, 0.57	—	—
		LTI	29.8	40.0	0.52, 0.52	0.52, 0.51	45.6	48.2
<b>Original</b>			79.8	85.0	0.65, 0.67	0.66, 0.66	—	—

**Table 5.7:** Results for *learn to deal* (LTD) and *learn to ignore* (LTI) strategies on the modified brute-force attack.

that even if the adversary is aware of the value of  $k$ , circumventing the proposed defense strategy is not trivial.

## 5.6 Summary

In this chapter, we proposed two defense strategies, namely *learn to deal* and *learn to ignore*, to enhance the robustness of the state-of-the-art NMT systems to invariance-based attack. We choose bruteforce attack for experimentation owing to its high success rate and number of replacements. The results demonstrate that the NMT systems trained under the two strategies are significantly more robust to invariance-based attack. The results also show that the *learn to ignore* strategy drastically reduces the potency of bruteforce attack. This suggests that it is easier to teach deep learning systems *what to do* rather than *what not to do*. The NMT systems trained under the two strategies also achieve a higher BLEU score than the original system. This shows that the noisy loss acts as a regularizer for the NMT system.

## Chapter 6

# Generalizability of Brute-force Attack: A case-study on TQA and SciQ dataset

*An idea is always a generalization, and  
generalization is a property of thinking.  
To generalize means to think.*

Georg Wilhelm Friedrich Hegel

This chapter explores the generalizability of brute-force attack, introduced in Chapter 5, by presenting a case study on two multiple choice QA datasets, namely, Textbook Question Answering (TQA) [64] and SciQ [133]. The case study evaluates the robustness of proposed multiple choice QA systems to brute-force attack. The goal of these systems is to answer a multiple choice question based on a given article. To this end, the QA system firstly chooses the *most relevant paragraph* in the article for a particular question. The sentences in the selected paragraph, and the question-option tuple are then embedded using either convolutional neural network (CNN) or gated recurrent unit (GRU). Apart from brute-force attack, we also study whether these systems generalize to different types of interventions on the input paragraph.

The rest of this chapter is organized as follows. Section 6.1 describes the proposed multiple choice QA systems in detail and presents its results on the two datasets. Section 6.2 explains the brute-force attack algorithm as well as different types

of interventions which are considered to study the robustness of QA systems. Section 6.3 presents the results of the bruteforce attack and the intervention-based study. Finally, Section 6.4 summarizes the chapter.

## 6.1 Background

In this section, we explain the proposed CNN-based multiple choice QA system in detail. The GRU-based QA system, denoted as  $GRU_{bl}$ , is considered as a baseline system. The proposed multiple choice QA system is also able to deal with options like none of the above, all of the above, both (a) and (b) etc. We refer to such options as *forbidden options*. We evaluate the performance of QA systems on two datasets, namely, Textbook Question Answering (TQA) [64] and SciQ [133]. The rest of this section is organized as follows. Section 6.1.1 explains paragraph selection for a particular question. Section 6.1.2 describes the architecture of the proposed system. Section 6.1.3 describes the proposed strategy for dealing with forbidden options. Section 6.1.4 presents the implementation details regarding the proposed system. Finally, Section 6.1.5 presents the results of the multiple-choice QA system on TQA and SciQ dataset.

### 6.1.1 Choosing the most relevant paragraph

Given a question based on an article, usually a small portion of the article is needed to answer the concerned question. Hence, it is not fruitful to give the entire article as input to the neural network. To select the most relevant paragraph in the article, we take both the question and the options into consideration instead of taking just the question into account for the same. The rationale behind this approach is to get the most relevant paragraphs in cases where the question is very general in nature. For example, consider that the article is about “*carbon*” and the question is “Which of the following statements is true about carbon?”. In such a scenario, it is not possible to choose the most relevant paragraph by just considering the question. The question along with the options forms the initial query. The most relevant paragraph is chosen in three stages:

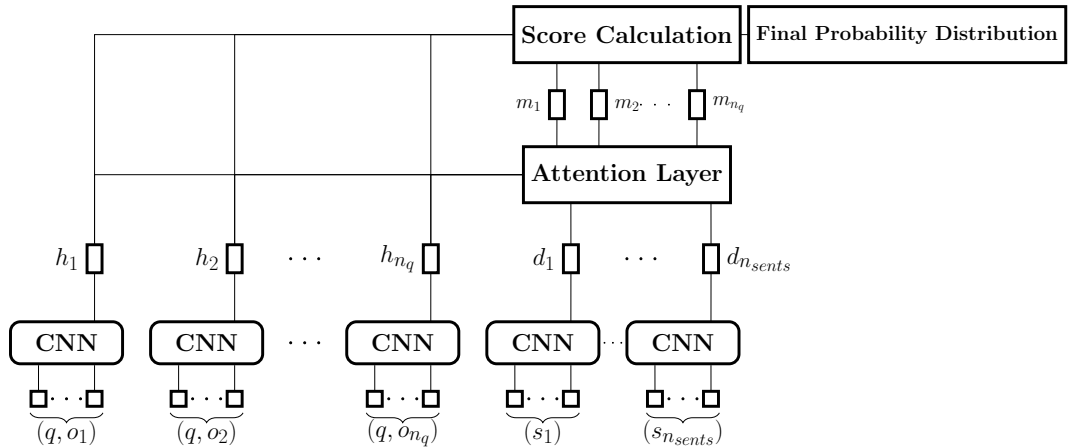
1. **Preprocessing the initial query:** In this stage, we remove the punctuation from the initial query string. Then, all the stop words are removed from the query.

