



Towards Robust Deep Neural Networks: Query Efficient Black-Box Adversarial Attacks and Defences

by

Viet Quoc Vo

M. Sc. (Electronic and Computer Engineering),
Royal Melbourne Institute of Technology, 2014

Dissertation submitted for the degree of

Doctor of Philosophy

in

School of Computer and Mathematical Sciences
Faculty of Sciences, Engineering, and Technology
The University of Adelaide

June 2023

Supervisors:

Associate Professor Damith Chinthana Ranasinghe ,
School of Computer and Mathematical Sciences,
The University of Adelaide

Doctor Ehsan Abbasnejad,
School of Computer and Mathematical Sciences,
The University of Adelaide

Contents

Contents	iii
Abstract	ix
Statement of Originality	xiii
Acknowledgements	xv
Dissertation Conventions	xvii
Acronyms	xix
Publications	xxi
List of Figures	xxiii
List of Tables	xxix
Chapter 1. Introduction	1
1.1 Introduction	2
1.2 Objectives	5
1.3 Challenges	5
1.4 Summary of Contributions	8
1.5 Dissertation Structure	11
Chapter 2. Background	15
2.1 Notations	16
2.2 Deep Neural Networks	16
2.3 Adversarial Attacks	19
2.4 Threat Models	20
2.4.1 Adversarial Capabilities	20

2.4.2	Similarity Measures	22
2.5	Adversarial Defense	23
2.6	Data sets	24
2.7	Evaluation Metrics	25
Chapter 3. RamBoAttack: A Dense Attack Under Decision-Based Settings		27
3.1	Motivation and Contribution	28
3.1.1	Chapter Overview	30
3.2	Investigation of Decision-Based Attacks	31
3.2.1	Adversarial Threat Model	31
3.2.2	Problem Formulation	31
3.2.3	Understanding Robustness	33
3.2.4	Observations from Assessing Attacks	34
3.2.5	An Intuition into Attack Methods	36
3.3	Proposed Attack Framework	37
3.3.1	Approach	39
3.3.2	BLOCKDESCENT	40
3.4	Experiments and Evaluations	44
3.4.1	Experiment Settings	44
3.4.2	Experimental Regime	45
3.4.3	Proposed Robustness Evaluation Protocol	47
3.4.4	Robustness of RAMBOATTACK	49
3.4.5	Attacking <i>Hard</i> Sets	49
3.4.6	Impact of Starting Images	54
3.4.7	Attack Insights	55
3.4.8	Attack Against Defense Mechanism	58
3.5	Conclusion	61
Chapter 4. SparseEvo: A Sparse Attack Under Decision-base Settings		63
4.1	Motivation and Contribution	64
4.1.1	Chapter Overview	66

4.2	Related Work on Sparse Attacks	66
4.3	Proposed Method	68
4.3.1	Problem Formulation	68
4.3.2	SPARSEEVO Attack Algorithm	69
4.4	Experiments and Evaluations	75
4.4.1	Experiment Settings	75
4.4.2	Experimental Regime	76
4.4.3	Attacks Against Convolutional Deep Neural Networks	77
4.4.4	Attacks Against a Vision Transformer	78
4.4.5	Attacks Against a CNN Model on the CIFAR10	79
4.4.6	Compare The Robustness of the Transformer and the CNN	80
4.4.7	Sparse Attacks Against an Adversarially Trained Model	83
4.5	Discussion and Conclusion	83
Chapter 5. BruSLeAttack: A Sparse Attack Under Score-base Settings		85
5.1	Motivation and Contribution	86
5.1.1	Chapter Overview	88
5.2	Related Work	89
5.3	Notation Table	90
5.4	Proposed Method	91
5.4.1	New Problem Formulation to Facilitate a Solution	91
5.4.2	A Probabilistic Framework for the l_0 Constrained Combinatorial Search	92
5.4.3	Sparse Attack Algorithm	96
5.5	Experiments and Evaluations	99
5.5.1	Experiment Settings	99
5.5.2	Experimental Regime	100
5.5.3	Attacking Transformers & Convolutional Nets	101
5.5.4	Comparing Performance with Prior Decision-Based and l_0 -Adapted Attack Algorithms	103
5.5.5	Attacking Defended Models	105
5.5.6	Attacking a Real-World System	106
5.6	Discussion and Conclusion	107

Chapter 6. Model Diversity: A Defense Approach Against Query-Based Attacks 109

- 6.1 Motivation and Contribution 110
 - 6.1.1 Chapter Overview 112
- 6.2 Related Work and Background 113
- 6.3 Proposed Method 114
 - 6.3.1 Achieving Model Output Uncertainty for Black-Box Attackers Through Randomness 114
 - 6.3.2 Proposed Method for Achieving Model Diversity 116
 - 6.3.3 Alternative Approaches to Promote Model Diversity 117
- 6.4 Experiments and Evaluations 119
 - 6.4.1 Experimental Regime 120
 - 6.4.2 Robustness to Black-box Attacks 121
 - 6.4.3 Diversity Analysis 124
 - 6.4.4 Effectiveness of the Proposed Learning Objective (Sample Loss) . 125
- 6.5 Conclusion 126

Chapter 7. Conclusion 131

- 7.1 Thesis Overview and Summary 132
- 7.2 Future Work 134

Appendix A. Chapter 3 Appendix 137

- A.1 Targeted Attacks on Balanced and Non-hard Sets 137
- A.2 Untargeted Attack Validation 140
- A.3 Attack Success Rates vs Query Budgets 142
- A.4 Impact of Starting Images Balance & Non-hard subset 142
- A.5 Robustness of RAMBOATTACK 143
- A.6 Perturbation Regions and Attack Insights 144
- A.7 Computation Time of Experiments 145
 - A.7.1 Hyper-parameters and Impacts 145
 - A.7.2 The impact of parameter λ : 145
- A.8 C&W Attack Configuration and Results Collection 147

Appendix B. Chapter 4 Appendix	149
B.1 Hyper-parameters	149
B.2 Investigate Hyper-Parameters, Recombination and Mutation	149
B.3 A Comparison with the Whitebox Baseline	151
B.4 Algorithmic Comparison with PointWise	151
B.5 Comparison with an Improved PointWise Algorithm	152
B.6 Comparison with Adapted l_0 Attacks	153
B.7 Illustration of Sparse Adversarial Examples	155
Appendix C. Chapter 5 Appendix	157
C.1 Sparse Attack Evaluations On ImageNet	159
C.2 Sparse Attack Evaluations on STL10 (Targeted Settings)	162
C.3 Sparse Attack Evaluations on CIFAR-10 (Targeted Settings)	162
C.4 Comparing BRUSLEATTACK With Other Attacks Adapted for Score-Based Sparse Attacks For Additional Baselines	163
C.4.1 Additional Evaluations With Decision-Based Sparse Attack Methods	163
C.4.2 l_0 Adaptations of Dense Attacks	164
C.4.3 Comparing BRUSLEATTACK With One-Pixel Attack	165
C.4.4 Bayesian Optimization	166
C.4.5 A Discussion Between BRUSLEATTACK (Adversarial Attack) and B3D (Black-box Backdoor Detection)	168
C.5 Evaluations Against l_2, l_∞ Robust Models From Robustbench and l_1 Robust Models	169
C.6 Reformulate the Optimization Problem	171
C.7 Analysis of Search Space Reformulation and Dimensionality Reduction .	172
C.8 Analysis of Synthetic Image Initialization	174
C.9 BRUSLEATTACK under Different Random Seeds	176
C.10 Effectiveness of Dissimilarity Map	177
C.11 Hyper-parameters, Initialization and Computation Resources	178
C.12 Hyper-Parameters Study	179

BIBLIOGRAPHY

C.12.1 The Impact of m_1, m_2	179
C.12.2 The Impact of λ_0	179
C.12.3 The Choice of α^{prior}	181
C.13 BRUSLEATTACK With Different Schedulers	182
C.14 Evaluation Protocol	183
C.15 Attack Against Google Cloud Vision	184
C.16 Visualizations of Dissimilarity Maps and Sparse Adversarial Examples .	187
Appendix D. Chapter 6 Appendix	193
D.1 Diverse Set of Models Against Black-box Attacks on MNIST	193
D.2 Accuracy of Non-defense versus Defense Models	194
Bibliography	197
Biography	211

Abstract

Deep neural networks (DNNs) have been recognized for their remarkable ability to achieve state-of-the-art performance across numerous machine learning tasks. However, DNN models are susceptible to attacks in the deployment phase, where Adversarial Examples (AEs) present significant threats. Generally, in the Computer Vision domain, adversarial examples are maliciously modified inputs that look similar to the original input and are constructed under white-box settings by adversaries with full knowledge and access to a victim model. But, recent studies have shown the ability to extract information *solely* from the output of a machine learning model to craft adversarial perturbations to black-box models is a *practical* threat against real-world systems. This is significant because of the growing numbers of Machine Learning as a Service (MLaaS) providers—including Google, Microsoft, IBM—and applications incorporating these models. Therefore, this dissertation studies the weaknesses of DNNs to attacks in black-box settings and seeks to develop mechanisms that can defend DNNs against these attacks.

Recognising the *practical* ability of adversaries to exploit simply the classification decision (*predicted label*) from a trained model's *access interface* distinguished as a *decision-based* attack, the research in Chapter 3 first delves into recent state-of-the-art decision-based attacks employing approximate gradient estimation or random search methods. These attacks aim at discovering $l_{p>0}$ constraint adversarial instances, dubbed *dense attacks*. The research then develops a *robust* class of *query efficient* attacks capable of avoiding entrapment in a local minimum and misdirection from noisy gradients seen in gradient estimation methods. The proposed attack method—RAMBOATTACK—exploits the notion of Randomized Block Coordinate Descent to explore the hidden classifier manifold, targeting perturbations to manipulate only localized input features to address the entrapment issues in local minima encountered by gradient estimation methods.

In contrast to *dense attacks*, recent studies have realised $l_{p=0}$ constraint adversarial instances, dubbed *sparse attacks* in white-box settings. This demonstrates that machine learning models are more vulnerable than we believe. However, these sparse attacks in the most challenging scenario—decision-based—have not been

well studied. Furthermore, the sparse attacks aim to *minimize the number of perturbed pixels*—measured by l_0 norm—leads to i) an NP-hard problem; and ii) a non-differentiable search space. Recognizing the shortage of study about sparse attacks in a decision-based setting and challenges of NP-hard problem and non-differential search space, the research in Chapter 4 explores decision-based sparse attacks and develops an evolution-based algorithm—SPARSEEVO—for handling these challenges. The results of comprehensive experiments in this research show that SPARSEEVO requires significantly fewer model queries than the state-of-the-art sparse attack for both untargeted and targeted attacks. Importantly, the query efficient SPARSEEVO, along with decision-based attacks, in general, raise new questions regarding the safety of deployed systems and poses new directions to study and understand the robustness of machine learning models.

Extracting information *solely* from the confidence score of a machine learning model can considerably reduce the required query budgets to attack a victim model. But similar to sparse attacks in decision-based settings, constructing sparse adversarial attacks, even when models opt to serve *confidence score information* to queries, is non-trivial because of the resulting NP-hard problem and the non-differentiable search space. To this end, the study in Chapter 5 develops the BRUSLEATTACK—a *new* algorithm built upon a Bayesian framework for the problem and evaluates against Convolutional Neural Networks, *Vision Transformers*, recent *Stylized ImageNet* models, *defense methods* and Machine Learning as a Service (MLaaS) offerings exemplified by **Google Cloud Vision**. Through extensive experiments, the proposed attack achieves *state-of-the-art attack success rates* and *query efficiency* on standard computer vision tasks across various models.

Understanding and recognizing the vulnerability of Deep Learning models to adversarial attacks in various black-box scenarios has compelled the exploration of mechanisms to defend Deep Learning models. Therefore, the research in Chapter 6 explores different defense approaches and proposes a more effective mechanism to defend against black-box attacks. Particularly, the research aims to integrate uncertainty into model outputs to mislead black-box attacks by randomly selecting a single or a subset of well-trained models to make predictions to query inputs. The uncertainty in the output scores to sequences of queries is able to hamper the attempt of attack algorithms at estimating gradients or searching directions toward an adversarial example. Since the uncertainty in the output scores can be improved through the

diversity of a model set, the research investigates different techniques to promote model diversity. Through comprehensive experiments, the research demonstrates that the Stein Variational Gradient Descent method with a novel sample loss objective encourages greater diversity than others. Overall, both introducing uncertainty into the output scores and prompting diversity of the model set studied in this research is able to greatly enhance the defense capability against black-box attacks with minimal impact on model performance.

Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

The author acknowledges that copyright of published works contained within this dissertation resides with the copyright holder(s) of those works.

I give permission for the digital version of my dissertation to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed

Date

31 / 07 / 2023

Acknowledgements

I would like to express my gratitude to my supervisors, Associate Professor Damith Chinthana Ranasinghe and Associate Professor Ehsan Abbasnejad, for their generous support, exceptional guidance, and endless inspiration. Without them, I would not have been able to complete my dissertation, and the knowledge gained from them is the most valuable asset in my life.

Thank A/Prof. Ranasinghe for pushing me out of my comfort zone and helping me become an independent researcher. Your constant encouragement and appreciation of my efforts to manage many challenges during my Ph.D. journey have been invaluable.

I also would like to thank A/Prof. Abbasnejad for introducing me to Machine Learning in my early candidature and helping me study and leverage the elegance and beauty of mathematics concepts.

I am grateful to my lab mates, especially Hoa from my day one in Australia, Bao, for his support and collaboration in my last chapter, and Yang Fei, Michael, Joshua, and Yang Su, for their generous support during my candidature.

I would like to extend my appreciation to the University of Adelaide for awarding me the University of Adelaide International Scholarship. This allowed me to study in such a beautiful city and gave me great chances to meet new friends. Mainly, I would like to thank all my Australian and Vietnamese friends in Adelaide who have provided me with countless support and helped me through academic and personal life difficulties.

Lastly, I would like to express my special thanks to my mother, Tuyet, and parents-in-law, Luong and Tran, for their love and support. Especially, I want to say tons of gratitude to my wife, Thao, the most gorgeous girl in the world, for always being there silently and taking care of me. Their unlimited support has been essential and motivated me to complete my research.

Dissertation Conventions

The following conventions have been adopted in this dissertation:

Typesetting

This document was compiled using L^AT_EX2_ε. Texstudio 2.12.22 was used as a text editor interfaced to L^AT_EX2_ε. Inkscape 0.92.3 was used to produce schematic diagrams and other drawings.

Spelling

Australian English spelling conventions have been used, as defined in the Macquarie English Dictionary—A. Delbridge (Ed.), Macquarie Library, North Ryde, NSW, Australia, 2001.

Referencing

The Harvard reference style is used for referencing and citation in this dissertation.

System of Units

The units comply with the international system of units recommended in an Australian Standard: AS ISO 1000-1998 (Standards Australia Committee ME/71, Quantities, Units and Conversions 1998).

Acronyms

AE	Adversarial Examples
AI	Artificial Intelligence
ASR	Attack Success Rate
BNN	Bayesian Neural Network
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
GA	Genetic algorithms
ML	Machine Learning
MLaaS	Machine Learning as a Service
SVGD	Stein Variational Gradient Descent
ViT	Vision Transformer

Publications

Conference Articles

- [1] Vo, V.Q., Abbasnejad, E., Ranasinghe, D.C. RamBoAttack: A Robust and Query Efficient Deep Neural Network Decision Exploit. *Network and Distributed System Security Symposium (NDSS'22)* (CORE Rank: A*, acceptance rate: 16.2%, 94 accepted out of 581 submissions).
- [2] Vo, V.Q., Abbasnejad, E., Ranasinghe, D.C. Query Efficient Decision Based Sparse Attacks Against Black-Box Machine Learning Models. *Conference on Learning Representations (ICLR'2022)* (CORE Rank: A*, acceptance rate: 32.9%, 1095 accepted out of 3328 submissions).

Under Preparation for Submissions

- [3] Vo, V.Q., Abbasnejad, E., Ranasinghe, D.C. BruSLeAttack: A Query-Efficient Score-Based Sparse Adversarial Attack.
- [4] Vo, V.Q., Abbasnejad, E., Ranasinghe, D.C. Model Diversity: A Defense Approach Against Query-Based Attacks.

List of Figures

1.1	An illustration of an attack to craft an adversarial example in a black-box scenario.	3
1.2	An illustration of adversarial attacks in different black-box settings. . . .	4
1.3	Outline of the dissertation	14
<hr/>		
2.1	An illustration of a deep neural network (DNN)	16
2.2	An illustration of Convolutional Neural Network.	17
2.3	An illustration of a vision transformer.	18
2.4	Attack Taxonomy	20
2.5	Black-box adversarial attacks categorized based on access level and knowledge.	21
2.6	Black-box adversarial attacks categorized based on similarity measures.	22
2.7	An illustration of various defense methods.	23
<hr/>		
3.1	An illustration of black-box attack in decision-based settings.	28
3.2	The number of hard cases from CIFAR10.	32
3.3	Demonstration of different adversarial examples found by different methods and an illustration of dependency of attacks methods on starting images.	34
3.4	An illustration of different attacks using a Toy model.	35
3.5	A pictorial illustration of RAMBOATTACK.	37
3.6	An illustration of our RAMBOATTACK against the toy model.	38
3.7	The proposed evaluation protocol for assessing robustness under an exhaustive evaluation setting.	47
3.8	The proposed evaluation protocol requires a balanced dataset including n source classes and a balance target set.	48
3.9	The number of <i>hard</i> cases found for Sign-OPT, HopSkipJump and RAMBOATTACK.	50

List of Figures

3.10	A distortion comparison versus queries for each method using their own <i>hard</i> versus <i>non-hard</i> cases.	50
3.11	Distortion (dist) on a \log_{10} scale vs number of queries on <i>hard-set</i> A.	51
3.12	Distortion (dist) on a \log_{10} scale vs number of queries on <i>hard-set</i> B.	51
3.13	The distortion distribution yielded by our RAMBOATTACKS on both <i>hard-set</i> A and B from CIFAR10.	52
3.14	Distortion in a \log_{10} scale vs number of queries on <i>hard-set-D</i>	52
3.15	Distortion in a \log_{10} scale vs number of queries on <i>hard</i> ImageNet evaluation sets.	53
3.16	The distortion distributions yielded by our RAMBOATTACKS on the <i>hard-set</i> selected from ImageNet.	54
3.17	An illustration of sensitivity of different attacks to various starting images.	55
3.18	Grad-CAM tool visualizes salient features of the starting image or target class.	56
3.19	An illustration of <i>hard</i> case (white stork to goldfish) versus <i>non-hard</i> case (white stork to digital watch) on ImageNet.	57
3.20	Performance comparison between different state-of-the-art attacks and RAMBOATTACK against a region-based classifier on CIFAR10.	59
3.21	ASR comparison between white-box (<i>employed as a baseline</i>) and current decision-based attacks versus our RAMBOATTACK against.	60
3.22	Upcoming chapter sneak peek.	61
<hr/>		
4.1	Malicious instances generated for a sparse attack with different query budgets using our SPARSEEVO employed on black-box models.	64
4.2	An illustration of SPARSEEVO algorithm. <i>Population Initialization</i> creates the first population generation.	70
4.3	The Binary Differential Recombination Algorithm	73
4.4	Evaluation set from ImageNet using the ResNet50 model.	77
4.5	Evaluation set from ImageNet using the ViT model.	79
4.6	Evaluation set from CIFAR10 using a ResNet18 model.	80
4.7	Attack success rate versus sparsity thresholds at different query budgets for the evaluation set from ImageNet with ViT vs ResNet.	82

4.8	Different sparse attacks against an adversarially trained model on the CIFAR10 task.	83
4.9	Upcoming chapter sneak peek.	84

5.1	Malicious instances are generated by BRUSLEATTACK with different perturbation budgets against three deep learning models on ImageNet .	86
5.2	A Sampling and Update illustration.	93
5.3	BRUSLEATTACK algorithm	96
5.4	fTargeted setting on ImageNet. ASR of different sparse attacks and accuracy of different models against BRUSLEATTACK.	101
5.5	Targeted attacks on the ImageNet task against ResNet-50.	103
5.6	Demonstration of sparse attacks against GCV in targeted settings with a budget of 5K queries and sparsity of 0.5%	107
5.7	Upcoming chapter sneak peek.	108

6.1	A comparison of robustness among defense methods on MNIST.	122
6.2	A robustness comparison among defense methods on CIFAR-10.	123
6.3	A robustness comparison among defense methods on STL-10.	124
6.4	A model diversity comparison using Jensen–Shannon divergence on CIFAR-10 and STL-10	125

A.1	A comparison between three current state-of-the-art attacks and RAMBOATTACK on a balance set selected from CIFAR10.	137
A.2	A comparison between three current state-of-the-art attacks and RAMBOATTACK on a large scale balance set selected from ImageNet. . .	138
A.3	A comparison between three current state-of-the-art attacks and RAMBOATTACK on a <i>non-hard</i> set C selected from CIFAR10.	139
A.4	A comparison between three current state-of-the-art attacks and RAMBOATTACK on a <i>non-hard</i> set selected from ImageNet.	139
A.5	Comparing between three current state-of-the-art attacks and RAMBOATTACK on the balance set selected from CIFAR10.	140

List of Figures

A.6	Comparing between three current state-of-the-art attacks and RAMBOATTACK on the balance set from ImageNet.	141
A.7	ASR vs. queries for our RAMBOATTACKS with respect to Boundary attack and with respect to HopSkipJump and Sign-OPT.	141
A.8	An illustration of the sensitivity of different attacks to various chosen starting images.	142
A.9	The number of <i>hard</i> cases on CIFAR10 obtained from different attack methods categorized by pairs of source and target classes.	143
A.10	Grad-CAM tool visualizes salient area of the starting image Staffordshire bull terrier.	144
A.11	An illustration of different distortion levels produced by RAMBOATTACK.	145
A.12	A comparison between RAMBOATTACK with different values of λ on 100 source and target class sample pairs selected from ImageNet.	146

B.1	Sparsity versus number of model queries on CIFAR10 with ResNet18 to show the impacts of different hyper-parameters on SPARSEEVO.	150
B.2	Visualisations from a targeted attack Settings.	156

C.1	Untargetted Setting. ASR versus the number of model queries against different Deep Learning models.	161
C.2	Targeted attacks on CIFAR-10 against ResNet-18.	164
C.3	Targeted attacks on CIFAR-10 with a query budget of 250.	167
C.4	ASR versus model queries on ImageNet.	181
C.5	Attack result demonstration against Google Cloud Vision.	185
C.6	Attack result demonstration against Google Cloud Vision.	186
C.7	Targeted Attack. Visualization of Adversarial examples.	188
C.8	Untargeted Attack. Visualization of Adversarial examples	188
C.9	Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 5000 queries.	189
C.10	Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 5000 queries.	190

C.11 Visualization of Dissimilarity Maps between a source image and a synthetic color image. 191