Thereafter, all the words are replaced with their corresponding lemma. Stop word removal and lemmatization is done using the NLTK toolkit [78] in Python.

2. **Query Expansion using word2vec:** In this stage, we expand our initial query using word2vec [85]. More specifically, for each word in the initial query, its closest neighbours (i.e. words having a cosine similarity greater than 0.6) are appended to the initial query. This forms our final expanded query.

3. **Paragraph Ranking:** Using Lucene<sup>1</sup> and the final expanded query, we rank the paragraphs present in the article based on tf-idf scores. The paragraph with the highest tf-idf score is chosen as the most relevant paragraph for the concerned question.

### 6.1.2 Neural Network Architecture



**Figure 6.1:** Architecture of the proposed system. Attention layer attends on sentence embeddings  $d_j$ 's using question-option tuple embeddings  $h_i$ 's. Score Calculation layer calculates the cosine similarity between  $m_i$  and  $h_i$  which is passed through softmax to get the final probability distribution.

We use word2vec [85] to encode the words present in question, option and the most relevant paragraph. As a result, each word is assigned a fixed  $d$ -dimensional representation. The proposed system architecture is shown in Figure 6.1. Let  $q, o_i$  denote the word embeddings of words present in the question and the  $i^{\text{th}}$  option respectively. Thus,  $q \in \mathbb{R}^{d \times l_q}$  and  $o_i \in \mathbb{R}^{d \times l_o}$  where  $l_q$  and  $l_o$  represent the number of words in the question and option respectively. The question-option tuple  $(q, o_i)$  is embedded using CNN consisting of a convolutional layer followed by average

<sup>1</sup>[https://lucene.apache.org/core/3\\_4\\_0/scoring.html](https://lucene.apache.org/core/3_4_0/scoring.html)



pooling. The convolution layer has three types of filters of sizes  $f_j \times d \forall j = 1, 2, 3$  with size of output channel of  $k$ . Each filter type  $j$  produces a feature map of shape  $(l_q + l_o - f_j + 1) \times k$  which is average pooled to generate a  $k$ -dimensional vector. The three  $k$ -dimensional vectors are concatenated to form  $3k$ -dimensional vector. We use average pooling to ensure different embedding for different question-option tuples. Hence,

$$h_i = CNN([q; o_i]) \quad \forall i = 1, 2, \dots, n_q \quad (6.1)$$

where  $n_q$  is the number of options,  $h_i$  is the output of CNN and  $[q; o_i]$  denotes the concatenation of  $q$  and  $o_i$  i.e.  $[q; o_i] \in \mathbb{R}^{d \times (l_q + l_o)}$ . The sentences in the most relevant paragraph are embedded using the same CNN. Let  $s_j$  denote the word embeddings of words present in the  $j^{\text{th}}$  sentence i.e.  $s_j \in \mathbb{R}^{d \times l_s}$  where  $l_s$  is the number of words in the sentence. Then,

$$d_j = CNN(s_j) \quad \forall j = 1, 2, \dots, n_{sents} \quad (6.2)$$

where  $n_{sents}$  is the number of sentences in the most relevant paragraph and  $d_j$  is the output of CNN. The rationale behind using the same CNN for embedding question-option tuple and sentences in the most relevant paragraph is to ensure similar embeddings for similar question-option tuple and sentences. Next, we use  $h_i$  to attend on the sentence embeddings. Formally,

$$a_{ij} = \frac{h_i \cdot d_j}{\|h_i\| \cdot \|d_j\|} \quad (6.3)$$

$$r_{ij} = \frac{\exp(a_{ij})}{\sum_{j=1}^{n_{sents}} \exp(a_{ij})} \quad (6.4)$$

$$m_i = \sum_{j=1}^{n_{sents}} r_{ij} d_j \quad (6.5)$$

where  $\|\cdot\|$  signifies the  $\ell_2$ -norm,  $\exp(x) = e^x$  and  $h_i \cdot d_j$  is the dot product between the two vectors. Since  $a_{ij}$  is the cosine similarity between  $h_i$  and  $d_j$ , the attention weights  $r_{ij}$  give more weighting to those sentences which are more relevant to the question. The attended vector  $m_i$  can be thought of as the *evidence* in favor of the

$i^{\text{th}}$  option. Hence, to give a score to the  $i^{\text{th}}$  option, we take the cosine similarity between  $h_i$  and  $m_i$  i.e.

$$\text{score}_i = \frac{h_i \cdot m_i}{\|h_i\| \cdot \|m_i\|} \quad (6.6)$$

Finally, the scores are normalized using softmax to get the final probability distribution.

$$p_i = \frac{\exp(\text{score}_i)}{\sum_{i=1}^{n_q} \exp(\text{score}_i)} \quad (6.7)$$

where  $p_i$  denotes the probability for the  $i^{\text{th}}$  option.

### 6.1.3 Dealing with forbidden options

As mentioned earlier, we refer to options like none of the above, two of the above, all of the above, both (a) and (b) as *forbidden options*. During training, the questions having a forbidden option as the correct option are not considered. Furthermore, if a question has a forbidden option, that particular question-option tuple is not taken into consideration. Let  $S = [\text{score}_i \forall i \mid i^{\text{th}} \text{ option not in forbidden options}]$  and  $|S| = k$ . During prediction, the questions having one of the forbidden options as an option are dealt with as follows:

1. **Questions with none of the above/ all of the above option:** If the  $\max(S) - \min(S) < \text{threshold}$  then the final option is the concerned forbidden option. Else, the final option is  $\text{argmax}(p_i)$ .
2. **Questions with two of the above option:** If the  $S_{(k)} - S_{(k-1)} < \text{threshold}$  where  $S_{(n)}$  denotes the  $n^{\text{th}}$  order statistic, then the final option is the concerned forbidden option. Else, the final option is  $\text{argmax}(p_i)$ .
3. **Questions with both (a) and (b) type option:** For these type of questions, let the corresponding scores for the two options be  $\text{score}_{i_1}$  and  $\text{score}_{i_2}$ . If the  $|\text{score}_{i_1} - \text{score}_{i_2}| < \text{threshold}$  then the final option is the concerned forbidden option. Else, the final option is  $\text{argmax}(p_i)$ .
4. **Questions with any of the above option:** Very few questions had this option. In this case, we always choose the concerned forbidden option.

Model	True-False (Correct/Total)	Multiple Choice (Correct/Total)
$GRU_{bl}$	536/994 (53.9%)	529/1530 (34.6%)
$CNN_{3,4,5}$	531/994 (52.4%)	531/1530 (34.7%)
$CNN_{2,3,4}$	537/994 (54.0%)	543/1530 (35.5%)

**Table 6.1:** Accuracy for true-false and multiple choice questions on validation set of TQA dataset.

We try different *threshold* values ranging from 0.0 to 1.0. Finally, the *threshold* is set to a value which gives the highest accuracy on the training set for these kind of questions.

#### 6.1.4 Implementation Details

We try two different CNN systems, one having  $f_j$ 's equal to 3,4,5 and other having  $f_j$ 's equal to 2,3,4. We refer to the two systems as  $CNN_{3,4,5}$  and  $CNN_{2,3,4}$  respectively. The values of hyperparameters are as follows:  $d = 300, k = 100, n_{sents} = 10$ . For baseline system, we replace CNN with Gated Recurrent Unit (GRU) [23] to embed question-option tuples and the sentences. The size of GRU cell is set to 100. The baseline system is denoted as  $GRU_{bl}$ . For TQA dataset, the value of *threshold* is 0.3, as per Section 6.1.3. The SciQ dataset does not contain any question with forbidden option. Every question in SciQ dataset contains 4 options, whereas the number of options vary from 2 to 7 in TQA dataset. Hence, the multiple-choice QA system generates the probability distribution over the set of available options. Similarly, the number of sentences in the most relevant paragraph can vary from question to question, so we set  $a_{ij} = -\infty$  whenever  $d_j$  is a zero vector. The standard cross-entropy loss function is minimized during training. The code for the proposed system is publicly available.<sup>2</sup>

#### 6.1.5 Results

Tables 6.1 and 6.2 show the accuracy of the proposed system on the validation set of TQA and SciQ dataset respectively. For SciQ dataset, we use the associated passage provided with the question as the *most relevant paragraph*. AS Reader [60]

<sup>2</sup><https://github.com/akshay107/CNN-QA/>

Model	Accuracy
$GRU_{bl}$	68.2%
$CNN_{3,4,5}$	87.1%
$CNN_{2,3,4}$	87.8%
$CNN_{2,3,4}$	84.7% (test-set)

**Table 6.2:** Accuracy of the QA systems on SciQ dataset. The first three accuracies are on validation set. The last accuracy is of  $CNN_{2,3,4}$  on the test set.

which encodes the question and the paragraph using GRU followed by attention mechanism achieved 74.1% accuracy on the SciQ test set. However, for a question, they used a different corpus to extract the text passage. Hence it is not judicious to compare the two systems. As can be seen from the Tables 6.1 and 6.2,  $CNN_{2,3,4}$  gives the best performance on the validation set of both the datasets. Note that  $GRU_{bl}$  highly overfits on the SciQ dataset which shows that CNN-based systems work better for datasets where long-term dependency is not a major concern. This rationale is also supported by the fact that  $CNN_{2,3,4}$  performs better than  $CNN_{3,4,5}$  on the two datasets.

**Baselines for TQA dataset:** Three baseline systems are mentioned in Kembhavi et al. [64]. These baseline systems rely on word-level attention and encoding question and options separately. The baseline systems are random model, Text-Only model and BiDAF [114]. Text-Only model is a variant of Memory network [134] where the paragraph, question and options are embedded separately using LSTM followed by attention mechanism. In BiDAF, character and word level embedding is used to encode the question and paragraph followed by bidirectional attention mechanism. This system predicts a span within the paragraph containing the answer. Hence, the predicted span is compared with each of the options to select the final option.

Note that the result of the baseline systems given in Kembhavi et al. [64] were on test set but the authors had used a different data split than the publicly released split. As per the suggestion of the authors, we evaluate  $CNN_{2,3,4}$  by combining validation and test set. The comparison with the baseline systems is given in Table 6.3. As can be seen from Table 6.3,  $CNN_{2,3,4}$  shows significant improvement over the baseline systems. We argue that our proposed system outperforms the

Model	True-False	Multiple Choice
Random*	50.0	22.7
Text-Only*	50.2	32.9
BiDAF*	50.4	32.2
$CNN_{2,3,4}$	<b>53.7</b>	<b>35.8</b>

**Table 6.3:** Accuracy of different systems for true-false and multiple choice questions. Results marked with (\*) are taken from Kembhavi et al. [64] and are on test set obtained using a different data split. Result of our proposed system is on publicly released validation and test set combined.