List of Tables

4.1	Median sparsity and ASR at different query budgets.	81
4.2	Accuracy of ResNet50 and ViT under attacks at different query budgets and sparsity thresholds.	81
5.1	Table of notation descriptions.	90
5.2	ASR comparison between our proposal and SPARSEEVO (Alternative Loss) on CIFAR-10.	104
5.3	A comparison of ASR between our proposal (Synthetic Color Image) and employing a starting image.	105
5.4	A robustness comparison (lower \downarrow is stronger) between SPARSE-RS and BRUSLEATTACK against undefended and defended models	106
6.1	A robustness comparison among diversity-promoting defense methods on MNIST. Training a set of 40 models.	127
6.2	A robustness comparison among diversity-promoting defense methods on CIFAR-10. Train a set of 10 models.	128
6.3	A comparison of robustness among diversity promotion defense methods on STL-10. Train a set of 10 models.	128
6.4	Clean accuracy of a set of 10 models trained simultaneously with and without sample loss on MNIST and CIFAR-10.	129
A.1	Comparison among attacks with RAMBOATTACK on small and large scale balance datasets.	138
A.2	Summary of computation time for each experiment	146
B.1	Hyper-parameters setting in our experiments	149
B.2	Mean sparsity measure at different queries (lower is better) for a targeted attack setting.	153
B.3	Mean sparsity measure at different queries (lower is better) for a targeted setting.	154

List of Tables

C.1	ASR at different sparsity levels across different queries (higher is better).	159
C.2	ASR at different sparsity levels across different queries (higher is better).	160
C.3	ASR (higher is better) at different sparsity levels in targeted settings on STL-10.	162
C.4	ASR (higher is better) at different sparsity thresholds in the targeted setting on CIFAR-10.	163
C.5	Mean sparsity at different queries for a targeted setting.	165
C.6	ASR comparison (higher \uparrow is stronger) between One-Pixel and BRUSLEATTACK against ResNet18 on CIFAR-10.	166
C.7	A robustness comparison between SPARSE-RS and BRUSLEATTACK against undefended and defended models employing l_∞, l_2 robust models	170
C.8	A robustness comparison between SPARSE-RS and BRUSLEATTACK against undefended and defended models employing l_1 robust models .	171
C.9	Target setting. ASR (higher is better) at different sparsity thresholds in the targeted setting on CIFAR-10.	174
C.10	ASR comparison between using a synthetic color image uniformly generated at random (our proposal) and maximizing dissimilarity on CIFAR-10.	176
C.11	ASR comparison between using a fixed random color search space (our proposal) and two or four random color search spaces on CIFAR-10. . . .	176
C.12	ASR (Min, Mean, Max and Standard Deviation) of our attack methods on CIFAR-10.	177
C.13	ASR comparison between with and without using Dissimilarity Map on CIFAR-10.	178
C.14	ASR at different sparsity thresholds and queries (higher is better) for a targeted setting on ImageNet.	178
C.15	Hyper-parameters setting in our experiments	179
C.16	ASR of BRUSLEATTACK with different values of m_1, m_2 on CIFAR-10. . .	180
C.17	ASR of BRUSLEATTACK with different values of λ_0 on CIFAR-10.	180
C.18	ASR at different sparsity levels and queries in a targeted setting on ImageNet.	180
C.19	ASR comparison between using a Power Step Decay (our proposal) and other schedulers on CIFAR-10.	183

C.20	Demonstration of sparse attacks against GCV in targeted settings.	185
C.21	Demonstration of sparse attacks against GCV in targeted settings.	186
D.1	A robustness comparison among diversity-promotion defense methods on MNIST. Train a set of 10 models.	193
D.2	A robustness comparison among diversity-promotion defense methods on MNIST. Train a set of 20 models.	194
D.3	Clean accuracy achieved by different defended models employing diversity-promotion techniques.	195
D.4	Clean accuracy achieved by non-defense models and defended models.	196

Chapter 1

Introduction

THE first chapter of this dissertation provides a concise introduction to the field of Machine Learning and discusses the scope and challenges of the research problems. The chapter also presents the motivations for the research and the objectives of this dissertation, as well as emphasizing the contributions made in the following chapters. This chapter concludes with a guide to the dissertation's structure.

1.1 Introduction

In recent years, deep neural networks (DNNs) have demonstrated remarkable performance on a variety of vision tasks, earning them enough trust to be deployed in what are often critical applications, such as self-driving cars (Chen et al., 2015) or disease diagnosis (Anwar et al., 2018). DNNs' superhuman performance on certain tasks has also led to the industrialization of machine learning, with a growing numbers of Machine Learning as a Service (MLaaS) providers—including Google Cloud Vision¹, IBM Watson Visual Recognition², Amazon Rekognition³ or Microsoft's Cognitive Services⁴—and a plethora of applications incorporating DNN models. Now, at the cost-per-service level, any system can easily integrate *intelligence* into applications. However, the increasingly inevitable and widespread proliferation of machine learning in systems is creating incentives and new attack surfaces to exploit for malevolent actors.

Extensive research assessing the vulnerabilities of deep learning systems have already shown that some models are critically susceptible to evasion attacks from *Adversarial Examples*. These attacks, in general, seek to craft malicious, imperceptible perturbations to be applied to model inputs in order to misguide or hijack the decision of the DNN model.

Adversarial examples attacks, or simply *adversarial attacks* henceforth, conducted under conditions of complete access to and knowledge of the target model (*i.e.* architecture and parameters) in so-called white-box settings are well-studied. However, in practical deployments of commercial and industrial machine learning systems, model information is highly restricted to external parties. Now, attacks must be conducted in black-box settings—with highly limited access to the model. In such practical settings, an attacker is limited to interacting with a model through a query-response mechanism and is only able to access the revealed model outputs, as illustrated in Figure 1.1.

Attacking these systems in such black-box scenarios is more practical and, therefore, interesting to study. Since these attacks are able to compromise the reliability and security of DNN models with limited access and information, they pose a significant threat to the safety of applications and systems relying on DNNs. Therefore, this

¹<https://cloud.google.com/vision>

²<https://www.ibm.com/cloud/machine-learning>

³<https://aws.amazon.com/machine-learning/>

⁴<https://azure.microsoft.com/en-us/services/cognitive-services/>

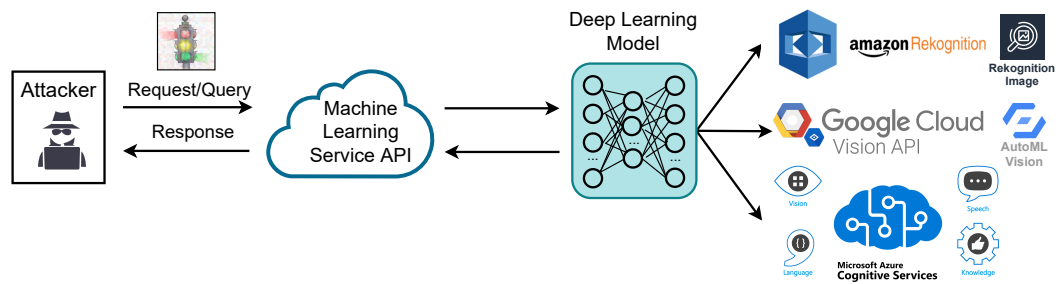


Figure 1.1. An illustration of an attack to craft an adversarial example in a black-box scenario in which an attacker aims to mislead a deep learning model using a query-response interaction via the publicly exposed application programming interface or API (*i.e.* Google Cloud Vision or Amazon Rekognition) of a MLaaS provider. Of particular concern, an attacker can query a model by sending a request to a machine learning service API and exploit the response from that model to make changes to the input data and craft an input that is capable of leading the model to make an incorrect decision.

dissertation aims to study the vulnerability of deep learning models to adversarial attacks in black-box scenarios.

Recently, research has demonstrated that adversarial attacks in black-box scenarios relying solely on the limited information from a model's output are feasible. The threat posed by these attacks can be characterized as *score-based* or *decision-based* attacks by examining the information revealed by a model to adversaries. Notably, in an effort to quantitatively assess the imperceptibility of perturbations, adversarial attacks can be categorized according to the l_p norm-constrained perturbation of *Dense Adversarial Examples* or *Sparse Adversarial Examples*. In general, in any attack against a DNN model, an adversary may aim to cause the models to fail to make a correct decision—referred to as an *untargeted attack*—or lead the DNN to make a malicious decision—referred to as a *targeted attack*. In practice, achieving a targeted attack is significantly harder than an untargeted attack. The details of black-box attack variations are illustrated in Figure 1.2 and explained further below:

- **Score-based versus decision-based attacks.** Query-based black-box attacks capable of exploiting only a model's output scores to craft adversarial examples are referred to as *score-based attacks*, while black-box attacks in the more restrictive setting relying *solely* on the label obtained from model queries are dubbed *decision-based attacks*. This attack setting is considered the *most* restricted threat model since the information exposed to an attacker is limited

1.1 Introduction

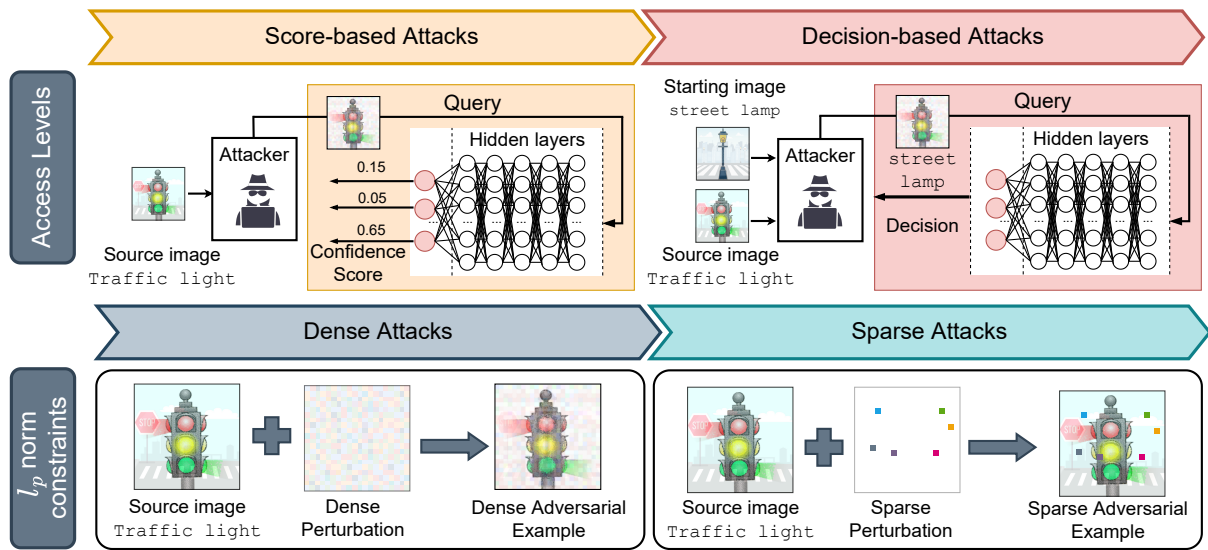


Figure 1.2. An illustration of adversarial attacks in different black-box settings in which attacks aim at manipulating a source image (*i.e.* traffic light image) to evade a black-box deep learning model. Specifically, in *Score-based* settings, the attack can merely access the output—*Confident Score* while in *Decision-based* settings, the attack only has access to the model's decision—predicted label. Moreover, a targeted attack in decision-based settings requires a starting image from a target class (*i.e.* street lamp). In *Dense* settings, an adversarial attack aims to perturb an entire source image, whereas in *Sparse* settings, the attack only alters a few pixels of the source image.

to the *hard-label*—the most confident label predicted or *decision*, for instance, as provided by the logo or landmark detection model services on Google Cloud Vision⁵.

- **Dense versus sparse attacks.** Based on a similarity measure, imperceptibility can describe an attack as a *dense attack*— l_2, l_∞ norm constrained adversarial attacks—or a *sparse attack*— l_0 norm constrained adversarial attacks. In the vision domain, a l_0 norm-constrained perturbation is equivalent to the number of pixels manipulated. As such, the main aim of sparse attacks is to minimize the number of perturbed pixels required to mislead the target machine learning model.

Since black-box scenarios are practical and provide insights into real-world security threats faced by machine learning systems, it is logical to explore and investigate adversarial attacks under these black-box scenarios. However, compared to the

⁵<https://cloud.google.com/vision>

extensive studies that have focused on crafting adversarial examples in white-box settings, there is not yet a comparable level of knowledge and understanding regarding attacks in black-box settings. Notably, only a handful of studies have explored sparse attacks and attacks under more challenging, decision-based settings. To this end, this dissertation focuses on understanding the vulnerability of DNN models to threats from lesser studied black-box attack variants and developing a means of defending against such attacks.

1.2 Objectives

In seeking to contribute to knowledge and understanding about black-box attacks and design defense mechanisms against them, this dissertation pursues the following two objectives:

Objective 1: Understand the practical threats to deep learning models from adversarial attacks under lesser studied black-box scenarios.

Objective 2: Develop a robust mechanism to defend against black-box adversarial attacks with a marginal accuracy trade-off.

1.3 Challenges

Prior research has investigated score-based and dense attacks to promote our understanding of deep learning model vulnerabilities. However, threats against deep learning models in other black-box settings such as decision-based dense, decision-based sparse or score-based sparse threat models have not been widely studied. Additionally, known adversarial attacks under black-box threat models require a large number of queries to deceive a learned model. Consequently, these attack methods do not appear to scale well, especially to high-resolution tasks such as ImageNet classification. Therefore, the practicability of the threat to real-world systems is unclear. This leads to an open question about the existence of *query efficient attacks* that can scale to threaten practical vision tasks.

For realistic attacks, achieving attack success with a limited query budget is important because: i) MLaaS providers limit the rate of queries to their services; ii) throttling

1.3 Challenges

at a service provider limits large-scale attacks; and iii) providers can recognize large number of queries with similar inputs made in rapid succession to detect malicious activity and thwart query attacks. Furthermore, from the perspective of both the attacker and the provider, reducing the number of queries *reduces the cost of mounting the attack* as well the time needed to evaluate the model and potential defenses⁶.

However, designing query-efficient adversarial attacks in black-box scenarios is a difficult task. Therefore, this dissertation argues that to foster a deeper understanding of black-box threat models, the query efficiency of attacks must be investigated further.

Challenges in Developing Query Efficient Black-box Attacks

Due to the lack of model knowledge and direct access to model gradients, formulating query-efficient attacks, especially for high-resolution images, is challenging. Because attack algorithms:

- Must estimate gradients or search for a direction toward adversarial examples in the absence of gradient feedback. They therefore suffer from algorithmic approximation to gradients that can hinder the efficiency of the attack. For instance, gradient estimation frameworks used in dense attacks may suffer from the problem of entrapment in local minima (Chen, Jordan and Wainwright, 2020).
- Encounter NP-hard problems. For example, sparse attacks are shown to be NP-hard problems (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020).
- Face a large and complex search space of possible adversarial examples. For instance, an attack in sparse settings has to search for an adversarial example over a mixed space encompassing continuous values for color and discrete values for pixel locations. Notably, when dealing with high-resolution data sets, this mixed search space expands into a high-dimensional space that is colossal in size, as discussed in Appendix C.7.

As such, crafting successful adversarial examples in black-box settings often necessitates a substantial number of queries, especially when targeting high-resolution data sets.

⁶For example, we consumed over 1,700 hours on two dedicated modern GPUs with 48 GB memory to curate the results in our study in Chapter 3

On the other hand, a deeper understanding and knowledge of model vulnerabilities to black-box adversarial attacks in different scenarios may allow us to establish the criticality of the threat and indicate a need to develop mechanisms to defend deep learning models. Developing robust defenses is an ongoing research challenge, that this dissertation attempts to confront.

Challenges in Developing Robust Defenses against Black-box Attacks

An intuitive approach to developing such defense mechanisms is to leverage existing countermeasures (Goodfellow, Shlens and Szegedy, 2014; Xie et al., 2018; Dhillon et al., 2018; Xie et al., 2019; Rakin, He and Fan, 2019) originally devised for white-box attacks in order to fortify DNN models against black-box attacks. While they can defend against these black-box attacks, they often entail a compromise with respect to accuracy (Tsipras et al., 2019; Yang et al., 2020b; Qin et al., 2021; Byun, Go and Kim, 2022). For instance, adversarial training-based methods, renowned for their effectiveness against white-box attacks (Athalye, Carlini and Wagner, 2018; Tramer et al., 2020), are associated with a considerable reduction in accuracy as shown by (Zhang et al., 2019; Zhang and Wang, 2019; Shafahi et al., 2019; Yang et al., 2020b; Doan et al., 2022a). Consequently, developing a defense method that achieves the objectives of both robustness and accuracy poses a formidable challenge.

Nevertheless, in contrast to their white-box counterparts, black-box attacks have limited access to deep learning models' output and are hindered by the lack of gradient information. Due to this constraint, black-box attacks necessitate myriad queries for sending manipulated input and observing the corresponding output that aims to approximate gradients or find proper search directions toward adversarial examples. Consequently, some defense methods (Qin et al., 2021; Byun, Go and Kim, 2022) tailored for countering black-box attacks exploit this intrinsic weakness instead of adopting adversarial training techniques to thwart black-box attacks. These defense methods add random noise to each queried input at the inference phase to hamper gradient estimation or random search. However, adding larger noise can result in a decrease in clean accuracy as the model is sensitive to added random noise (Qin et al., 2021) and does not see noisy images during training (Cohen, Rosenfeld and Kolter, 2019).

We can see that existing defenses compromise accuracy to achieve robustness (*i.e.* robust at higher input distortion levels). Consequently, this leaves open the

question of how to obtain the objectives of both robustness and clean accuracy when fortifying deep learning models against black-box attacks.

1.4 Summary of Contributions

To achieve the objectives of this dissertation, as outlined in Section 1.2, the research presented here has made several original contributions in addressing the challenges outlined in Section 1.3 to enhance knowledge of query-based black-box attacks and efficient defense mechanisms against such attacks. The key developments and contributions made in this dissertation can be succinctly summarized as follows:

1. **(A Decision-Based Dense Attack)** To improve knowledge of deep learning model vulnerabilities in practical *decision-based* and *dense* scenarios, the study in Chapter 3 analyzes different decision-based attack algorithms and investigates a challenging optimization problem (*i.e.* the entrapment in local minima) encountered by these attacks. First, this study presents a systematic investigation of state-of-the-art decision-based attacks to demonstrate their robustness. Through extensive experiments, this study uncovers the existence of challenging instances, known as *hard cases*, where attack algorithms struggle to flip the prediction of input towards a desired target class, even under an extremely high number of model queries (*i.e.* high query budgets). We hypothesize that these hard cases stem from entrapment in various local minima. Secondly, this study introduces a novel attack method—RAMBOATTACK—which leverages a search algorithm inspired by Randomized Block Coordinate Descent, referred to as BLOCKDESCENT, to overcome the entrapment problem when gradient estimation fails to guide the attack. Thirdly, the study provides new insights into query-efficient mechanisms to craft adversarial perturbations. Unlike existing techniques, the BLOCKDESCENT component of RAMBOATTACK focuses on altering local regions of the input commensurate with the filter sizes employed by deep neural networks (DNNs) to generate adversarial examples. This proposed mechanism, in the *hard cases*, allows the discovery of potential adversarial perturbations to exploit the model’s reliance on salient features of the target class for classifying an input. The study demonstrates clear connections between added perturbations and salient regions in images from the target class using a visual explanation tool. Overall, RAMBOATTACK emerges as a more

robust and query-efficient method for crafting adversarial examples than other decision-based attacks. Notably, RAMBOATTACK demonstrates significantly less sensitivity to the choice of a starting image from the target class when compared with existing attacks in decision-based settings. This work has been accepted for publication at the 29th annual Network and Distributed System Security Symposium (NDSS'22) under the title "RamBoAttack: A Robust Query Efficient Deep Neural Network Decision Exploit" and contributes to achieving **Objective 1**.

2. **(A Decision-Based Sparse Attack)** To improve knowledge of deep learning model vulnerabilities in a practical *decision-based* and *sparse* setting, the work in Chapter 4 studies sparse attacks and the resulting NP-hard optimization problem. Firstly, this work examines the effectiveness of sparse attacks and introduces a novel sparse attack—SPARSEEVO—an evolution-based algorithm, to mitigate the complexity posed by the NP-hard problem. The proposed method is capable of exploiting access to solely the top-1 predicted label or model decision to search for an adversarial example while minimizing the number of perturbed pixels required to deceive the model. Secondly, for the first time, this work assesses the vulnerability of transformer-based models against decision-based sparse attacks on the standard computer vision task ImageNet and its relative robustness to a convolutional-based model. Thirdly, through extensive experiments, the proposed attack algorithm demonstrates a significant reduction in the number of model queries when compared to a state-of-the-art decision-based sparse attack. Interestingly, SPARSEEVO achieves comparable levels of success to a state-of-the-art *white-box* attack, even when operating under a limited query budget. Overall, SPARSEEVO emerges as a significantly more query-efficient method compared to state-of-the-art algorithms for generating sparse adversarial examples under decision-based sparse settings. This work has been accepted for publication at the Tenth International Conference on Learning Representations (ICLR'22) under the title "Query Efficient Decision-Based Sparse Attacks Against Black-Box Machine Learning Models" and contributes to achieving **Objective 1**.
3. **(A Score-Based Sparse Attack)** To provide new perspectives for understanding and mitigating the vulnerabilities of DNNs in a practical *score-based* and *sparse* setting, the research in Chapter 5 studies crafting sparse

adversarial perturbations where models reveal scores and the resulting NP-hard optimization problem. Firstly, the study proposes a new sparse attack method—BRUSLEATTACK—in the score-based setting. The algorithm leverages the knowledge of output scores and intuitions used to learn influential pixel information from past pixel manipulations and to select pixel perturbations based on pixel dissimilarity between a search space prior and a source image. These strategies aim to remedy the NP-hard problem and accelerate the process of searching for a sparse adversarial example. Secondly, comprehensive experiments demonstrate that the proposed attack is more query-efficient than the state-of-the-art methods across different data sets, various deep learning models, defense mechanisms and attacks against Google Cloud Vision in terms of attack success rate (ASR) and sparsity within a limited query budget of 10K queries. Thirdly, for the first time, this work assesses the vulnerability of transformer-based models against score-based sparse attacks on the high-resolution dataset Imagenet and its relative robustness to convolutional-based models. Overall, BRUSLEATTACK is a more query-efficient attack algorithm for yielding sparse adversarial examples than other score-based sparse attacks. This work is currently under review for the 37th Conference on Neural Information Processing Systems (NeurIPS'23) under the title "A Query-Efficient Score-Based Sparse Adversarial Attack" and contributes to achieving **Objective 1**.

4. (**A Defence Against Black-Box Attacks**) The study in Chapter 6 aims to develop an effective defense mechanism against query-based black-box adversarial attacks whilst maintaining high clean high accuracy. Firstly, this study introduces an intuitive countermeasure that incorporates uncertainty in model outputs to a black-box attacker by randomly selecting a subset of well-trained models to make a prediction at test time. The aim is to mislead attack algorithms seeking to estimate gradient feedback from model outputs to craft adversarial perturbations. An extensive empirical study regime across a range of data sets and models confirms that the method of randomness incorporation possesses a strong capability to mislead black-box adversarial attacks. Secondly, to enhance the defense capability, the study explores existing approaches for promoting a diverse set of well-trained models to effectively increase the diversity in model outputs. Through extensive experiments, the study shows a Bayesian

learning method that pushes the model parameters of each model apart using Stein Variational Gradient Descent (SVGD) along with the *newly proposed sample loss objective* can encourage more diverse models and, consequently, model outputs. Overall, the proposed defense incorporating the model of diversity and randomness is able to achieve the greatest level of robustness with minimal impact on clean accuracy compared to current defense methods. This work is expected to be published after the submission of this dissertation and contributes to achieving **Objective 2**.