Text-Only model because of three reasons (i) sentence level attention, (ii) question-option tuple as input, and (iii) ability to tackle forbidden options. Sentence level attention leads to better attention weights, especially in cases where a single sentence suffices to answer the question. Furthermore, if question is given as input to the system, then it has to extract the embedding of the answer whereas giving question-option tuple as input simplifies the task to comparison between the two embeddings.

As mentioned earlier, SciQ dataset does not have any questions with forbidden options. The validation set of TQA has 433 questions with forbidden options. Using the proposed threshold strategy for tackling forbidden options,  $CNN_{2,3,4}$  gets 188 out of 433 questions correct. Without using this strategy and giving every question-option tuple as input,  $CNN_{2,3,4}$  gets 109 out of 433 questions correct.

## 6.2 Bruteforce Attack and Types of Intervention

As we can see from Table 6.1, the performance of the three multiple-choice QA systems is quite low on the TQA dataset. This is because several instances of TQA dataset require multi-hop reasoning across paragraphs in order to answer a question correctly. Hence, to evaluate the robustness of these multiple-choice QA systems, we perform experiments on SciQ dataset. We randomly pick 100 samples from SciQ validation set where all the three multiple-choice QA systems gave the correct prediction. For these samples, we manually annotate the portion of the paragraph responsible for the answer (i.e. *rationale*). Table 6.4 shows one such example where the rationale is marked in blue. Note that the second mention of “barrier island” in the paragraph of Table 6.4 is not responsible for the answer.

<b>Paragraph</b>	A <b>barrier island</b> is a long strip of sand. The sand naturally moves in the local currents. People try to build on barrier islands.
<b>Question</b>	A long strip of sand is referred to as what?
<b>Options</b>	(a) a volcano (b) a component island (c) a composition island (d) a barrier island

**Table 6.4:** Example from SciQ validation set. We manually annotate the portion of paragraph responsible for the answer (shown in blue).

After annotating the rationales, we run the bruteforce attack with the goal of making as many replacements as possible within the rationale, keeping the rest of the paragraph unchanged. Apart from this constraint, the attack algorithm is similar to Algorithm 5.1. We also evaluate the robustness of multiple-choice QA systems by performing interventions on the input paragraph. The intervention-based study allows us to evaluate the ability of QA systems to deduce *logical consequence*. We experiment with the following two types of interventions:

1. **Mask intervention:** In mask intervention, we remove the rationale from the input paragraph. For example, in Table 6.4, mask intervention will change the sentence “A barrier island is a long strip of sand” to “is a long strip of sand”. Since in this scenario, there is no evidence for any of the options, we expect the QA system to predict an option randomly. We refer to this type of intervention as *Mask*.
2. **Option-specific intervention:** In option-specific intervention, we replace the rationale with an incorrect option in the input paragraph. For example, in Table 6.4, option-specific intervention will change the sentence “A barrier island is a long strip of sand” to “A volcano is a long strip of sand”. In this scenario, we expect the QA system to change its prediction to the corresponding option. Note that option (d) is always the correct option for SciQ dataset. Hence, we refer to this type of intervention as *Option A/B/C* depending on which incorrect option is used for replacement.

## 6.3 Results

Table 6.5 shows the success rate and percentage of replacements of bruteforce attack against the three multiple choice QA systems. From the table, we can see

Model	Success Rate	Replacement %
$GRU_{bl}$	99.0%	133/142 (93.7%)
$CNN_{3,4,5}$	100.0%	136/142 (95.8%)
$CNN_{2,3,4}$	99.0%	135/142 (95.1%)

**Table 6.5:** Success Rate of Brute-force-Attack

Source \ Target	$GRU_{bl}$	$CNN_{3,4,5}$	$CNN_{2,3,4}$
$GRU_{bl}$	-	82.0%	76.0%
$CNN_{3,4,5}$	89.0%	-	89.0%
$CNN_{2,3,4}$	83.0%	94.0%	-

**Table 6.6:** Transferability of Brute-force-Attack. The adversarial example obtained for the Source QA system is given as input to the Target QA system.

that brute-force attack achieves very high success rate for all the three QA systems. Furthermore, the percentage of replacement is also significantly high. This clearly showcases that the brute-force attack is able to generalize to multiple choice QA systems as well. Furthermore, Table 6.6 shows transferability of brute-force attack across QA systems. Unlike NMT systems, we observe that the adversarial examples obtained via brute-force attack have high transferability across QA systems.

Table 6.7 shows the prediction count for all the options across QA systems and types of intervention. For *mask intervention*, all the three systems still predict the same option, i.e. option (d), in majority of cases. This shows that when the annotated rational is masked, the QA systems mostly rely on superficial cues (such as attending on second mention of “barrier island” for Table 6.4) for prediction. This reliance on superficial cues along with the fact that the three QA systems share the same embedding matrix also explains the high transferability of brute-force attack observed in Table 6.6. For *option-specific intervention*, we observe that the prediction count for the desired option is significantly higher in CNN-based QA systems than  $GRU_{bl}$ . Hence, CNN-based QA systems generalize better to *option-specific intervention*. This finding can be attributed to the fact that, unlike  $GRU_{bl}$ , the proposed CNN-based QA systems rely on *n-gram* features and hence are more sensitive to minor changes in the input paragraph.