5. (**Open Source Code Releases**) Through all the extensive studies, this dissertation contributes three open-source code repositories to the research community:

- The source code for a query-efficient dense attack under decision-based settings in Chapter 3: "A Robust Query Efficient Attack against Deep Neural" is available at <https://github.com/RamBoAttack/RamBoAttack.github.io> (RAMBOATTACK).
- The source code for a query-efficient sparse attack in the decision-based scenario in Chapter 4: "A Query Efficient Sparse Attack In Decision-base Settings" is available at <https://github.com/SparseEvoAttack/SparseEvoAttack.github.io> (SPARSEEVO).
- The source code for a query-efficient sparse attack in the score-based scenario in Chapter 5: "A Query-Efficient Score-Based Sparse Adversarial Attack" is available at <https://github.com/BruSLiAttack/BruSLiAttack.github.io> (BRUSLEATTACK).

1.5 Dissertation Structure

The dissertation structure is outlined in Figure 1.3 and is presented as follows:

1. Chapter 1 and Chapter 2 present a concise introduction to the fundamentals of DNNs, potential threats and countermeasures in various scenarios. These

- chapters also delve deeper into threat models, examine various attacks, explore defense mechanisms that can be employed to bolster resilience and discuss evaluation metrics.
2. Chapter 3 explores dense attacks and the vulnerability of DNN models in a decision-based scenario, highlighting the entrapment problem encountered by these methods. This chapter also focuses on developing a novel attack method that can be incorporated with other gradient estimation methods to overcome the entrapment problem as well as improve query efficiency. The efficiency of existing defense strategies against various decision-based dense attacks is also examined here.
 3. Chapter 4 considers the problem of searching sparse adversarial examples to mislead DNN models in decision-based settings. Moreover, the chapter considers the challenges posed by sparse attacks, *i.e.* non-differentiable search space, and the NP-Hard problem and proposes an evolution-based algorithm—SPARSEEVO—to alleviate these challenges. We also demonstrate the practical threats of sparse attacks against both transformer-based and convolutional-based architectures in computer vision tasks and analyze the relative robustness of each.
 4. Chapter 5 investigates the robustness of DNN models and the potential threat posed by sparse attacks in score-based settings. This chapter introduces a novel Bayesian-based algorithm—BRUSLEATTACK—that leverages some prior knowledge to search for sparse adversarial perturbations. We also demonstrate the efficiency of BRUSLEATTACK against various deep learning models including transformer-based and convolutional-based models, as well as defense mechanisms across different data sets.
 5. Chapter 6 examines the effectiveness and robustness of different defense mechanisms against query-based black-box attacks. In addition, it explores different methods to encourage the diversity of a set of models to hinder the progress of query-based attacks. This chapter later proposes a new method that leverages a diverse set of models and random model selection to defend against query-based black-box attacks and demonstrate the high performance of the proposed defense method against various black-box adversarial attacks across different data sets.

6. Chapter 7 gives a conclusive summary of the studies conducted for this dissertation, summarizing the findings and discussing potential areas for future exploration.

1.5 Dissertation Structure

Overview	Chapter 1	<ul style="list-style-type: none"> • Introduction • Challenges • Summary of Contributions
	Chapter 2	<ul style="list-style-type: none"> • Background on Deep Neural Networks • Fundamental of adversarial attacks and threat models • Background on defenses • Dataset and Metrics to evaluate attacks' performance and model's robustness
Black-box Adversarial Attacks	Chapter 3	<ul style="list-style-type: none"> • Investigate dense decision-based attacks and highlight the <i>hard cases</i> possibly stemming from the entrapment in local minima. • Develop a new attack method—RAMBOATTACK—to address the entrapment problem and provides new insights into the method. • Demonstrate the robustness and query efficiency of the proposed attack when dealing with <i>hard cases</i> in different datasets.
	Chapter 4	<ul style="list-style-type: none"> • Investigate sparse attacks in decision-based settings. • Introduce a new sparse attack—SPARSEEVO—with only access to a predicted label from a Deep Learning model. • Demonstrate the effectiveness of the proposed attack method against both Transformer-based and Convolutional-based models across different datasets.
	Chapter 5	<ul style="list-style-type: none"> • Investigate sparse attacks in score-based settings. • Formulate a new sparse attack—BRUSLEATTACK—in the score-based setting. • Demonstrate the efficiency of the proposed method against various Deep Learning models across different datasets, defense mechanisms and Google Cloud Vision.
Defense	Chapter 6	<ul style="list-style-type: none"> • Investigate a new defense incorporating uncertainty into model outputs to mislead black-box attacks by randomly selecting a subset of well-trained models to make predictions. • Study existing approaches for promoting diverse sets of models to enhance defense capability. • Demonstrate the efficiency of the proposed method against black-box attacks across different datasets.
Conclusion	Chapter 7	<ul style="list-style-type: none"> • Summary • Future research directions

Figure 1.3. Outline of the dissertation.

Chapter 2

Background

THIS chapter introduces the literature on deep neural networks, their vulnerabilities to attack in different scenarios and the current defense mechanisms. The chapter presents a generic formulation for the attack problem and notations, as well as exploring two main architectures of deep neural networks (DNNs) in the vision domain. Additionally, common attacks and countermeasures are discussed with the aim of establishing a solid foundation and thorough understanding of the subject matter. This section provides a brief overview of common evaluation metrics used for quantifying experimental results, which are crucial for drawing valid conclusions from the findings of this research.

2.1 Notations

For notational consistency, lowercase bold letters (*i.e.* \mathbf{x}) denote vectors, uppercase bold typeface letters (*i.e.* \mathbf{X}) represent matrices and lowercase letters (*i.e.* x) represent random variables. Let $\|x\|_p$ denote l_p norm of x , $A \odot B$ denote the element-wise (Hadamard) product of A and B . $f(\mathbf{x}; \boldsymbol{\theta})$ denotes a function of \mathbf{x} parametrized by $\boldsymbol{\theta}$; to simplify notation, the argument $\boldsymbol{\theta}$ can be omitted as $f(\mathbf{x})$. $\ell(f(\mathbf{x}; \boldsymbol{\theta}), y)$ denotes the loss between model output $f(\mathbf{x}; \boldsymbol{\theta})$ and the ground-truth y . To simplify the notation, $\boldsymbol{\theta}$ can be dropped and $f(\mathbf{x})$ is used rather than $f(\mathbf{x}; \boldsymbol{\theta})$.

2.2 Deep Neural Networks

This section briefly introduces machine learning and deep learning models (illustrated in Figure 2.1) and then presents the two most widely used deep learning models in different vision tasks—convolutional neural networks (CNNs) and vision transformer (ViT).

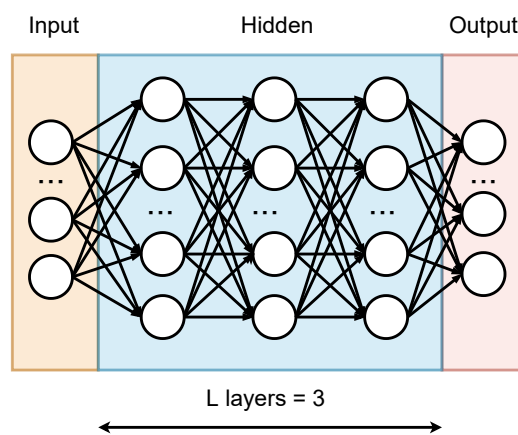


Figure 2.1. An illustration of a deep neural network (DNN)

Machine Learning Models. Machine learning (ML) models (*i.e.* a support vector machine, linear neural network or multilayer perceptron) can be defined as a parameterized function that learns patterns and relationships from data, and then uses this knowledge to make decisions on new data (*i.e.* predictions or classifications). Concretely, $f(\mathbf{x}, \boldsymbol{\theta})$ maps input data $\mathbf{x} \in \mathbf{X}$ to a particular output $y \in \mathbf{Y}$ (*i.e.* an image of traffic sign is mapped into a label of traffic sign in a classification task), where $\boldsymbol{\theta}$ denotes the parameter set. Based on a given collection of data—training

data—the parameterized function can automatically learn and update its parameter set θ by optimizing an objective function (*i.e.* minimize loss ℓ between model output $f(x, \theta)$ and the ground-truth y specified by the data set). In practice, a widely used technique—gradient descent—is leveraged to optimize this loss and then update the parameter set of a ML model.

Deep Learning Models. Deep learning models, DNNs, are a specialized machine learning model and are composed of multiple layers of interconnected processing nodes, called neurons. These layers include one input layer, one output layer and L hidden layers. Learning a DNN is similar to learning a ML model by optimizing the loss ℓ as the following:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \ell(f(x_i, \theta), y_i) \quad (2.1)$$

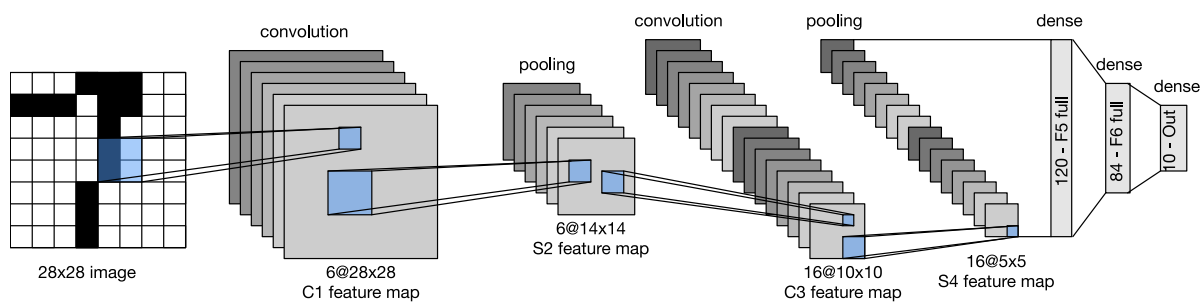


Figure 2.2. An illustration of data flow in a Convolutional Neural Network (CNN)—LeNet (Lecun et al., 1998). The input—a handwritten digit—goes through convolutional, pooling and dense layers to extract the feature of the input. The output layer has 10 possible outcomes, which are probability. Image from Zhang et al. (2021a), https://d21.ai/chapter_convolutional-neural-networks/lenet.html.

Convolutional Neural Networks. Convolutional neural networks (CNNs) (LeCun, Bengio and Hinton, 2015) are one of the most dominant DNNs in the vision domain and are designed for processing grid pattern data such as images (Goodfellow, Bengio and Courville, 2016). CNNs are able to adaptively learn spatial hierarchies

2.2 Deep Neural Networks

of features from low-to high-level patterns (Yamashita et al., 2018). Generally, CNNs are constructed by sequentially stacking different building blocks consisting of convolutional, pooling and fully connected layers. Each convolutional layer comprises a set of filters, named kernel, sliding over the image and performing a dot product operation between each filter with the pixel values in the region of an image covered by the filter. Pooling layers downsample or reduce the size of the feature maps, which are the outputs from convolutional layers, so that CNNs are more robust to variations in the input image. In a forward pass, as shown in Figure 2.2, the feature of an input image is extracted by convolutional and pooling layers, while fully connected layers combine and map the extracted features into outputs which are then used for classification tasks.

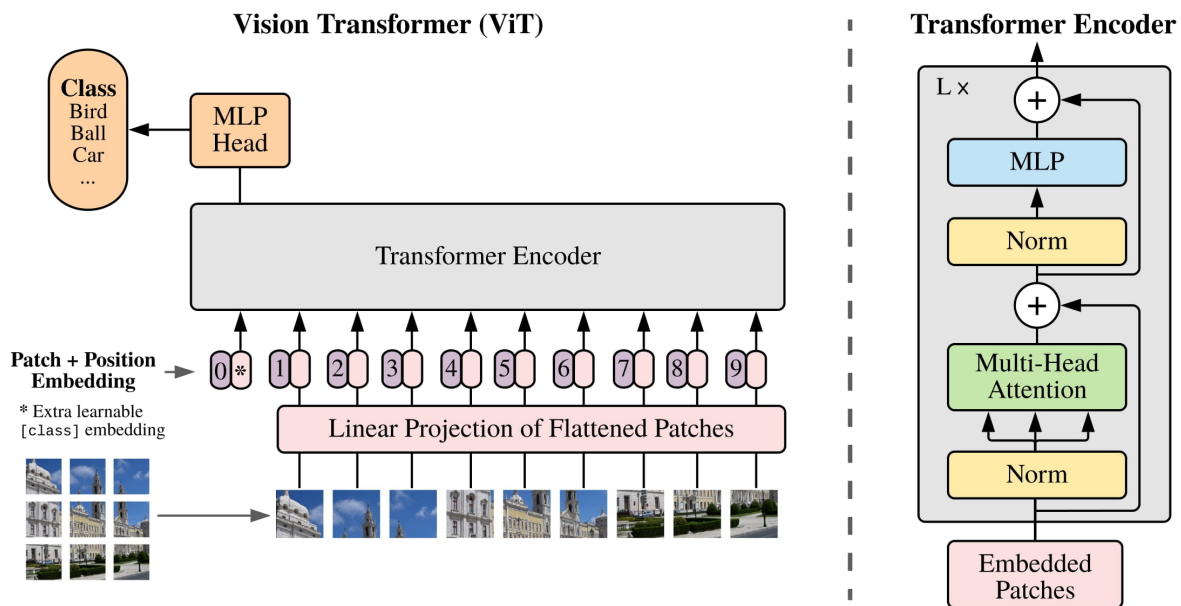


Figure 2.3. An illustration of vision transformer (ViT). The ViT comprises an embedded patch block and a transformer encoder that is constructed by multiple stacks of multi-head self-attention and feedforward neural network layers (Dosovitskiy et al., 2021). Image from https://github.com/google-research/vision_transformer.

Vision Transformer. Vision transformer (ViT) (Dosovitskiy et al., 2021) is a new type of DNN that applies transformer architecture, which was originally designed for natural language processing (NLP), to vision tasks. The transformer architecture is based on attention mechanisms without any convolutional blocks. It comprises an embedded patch block, a stack of multi-head self-attention and feedforward neural network layers that allow the ViT to attend and capture the global dependencies between

different parts of the input, as well as map the input's features to higher-dimensional representations. In a forward pass, as shown in Figure 2.3, an input image is first split and flattened into fixed-size patches in an embedded patch block. Each patch has learnable positional embeddings to encode its position in the image. These patch embeddings are then sequentially fed into the transformer encoder to extract image features and classify the input image.

2.3 Adversarial Attacks

Adversarial attacks are a class of threats that intentionally deceive a victim model through such methods as hijacking the model's prediction or evading the model's recognition or classification. This attack was first investigated in (Szegedy et al., 2014) and has been studied by substantial research (Goodfellow, Shlens and Szegedy, 2014; Carlini and Wagner, 2017; Madry et al., 2018; Alzantot et al., 2019; Cheng et al., 2020). Generally, in the vision domain, adversarial attacks aim to craft an adversarial example to cause the model to misclassify an input image at test time by carefully adding an imperceptible perturbation to the input image. Concretely, this malicious objective for *targeted attacks* in *white-box* and *score-based* settings across different perturbation regimes can be formulated as the following:

$$\mathbf{x}^* = \arg \min_{\tilde{\mathbf{x}}} \ell(f(\tilde{\mathbf{x}}), y_{\text{target}}) \text{ s.t. } \|\mathbf{x} - \tilde{\mathbf{x}}\|_p \leq \epsilon, \quad (2.2)$$

where p is the norm, ϵ is the perturbation budget, ℓ is the loss function (typically *cross-entropy*), f is the network, x is the input, θ is the network parameter, and y_{target} is the desired class label—target class. However, for *untargeted attacks*, this objective is different and formulated as follows:

$$\mathbf{x}^* = \arg \max_{\tilde{\mathbf{x}}} \ell(f(\tilde{\mathbf{x}}), y) \text{ s.t. } \|\mathbf{x} - \tilde{\mathbf{x}}\|_p \leq \epsilon, \quad (2.3)$$

where y is the ground-truth label. In contrast to white-box and score-based settings, the malicious objective for *targeted attacks* in decision-based settings across different perturbation regimes can be formulated as follow:

$$\mathbf{x}^* = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{x} - \tilde{\mathbf{x}}\|_p \text{ s.t. } f(\tilde{\mathbf{x}}) = y_{\text{target}}, \quad (2.4)$$

Likewise, for *untargeted attack* in decision-based settings, the objective is formulated as the following:

$$\mathbf{x}^* = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{x} - \tilde{\mathbf{x}}\|_p \text{ s.t. } f(\tilde{\mathbf{x}}) \neq y, \quad (2.5)$$

2.4 Threat Models

This section presents the taxonomy of widely studied threat models of adversarial attacks as shown in Figure 2.7. Generally, a threat model can be categorized based on: i) adversary capabilities (*i.e.* prior knowledge and access levels to a deep learning model); and ii) similarity measure (*i.e.* l_p norm-constraints). The details of attack taxonomy are described in Section 2.4.1 and 2.4.2.

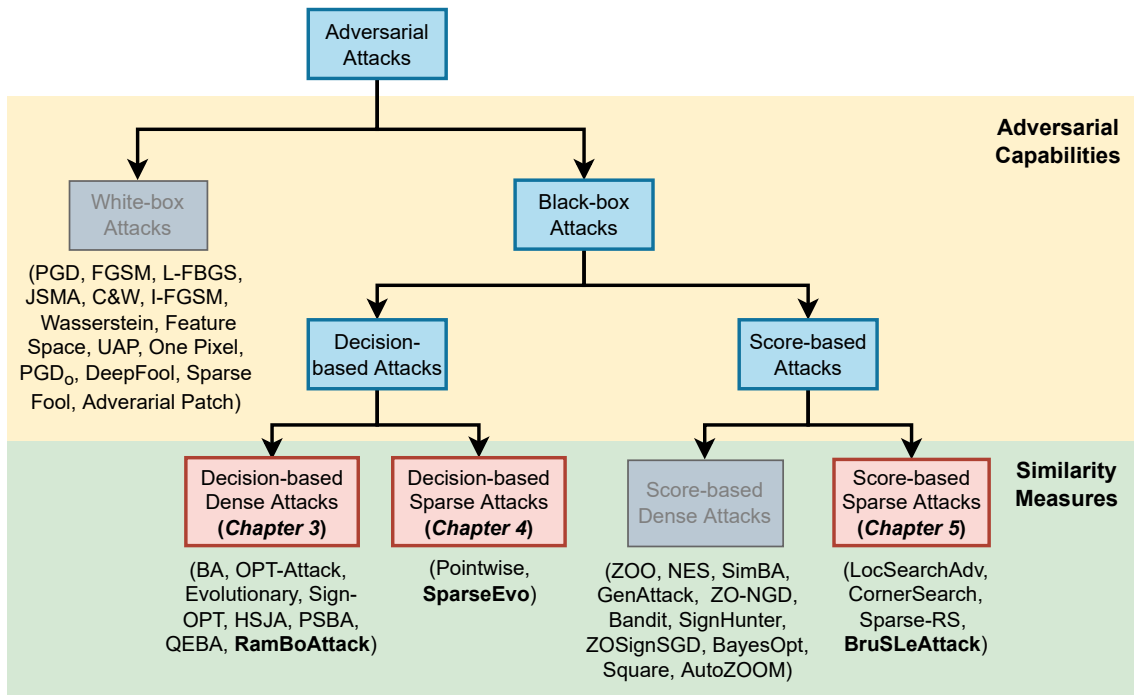


Figure 2.4. An attack taxonomy for various adversarial attack methods in different threat models. The red boxes indicate the threat models explored in this dissertation. The attack methods proposed in this dissertation are **RamBoAttack**, **SparseEvo** and **BruSLeAttack**.

2.4.1 Adversarial Capabilities

White-box Settings. An adversarial perturbation is an imperceptible noise when added to an input cause a failure—simply misclassifying the input in an *untargeted attack* or hijacking the decision of a model to generate a decision preselected by the adversary (Szegedy et al., 2014) in a *targeted attack*. In *white-box* attacks, adversaries may have full knowledge and access to the machine learning models (*i.e.* model architecture, parameters, weights or objective loss) to effectively generate adversarial

examples (Goodfellow, Shlens and Szegedy, 2014; Papernot et al., 2016a; Madry et al., 2018; Carlini and Wagner, 2017; Xu et al., 2019).

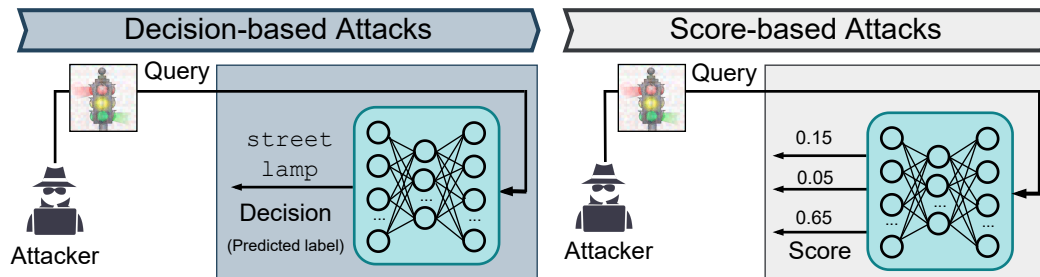


Figure 2.5. An illustration of black-box adversarial attacks categorized based on access level and knowledge including score-based and decision-based threat models.

Black-box Settings. In contrast to white-box attacks, on commercial and industrial systems, attackers have limited or no knowledge of the model (*i.e.* model architecture, parameters or weights). Access may be limited to only the full or partial output information of the models (*i.e.* a probability distribution, confidence score or even only top-K predicted labels). In practice, attackers can query a crafted adversarial example to extract the information returned from the target model and then exploit this information to achieve their objectives, as shown in Figure 2.5.

- **Decision-based Settings.** In some commercial and industrial machine learning systems the information exposed to an attacker is limited to the *hard-label* only—the most confident label predicted or *decision*, for instance logo or landmark detection on Google Cloud Vision. This is the *most* restricted threat model and recent studies (Brendel, Rauber and Bethge, 2018; Chen, Jordan and Wainwright, 2020) have demonstrated the practicability of black-box attacks under these restrictive *decision-based* settings (investigated in Chapters 3 and 4).
- **Score-based Settings.** Attackers in these settings may have access to full or partial output scores from a target model but no model knowledge is exposed to them (Chen et al., 2017; Ilyas et al., 2018) (investigated in Chapter 5).

Overall, score-based and decision-based settings are restrictive and challenging attack settings given the limited access to information but present a *practical threat model* for deployed systems. Moreover, the latter is particularly more threatening to model owners and applications because an adversary is still capable of exploiting

2.4.2 Similarity Measures

the very minimal information exposed—the *top-1 predicted label*—for constructing a perturbation.

2.4.2 Similarity Measures

The similarity measures— l_p norm—can be used to describe the imperceptibility of the perturbation and categorize adversarial examples quantitatively as illustrated in Figure 2.6. Particularly, l_2 and l_∞ norm is used to quantify dense perturbations whereas l_0 norm quantitatively describes sparse perturbation for adversarial attacks. Formally, the similarity measures l_p norm can be formulated as the following:

$$\|x\|_p = \left(\sum_i^n \|x_i\|^p \right)^{\frac{1}{p}}, \quad (2.6)$$

where n is the number of elements of x .

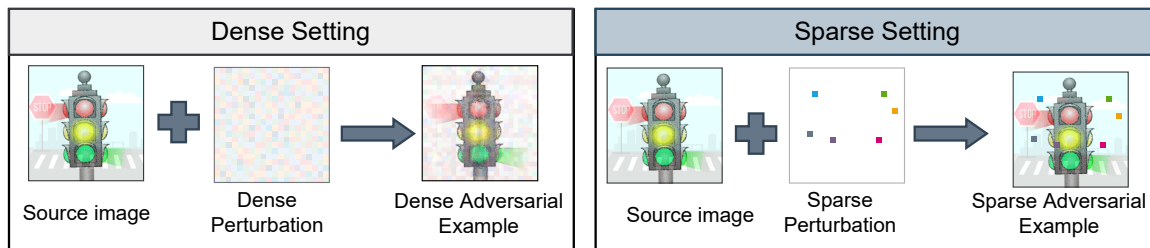


Figure 2.6. An illustration of black-box adversarial attacks categorized based on similarity measures including dense and sparse settings.

- **Dense Settings.** The attacks in dense settings aim to search for a dense adversarial perturbation whose all pixels are altered to fool a victim model. There is a large body of work investigating dense attacks (Carlini and Wagner, 2017; Papernot et al., 2017; Ilyas et al., 2018; Alzantot et al., 2019; Li et al., 2021a; Zhang et al., 2021b), (investigated in Chapter 3).
- **Sparse Settings.** The main aim of sparse attacks is to minimize the number of perturbed pixels required to mislead a target machine learning model. Only a handful of works have investigated sparse attacks and these works can be broadly categorized based on various degrees of adversarial access to a model (*i.e.* white-box, score-based or decision-based settings) (Modas, Moosavi-Dezfooli and Frossard, 2019; Croce and Hein, 2019; Schott et al., 2019; Croce et al., 2022) (investigated in Chapter 4 and 5).

2.5 Adversarial Defense

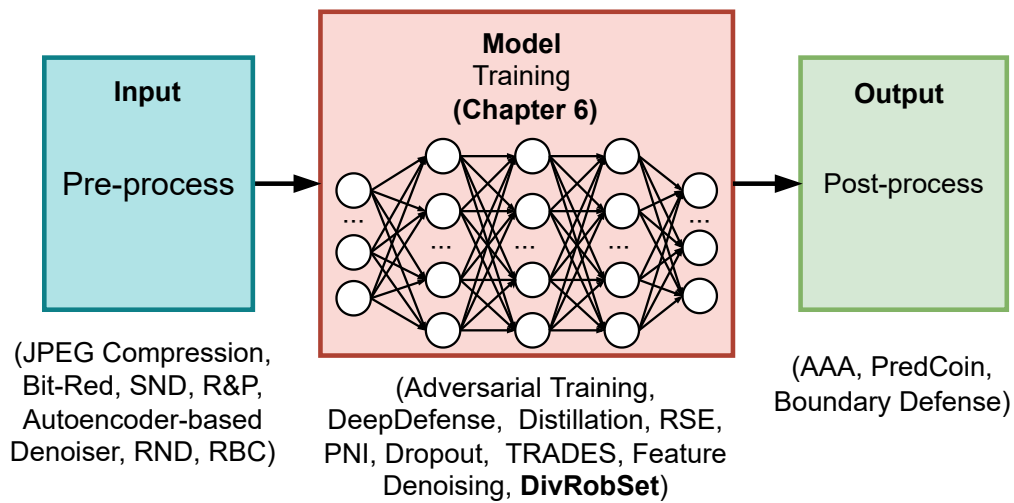


Figure 2.7. An illustration of various defense methods based on where they can apply in inference. The defense mechanism proposed in Chapter 6 is **DivRobSet**.

Due to the threats and potential impacts of adversarial attacks, particularly in white-box settings, significant research efforts have been dedicated to exploring and developing defense mechanisms against these attacks. These mechanisms encompass a wide range of approaches but we can differentiate them based on the area of focus along a deep learning pipeline. The methods focusing on pre-processing inputs (Cao and Gong, 2017; Liu et al., 2018b; Xu, Evans and Qi, 2018) aim to discard adversarial perturbations or make the output information incorrect to be exploited. Another line of research proposed to train a model to make it more robust against adversarial examples (Tramèr et al., 2018; Xie et al., 2019; Zhang and Wang, 2019; Zhang et al., 2022; Wang and Wang, 2022) while the approaches concentrating on post-processing output scores (Chen et al., 2022a; Aithal and Li, 2022) aim to misguide attackers.

In pre-process approaches, defenders transform or manipulate the inputs before feeding them into the model. These methods aim, firstly, to clean or alter added perturbations that may exist in the inputs and, secondly, to mislead attackers when they exploit output information to create a perturbation. Similarly, post-processing approaches aim to misguide attack algorithms by purposefully modifying the output information (*i.e.* confidence scores). When the output information from a model is intentionally altered, it does not provide useful information for attack algorithms built to search for a perturbation and hampers the progress of attacks. In contrast

2.6 Data sets

to the pre- and post-processing methods, the principle of robust training methods is to enhance the model's robustness to adversarial inputs through model training with augmented data, new learning objectives or regularization methods. Although methods designed for white-box attacks provide a defense against black-box attack methods, the proposed methods trade off robustness for clean accuracy (Tsipras et al., 2019; Qin et al., 2021). Interestingly, while most studies focus on white-box attacks, research aiming to study the problem of developing defenses to black-box attack methods, in particular, has only recently emerged (Pang et al., 2020; Qin et al., 2021). The second research objective in this dissertation is to develop a robust mechanism to defend against *black-box* adversarial attackers.

2.6 Data sets

This section describes various data sets in the vision domain used in the studies conducted for this dissertation. These data sets have different sizes, resolutions and number of classes.

- MNIST. (Lecun et al., 1998) MNIST stands for Modified National Institute of Standards and Technology and is a widely used benchmark data set in the field of machine learning and computer vision. This data set consists of a large collection of gray-scale images, handwritten digits from 0 to 9. Each image in MNIST has a resolution of 28×28 pixels. The data set has a training set (60,000 examples) and a test set (10,000 examples) for training and evaluating the performance of machine learning algorithms. This data set is used in Chapter 6.
- CIFAR-10. (Krizhevsky, Nair and Hinton, n.d.) CIFAR-10 stands for the Canadian Institute for Advanced Research 10 and is commonly used for image classification tasks. CIFAR-10 also serves as a benchmark for evaluating the performance of various machine learning algorithms and provides a more challenging task than MNIST due to the complexity of color images. This data set consists of 60,000 color images in 10 classes which are mutually exclusive. Each class has 6000 images and each image has a resolution of 32×32 . The data set is divided into a training set (50,000 images) and a test set (10,000 images) for training and evaluating different machine learning algorithms. This data set is used in Chapter 3, 4, 5 and 6.

- **STL-10.** (Coates, Lee and Ng, 2011) STL-10 stands for Stanford Large-Scale 10-class and is a popular image data set for developing unsupervised feature learning, deep learning and self-taught learning algorithms. This data set has ten classes and is divided into a training set (5,000 labeled images) and a test set (8,000 labeled images). Moreover, to serve unsupervised learning methods, it provides 100,000 unlabeled images extracted from a similar but broader distribution of images. The primary challenge is to make use of the unlabeled data to build a useful image model prior to supervised training. This data set has a higher resolution than CIFAR-10 (96×96) and is a more challenging benchmark for developing more scaleable unsupervised learning methods and advanced learning algorithms. This data set is used in Chapter 5 and 6.
- **ImageNet.** (Deng et al., 2009) ImageNet is a large-scale visual database spanning 1000 object classes and provides a comprehensive data set that covers a wide range of objects and scenes. This data set is divided into a training set (1,281,167 images), a validation set (50,000 images) and a test set (100,000 images) for training and evaluating different deep learning algorithms. The data set is widely used for object recognition and image classification tasks and has played a crucial role in the development and advancement of deep learning algorithms for image understanding. This data set is used in Chapter 3, 4, 5. In addition, ImageNet-10 is a subset of ImageNet with 10 classes and is used in Chapter 6.

2.7 Evaluation Metrics

This section describes various evaluation metrics used in Chapter 3, 4, 5 and 6 to evaluate the efficiency of different attack methods and the robustness of different defense mechanisms.

Accuracy. Accuracy (Acc) for a single model without attack, known as clean accuracy or standard accuracy, is calculated as the following:

$$\text{Acc} = \frac{c}{\text{Total}}, \quad (2.7)$$

where *Total* represents the total number of samples from the evaluation set, and *c* is the number of correct predictions.

Attack Success Rate (ASR). It is calculated as the following:

$$\text{ASR} = \frac{m}{M}, \quad (2.8)$$

2.7 Evaluation Metrics

where m is the number of incorrect predictions and M denotes the number of samples used for the attack evaluation.

Robustness. This metric quantifies the accuracy of a Deep Learning model under attacks. The metric is formulated as follows:

$$\text{Robustness} = \text{Acc}(x_{\text{adv}}), x_{\text{adv}} \sim \mathcal{D}_{\text{ADV}} \quad (2.9)$$

where x_{adv} denotes an adversarial example from a set of adversarial examples \mathcal{D}_{ADV} .

Sparsity Level. Sparsity level is the number of altered pixels over the total number of pixels of an input image and is computed as follows:

$$\text{Sparsity} = \frac{r}{R}, \quad (2.10)$$

where r represents the number of altered pixels while R denotes the resolution (*i.e.* the number of pixels) of an input image.

Chapter 3

RamBoAttack: A Dense Attack Under Decision-Based Settings

THIS chapter considers the problem of designing a query-efficient dense adversarial attack— l_2 norm-constraint—in a decision-based setting. Machine learning models are critically susceptible to evasion attacks from adversarial examples. Recent black-box attacks which require *only* the predicted label from a model (distinguished as a *decision-based* attack) have shown a remarkable reduction in the number of queries to craft adversarial examples. Particularly alarming is the *practical* ability to exploit the classification decision (*hard label*) from a trained model’s *access interface* provided by a growing number of Machine Learning as a Service (MLaaS) providers (*i.e.* Google, Microsoft or IBM) and used by a plethora of applications. The study in this chapter highlights the costly nature of discovering low distortion adversarial employing approximate gradient estimation methods. It then introduces a *robust query efficient* attack—BLOCKDESCENT—capable of avoiding entrapment in a local minimum and misdirection from noisy gradients seen in gradient estimation methods. The proposed attack method exploits the notion of Randomized Block Coordinate Descent to explore the hidden classifier manifold, targeting perturbations to manipulate only localized input features to address the issues of gradient estimation methods. Overall, for a given target class, BLOCKDESCENT is demonstrated to be more robust at achieving a lower distortion within a query budget.

3.1 Motivation and Contribution

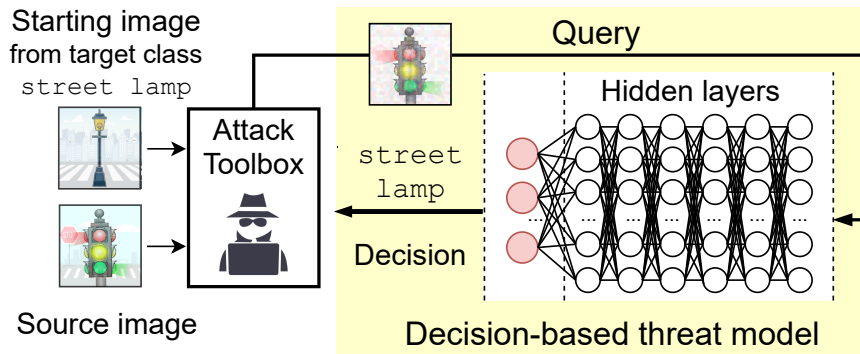


Figure 3.1. An illustration of black-box attack in the severely restricted threat model of a decision-based attack. In a decision-based threat model, an adversary with a source image and starting image from the target class, craft a sample, queries the model and observes the decision returned by the model.

In black-box scenarios, without access to model gradient and model knowledge, attacking machine learning systems is considerably challenging. To tackle these challenges, recent studies have formulated the decision-based attack as an optimization problem to propose algorithms based on gradient estimation methods (Cheng et al., 2020; Chen, Jordan and Wainwright, 2020) and demonstrated attacks with significantly fewer query numbers. However, the existing attacks suffer from the following problems:

- *Entrapment in local minima.* In gradient estimation methods, as eluded to by (Cheng et al., 2020), the search for an adversarial example can experience an entrapment problem in a local minimum where extra queries expended by the attacker fails to achieve a lower distortion adversarial example.
- *Unreliability of gradient estimations.* Further, as the magnitude of estimated gradients diminishes on approach to a local minimum or a plateau, the estimated gradients may become noisy and susceptible to misdirection.
- *Sensitivity to the starting image.* Then, intuitively, it can be expected that the initialization of optimization frameworks with an *available* or intended starting image, a *necessity* in decision-based attacks, to hinder an attacker from reaching an imperceptible adversarial example. But, there is no known method to determine a *good starting image prior to an attack*. Thus, the success of an attack can

be expected to be sensitive to the available starting image; an attempt to discover a better-starting image or target class through *trial and error* can not only lead to detection and discovery by effectively increasing the numbers of queries needed but also limit the scope of the attack by reducing the number of classes that can be targeted.

In general, developing decision-based attacks poses a challenging optimization problem because only binary information from output labels are available to us from the target model as opposed to output values from a function.

Therefore, the study in this chapter seeks to understand the fragility of gradient estimation methods and develop a more robust and query-efficient attack. Consequently, the study aims to answer the following research questions (RQ).

RQ1: How can we assess the robustness of decision-based black-box attacks to understand their fragility? (Section 3.2.3)

RQ2: What is the impact of the source and starting target class images accessible to an adversary on the success of an attack? (Section 3.2.4 & extensive results in Section 3.4.6)

RQ3: How can an adversary construct a robust and query-efficient attack for achieving low distortion adversarial examples for any starting image from the targeted class and avoid the pitfalls of gradient estimation-based attack methods? (Section 3.3 & 3.4)

Main Contributions. This chapter aims to: i) address the research questions; ii) better understand and assess the vulnerabilities of DNNs to adversarial attacks in the pragmatic decision-based threat model; and iii) explore more robust attack methods. The contributions of this chapter are summarised below:

1. The study presents the *first* systematic investigation of state-of-the-art decision-based attacks to understand their robustness. Through extensive experiments, this study highlights the problem of *hard* cases where attackers struggle to flip the prediction of images towards a chosen target class, even with increasing query budgets—see Figure 3.2. As summarized in Table A.7, all comprehensive experiments in this study consume over 1800 computation hours with 2 GPUs to curate results.

3.1.1 Chapter Overview

2. Motivated by the findings, the study *proposes a new attack*—RAMBOATTACK—that is demonstrably more robust. A search algorithm analogous to Randomized Block Coordinate Descent—BLOCKDESCENT—is proposed to address the entrapment problem where gradient estimation fails to provide a useful direction to descend and combine BLOCKDESCENT with gradient estimation frameworks to attain query efficiency. In contrast to existing approaches, BLOCKDESCENT focuses on altering local regions of the input commensurate with the filter sizes employed by DNNs to forge adversarial examples.
3. The study provides new insights into query-efficient mechanisms for crafting adversarial perturbation to attack DNNs. The proposed decision-based black box attack method relying on localized alterations to inputs discovers effective adversarial perturbations attempting to exploit the model’s reliance on salient features of the target class to correctly classify an input to a target label in the *hard* cases. Clear correlations between perturbations found and added to inputs, and salient regions on target class images are illustrated with the aid of a visual explanation tool.
4. Overall, RAMBOATTACK is a more robust and query-efficient approach for generating an adversarial example of a high attack success rate compared to existing counterparts. Importantly, the proposed attack method is *significantly less impacted by a starting image* from a target class accessible to an adversary.
5. The need for reliable and reproducible attack evaluation strategies is recognized and two evaluation protocols applied across CIFAR10 and ImageNet is introduced. *The dataset constructed through our extensive study is released to support future benchmarking of black-box attacks* under a decision-based setting.

3.1.1 Chapter Overview

Section 3.2 presents a threat model, problem formulation, robustness understanding and intuition into the proposed attack method in this chapter; Section 3.3 details the proposed attack framework and an end-to-end implementation; Section 3.4 evaluates and analyzes the performance of different attacks across different datasets. Section 3.5 concludes this chapter.

3.2 Investigation of Decision-Based Attacks

This section: i) formalizes an adversarial attack as an optimization problem; ii) revisits current state-of-the-art methods; and iii) analyzes the results to present our intuitions into the state-of-the-art attacks based on our observations.

3.2.1 Adversarial Threat Model

We adopt the threat model proposed in prior works (Brendel, Rauber and Bethge, 2018; Cheng et al., 2019a; Chen, Jordan and Wainwright, 2020). Under the decision-based black-box setting, adversaries have no prior knowledge such as model architecture or parameters but have limited access to the output of a victim model—the *model’s decision* as illustrated in Figure 3.1. Furthermore, an adversary can make numerous queries to a victim’s machine learning model via an access interface and receive the model’s decision. The adversary must have *at least one image from a target class that is classified correctly by the victim model if the adversary aims to carry out a targeted attack*. This image is the *starting image* used to initialize the attack. The adversary also holds at least one image from a *source class* correctly classified by the model. The *objective* of the adversary is to discover the minimum (imperceptible) perturbation—quantitatively measured by the common distortion measure adopted in recent studies—to flip the decision for the source image to the targeted class using the minimum number of queries to the model.

3.2.2 Problem Formulation

Given a source image $x \in \mathbb{R}^{C \times W \times H}$ its ground truth label y from the label set $\mathcal{Y} = \{1, 2, \dots, K\}$, where K denote the number of classes, C , W and H denotes the number of channels, width and height of an image, respectively. Given a pre-trained multi-class classification model $f : x \rightarrow y$ so that $f(x) = y$, in a targeted attack, an adversary aims to modify an input x to craft an optimal adversarial example $x^* \in \mathbb{R}^{C \times W \times H}$ that is classified as the class label desired by the adversary when used as an input for the victim model. In an untargeted attack, an adversary manipulates input x to change the decision of a classifier to any class label other than its ground-truth label. To simplify the descriptions, we refer to the desired class label as the *target class* while the class of the input x is called the *source class*.

3.2.2 Problem Formulation

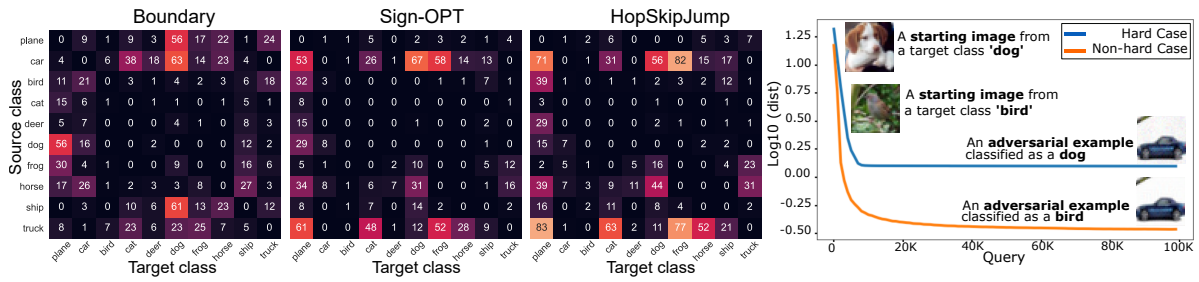


Figure 3.2. (Left) The number of *hard* cases from CIFAR10 found for Boundary Attack (BA), Sign-OPT and HopSkipJump categorized by different pairs of a source and target class (starting image) at a distortion threshold = 0.9 and a query budget of 50,000. (Right) The line chart illustrates a significant difference between a *hard* versus *non-hard* case—*interestingly increasing the query budget to even 100,000 does not yield a lower distortion adversarial example.*

Measuring Distortion. l_2 -norm is widely adopted, in *all of the recent works* as in (Brendel, Rauber and Bethge, 2018; Brunner et al., 2019; Cheng et al., 2019b, 2020; Chen et al., 2017), to measure the distortion and similarity between a generated adversarial example and the source sample. Therefore, in this chapter, our approach focuses on l_2 -norm. Then, let $D(x, x^*)$ be the l_2 -distance that measures the similarity between x and x^* .

Optimization Problem. The main aim of adversarial attacks is to minimize the distortion measured by D while ensuring the perturbed input data is classified as a target class—for a targeted attack—or non-source class—for an untargeted attack. Therefore, an adversarial attack can be formulated as a constrained optimization problem:

$$\begin{aligned}
 \min_{x^*} \quad & D(x, x^*) \\
 \text{s.t.} \quad & \mathcal{C}(f(x^*)) = 1, \\
 & x, x^* \in [0, 1]^{C \times W \times H},
 \end{aligned} \tag{3.1}$$

Here, $\mathcal{C}(f(x^*))$ is an adversarial criterion that takes the value 1 if the attack requirement is satisfied and 0 otherwise. This requirement is satisfied if $f(x^*) \neq y$ for an untargeted attack or $f(x^*) = y^*$ for a targeted attack (i.e. for the instance x^* to be misclassified as targeted class label y^*).

3.2.3 Understanding Robustness

The two current query-efficient attack methods employ gradient approximation frameworks, whilst the earlier method relied on a stochastic approach. We briefly summarize these methods before delving into our systematic study to understand their robustness.

Random Walk along a Decision Boundary. The first attack under a decision-based threat model proposed by Brendel et al. (Brendel, Rauber and Bethge, 2018) initialized an image in a target class and in each iteration, sampled a perturbation from a Gaussian distribution and projected the perturbation onto a sphere around a source image. If this perturbation yields an adversarial example, the attack makes a small movement toward the source image and repeats these steps until the decision boundary is reached. Subsequently, by traveling along the decision boundary based on sampling, projecting and moving toward the source image, the adversarial example is refined until an adversarial example with a desirable distortion is discovered.

Optimization Frameworks. In the absence of a means for computing the gradient for solving Equation 4.1, the attacks in (Cheng et al., 2019b) and (Cheng et al., 2020) attempt to solve the optimization problem using methods to estimate the gradient. (Cheng et al., 2019b) samples directions from a Gaussian distribution and applies a zeroth-order gradient estimation method in their OPT-attack, then (Cheng et al., 2020) leveraged their former optimization framework and proposed a zeroth-order optimization algorithm called Sign-OPT that is much faster to converge. (Chen, Jordan and Wainwright, 2020) introduced a different optimization framework named HopSkipJumpAttack using a Monte Carlo method to find the approximate gradient direction to descend.