System	Intervention Type	Prediction Count
$GRU_{bl}$	Mask	12, 6, 9, 73
	Option A	<b>42</b> , 1, 1, 56
	Option B	1, <b>44</b> , 0, 55
	Option C	1, 2, <b>43</b> , 54
$CNN_{2,3,4}$	Mask	8, 13, 8, 71
	Option A	<b>57</b> , 2, 2, 39
	Option B	0, <b>60</b> , 1, 39
	Option C	2, 1, <b>61</b> , 36
$CNN_{3,4,5}$	Mask	11, 10, 9, 70
	Option A	<b>55</b> , 1, 2, 42
	Option B	0, <b>53</b> , 4, 43
	Option C	1, 0, <b>61</b> , 38

**Table 6.7:** Results for mask and option-specific interventions. Prediction count shows the number of times each of the option is predicted by the QA system. For *option-specific intervention*, the prediction count of the *desired option* is marked in bold.

## 6.4 Summary

This chapter explored the robustness of multiple-choice QA systems to brute-force attack and two types of input intervention, namely, mask intervention and option-specific intervention. These interventions were performed on the annotated rationale of the input paragraph. The results showed that the brute-force attack achieves high success rate along with high percentage of replacements for all the QA systems. We also observed high transferability of adversarial examples across the three QA systems. For intervention-based study, we observed that the QA systems do not generalize well to mask intervention. However, for option-specific intervention, CNN-based systems generalize better than their GRU counterpart.



# Chapter 7

## Conclusion

*Amplify, clarify, and punctuate, and let the viewer draw his or her own conclusion.*

Keith Jackson

The goal of this thesis was to study the adversarial robustness of deep learning systems. In this thesis, we analysed the robustness of several state-of-the-art deep learning systems across various NLP and vision tasks. In Chapter 2, we looked at the robustness of VQA systems to adversarial background noise. The results showed that by adding minimal background noise to the image, these systems can be fooled both in the *same-category* and *different-category* setting. This holds true for toy datasets, where VQA systems have very high accuracy, as well as real-world dataset, where a very tiny fraction of the image is modified during the attack. However, as shown in Chapter 2, category-specific answer modules significantly enhance the robustness of N2NMN against *different-category* attack. In Chapter 3, we proposed a *task-agnostic* attack, named *Mimic and Fool*, against vision systems. Mimic and Fool relies on the idea that if two images are *indistinguishable* for the feature extractor then they will be *indistinguishable* for the model as well. Keeping this in mind, Mimic and Fool solely relies on fooling the *feature extractor* to fool the underlying vision system. Both the attacks in Chapters 2 and 3 can be considered as adversarial attacks in a *constrained setting*. While the adversarial attack in Chapter 2 only modifies the image background, Mimic and Fool is a gray-box attack as it only requires the *knowledge* of the feature extractor of the vision system. Even in a *constrained setting*, these attacks achieve high success rate against deep learning-based vision systems.

In Chapter 4, we explored the robustness of NMT systems to invariance-based adversarial attack. In a low-resource setting, we observed that the NMT systems are unable to capture the semantics as they predict the same translation for two completely different source sentences. However, in a high-resource setting, NMT systems are significantly more robust to the invariance-based attack. In Chapter 5, we explored two defense strategies to counter bruteforce attack: *learn to deal* and *learn to ignore*. The results showed that *learn to ignore* strategy is able to significantly reduce the effectiveness of bruteforce attack. In Chapter 6, we explored the generalizability of the bruteforce attack to multiple-choice QA systems. We observed that the bruteforce attack achieves a very high success rate for both CNN and GRU-based multiple-choice QA systems. In this chapter, we also explored the ability ability of QA systems to deduce *logical consequence* by performing two types of interventions on the input paragraph, namely, mask intervention and option-specific intervention. The low generalizability of QA systems to mask intervention showcased that, in absence of evidence, these system rely on superficial cues for answering a question. However, for option-specific intervention, we observe that CNN-based multiple-choice QA systems generalize better than their GRU counterpart.

Overall, the findings of this thesis show that state-of-the-art deep learning systems across NLP and vision lack adversarial robustness. Deep learning-based vision systems were shown to be vulnerable to adversarial attacks, even in a constrained setting. *Mimic and Fool* showcases the drawbacks of commonly-used feature extractors in deep learning-based vision systems. For NLP systems, we took an orthogonal approach from previous works for designing adversarial attacks. We designed invariance-based adversarial attacks which make multiple changes to the input sentence with the goal of keeping the prediction unchanged. The invariance-based adversarial attack was shown to be effective against NMT system in a low-resource setting. However, in a high-resource setting, the adversarial robustness of the NMT system against such attacks is significantly enhanced. This shows that NMT systems have more adversarial robustness in comparison to deep learning-based vision systems.

The future scope of this thesis is as follows:

1. As shown in Chapter 2, VQA systems are vulnerable to adversarial background noise. Recent VQA systems rely on bottom-up features obtained

from Faster R-CNN [2, 148] to capture information about an image. Robustness of such systems to adversarial background noise can be explored in future. However, given previous works on adversarial attacks against object detectors, it is highly unlikely that such VQA systems are resilient to adversarial background noise. Hence, for future work, one can design VQA systems which can effectively learn to ignore the background noise especially for toy-datasets where background is easily identifiable.