Evaluating Robustness. To understand the robustness of recent attack methods and illustrate the costly nature of discovering low distortion adversarial examples with these attacks, we propose an *exhaustive but tractable* experiment using the relatively small number of classes albeit with a significantly large validation set offered by CIFAR10 dataset. The protocol for assessing the robustness of each state-of-the-art method described is carefully described in Section 3.4.3.

Hard Cases. Empirically, we define a *hard* case as a pair of source and starting images—the starting image is from a given target class—where a given decision-based

3.2.4 Observations from Assessing Attacks

attack fails to yield an adversarial example with distortion below a desirable threshold using a set query budget.

3.2.4 Observations from Assessing Attacks

We make the following observations from our results summarized in Figures 3.2 and 3.3.

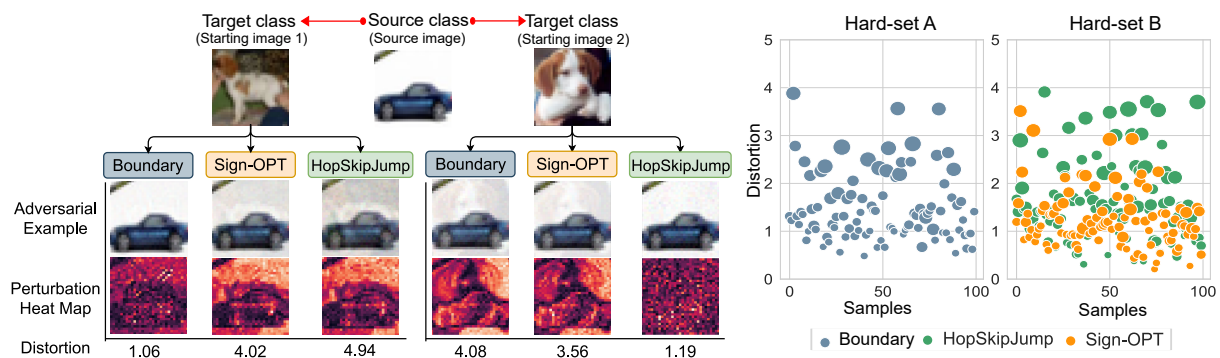


Figure 3.3. (Left) Consider an attack to discover an adversarial example for the source image of class car such that the car is predicted as belonging to the target class dog. We demonstrate the very different results an adversary can obtain based on the availability of a target class *image 1* and *image 2*. The attacker initializes the attack for Boundary, starting *image 1* is a better initialization. In contrast, Sign-OPT and HopSkipJump discover better adversarial examples if they are initialized with starting *image 2*. (Right) The scatter plot illustrates this attack scenario with 100 samples randomly selected from their own *hard* set. The *y*-axis denotes the average distortion and the size of each bubble denotes the variation in distortion for each source image with respect to 10 different starting images from hard targets. It shows that all these methods are highly dependent on a starting image in *hard* cases.

Observation 1: Hard Cases. *In decision-based attacks, specific classes and/or samples from classes are more difficult to attack than others. As illustrated in Figure 3.2 (left), the current attack methods are not uniformly effective against all pairs of source and starting images from target classes. Interestingly, any of the gradient estimation methods can approximate the true gradient given enough queries (or samples) to the target model. However, solutions can become entrapped in various local minima. Further, approaching a local minimum or a plateau can considerably undermine the quality of that approximation; for instance, estimated gradients may become noisy when the*

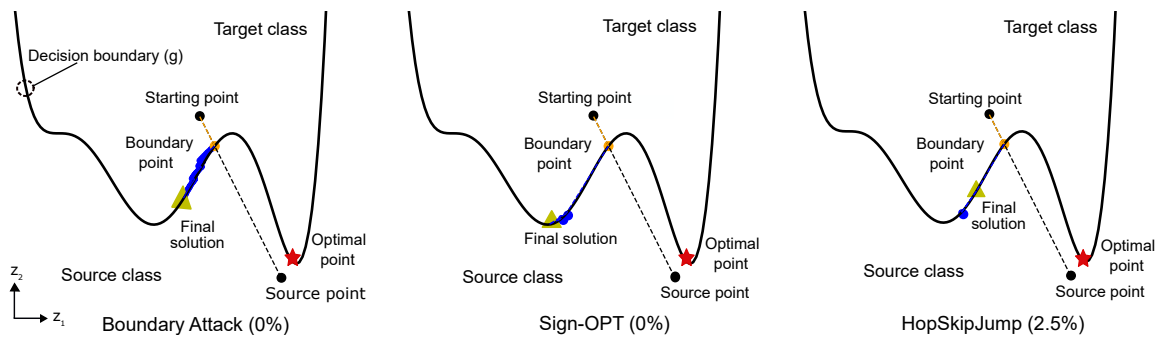


Figure 3.4. 2D (z_1 and z_2) Input Space Example. An illustration of the execution of the three different decision-based attack methods (Boundary, Sign-OPT and HopSkipJump) to attack a toy model employing 2D inputs. The attacks result in different final solutions denoted by a yellow triangle (\blacktriangle). We executed the algorithms 100,000 times; both Boundary Attack and Sign-OPT failed to find the global minimum (the Optimal Point *closest* to the Source point) and HopSkipJump only found the global minimum 2.5% of the time. This illustrates the *problem faced by current attack methods* when attacking a machine learning model *whose decision boundary in the input space is multi-dimensional and highly complex* for realistic and practical image inputs.

gradient magnitude diminishes whilst approaching a local minimum. As shown in Figure 3.2 (right), even with 100K queries, the solutions based on the gradient direction estimation methods do not improve the distortion of the adversarial sample for the car classified as a dog (Hard case).

Observation 2: Attack initialization. *An attack algorithm’s ability to find a low distortion adversarial example with a given query budget is dependent on the starting image from a target class selected for initializing the attack algorithm.* Interestingly, (Chen, Jordan and Wainwright, 2020) in their S&P2020 paper briefly noted the potential for an algorithm to get trapped in a bad local minimum based on the starting image used to initialize an attack. Our systematic study confirms this intuition.

In this case, the achievable distortion of an adversarial example is highly dependent on the starting image and the behavior of the algorithm. This observation is illustrated by comparing the results of starting *image 1* with *image 2* for different attack methods in Figure 3.3 and by 100 samples randomly selected from the *hard* set of each method—see Section 3.4.5 and 3.4.6 for more details. *The results demonstrate the dependence of attack success on the starting image accessible to an adversary.*

3.2.5 An Intuition into Attack Methods

Currently, there is no effective initialization method to determine a good starting image, prior to mounting an attack. Therefore, developing a robust attack that is less sensitive to the choice of starting image remains an open challenge.

Observation 3: Perturbation Region. *Current attack approaches aim to perturb the whole image to traverse the decision boundary to find an adversarial example with minimum distortion. In other words, these methods always manipulate the whole image at a time and result in perturbations that is spread over the entire image as illustrated by perturbation heat maps in Figure 3.3. Another interesting remark drawn from these figures is that the main features (for example edges) of the starting image remains super-imposed in an adversarial example. However, most of the state-of-the-art classifiers in computer vision utilize convolutional filters to extract local patterns in an image; further, visual explanation tools demonstrate the reliance of classifiers on key salient features of an image. Therefore, whether an attack could achieve a lower distortion adversarial example by targeting the filter operation over local features in contrast to manipulation of the whole image remains an interesting direction to explore.*

3.2.5 An Intuition into Attack Methods

To understand and illustrate the underlying cause of the first two observations, this study uses Boundary attack (BA) (Brendel, Rauber and Bethge, 2018), Sign-OPT (Cheng et al., 2020) and HopSkipJump (Chen, Jordan and Wainwright, 2020) to attack a Toy model. The *decision boundary* of the Toy model in a 2D *input space* illustrates a constraint of the optimization problem in Equation 4.1. This decision boundary is represented by $g(z_1, z_2) = (z_1 - 2)(z_1 - 1)^2(z_1 + 1)^3 - z_2 = 0$ where z_1 and z_2 denote two coordinates of a point such as a starting point or a source point as illustrated in Figure 3.4. A point above the boundary is classified as in the target class; otherwise, it belongs to the source class. The black dot (●) *source point* denotes a source class example whilst the black dot (●) *starting point* denotes a starting target class example. All three methods are initialized with the same starting point, this study then employs the attacks to search for an adversarial point within the target class and closest to the source point; alternatively, the study aims to solve the optimization problem in Equation 4.1, where the objective is to minimize the l_2 distance to the source point subject to the constraint imposed by the decision boundary, using these attack algorithms.

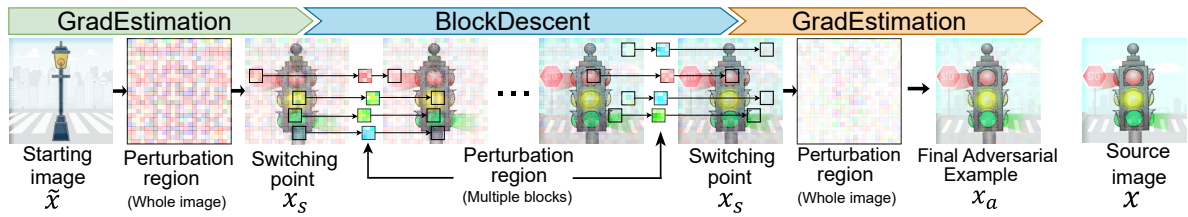


Figure 3.5. A pictorial illustration of RAMBOAttack to craft an adversarial example. In a targeted attack, the first component (GradEstimation) initializes an attack with a starting image from a target class (e.g. this study uses a clip art street lamp for illustration) and then manipulates this image to search for adversarial examples that looks like an image from source class e.g traffic light. The attack switches to the second component, BLOCKDESCENT, when it reaches its own local minimum. BLOCKDESCENT helps to redirect away from that local minimum by manipulating multiple blocks—or making local changes to the current adversarial example. Subsequently, the adversarial example crafted by BLOCKDESCENT is refined by the third component (GradEstimation).

Figure 3.4 illustrates several intermediate adversarial example points denoted by blue dots and a final adversarial example achieved by each method denoted by a yellow triangle for one example attack execution. Given the stochastic nature of the algorithms, this study executes each attack 100,000 times with different random seeds. All of the methods, except HopSkipJump, fail to find the optimal solution—global minimum—and HopSkipJump only managed to reach the optimal solution in 2.5 % of the attempts. As illustrated in Figure 3.4, the approximate gradient appears to be noisy and the methods traverse the decision boundary in an incorrect direction towards the local minimum rather than the global minimum. Although not illustrated here, changing the starting coordinate can lead all of these methods to discover the global minimum.

3.3 Proposed Attack Framework

The study in this chapter observes that: i) gradient estimation methods in attacks face an entrapment problem in a highly complex loss landscape; ii) current attacks focus on altering all of the coordinates of an image simultaneously to forge a perturbation; and iii) the success of current attacks are sensitive to the chosen or available starting image possessed by an adversary.

The study proposes an analogous Randomized Block Coordinate Descent method—named BLOCKDESCENT—that aims to manipulate local features and target

3.3 Proposed Attack Framework

convolutional filter outputs by modifying values of coordinates in a square-block region and in different color channels with targeted perturbations. The study proposes localized changes to affect convolutional filter outputs and pixel values as a means of impacting salient features and may even mimic salient features of the target. This leads to potential redirection and escapes from entrapment in a bad local minimum with minimal but effective changes to the image to mislead the classifier. In other words, this study proposes taking a direct path along some coordinates towards a source image whilst retaining the target class label to prevent the problem encountered by gradient estimation methods—entrapment in a local minimum as shown in Figure 3.4.

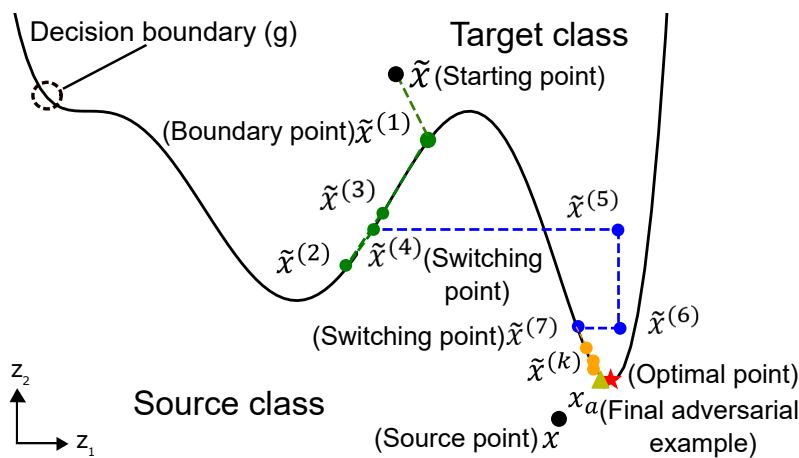


Figure 3.6. 2D (z_1 and z_2) Input Space Example. An illustration of our RAMBOATTACK against the toy model in Figure 3.4. If the first gradient estimation method—GRADESTIMATION in Algorithm 3.1—leads to entrapment in a local minimum—denoted by $\tilde{x}^{(1)}, \dots, \tilde{x}^{(4)}$ at the start—there is no effective mechanism to escape. However BLOCKDESCENT moves away from the local minimum. This is illustrated by $\tilde{x}^{(5)}, \dots, \tilde{x}^{(7)}$ when the number of modified coordinates is one in the 2D input space. Subsequently, the third component applying a gradient estimation method searches for a better adversarial example $\tilde{x}^{(k)}$ in the neighborhood region and reaches the nearly optimal solution x_a . In contrast to results in Figure 3.4, when evaluating RAMBOATTACK over 100,000 runs against the Toy model, the study observed the proposed attack to **always** find the optimal or near optimal solution.

Further, when employing gradient estimation methods, the gradient values decrease as the crafted adversarial example moves closer to the source image leading to the increasingly larger number of perturbations needed to converge. This issue is exacerbated if there is a plateau in the decision boundary; now the gradient estimation methods are as effective as a random search. We conjecture that the *hard* cases are examples of where the gradient of the distortions is generally small and, thus, leads to

local optima. However, we observe that the gradient estimation methods are effective in two cases: **(a)** initial stages of optimizing Equation (4.1) or **(b)** at close proximity to the source image. In (a), the gradients are sufficiently large to be estimated effectively, and in (b) small changes and refinements (*i.e.* few perturbation iterations) facilitate a decent to the optimum.

Consequently, we propose a new framework using gradient estimation for the initial descent—case (a)—supported by BLOCKDESCENT to escape entrapment and noisy gradient problems and refining the adversarial example supported by a gradient estimation-based descent to forge a robust and query-efficient attack. Importantly, BLOCKDESCENT is *insensitive to the choice of starting images*, although it is effectively initialized with a gradient estimation, because BLOCKDESCENT manipulates blocks that cause a move away from the direction set by a starting image. The new framework we propose, RAMBOATTACK, is illustrated in Figure 3.5.

Summary. *Gradient estimation methods are fast but face the potential problem of getting trapped in a bad local minimum, particularly in hard cases. BLOCKDESCENT, on the other hand, is slower—selecting to manipulate local regions—but is capable of tackling the problems faced by gradient estimation attacks. Therefore, we develop a hybrid framework called RAMBOATTACK for query-efficient decision-based attacks that can exploit the merits of both approaches. In particular, our derivative-free optimization method considers, for the first time, an approach to manipulate blocks of coordinates in the input image to influence the outcome of convolution operations used in deep neural networks as a means for misguiding a networks decision and generating adversarial examples with minimal manipulations.*

3.3.1 Approach

Our proposed attack thus comprises BLOCKDESCENT and two components of gradient estimation—GradEstimation—as shown in Figure 3.5 and described in Algorithm 3.1. The gradient estimation algorithms used by these two components can be the same or different from each other. When starting an attack, particularly in targeted settings, the first component is initialized with a starting image \tilde{x} from a target class and approaches the decision boundary via a binary search—the first step in a gradient estimation method. We employ the gradient estimation method to search for adversarial examples until reaching its own local minimum. We call it a switching point x_s because, from this point, the gradient estimation method switches to BLOCKDESCENT. If the gradient

3.3.2 BlockDescent

estimation method is entrapped in a local minimum, BLOCKDESCENT helps to move away from that local minimum. Subsequently, when local changes are insufficient, the attack switches to the third component to refine the adversarial example crafted by BLOCKDESCENT which is considered as the second switching point. This refinement aims to search for the final adversarial example x_a with lower distortion.

Figure 3.6 illustrates RAMBOATTACK against the Toy model used in Section 3.2.5 and demonstrates the effectiveness of the attack we propose. Particularly, the first gradient estimation approach searches for and reaches the adversarial examples $\tilde{x}^{(1)}$, $\tilde{x}^{(2)}$, $\tilde{x}^{(3)}$ at different steps towards approaching the source point but is stuck at $\tilde{x}^{(4)}$ which is a local minimum of the objective function $D(x, x^*)$ subject to the constraint defined by the decision boundary $g(z_1, z_2)$. Henceforth, BLOCKDESCENT searches for next adversarial examples $\tilde{x}^{(5)}, \dots, \tilde{x}^{(7)}$ by modifying one coordinate at a time—in this 2D example—by applying δ changes. Subsequently, the second gradient estimation method continues searching for adversarial examples $\tilde{x}^{(k)}$ in the neighborhood areas until reaching the near optimal x_a . Most importantly, in contrast to the experiment in Figure 3.4 when evaluating RAMBOATTACK over 100,000 attacks on the Toy model, our proposed attack always reached the optimal or near optimal solution.

When to switch to BLOCKDESCENT? The gradient estimation methods are designed to work alone rather than with other methods. Therefore, we develop a sub-module GRADESTIMATION to call these methods and determine when to switch from a gradient estimation method to BLOCKDESCENT. Empirically, gradient estimation methods reach their local minimum when they cannot find any better adversarial example after several steps of searching. In practice, this can be determined by the distortion reduction rate Δ after every T queries—a time frame to calculate Δ . However, in gradient estimation methods, the number of queries per iteration is varied so we relax this by accumulating the number of queries after each iteration. Whenever it exceeds T , we compute Δ and if this distortion reduction rate is below a switching threshold ϵ_s , it switches to BLOCKDESCENT (see Algorithm 3.2).

3.3.2 BlockDescent

We recognize that the architecture of most machine learning models in computer vision is based on a Convolutional Neural Network (CNN) built on convolution operations. These convolution operations are defined as $c \times q \times q$ where q is the size

Algorithm 3.1: RAMBOATTACK

Input: source image x , starting image \tilde{x} , model f
 gradient estimation function g_1, g_2 , reduction scale λ ,
 input dimensions N , square extension n ,
 block number m , query number T_1, T_2

- 1 $x_s \leftarrow \text{GRADESTIMATION}(x, \tilde{x}, f, g_1, T_1)$
- 2 $x_s \leftarrow \text{BLOCKDESCENT}(x, x_s, f, \lambda, N, n, m)$
- 3 $x_a \leftarrow \text{GRADESTIMATION}(x, x_s, f, g_2, T_2)$
- 4 **return** x_a

of the filter and c is the number of channels to extract local patterns of an image. Consequently, we hypothesize that altering a block of coordinates as a square-shaped region with an appropriate size can target significant filter outputs potentially having a significant impact on the network’s decision. Perturbing these coordinates can result in an adversarial example with fewer queries since we target regions of the input to impact actual convolutional filters and potentially discover salient features to mimic. Inspired by this, we adopt a notion of square-block perturbation regions and introduce BLOCKDESCENT that manipulates blocks of size n . BLOCKDESCENT has two stages: i) crafting a sample; and ii) its evaluation as described in Algorithm 3.3.

Crafting a Sample. In each iteration, the first stage of BLOCKDESCENT aims to yield a sample x' that is initialized with $x^{(k)}$ which is an adversarial example at k -th step. To increase the convergent rate and reduce query number, BLOCKDESCENT modifies several blocks of coordinates concurrently. It firstly selects m different coordinates across different channels (R, G, B) of an image by choosing a set $S = \{S_1, S_2, \dots, S_m\}$ where $S_t = \{c_t, w_t, h_t\}$ is selected uniformly at random such that $c_t \in [1, C]$, $w_t \in [1, W]$ and $h_t \in [1, H]$, where $t = 1, 2, \dots, m$ and C, W, H denote the number of channel, width and height of an image. This random selection is sampling without replacement and each selected coordinate $x'_{c,w,h}$ is a center of a square block x'_{B_t} , where x'_{B_t} represents $x'_{[c_t, w_t - n : w_t + n, h_t - n : h_t + n]}$. Likewise, m corresponding blocks x_{B_t} are yielded from the source image x . A mask M with the same size as x'_{B_t} can be defined as $M = \text{sign}(x_{B_t} - x'_{B_t})$. This mask is used to identify the direction of perturbation for each element of a block x'_{B_t} . When each element of a block which is a coordinate of an image is manipulated to move along this direction, it tends to move towards to its corresponding element in the source image. The sample x' is crafted when each of m

3.3.2 BlockDescent

Algorithm 3.2: GRADESTIMATION

Input: source image x , switching image x_s , model f
gradient estimation function g , query number T

```

1  $n_q \leftarrow 0, switch \leftarrow False$ 
2  $d \leftarrow D(x, x')$ 
3 while not ( $switch$ ) do
4    $x', i \leftarrow g(f, x, x')$ 
5    $n_q \leftarrow n_q + i$ 
6   if  $n_q > T$  then
7      $\Delta \leftarrow d - D(x, x')$ 
8      $d \leftarrow D(x, x')$ 
9      $n_q \leftarrow 0$ 
10    if  $\Delta < \epsilon_s$  then
11       $switch \leftarrow True$ 
12 end while
13 return  $x'$ 

```

blocks of coordinates is updated as below:

$$x'_{B_t} \leftarrow x'_{B_t} + M \times \delta \quad (3.2)$$

Where δ is a scalar that denotes an amount of perturbation for each element and it reduces by λ after each cycle. One cycle is ended when all coordinates are selected for perturbation. If δ is initialized with a small value, it is slow convergent and results in query inefficiency from the beginning. Whilst, for large initial δ , modifying blocks of coordinate almost leads to a sample moving further from the source image from the beginning rather than moving closer. Consequently, it requires several cycles until δ reduces to a suitable value. To tackle this issue, we exploit the distribution of the absolute difference between all coordinates of a sample and their corresponding coordinate in a source image and use i -th percentile P_i of this distribution to specify a proper initial δ . In Equation 3.2, only selected square blocks are perturbed while the rest of \tilde{x} remains unchanged.