2. In Chapter 3, the proposed adversarial attack, Mimic and Fool, showcases the limitation of current feature extractors which are widely used in deep learning-based vision systems. Hence, a possible future work will be to develop feature extractors which are robust to Mimic and Fool and, at the same time, lead to vision systems which are at par with current systems in terms of performance. Another possible scope of future work, from an attack perspective, is to design task-agnostic adversarial attacks which requires access to only the pretrained weights (instead of fine-tuned weights as is the case with Mimic and Fool) of the feature extractor.
3. Chapter 4, as mentioned earlier, showed that NMT systems are vulnerable to invariance-based adversarial attacks, especially in a low-resource setting. We also discussed several metrics to evaluate the efficiency of the proposed attack. To build trustworthy MT systems, it is important to benchmark progress of NMT systems not only on BLEU score on test set but also on such robustness metrics.
4. In Chapter 5, we explored several defense strategies to tackle invariance-based attack on NMT systems. Different from prior works where minimal changes are made to the source sentence to change the predicted translation, invariance-based attack makes multiple changes to the source sentence with the goal of keeping the predicted translation unchanged. Hence, a possible future work would be to design NMT systems which are robust to invariance-based attacks as well as prior attacks.
5. Chapter 6 showed that, on SciQ dataset, CNN-based multiple-choice QA systems are better at generalizing to option-specific interventions in comparison to GRU-based systems. However, in absolute terms, this generalizability for both CNN and GRU based multiple-choice QA systems is quite low. The low generalizability showcases the inability of multiple-choice QA

systems to learn logical consequences. Current QA systems in NLP rely on finetuning Transformer-based architectures such as BERT [30], XLNet [146] etc. Robustness of such systems to invariance-based attacks as well as different types of interventions can be explored in future. In the literature, intervention-based techniques have been extensively studied to learn causal structures [104]. Keeping this in mind, a possible future work is to explore training paradigms which use the notion of intervention to teach logical consequences to QA systems. Such a training paradigm can later be extended to other NLP and vision tasks as well.

6. While the thesis work highlighted the lack of robustness of current deep learning systems, adversarial attacks can also be used in a positive setting such as for developing privacy preserving applications [86]. However, the AI community should be aware of the ethical issues in regard to use of adversarial attacks. For instance, selective bias (such as racial or gender bias) can be injected in a system through poisoning attacks.

# Publication related to this thesis

## Journal

- A. Chaturvedi, and U. Garain: *Attacking VQA Systems via Adversarial Background Noise*, IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 4, pp. 490-499, August 2020.
- A. Chaturvedi, and U. Garain: *Mimic and Fool: A Task Agnostic Adversarial Attack*, IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 4, pp. 1801-1808, April 2021.
- A. Chaturvedi, A. Chakrabarty, M. Utiyama, E. Sumita, and U. Garain: *Ignorance is Bliss: Exploring Defenses Against Invariance-based Attacks on Neural Machine Translation Systems*, IEEE Transactions on Artificial Intelligence, doi: 10.1109/TAI.2021.3123931.

## Conference

- A. Chaturvedi, O. Pandit, and U. Garain: *CNN for Text-Based Multiple Choice Question Answering*, Association for Computational Linguistics (ACL), 2018 pp. 272-277.

## Preprint

- A. Chaturvedi, Abijith KP, and U. Garain: *Exploring the Robustness of NMT Systems to Nonsensical Inputs*, arXiv:1908.01165v3.

# Bibliography

- [1] Akhtar, N., Liu, J., and Mian, A. (2018). Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [2] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [3] Andriushchenko, M. and Flammarion, N. (2020). Understanding and improving fast adversarial training. In *Advances in Neural Information Processing Systems*.
- [4] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.
- [5] Athalye, A., Carlini, N., and Wagner, D. (2018a). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmässan, Stockholm Sweden. PMLR.
- [6] Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. (2018b). Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293, Stockholmsmässan, Stockholm Sweden. PMLR.
- [7] Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. (2010). The security of machine learning. *Machine Learning*, 81(2):121–148.

- 
- [8] Belinkov, Y. and Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.
- [9] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer.
- [10] Biggio, B. and Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317 – 331.
- [11] Blohm, M., Jagfeld, G., Sood, E., Yu, X., and Vu, N. T. (2018). Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 108–118, Brussels, Belgium. Association for Computational Linguistics.
- [12] Brendel, W., Rauber, J., and Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*.
- [13] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.
- [14] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. (2019). On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.
- [15] Carlini, N. and Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec '17*, page 3–14, New York, NY, USA. Association for Computing Machinery.
- [16] Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57.
- [17] Chen, H., Zhang, H., Chen, P.-Y., Yi, J., and Hsieh, C.-J. (2018a). Attacking visual language grounding with adversarial examples: A case study on neural

- image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2587–2597, Melbourne, Australia. Association for Computational Linguistics.
- [18] Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec '17*, page 15–26, New York, NY, USA. Association for Computing Machinery.
- [19] Chen, S.-T., Cornelius, C., Martin, J., and Chau, D. H. P. (2018b). Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer.
- [20] Cheng, M., Yi, J., Chen, P.-Y., Zhang, H., and Hsieh, C.-J. (2020). Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3601–3608.
- [21] Cheng, Y., Jiang, L., and Macherey, W. (2019). Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.
- [22] Cheng, Y., Tu, Z., Meng, F., Zhai, J., and Liu, Y. (2018). Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.
- [23] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- [24] Chollet, F. et al. (2015). Keras. <https://keras.io>.
- [25] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.



- [26] Croce, F. and Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR.
- [27] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, page 886–893, USA. IEEE Computer Society.
- [28] Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D. (2004). Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, page 99–108, New York, NY, USA. Association for Computing Machinery.
- [29] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- [30] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- [31] Ebrahimi, J., Lowd, D., and Dou, D. (2018a). On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- [32] Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. (2018b). HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- [33] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramèr, F., Prakash, A., Kohno, T., and Song, D. (2018). Physical adversarial examples for

- object detectors. In *Proceedings of the 12th USENIX Conference on Offensive Technologies*, WOOT'18, page 1, USA. USENIX Association.
- [34] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634.
- [35] Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. (2017). Detecting Adversarial Samples from Artifacts. In *arXiv preprint arXiv:1703.00410*.
- [36] Feng, S., Wallace, E., Grissom II, A., Iyyer, M., Rodriguez, P., and Boyd-Graber, J. (2018). Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.
- [37] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.
- [38] Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D., and Dahl, G. E. (2018). Motivating the rules of the game for adversarial example research. In *arXiv preprint arXiv:1807.06732*.
- [39] Golan, I. and El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 9758–9769. Curran Associates, Inc.
- [40] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [41] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.

- [42] Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. (2019). Scalable verified training for provably robust image classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4841–4850.
- [43] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [44] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.
- [45] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.
- [46] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- [47] He, T. and Glass, J. (2019). Detecting egregious responses in neural sequence-to-sequence models. In *International Conference on Learning Representations*.
- [48] He, W., Wei, J., Chen, X., Carlini, N., and Song, D. (2017). Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC. USENIX Association.
- [49] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.
- [50] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [51] Hu, R., Andreas, J., Rohrbach, M., Darrell, T., and Saenko, K. (2017). Learning to reason: End-to-end module networks for visual question answering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 804–813.

- [52] Hudson, D. A. and Manning, C. D. (2018). Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*.
- [53] Ilyas, A., Engstrom, L., and Madry, A. (2019). Prior convictions: Black-box adversarial attacks with bandits and priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- [54] Jacobsen, J.-H., Smeulders, A. W. M., and Oyallon, E. (2018). i-revnet: Deep invertible networks. In *International Conference on Learning Representations (ICLR)*.
- [55] Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.
- [56] Jia, Y., Lu, Y., Shen, J., Chen, Q. A., Chen, H., Zhong, Z., and Wei, T. (2020). Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations*.
- [57] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997. IEEE Computer Society.
- [58] Johnson, J., Karpathy, A., and Fei-Fei, L. (2016). Densecap: Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574.
- [59] Joseph, A. D., Nelson, B., Rubinstein, B. I. P., and Tygar, J. D. (2019). *Adversarial Machine Learning*. Cambridge University Press.
- [60] Kadlec, R., Schmid, M., Bajgar, O., and Kleindienst, J. (2016). Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

- [61] Kafle, K. and Kanan, C. (2016). Answer-type prediction for visual question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4976–4984.
- [62] Kannan, H., Kurakin, A., and Goodfellow, I. (2018). Adversarial logit pairing. *arXiv preprint arXiv: 1803.06373*.
- [63] Karmon, D., Zoran, D., and Goldberg, Y. (2018). LaVAN: Localized and visible adversarial noise. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2507–2515, Stockholmsmässan, Stockholm Sweden. PMLR.
- [64] Kembhavi, A., Seo, M., Schwenk, D., Choi, J., Farhadi, A., and Hajishirzi, H. (2017). Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [65] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- [66] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- [67] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.
- [68] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.
- [69] Kurakin, A., Goodfellow, I., and Bengio, S. (2017a). Adversarial examples in the physical world. *ICLR Workshop*.

- [70] Kurakin, A., Goodfellow, I. J., and Bengio, S. (2017b). Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- [71] Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., and Hubbard, W. (1989). Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46.
- [72] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [73] Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672.
- [74] Liang, B., Li, H., Su, M., Bian, P., Li, X., and Shi, W. (2018). Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4208–4215. International Joint Conferences on Artificial Intelligence Organization.
- [75] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1–3):503–528.
- [76] Liu, H., Ma, M., Huang, L., Xiong, H., and He, Z. (2019). Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.
- [77] Liu, Y., Chen, X., Liu, C., and Song, D. (2017). Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- [78] Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Philadelphia: Association for Computational Linguistics.

- [79] Lowd, D. and Meek, C. (2005). Good word attacks on statistical spam filters. In *In Proceedings of the Second Conference on Email and Anti-Spam (CEAS)*.
- [80] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.
- [81] Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.
- [82] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- [83] Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196.
- [84] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. (2017). On detecting adversarial perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.
- [85] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.
- [86] Mirjalili, V., Raschka, S., and Ross, A. (2020). Privacynet: Semi-adversarial networks for multi-attribute face privacy. *IEEE Transactions on Image Processing*, 29:9400–9412.
- [87] Mohseni, S., Pitale, M., Yadawa, J., and Wang, Z. (2020). Self-supervised learning for generalizable out-of-distribution detection. In *AAAI 2020*.
- [88] Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94.

- [89] Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2016). Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582.
- [90] Mosbach, M., Andriushchenko, M., Trost, T., Hein, M., and Klakow, D. (2018). Logit pairing methods can fool gradient-based attacks. In *NeurIPS 2018 Workshop on Security in Machine Learning*.
- [91] Nelson, B., Barreno, M., Chi, F. J., Joseph, A. D., Rubinstein, B. I. P., Saini, U., Sutton, C., Tygar, J. D., and Xia, K. (2008). Exploiting machine learning to subvert your spam filter. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, LEET'08*, USA. USENIX Association.
- [92] Ng, N., Yee, K., Baeviski, A., Ott, M., Auli, M., and Edunov, S. (2019). Facebook FAIR’s WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.
- [93] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436.
- [94] Oord, A. V., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA. PMLR.
- [95] Ott, M., Edunov, S., Baeviski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- [96] Ott, M., Edunov, S., Grangier, D., and Auli, M. (2018). Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*.
- [97] Papernot, N., McDaniel, P., and Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.