Evaluate Crafted Sample. In the second stage, to ensure a descent of distortion and improve query efficiency, a crafted sample x' is merely evaluated by the victim model if it moves closer to x . If the adversarial criteria is then satisfied ($\mathcal{C}(f(x')) = 1$), the

Algorithm 3.3: BLOCKDESCENT

Input: source image x , switching image x_s , model f
reduction scale λ , input dimension N
square extension n , block number m

- 1 $k \leftarrow 0, n_q \leftarrow 0, switch \leftarrow False$
- 2 $\delta \leftarrow P_1(|x - x_s|), \tilde{x}^{(k)} \leftarrow x_s, D_{n_q} \leftarrow D(x, \tilde{x}^{(k)})$
- 3 **while** *not* (*switch*) **do**
- 4 $j \leftarrow 0$
- 5 **while** $j < N$ *and* *not* (*switch*) **do**
- 6 /* Craft a new sample */
- 7 $x' \leftarrow \tilde{x}^{(k)}$
- 8 **for** $t = 1, 2, \dots, m$ **do**
- 9 Uniformly select a set $\{c, w, h\}$ at random without replacement
- 10 $x'_{B_t} \leftarrow x'_{[c, w-n:w+n, h-n:h+n]}$
- 11 $x_{B_t} \leftarrow x_{[c, w-n:w+n, h-n:h+n]}$
- 12 /* Perturbation region */
- 13 $M \leftarrow \text{sign}(x_{B_t} - x'_{B_t})$
- 14 $x'_{B_t} \leftarrow x'_{B_t} + M \times \delta$
- 15 **end for**
- 16 /* Evaluate crafted sample */
- 17 **if** $D_{n_q} > D(x, x')$ **then**
- 18 $n_q \leftarrow n_q + 1$
- 19 **if** $\mathcal{C}(f(x')) = 1$ **then**
- 20 $\tilde{x}^{(k+1)} \leftarrow x'$
- 21 $k \leftarrow k + 1$
- 22 $D_{n_q} \leftarrow D(x, \tilde{x}^{(k)})$
- 23 Compute Δ using Equation 3.3
- 24 **if** $\Delta < \epsilon_s$ **then**
- 25 $switch \leftarrow True$
- 26 $j \leftarrow j + m$
- 27 **end while**
- 28 $\delta \leftarrow \frac{\delta}{\lambda}$
- 29 **end while**
- 30 **return** $\tilde{x}^{(k)}$

3.4 Experiments and Evaluations

perturbation will make a change to update the next adversarial example as $\tilde{x}^{(k+1)} = x'$. Otherwise, the perturbation will be discarded.

Determining when to switch to the next component. Similar to the switching criterion of gradient estimation methods, BLOCKDESCENT should switch to the next component when it cannot find any better adversarial example that can be empirically measured by distortion reduction rate Δ per T queries. However, we observe that BLOCKDESCENT is a gradient-free optimization so Δ is highly varied for each subsequent query. As such we cannot simply apply the same criterion as gradient estimation methods. Consequently, to determine a better switching criterion for BLOCKDESCENT, we adopt a smoothing technique based on Simple Moving Average to measure the distortion reduction rate Δ . In practice, Δ is computed as follows:

$$\Delta \leftarrow \frac{1}{T} \sum_{l=n_q-2T}^{n_q-T} (D_l - D_{(l+T)}) \quad (3.3)$$

where D_l is a distance between x and $\tilde{x}^{(k)}$ at query l , n_q is n_q -th query. If Δ is smaller than a switching threshold ϵ_s , BLOCKDESCENT switch to the next component.

3.4 Experiments and Evaluations

This section evaluates the effectiveness of RAMBOATTACK versus current state-of-the-art attacks:

- Boundary Attack (Boundary) (Brendel, Rauber and Bethge, 2018)
- Sign-OPT (Cheng et al., 2020) and
- HopSkipJump (Chen, Jordan and Wainwright, 2020)

The attacks are evaluated on two standard vision tasks: CIFAR10 (Krizhevsky, Nair and Hinton, n.d.) and ImageNet (Deng et al., 2009).

3.4.1 Experiment Settings

Models and Hyperparameters. For a fair comparison, for CIFAR10, we use the same CNN architecture used by Cheng et al. (Cheng et al., 2019b, 2020). This

network comprises of four convolutional layers, two max-pooling layers and two fully connected layers. For evaluation on ImageNet, we use a pre-trained ResNet-50 (He et al., 2016) provided by torchvision (Marcel and Rodriguez, 2010) which obtains 76.15% Top-1 test accuracy. All images are normalized into pixel scale of $[0, 1]$.

All hyper-parameters of our RAMBOATTACK are described in Appendix A.7.1 and all of the evaluation sets are described in Section 3.4.4, 3.4.5, 3.4.3 and Appendix A.1.

Evaluation Measures. To evaluate the performance of a method, prior works use different metrics such as a score based on the median squared l_2 -norm (Brendel, Rauber and Bethge, 2018) and median l_2 -norm distortion versus the number of queries (Cheng et al., 2020; Chen, Jordan and Wainwright, 2020). However, the median metric is not able to highlight the existence of the so-called *hard* cases and their impact on the performance of an attack so the evaluation may be less reliable. Therefore, in addition to the median, we report average l_2 -norm distortion. We also adopt Attack Success Rate (ASR) used in (Chen, Jordan and Wainwright, 2020) to measure the attack success of crafted adversarial samples, obtained with a given query budget, at various distortion limits.

Gradient Estimation Selection for RAMBOATTACK. We apply two state-of-the-art gradient estimation methods, HopSkipJump and Sign-OPT, and derive two RAMBOATTACK attacks: i) RAMBOATTACK (HSJA), composed of HopSkipJump, BLOCKDESCENT and Sign-OPT; and ii) RAMBOATTACK (SOPT), composed of Sign-OPT and BLOCKDESCENT. We do not use HopSkipJump for the second gradient descent stage because we observe Sign-OPT to be more effective at refining adversarial samples—as also observed in (Cheng et al., 2020).

3.4.2 Experimental Regime

This section summarizes all extensive experiments conducted on CIFAR-10 and ImageNet datasets with different sparse attacks in decision-based settings.

- *Evaluation Protocol:* Section 3.4.3 proposes two different evaluation protocols which are used in exhaustive experiments on *hard-set* and balance sets. Especially, the exhaustive evaluation protocol is designed to explore the existence of a *hard* set for decision-based attacks as observed in Section 3.2.4.

3.4.2 Experimental Regime

- *Robustness of RAMBOATTACK*: Section 3.4.4 investigates the robustness of our RAMBOATTACK and other methods by assessing the existence of a *hard* set for our RAMBOATTACK and compares its performance with the state-of-the-art attacks.
- *Attacking Hard Sets*: Most attacks demonstrate their impressive performance in *non-hard* cases whilst struggling with *hard* cases. Therefore, Section 3.4.5 compares and demonstrates the performance differences—in terms of query efficiency, attack success rate and distortion—on *hard* evaluation sets.
- *Impact of the Starting Image*: The impact of the starting image from the target class on the success of the attack is observed in Section 3.2.4. Hence, the exhaustive experimental evaluations in Section 3.4.6 explore the sensitivity of an attack’s success to the choice of the attacker’s starting image. An important consideration to evade detection is through trial-and-error testing of starting images to find easy samples or when access to samples (source or target class) is restricted.
- *Attack Insights*: Clear correlations between perturbations yielded by our RAMBOATTACK and salient regions of target images embedded inconspicuously in adversarial examples is observed and investigated in Section 3.4.7. These artifacts result from the localized perturbation method in BlockDescent.
- *Attacks Against Defended Models*: Decision-based attacks are able to fool standard models. This naturally leads to the critical question of whether or not such attacks are able to bypass defended models. Thus, the experiments in Section 3.4.8 aim to investigate the robustness of decision-based attacks against defense mechanisms.
- *Validation on Balance Datasets*: Constructing *hard* and *non-hard* sets for all decision-based attack methods through exhaustive evaluations to assess robustness is extremely time-consuming. Therefore, a reliable and reproducible attack evaluation strategy is proposed to validate attack robustness. The proposed evaluation protocol and results are deferred to Appendix A.1 and all of the constructed sets for comparisons are released in future studies.
- *Untargeted Attack Validation*: In addition to targeted attacks, for completeness, RAMBOATTACK and other state-of-the-art attacks are evaluated on CIFAR10 and ImageNet under the untargeted attack setting in Appendix A.2.

3.4.3 Proposed Robustness Evaluation Protocol

This section introduces two evaluation protocols for exhaustive experiments on *hard-set* and balance sets. While the exhaustive evaluation protocol is designed to discover the existence of a *hard* set in decision-based attacks, the balance evaluation protocol aims at providing a fair comparison and reliable benchmark.

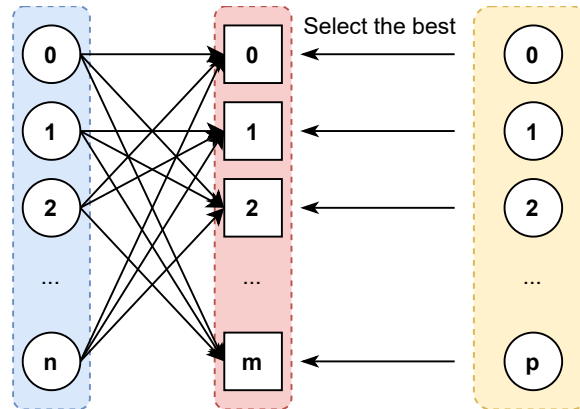


Figure 3.7. The proposed evaluation protocol for assessing robustness under an exhaustive evaluation setting. In this mode, each sample from a dataset with the size of n is evaluated to obtain an adversarial example for that sample capable of flipping its predicted label to m different target classes from that dataset. For each attack, a starting image is selected from a pool of p starting images.

Protocol for exhaustive evaluations

An attack method is mounted to change the true prediction of the DNN from its ground truth label for a given source sample image to each of the different target classes. For CIFAR10 with ten classes and 10,000 samples from a testset, an attack method selects each of the 1000 testset samples from a given class as a source image and attempts to find an adversarial example for each of the other target classes (of which there are 9). Consequently, we evaluate 90,000 pairs of source and starting images. Since there is no effective method to choose a starting image from a target class, for a fair evaluation, we apply the same protocol used in (Cheng et al., 2019b, 2020) to initialize an attack for each method. We execute each attack with a query budget of 50,000 queries. Then we identify *hard* cases of each attack method against the victim model (detailed in Section 4.4.1). This protocol can be generalized to other datasets by choosing n samples and m different target classes from that dataset where each target class has its own starting image as shown in Figure 3.7.

3.4.3 Proposed Robustness Evaluation Protocol

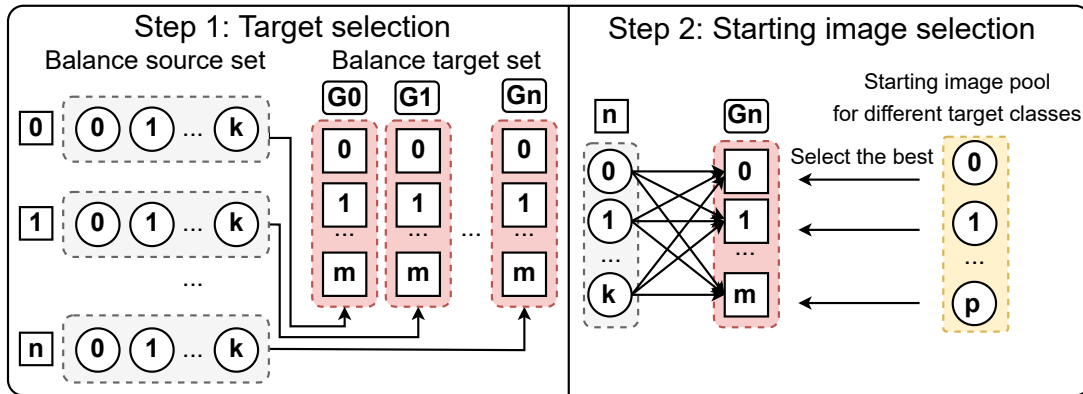


Figure 3.8. The proposed evaluation protocol requires a balanced dataset including n source classes and a balanced target set comprising of n corresponding groups. On the balance source set, all source classes have an equal number of samples (k) while all n corresponding groups have an equal number of target classes (m). These target classes are different within a group but can be repeated in other groups. From these groups G_n , a starting image is selected from a pool of p starting images.

Evaluation protocol for balance sets

The second research question highlights a need to evaluate the overall performance of various black-box attacks under decision-based settings reliably. On CIFAR10, most previous works propose to choose a random evaluation set with randomly sampled images with label y and select a random target label \tilde{y} (Cheng et al., 2020) or set $\tilde{y} = (y + 1) \bmod 10$ (Brendel, Rauber and Bethge, 2018; Brunner et al., 2019; Cheng et al., 2019b). Nonetheless, these selection schemes may lead to an imbalanced dataset that is insufficient to evaluate the effectiveness of the attack since it may lack the so-called *hard* cases that occur more frequently with specific pairs of classes. As a result, it may lead to a bias in evaluation results and fail to highlight potential weaknesses of an attack. Consequently, we were motivated to propose a more robust and reliable evaluation protocol and illustrate it in Figure 3.8.

A balance set comprises a balanced source set and a balanced target set. Both sets are composed of N different source classes and N corresponding groups. Each group is composed of m different target classes and all source and target classes are randomly chosen from all classes of a test set. In addition, all target classes are different within a group but can be repeated in other groups. Each source class has n samples selected randomly from a test set. Adversaries may have one or several images from each target class and select one to initialize an attack. Each attack method aims to craft an adversarial example for every selected sample from each source class and flip

its true prediction towards every target class given in the corresponding group of balanced target set. The total number of evaluation pairs is $N \times n \times m$. For instance, every sample of source class i (img: i_1, i_2, \dots, i_n) is flipped towards each target class (class: i_1, i_2, \dots, i_m) in the corresponding group i (see Figure 3.8).

3.4.4 Robustness of RamBoAttack

We carry out a comprehensive experiment, similar to that in Section 3.2.4. In this experiment, we use a range of distortion thresholds of 0.7 to 1.1. Notably, both (Chen, Jordan and Wainwright, 2020) and (Cheng et al., 2020) reported their methods to achieve a distortion level below 0.3 after 10,000 queries; hence our proposed values are not guaranteed to discover *hard* cases because the smallest value, 0.7, is much higher than 0.3 achieved in other studies. The main aim is to illustrate how our RAMBOATTACK are able to craft more adversarial examples with distortions below a range of distortions from 0.7 to 1.1 for each sample of the entire CIFAR10 test set. We compare the performance of the RAMBOATTACK with Sign-OPT and HopSkipJump. Figure 3.9 shows a remarkably low number of *hard* cases for the RAMBOATTACK. The total number of hard cases achieved for our RAMBOATTACK is approximately 10 times lower for the distortion ranges from 0.9 to 1.1. For distortion at 0.7 and 0.8, the number of *hard* cases drops approximately 2 times and 5 times, respectively in comparison with the other attack methods. Interestingly, as expected, *hard* pairs encountered by Sign-OPT and HopSkipJump are resolved with RAMBOATTACK as shown in Appendix A.5.

3.4.5 Attacking Hard Sets

This section analyses the performance difference in terms of query efficiency, attack success rate and distortion on *hard* evaluation sets.

Evaluations on CIFAR10. From CIFAR10 test set, we generate a *hard* set for Boundary Attack called *hard-set A* and another *hard* set for both Sign-OPT and HopSkipJump called *hard-set B*. The *hard-set A* and *B* are composed of 400 *hard* sample pairs of a source image and a starting image. A *hard* sample is selected when a distortion between a source image and its adversarial example found after 50K queries is larger than or equal to 0.9. For a fair comparison, each method is employed to craft an adversarial example for each source image initialized with a given starting image. In addition,

3.4.5 Attacking Hard Sets

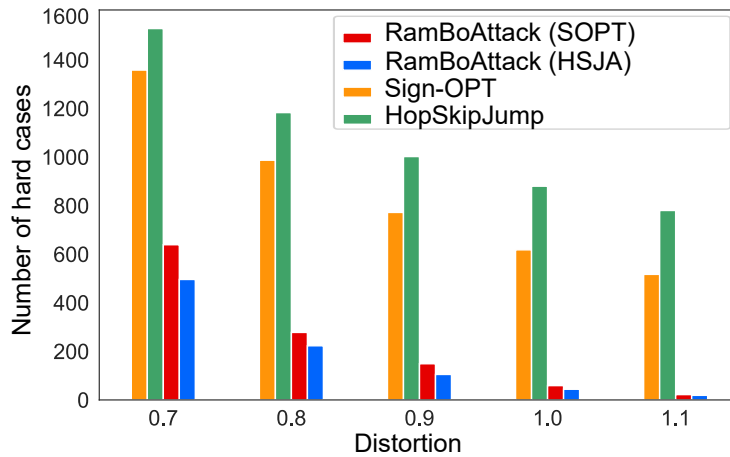


Figure 3.9. The number of *hard* cases found for Sign-OPT, HopSkipJump and RAMBOATTACK with a range of distortion threshold from 0.7 to 1.1 using a budget of 50,000 queries (see detailed results in Appendix A.5 and Figure A.9).

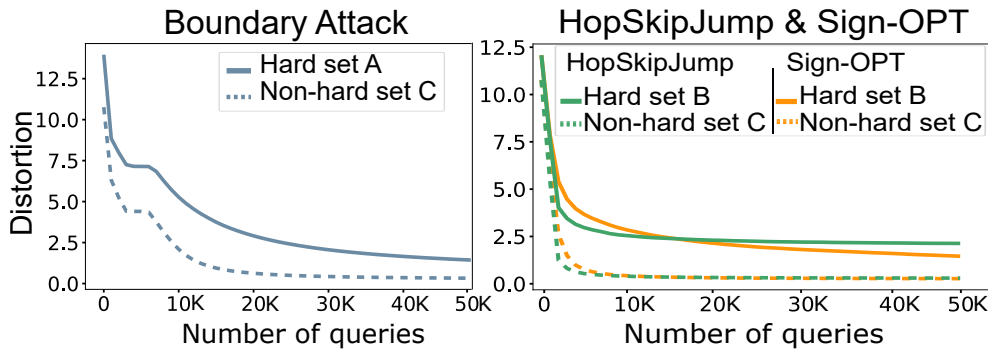


Figure 3.10. A distortion comparison versus queries for each method using their own *hard* versus *non-hard* cases.

we also construct a common *non-hard* set for all three attacks called *non-hard set C* to compare and highlight the significant difference between evaluation results on *hard* and *non-hard* sets as shown in Figure 3.10. Particularly, it shows that the average distortion versus queries on the common *non-hard set C* achieved by these methods is significantly lower than that obtained on their own *hard* set after 50K queries.

We evaluate our RAMBOATTACK on *hard-set A* & B. Figure 3.11 and 3.12 show that Boundary Attack, Sign-OPT and HopSkipJump do not efficiently find an adversarial example with low distortion; however, RAMBOATTACK can achieve better performance on the *hard-sets*. We defer detailed evaluations on *non-hard-sets* to Appendix A.1; as expected, RAMBOATTACK performs comparably well on these sets. Histogram charts in Figure 3.13 demonstrate that for each *hard-set*, our attacks are able to find lower distortion adversarial examples for most *hard* cases and the distortion

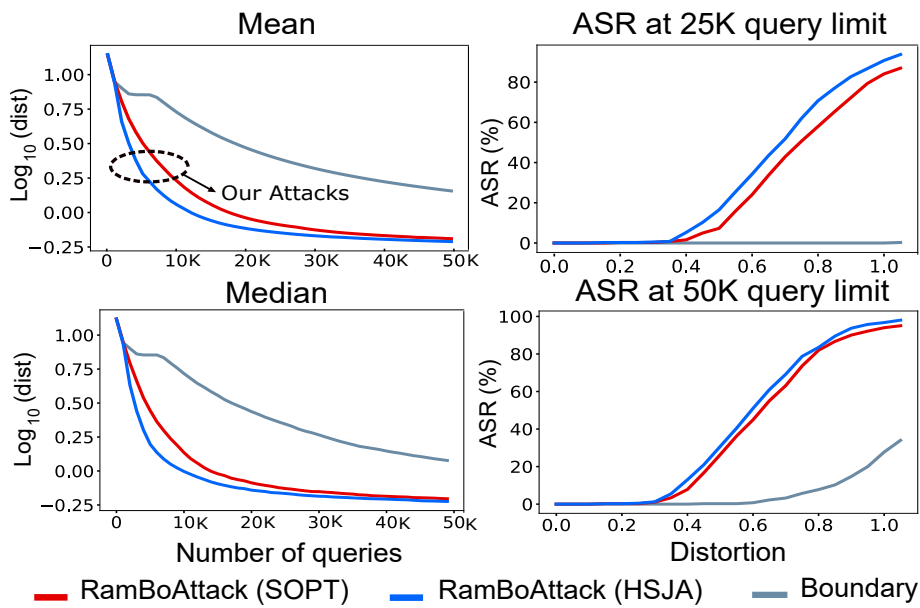


Figure 3.11. Distortion (dist) on a \log_{10} scale vs number of queries. It shows the results for our RAMBOATTACKS versus Boundary attack on **hard-set A**. Our RAMBOATTACKS are **more query efficient** in *hard* cases.

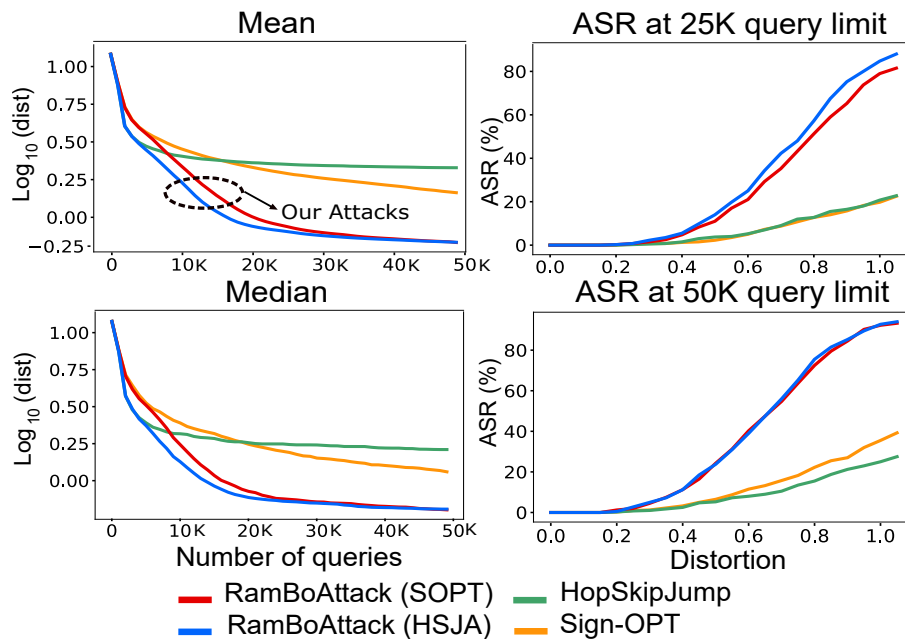


Figure 3.12. Distortion (dist) on a \log_{10} scale vs number of queries. It shows the results for our RAMBOATTACKS versus Boundary attack on **hard-set B**. Our RAMBOATTACKS are **more query efficient** in *hard* cases. Hence our attack is demonstrably more robust and query efficient.

distribution on both *hard-sets*: i) are shifted to smaller distortion regions; and ii) show significantly smaller spread or variance.