- [98] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017a). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17*, page 506–519, New York, NY, USA. Association for Computing Machinery.
- [99] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016a). The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387.
- [100] Papernot, N., McDaniel, P., Swami, A., and Harang, R. (2016b). Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 49–54.
- [101] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016c). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597.
- [102] Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. (2017b). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM.
- [103] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
- [104] Pearl, J. and Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition.
- [105] Qi, Y., Sachan, D., Felix, M., Padmanabhan, S., and Neubig, G. (2018). When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.
- [106] Raghunathan, A., Steinhardt, J., and Liang, P. (2018). Certified defenses against adversarial examples. In *International Conference on Learning Representations*.

- [107] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 91–99. Curran Associates, Inc.
- [108] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.
- [109] Rubinstein, B. I., Nelson, B., Huang, L., Joseph, A. D., Lau, S.-h., Rao, S., Taft, N., and Tygar, J. D. (2009). Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, IMC '09*, page 1–14, New York, NY, USA. Association for Computing Machinery.
- [110] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536.
- [111] Sachan, D. and Neubig, G. (2018). Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics.
- [112] Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pix-*elcnn++*: Improving the *pix-*elcnn** with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- [113] Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- [114] Seo, M. J., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.
- [115] Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. (2019). Adversarial training for free! In *Advances in Neural Information Processing Systems*, volume 32, pages 3358–3369. Curran Associates, Inc.

- [116] Sharma, S., El Asri, L., Schulz, H., and Zumer, J. (2017). Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.
- [117] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [118] Sinha, A., Namkoong, H., and Duchi, J. (2018). Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*.
- [119] Smutz, C. and Stavrou, A. (2012). Malicious pdf detection using metadata and structural features. In *Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC '12*, page 239–248, New York, NY, USA. Association for Computing Machinery.
- [120] Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. (2018). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- [121] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- [122] Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.
- [123] Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841.
- [124] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

- [125] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations*.
- [126] Uesato, J., O’Donoghue, B., Kohli, P., and van den Oord, A. (2018). Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5025–5034, Stockholmsmässan, Stockholm Sweden. PMLR.
- [127] Šrndić, N. and Laskov, P. (2014). Practical evasion of a learning-based classifier: A case study. In *2014 IEEE Symposium on Security and Privacy*, pages 197–211.
- [128] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.
- [129] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.
- [130] Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., and Willke, T. L. (2018). Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 560–574. Springer.
- [131] Wang, Y. and Bansal, M. (2018). Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana. Association for Computational Linguistics.
- [132] Wei, X., Liang, S., Chen, N., and Cao, X. (2019). Transferable adversarial attacks for image and video object detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 954–960. International Joint Conferences on Artificial Intelligence Organization.

- [133] Welbl, J., Liu, N. F., and Gardner, M. (2017). Crowdsourcing multiple choice science questions. *CoRR*, abs/1707.06209.
- [134] Weston, J., Chopra, S., and Bordes, A. (2014). Memory networks. *CoRR*, abs/1410.3916.
- [135] Williams, R. J. (1990). *Adaptive State Representation and Estimation Using Recurrent Connectionist Networks*, page 97–114. MIT Press, Cambridge, MA, USA.
- [136] Williams, R. J. and Zipser, D. (1992). Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Backpropagation: Theory, Architectures and Applications*. Erlbaum, Hillsdale, NJ.
- [137] Wittel, G. and Wu, S. (2004). On Attacking Statistical Spam Filters. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, USA.
- [138] Wong, E. and Kolter, J. Z. (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5283–5292. PMLR.
- [139] Wong, E., Rice, L., and Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.
- [140] Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *International Conference on Computer Vision*. IEEE.
- [141] Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K. (2019). Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [142] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of 32nd International Conference on Machine Learning (ICML)*, pages 2048–2057.

- [143] Xu, W., Evans, D., and Qi, Y. (2018a). Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.
- [144] Xu, X., Chen, X., Liu, C., Rohrbach, A., Darrell, T., and Song, D. (2018b). Fooling vision and language models despite localization and attention mechanism. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [145] Xu, Y., Wu, B., Shen, F., Fan, Y., Zhang, Y., Shen, H., and Liu, W. (2019). Exact adversarial attack to image captioning via structured output learning with latent variables. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4130–4139.
- [146] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- [147] Yu, C.-N. J. and Joachims, T. (2009). Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, page 1169–1176, New York, NY, USA. Association for Computing Machinery.
- [148] Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. (2019). Deep modular co-attention networks for visual question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283.
- [149] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.
- [150] Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. (2019a). You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems*, volume 32, pages 227–238. Curran Associates, Inc.
- [151] Zhang, H. and Wang, J. (2019). Towards adversarially robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

- 
- [152] Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. (2019b). Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482, Long Beach, California, USA. PMLR.
- [153] Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., and Chen, K. (2019). Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 1989–2004. ACM.
- [154] Zhao, Z., Dua, D., and Singh, S. (2018). Generating natural adversarial examples. In *International Conference on Learning Representations*.
- [155] Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- [156] Zou, W., Huang, S., Xie, J., Dai, X., and Chen, J. (2020). A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3486–3497, Online. Association for Computational Linguistics.