3.4.5 Attacking Hard Sets

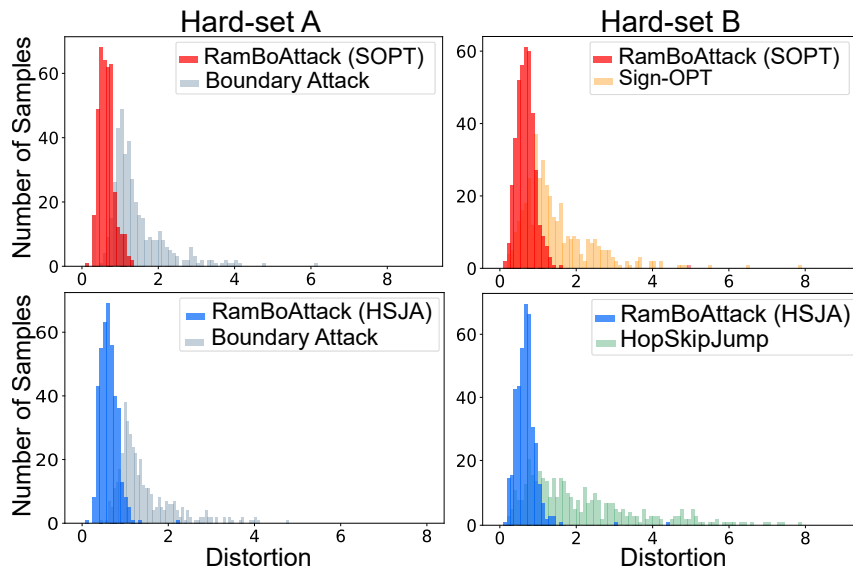


Figure 3.13. On both *hard-set A* and *B* selected from CIFAR10, the distortion distribution yielded by RAMBOATTACKS are shifted left and indicates an overall smaller distortion than other attacks.

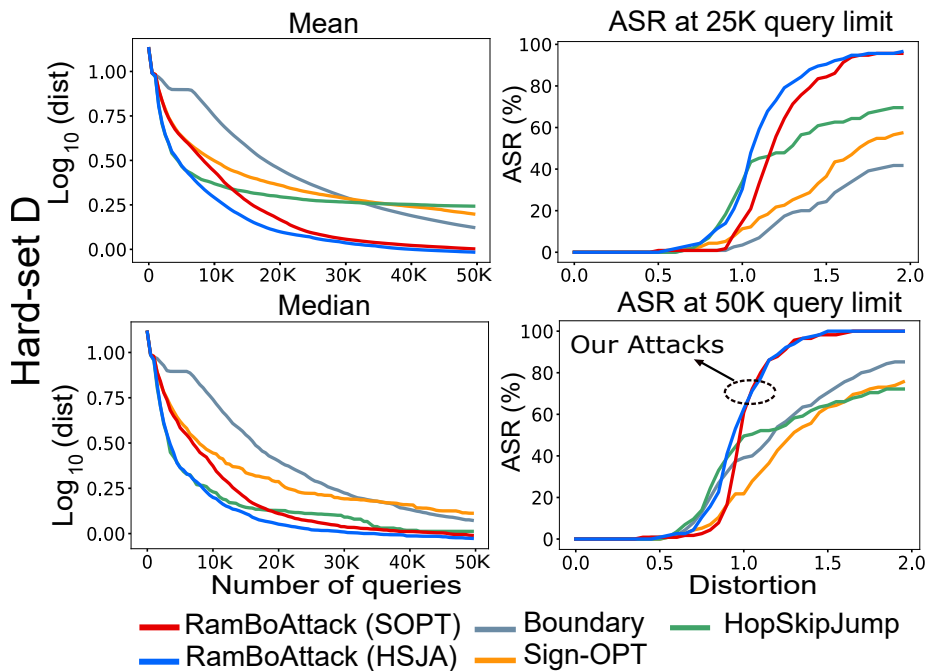


Figure 3.14. Distortion in a \log_{10} scale vs number of queries on *hard-set-D*. Our RAMBOATTACK is **more query efficient** and achieves a higher ASR on this *hard-set*. Hence, our attack is demonstrably more robust and query efficient.

Although we observe RAMBOATTACK to result in fewer hard samples in comparison to other methods at various distortion thresholds, we construct a *hard set* for RAMBOATTACK called *hard-set D* based on the same criteria used to generate *hard-set A*

and B to assess if the *hard-set* for RAMBOATTACK could somehow be easier for the other attack methods. The total number of samples for this set is 115 sample pairs because RAMBOATTACK has a much lower number of *hard* cases than their counterparts (namely BA, HopSkipJump and Sign-OPT) at a given distortion threshold as illustrated in Figure 3.9. We summarize the results from our evaluations in Figure 3.14. As expected, RAMBOATTACKS are more query efficient and are able to craft lower mean and median distortion adversarial examples as well as achieve higher attack success rates at both query budgets. In particular, at distortion levels above 1.0, in comparison to other attacks, RAMBOATTACKS obtain much higher attack success rates—notably, with significant margins at the lower query budget of 25K, since RAMBOATTACKS employ BLOCKDESCENT when the gradient estimation method is unable to make progress (potentially being stuck in a bad local minimum), to discover better solutions and craft lower distortion adversarial samples.

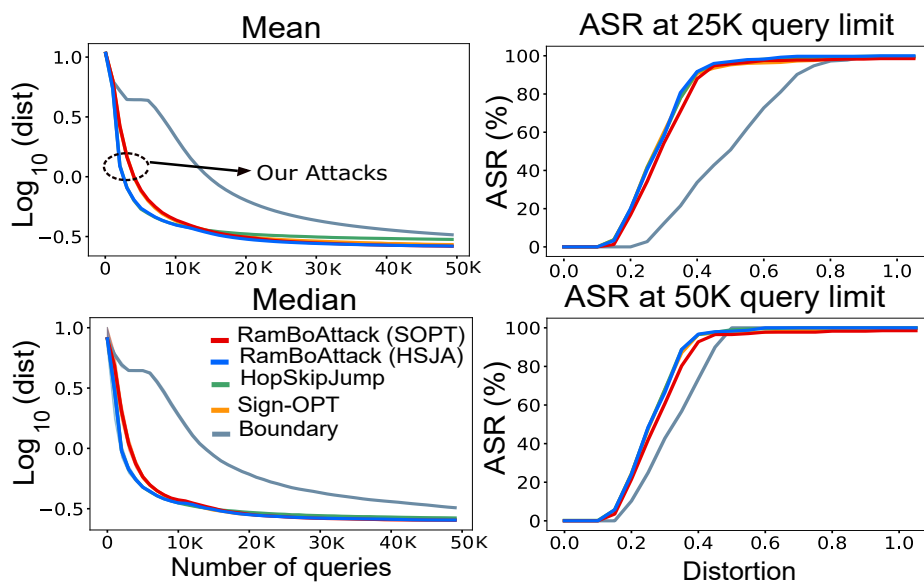


Figure 3.15. Distortion in a \log_{10} scale vs number of queries on *hard* ImageNet evaluation sets. It shows the results on the *hard-set* and our RAMBOATTACKS are **more query efficient**. Hence our attack is demonstrably more robust and query efficient.

Evaluation on ImageNet. To demonstrate the robustness of our attacks on a large-scale model and dataset, we compose a *hard-set* with 120 *hard* sample pairs from ImageNet. A *hard* sample is selected when a distortion between a source image and its adversarial example found after 50,000 queries by Sign-OPT and HopSkipJump is larger than or equal to 15. Notably, we do not compose a *hard* set for Boundary Attack because it cannot yield low distortion adversarial examples efficiently on large scale datasets.

3.4.6 Impact of Starting Images

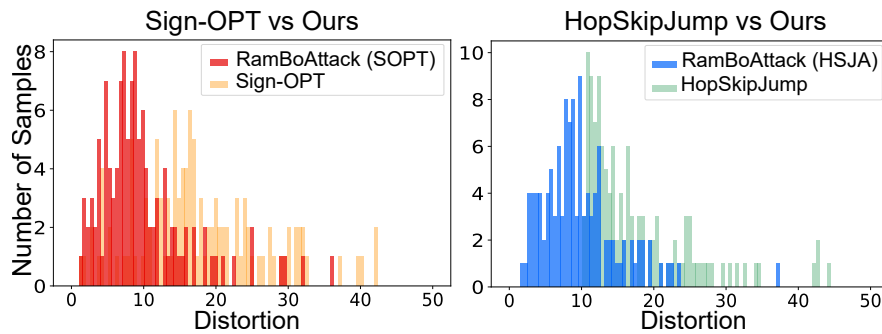


Figure 3.16. On the *hard-set* selected from ImageNet, the distortion distributions yielded by our RAMBOATTACKS indicate an overall smaller distortion compared to other attacks. The distributions is shifted to the left and has significantly less variance compared to other attacks.

Figure 3.15 demonstrates that our RAMBOATTACKS outperform both Sign-OPT and HopSkipJump on the *hard-set*. We defer detailed evaluations on *non-hard-sets* to Appendix A.1; notably, RAMBOATTACKS achieve improved results on the more complex ImageNet dataset. The histograms in Figure 3.16 show distortion distributions for our attacks shifted significantly to smaller distortion regions with smaller variance and fewer outliers compared to other attacks.

3.4.6 Impact of Starting Images

In this experiment, we first compose *subset A* and *B* by selecting 100 random *hard* sample pairs from *hard-set A* and *B*, respectively (see Section 3.4.5 for these sets). Our RAMBOATTACKS are compared with Boundary attack on subset *A* and with Sign-OPT and HopSkipJump, on subset *B*. In Section 3.4.5, each method needs to yield an adversarial example for a pair of a given source image and a given starting image. In contrast, in this experiment, the given starting image is replaced by 10 starting images randomly selected from the CIFAR10 evaluation set and correctly classified by the model. All evaluations are executed with a 50,000 query budget.

In Figure 3.17, the size of each bubble denotes the standard deviation while y-axis value indicates average distortion. We can see that our RAMBOATTACKS almost achieve smaller mean and standard deviation than Sign-OPT, HopSkipJump and Boundary Attack on subset *A* and *B*. A robust method should be less susceptible to the selection of a starting image and yield a low distortion adversarial example most chosen starting images. We can observe from Figure 3.17 that our RAMBOATTACKS are

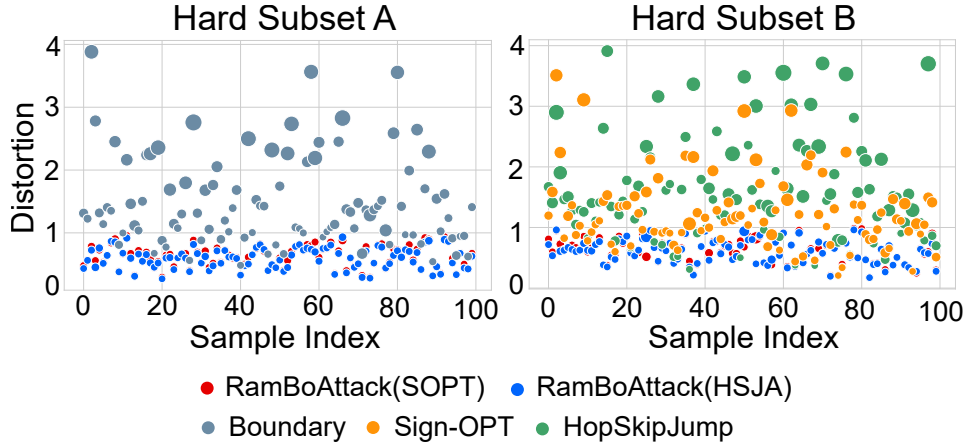


Figure 3.17. An illustration of sensitivity of different attacks to various starting images. Each method is evaluated on each subset and the charts show the average and variance of distortion for each case of each subset achieved by different methods. y -axis denotes the average distortion while the size of each bubble denotes the distortion variation. Compared with Boundary, Sign-OPT and HopSkipJump attacks, our RAMBOATTACKS are **much less sensitive to the choice of a starting image**.

more robust than Sign-OPT, HopSkipJump and Boundary attacks as a consequence of being less sensitive to the chosen starting images.

3.4.7 Attack Insights

This section investigates correlations between perturbations yielded by our attack and salient regions of target images embedded inconspicuously in adversarial example.

Perturbation Regions

First, we develop a simple technique to transform a perturbation with size $C \times W \times H$ to a Perturbation Heat Map (PHM) with size $W \times H$ that is able to visualize perturbation magnitude of each pixel. This transformation is defined as:

$$PHM_{i,j} \leftarrow \frac{A_{i,j}}{\max(A)}, \quad (3.4)$$

where $A_{i,j} = \sum_{c=1}^C |(x - x_a)_{c,i,j}|$; $c \in [1, C]$, $i \in [1, W]$ and $j \in [1, H]$. Second, since Grad-CAM (Selvaraju et al., 2017) is a popular visual explanation technique for visualizing salient features in an input image to understand a CNN model’s decision,

3.4.7 Attack Insights

we use it to investigate the adversarial perturbations generated by our attack and the salient features in the target image largely responsible for a model’s decision for the classification of an input to a target class.

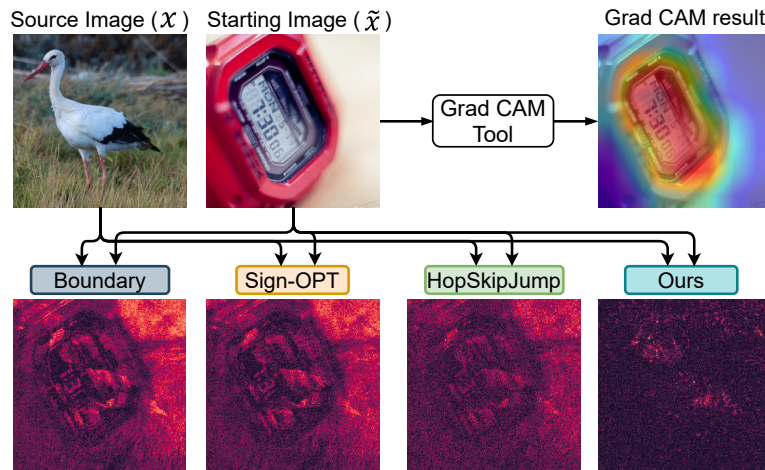


Figure 3.18. Grad-CAM tool visualizes salient features of the starting image or target class: digital watch. A perturbation heat map (PHM) visualizes the normalized perturbation magnitude at each pixel. Comparing different perturbations crafted by different attacks highlights that the localized perturbations yielded by RAMBOATTACK concentrate on salient areas illustrated by GRAD-CAM and embed these targeted perturbations in the source image to fool the classifier to predict the target class; even though, RAMBOATTACK does not exploit the knowledge of salient regions to generate perturbations—additional examples in Appendix A.6, Figure A.10

In all of the attack methods, we observe the attacks to embed the target image in the source image in a deceptive manner. However, in *hard* cases, based on PHM and Grad-CAM outcomes, we observe a strong connection between adversarial perturbations found and salient regions in starting images as illustrated in Figure 3.18 for RAMBOATTACKS. It shows that our RAMBOATTACKS are able to discover and limit manipulations of pixels to salient regions responsible for determining the classification decision of an input image to the target class to craft adversarial examples. This salient region consists of the most discriminative local structures of a starting image against a source image. Because BlockDescent is able to manipulate local regions, RAMBOATTACKS are able to exploit only this discriminative region and employ less adversarial perturbations than Sign-OPT and HopSkipJump to promote features of a starting image and suppress the feature of the source image. Therefore, it may shed light on why RAMBOATTACK with the core component BLOCKDESCENT is able to tackle the so-called *hard* cases. Moreover, in these *hard* cases, we observe that

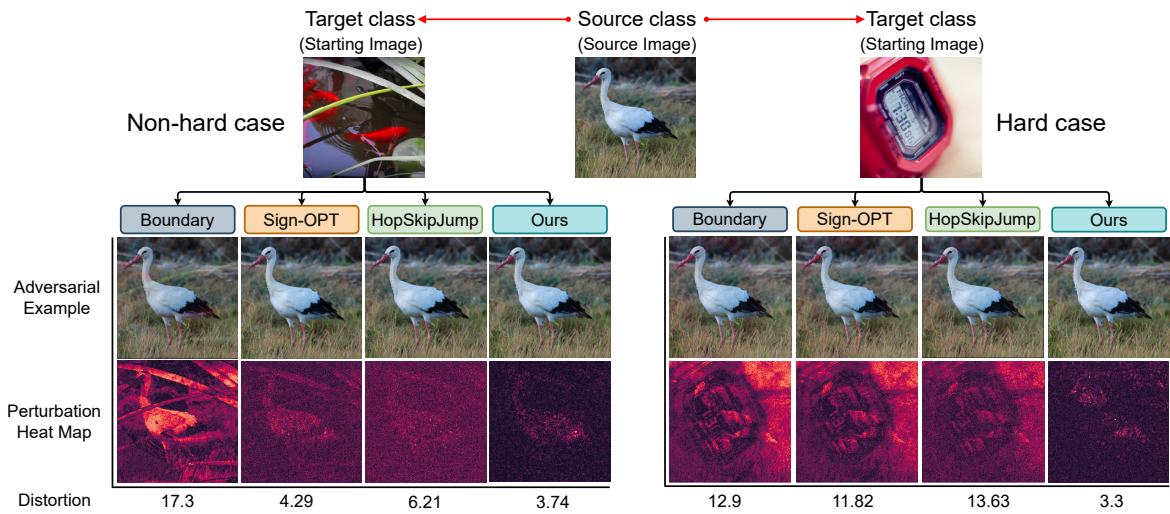


Figure 3.19. An illustration of *hard* case (white stork to goldfish) versus *non-hard* case (white stork to digital watch) on ImageNet. Adversarial examples in *non-hard* cases and *hard* cases are yielded after 50K and 100K queries, respectively. Except for the Boundary attack, adversarial examples crafted by different attacks in *non-hard* cases are slightly different whilst in the *hard* case, our RAMBOATTACK is able to craft an adversarial example with much smaller distortion than other attacks due to the ability of our BlockDescent formulation to *target effective localized perturbations*.

our RAMBOATTACK is able to yield perturbations with more semantic structure if compared with Sign-OPT or HopSkipJump.

Visualization of ImageNet Hard versus Non-hard Cases

Figure 3.19 illustrates adversarial examples in *non-hard* cases and *hard* cases yielded by Boundary Attack, Sign-OPT, HopSkipJump and our RAMBOATTACK (HSJA) after 50K and 100K queries, respectively. The second row of Figure 3.19 shows each corresponding adversarial example and the third row illustrates PHM of each adversarial example. The last row shows the l_2 distortion between each adversarial example and the source image.

For the adversarial example of *non-hard* cases, all methods are able to craft low-distortion adversarial examples except Boundary attack. These adversarial examples and their corresponding distortions are comparable. On the contrary, adversarial examples in *hard* cases yielded by Boundary Attack, Sign-OPT and HopSkipJump have *noticeably higher distortion* than the one crafted by our attack. We observe Boundary Attack, Sign-OPT and HopSkipJump to experience potential

3.4.8 Attack Against Defense Mechanism

entrapment when searching for a low-distortion adversarial example, even when the budget is increased to 100K queries.

Convergence Analysis

The problem considered in this chapter is non-convex and non-differentiable. As such, providing a guaranteed global minimum is not possible. However, our insight is that the gradient estimation in black-box attacks is unreliable, particularly in the vicinity of the local minima. To remedy the problem, we propose RAMBOATTACK as a generic method to overcome this issue. We employ a gradient estimation method in the initial descent using any of the existing alternatives (before BLOCKDESCENT) and subsequently in the refinement stage (after BLOCKDESCENT). Hence, employing the gradient estimation in (Cheng et al., 2020), for instance, would imply that the theoretical convergence analysis therein is still valid for our method.

3.4.8 Attack Against Defense Mechanism

In this section, we evaluate the robustness of various attacks against three different defense mechanisms including region-based classification, adversarial training and defensive distillation. These defense methods are selected due to their own strength. Region-based classifiers can pragmatically alleviate various adversarial attacks without sacrificing classification accuracy on benign inputs (Cao and Gong, 2017) whilst adversarial training (Goodfellow, Shlens and Szegedy, 2014; Madry et al., 2018; Tramèr et al., 2018) is one of the most effective defense mechanisms against adversarial attacks (Athalye, Carlini and Wagner, 2018) and defensive distillation (Papernot et al., 2016b) employ's a form of gradient masking.

For a *baseline*, we choose C&W attack ((Carlini and Wagner, 2017)), a state-of-the-art *white-box attack*. The adversarial training-based models used in this experiment are trained with Projected Gradient Descent (PGD) adversarial training proposed in (Madry et al., 2018). We evaluate our RAMBOATTACK and current state-of-the-art decision-based attacks at different query budgets: 5K, 10K, 25K and 50K.

Results for Attacking against a Region-based Classifier

Figure 3.20 shows that the average and median distortion (on a \log_{10} scale) achieved by RAMBOATTACKS are significantly lower than BA, Sign-OPT and HopSkipJump. In

addition, our attack outperforms others in terms of attack success rate (ASR) at 25K and 50K query budgets—i.e. achieve *higher* ASR on defended models under different query budgets and distortion thresholds. Based on these results, *we observe our attack to be more robust than exiting attacks when mounting an attack against region-based classifiers.*

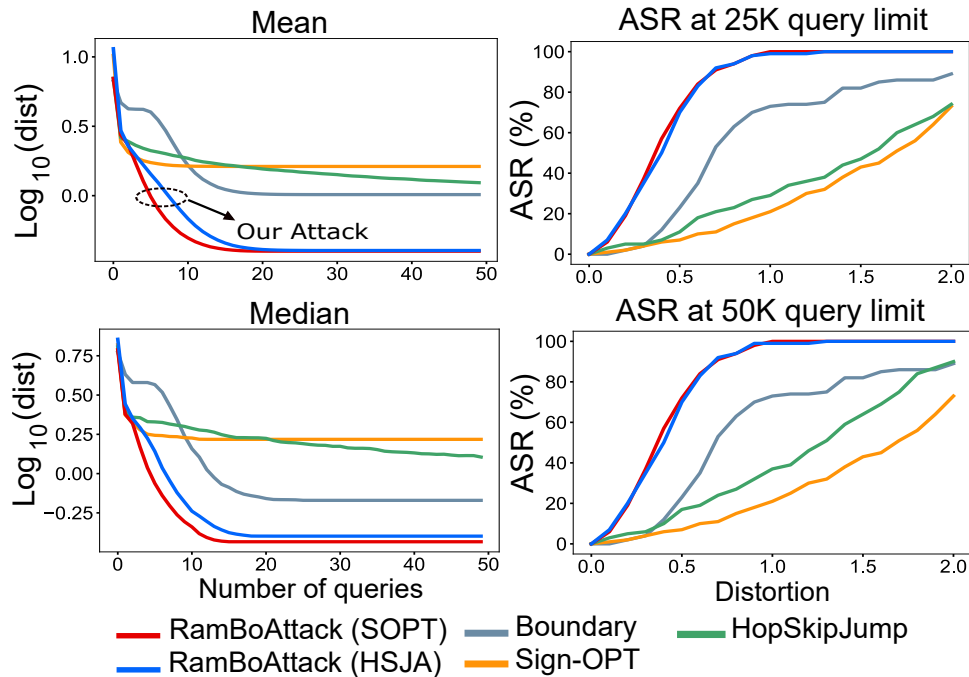


Figure 3.20. Performance comparison between different state-of-the-art attacks and RAMBOATTACK against a **Region-based Classifier** on CIFAR10. RAMBOATTACK outperforms other black-box attacks and is able to craft *significantly more effective* adversarial examples of lower distortion against the defense method as seen by the higher ASR results against the defended models from RAMBOATTACK across all of the evaluations.

Analysis. The reason for this is that existing attack methods need to follow the decision boundary where region-based classifiers are capable of correcting their prediction by uniformly generating a large amount of data points at random and returning the most frequently predicted label. This capability of region-based classifiers prevents binary search in Sign-OPT and HopSkipJump from specifying the boundary exactly and results in noisy and coarse boundary estimations that cause all attack methods aiming to walk along the boundary to fail to estimate a useful gradient direction. Nevertheless, our RAMBOATTACKS are able to break this defense mechanism because the core component, BLOCKDESCENT, is a derivative-free optimization that does not need to determine the boundary and estimate a gradient direction to descend.

3.4.8 Attack Against Defense Mechanism

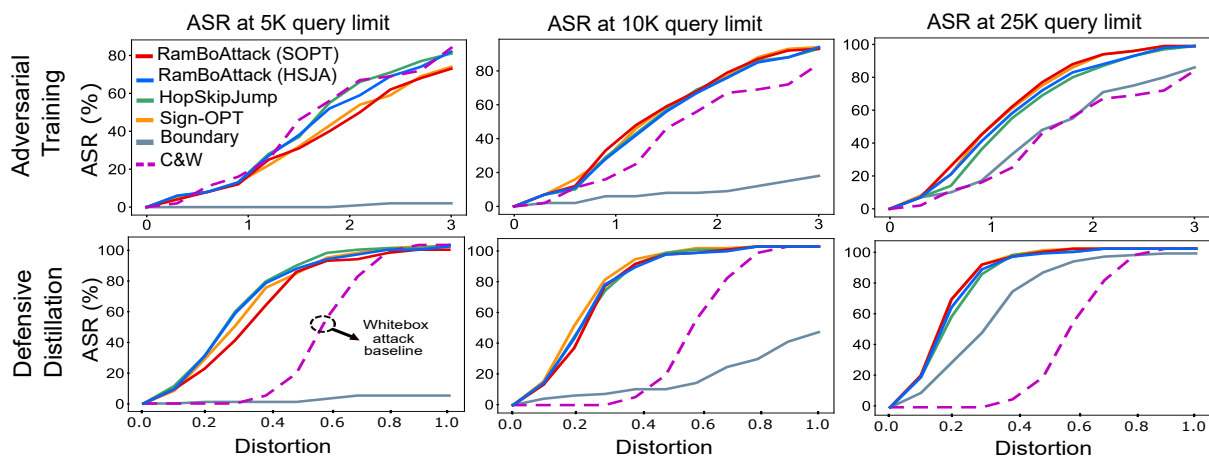


Figure 3.21. ASR comparison between white-box (*employed as a baseline*) and current decision-based attacks versus our RAMBOATTACK against Adversarial Training model and Defensive Distillation on CIFAR10 (using the balanced set). Interestingly, RAMBOATTACKS are more effective than the white-box attack method baseline and are slightly more robust under different query settings when compared to other decision-based black-box attacks.

Results for Attacking against Adversarial Training and Distillation

Figure 3.21 shows the attack success rate (ASR) at different distortion levels and query limits for various attack methods against an adversarially trained model and defensive distillation model. Particularly, for adversarial training, our RAMBOATTACKS can achieve comparable performance with Sign-OPT and HopSkipJump while outperforming Boundary attack within the query limits of 5K, 10K or 25K. In addition, we compare the performance of our attack at different query budgets with the white-box attack—C&W—used as a baseline for comparison. Notably, we do not execute C&W attack at different query settings because it is a white-box method and use the best result produced by this attack.

Analysis. We observe that our attacks are able to obtain a comparable performance with the C&W attack at the 5K query budget. When the query limit is up to 10K and higher, our RAMBOATTACKS outperform the white-box C&W baseline attack method. Nevertheless, Adversarial Training is still effective at reducing the ASR achieved by our method, even with a 25K query budget. Success falls from around 99% (see Figure A.5) to approximately 43% (see Figure 3.21) at a distortion of 1.0 (l_2 norm). Similarly, at a distortion of 0.3, the ASR decreases from about 60% (see Figure A.5) to approximately 10% (see Figure 3.21). However, what we can observe is that as the

distortion increases, the attack is more effective. This is expected because the attack budget of the adversary is increased above and beyond the budget used for generating the adversarial examples used for building the adversarially trained model.

Likewise, for defensive distillation, our RAMBOATTACKS can achieve comparable performance with Sign-OPT and HoSkipJump whilst outperforming Boundary attack and C&W whitebox baseline attack at different query budgets. These results confirm the results and findings presented in (Chen, Jordan and Wainwright, 2020).

3.5 Conclusion

This chapter proposes a new attack method in a decision-based setting; RAMBOATTACK. In contrast to modifying a whole image as in current attacks, the proposed attack exploits localized perturbations to yield more effective and low-distortion adversarial examples in the so-called *hard* cases. The comprehensive empirical results demonstrate that the proposed attack outperforms current state-of-the-art attacks. Interestingly, while the main proposed component, BLOCKDESCENT, is able to significantly improve the performance and robustness of attacks in the so-called *hard* cases, it does not degrade performance in *non-hard* cases. As a result, validation results on small and large-scale evaluation sets demonstrate that RAMBOATTACK is *more robust and query efficient* than current state-of-the-art attacks. Notably, whilst an extensive set of results is presented in the main chapter, additional results to support the study are in Appendix A.

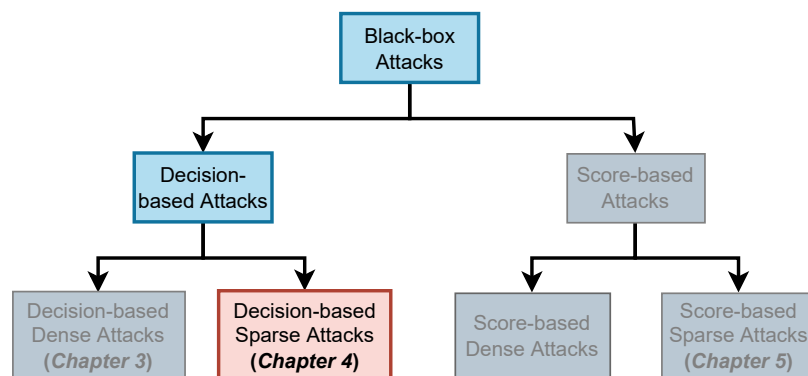


Figure 3.22. Upcoming chapter sneak peek.

The practicality of decision-based dense adversarial attacks, which manipulate an entire input (*i.e.* image) with only access to models' decision (*i.e.* predicted labels),

3.5 Conclusion

presented in this chapter poses a critical threat to Deep Learning models employed in real-world systems. This then raises a pertinent question what if manipulating solely some pixels in the input is able to deceive Deep Learning models. To address this concern, the upcoming chapter as depicted in Figure 3.22 will explore a new threat—a sparse adversarial attack—against DNN models. This threat is crucial to investigate because it demonstrates that DNN models are more susceptible to subtle changes in the input than we believe. Further, this type of threat has not drawn much attention and manifests as an inadequacy in our knowledge about the weaknesses of DNN models. To this end, the next chapter will discuss the challenging problem associated with this type of attack and propose a new sparse attack algorithm that is significantly more query efficient than the existing methods.

Chapter 4

SparseEvo: A Sparse Attack Under Decision-base Settings

THIS chapter considers the challenging problem of designing a query-efficient sparse adversarial attack— l_0 norm-constraint—in decision-based settings. In contrast, the previous chapter investigates the vulnerability of DNN models to dense attacks (l_2 norm-constraint). The realisation of sparse attacks against black-box models now demonstrates that machine learning models are more vulnerable than we believe. Because these attacks are able to *minimize the number of perturbed pixels*—measured by l_0 norm—required to mislead a model by *solely* observing the decision (*the predicted label*). But, such an attack leads to an NP-hard optimization problem. The study in this chapter proposes an evolution-based algorithm—SPARSEEVO—for the problem and evaluates against both convolutional deep neural networks and *vision transformers*. Notably, vision transformers are *yet* to be investigated under a decision-based attack setting. Although conceptually simple, the proposed attack with only a limited query budget outweighs the state-of-the-art decision-based sparse attack *Pointwise* and is competitive with the *whitebox* sparse attacks in standard computer vision tasks. Importantly, the query efficient SPARSEEVO, along with decision-based attacks, in general, raise new questions regarding the safety of deployed systems and poses new directions to study and understand the robustness of machine learning models.

4.1 Motivation and Contribution

Unlike the l_2 norm-constrained adversarial attack—*Dense Attack*—in Chapter 3, this chapter introduces a new l_0 norm-constrained adversarial attack—*Sparse Attack*. While dense attacks (Athalye, Carlini and Wagner, 2018; Ilyas et al., 2018; Shukla et al., 2021) are widely explored, *sparse attacks have not drawn much attention*. This potentially leads to a lack of knowledge on model vulnerabilities to this perturbation regime. From a security standpoint, sparse attacks are particularly as threatening as dense attacks. Therefore, investigating sparse perturbation regimes is as pivotal and necessary as dense perturbation counterparts. To this end, the study in this Chapter extensively investigates the robustness of DNNs against *Sparse Attacks* and proposes a new attack algorithm.

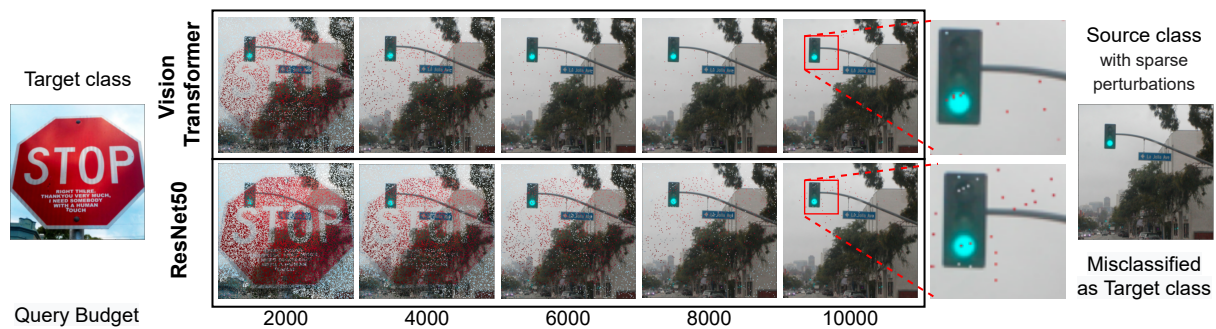


Figure 4.1. Targeted Attack. Malicious instances generated for a sparse attack with different query budgets using our SPARSEEVO attack algorithm employed on black-box models built for the ImageNet task. With an extremely sparse perturbation (*78 perturbed pixels over a total of 50,176 pixels*), an image with ground-truth label *traffic light* is misclassified as a *street sign*.

To explore the robustness of DNNs against *Sparse Attacks*, the study in this chapter will focus on convolution-based and Attention-based architectures introduced by (Ramachandran et al., 2019; Cordonnier, Loukas and Jaggi, 2020; Touvron et al., 2021), particularly the *Vision Transformer* (ViT) proposed by (Dosovitskiy et al., 2021) which is competitive or even outperform convolution-based network (Carion et al., 2020; Bhojanapalli et al., 2021). Existing studies have *not* considered adversarial attacks in l_0 norm constraint-based perturbation regimes against ViT, although a few studies have explored robustness against l_2 and l_∞ norm constraints (Shao et al., 2021). This raises a critical security concern for the reliable deployment of real-world applications based on vision transformers. Therefore, the study will focus on investigating a method

capable of evaluating the robustness of convolutional DNNs as well as transformer networks to understand the fragility of ViT in relation to CNNs under l_0 norm adversarial attacks.

Yielding sparse perturbations is incredibly difficult as minimizing l_0 norm leads to an NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020). Existing sparse attacks in black-box settings, particularly in decision-based scenarios, have a key shortcoming—the algorithms require a large number of model queries to achieve sparsity and invisibility. Consequently, this study proposes a novel evolutionary algorithm-based sparse attack method in the decision-based setting, referred to as SPARSEEVO. Because the evolutionary algorithm is a derivative-free method, it is able to handle the NP-hard problem significantly more effectively and is more query efficient than the state-of-the-art counterpart—Pointwise (Schott et al., 2019). An example of a targeted attack with the proposed algorithm is illustrated in Figure 4.1 on the standard computer vision task, ImageNet.

To understand the fragility of different Deep Learning models to sparse adversarial attacks in decision-based settings and examine the query efficiency of these sparse attacks, this study aims to answer the following research questions (RQ).

RQ1: How can an adversary construct a robust and query-efficient decision-based attack for achieving highly sparse adversarial perturbations in high-dimensional spaces? This question will be explored in Section 4.3.

RQ2: How successful are decision-based sparse attacks against Convolutional Neural Networks, Vision Transformers and defended models? And how do Vision Transformers compare with CNNs in terms of robustness? This question will be explored in Section 4.4.

Main Contributions. The contributions of this chapter are summarised below:

- A novel sparse attack—SPARSEEVO—an evolution-based algorithm capable of exploiting access to solely the *top-1 predicted* label from a model is formulated to search for an adversarial example in the model’s input space whilst minimizing the number of perturbed pixels required to mislead the model.
- The proposed attack algorithm can significantly reduce the number of model queries compared with the state-of-the-art counterpart, Pointwise. Further,

4.1.1 Chapter Overview

SPARSEEVO achieves comparable success to PGD_0 —the state-of-the-art *white-box* attack—in terms of attack success rate with a limited query budget.

- The *first* vulnerability evaluation of a Vision Transformer (ViT) on the standard computer vision task ImageNet in a decision-based and l_0 norm-constrained setting is conducted and compared with ResNet to assess the relative robustness of the ViT model.

4.1.1 Chapter Overview

Section 4.2 presents the related work on decision-based sparse attacks; Section 4.3 introduces the problem formulation and details the proposed attack algorithm; Section 4.4 evaluates and discusses the performance of different sparse attacks across different datasets. Section 4.5 gives a conclusion of this chapter.

4.2 Related Work on Sparse Attacks

This section discusses prior works in the area of sparse adversarial attacks in decision-based scenarios. It first presents sparse attacks in white-box settings and then briefly criticizes sparse attack methods under score-based and decision-based scenarios.

Sparse Attacks. The main aim of sparse attacks is to minimize the number of perturbed pixels required to mislead a target machine learning model. Only a handful of works have investigated sparse attacks and these works can be broadly categorised based on various degrees of adversarial access to a model.

White-box methods. To realize sparse attacks in a white-box setting, SparseFool attack introduced by (Modas, Moosavi-Dezfooli and Frossard, 2019) employed the idea of l_1 relaxation from (Andrei and Ion, 2015) and exploited low mean curvature of decision boundaries for l_0 minimization. JSMA (Papernot et al., 2017) constructed a saliency map for input to search for high-impact pixels on the model’s decision. Recently, (Croce and Hein, 2019) introduced PGD_0 that projects the adversarial perturbation yielded by PGD (Madry et al., 2018) to the l_0 ball. This attack method is capable of generating significantly lower l_0 perturbation and was shown to outperform other white-box algorithms. Therefore, we use the PGD_0 algorithm as *an ideal case baseline* to compare the success achievable in a black-box setting.

Score-based methods. (Su, Vargas and Sakurai, 2019) proposed the One-Pixel attack based on a differential evolutionary algorithm. Although the One-Pixel method is capable of searching and obtaining the most sparse perturbation, its attack success rate (ASR) on large neural networks and high-resolution images is relatively low. Importantly, the method requires a significant number of queries because it modifies one pixel at a time while the input search space, dependent on image resolution, can be enormous. Score-based methods exploit information exposed from a change in confident score to alter a pixel subset in an input image; a model owner may prevent this leakage by only exposing the top-1 predicted label to a model query.

Decision-based methods. In the decision-based setting, only the top-1 predicted label of a DNN model is exposed to adversaries. Now, perturbing an input image slightly will not expose subtle changes in the output corresponding to the perturbation; since only the predicted class label is revealed. Therefore, a decision-based attack is the most restrictive and challenging scenario. Most existing decision-based attack algorithms are *dense attacks* (the objective is to minimise l_2 or l_∞ distortion). Interestingly, these methods, including BA (Brendel, Rauber and Bethge, 2018), HSJA (Chen, Jordan and Wainwright, 2020), QEBA (Li et al., 2020), NLBA (Li et al., 2021a), PSBA (Zhang et al., 2021b), Sign-OPT (Cheng et al., 2020) or the covariance matrix adaptation evolution strategy (CMA-ES) based method for face recognition tasks in (Dong et al., 2019), can be adapted to a sparse attack setting by a projection to l_0 -ball; however, this is not effective, as we show later in Appendix B.5. Although CMA-ES (Dong et al., 2019) is an evolutionary algorithm, albeit for a dense attack, the formulation requires individuals of a population to be real number vectors that can be sampled from a Gaussian distribution. Thus, CMA-ES is well suited to the problem of dense attacks. In contrast, the optimization problem in a sparse attack (l_0 constrained) aims to minimize the *number of perturbed pixels*. Importantly, the discrete search space encountered in a sparse attack hinders the adoption of these dense attack algorithms to search for a sparse adversarial example, efficiently.

To the best of our knowledge, the recent attack—Pointwise (Schott et al., 2019)—applying a greedy search method to find sparse adversarial perturbations is the first decision-based sparse method. This method is effective in untargeted settings and on low-resolution datasets, but it is seen to require a prohibitively large number of queries to achieve low sparse adversarial perturbations on large-scale datasets and in a targeted attack setting (as seen in Section 4.4). *In summary, the current black-box,*

4.3 Proposed Method

sparse adversarial attack approaches still have shortcomings in sparsity and query efficiency. Developing decision-based sparse attacks poses a challenging optimization problem because of: i) limited access to only the decision of a target model; and ii) the NP-hard problem of l_0 norm constrained optimization.

4.3 Proposed Method

This section first formalizes a sparse adversarial attack as a combinatorial optimization problem. It then describes the proposed attack—SPARSEEVO—an Evolutionary-based method.

4.3.1 Problem Formulation

In the sparse attack setting, giving a normalized source image $\mathbf{x} \in [0, 1]^{C \times W \times H}$ and its corresponding ground truth label y from the label set $\mathbb{Y} = \{1, 2, \dots, K\}$ where K denotes the number of classes, C , W and H denotes the number of channels, width and height of an image, respectively. The classifier that we aim to attack is $f : \mathbb{R}^{C \times W \times H} \rightarrow \mathbb{Y}$; our access is limited to its output label. In a targeted setting, \mathbf{x} is perturbed such that the instance $\tilde{\mathbf{x}} \in \mathbb{R}^{C \times W \times H}$ obtained is misclassified to a desired class label $\tilde{y} \in \mathbb{Y}$ selected by the adversary. We refer to the desired class of the input \mathbf{x} as the *target class* and its ground-truth class as the *source class*. In an untargeted setting, the adversary manipulates input \mathbf{x} to change the decision of the classifier to any class label other than its ground-truth, i.e. $\tilde{y} \in \mathbb{Y}$ where $\tilde{y} \neq y$. Formally, a sparse adversarial attack (either targeted or untargeted) to find the best adversarial instance \mathbf{x}^* can be formulated as a constrained optimization problem:

$$\mathbf{x}^* = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \quad \text{s.t.} \quad f(\mathbf{x}^*) = \tilde{y}. \quad (4.1)$$

where $\|\cdot\|_0$ is the l_0 norm denoting the number of perturbed pixels. The optimization problem in Equation 4.1 aiming to minimize the number of perturbed pixels leads to an NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020). Thus, the solution to the optimisation problem is non-trivial given the constraint and the fact that f is not differentiable in this setting.

4.3.2 SparseEvo Attack Algorithm

We devise an efficient parametric search method—SPARSEEVO—based on an evolutionary algorithm approach to search for a desirable solution through an iterative process of improving upon potential solutions. Through a process of recombination, mutation, fitness evaluation and selection, the quality of a population improves over time to yield a desirable solution. Importantly, our evolution-based search method does not require prior knowledge about the underlying target model, such as model architecture or model parameters to construct a fitness function for assessing potential solutions. Consequently, this method detailed in Algorithm 4.1 and Figure 4.2 is well-suited for solving the non-trivial optimization problem in Equation 4.1 in a black-box setting and provides a possible remedy for the NP-hard problem. We detail the formulation of the algorithm in the following.

Defining a Dimensionality Reduced Search Space. In applying a parametric search method to the problem, each *candidate solution* can be defined as a *parameter set* consisting of coordinates and RGB values defining all perturbed pixels of an adversarial input in the search space $\mathbb{R}^{C \times W \times H}$. Naively applying a generic parametric search method to seek potential solutions—parameter sets—as observed in One-pixel algorithm (Su, Vargas and Sakurai, 2019), is not effective because the number of queries to the model grows rapidly with respect to the input image size and the number of perturbed pixels. We propose two techniques to reduce the search space. To facilitate a parametric search method, instead of searching for parameters defining coordinates and RGB values of each perturbed pixel, we propose to solely search for parameters defining coordinates of pixels in the source image to perturb—i.e. image we aim to craft adversarial perturbations for.

Constructing all candidate solutions which are parameter sets in the form of coordinate values is dependent on the number of perturbed pixels and hinders the method implementation. Therefore, we vectorize each *candidate solution* in a population as a *binary vector* $v \in \{0, 1\}^N$ where 0-bits and 1-bits denote non-perturbed and perturbed pixels respectively and N is the total number of pixels of an image. Each element of v corresponds to a pixel and the position i of each element is identified by a mapping function $\phi(n, m)$. Here, we employ a simple flattening technique defined by a mapping function $\phi(n, m) = n + W \times (m - 1)$ where n, m are coordinates of a pixel, and W is the width of an image to reduce the search space further. For the color values of these perturbed pixels, we select RGB values from their corresponding pixels in a starting

4.3.2 SparseEvo Attack Algorithm

image from the target class (we aim to misclassify the source image to the target class in a targeted attack). We illustrate a source image and a starting image in the context of the algorithm in Figure 4.2. All candidate solutions—*binary vectors*—can be changed and evolved over iterations until a desirable solution is reached. *Thus our parametric search method essentially transforms to one that will discover the minimum set of most effective pixels to inject into the source image to construct an adversarial example.* Surprisingly, this method is shown to be an extremely effective strategy for a decision-based sparse attack.

The original search space $\mathbb{R}^{C \times W \times H}$ is now transformed to the new search space $\{0, 1\}^N$ where $N = WH$ is the total number of pixels. In other words, a search space on RGB values and n, m coordinates is transformed into a search space on $i = \phi(n, m)$ without exploring RGB values. As a result, these techniques lead to a reduction in the size of the search space when compared with the original search space.

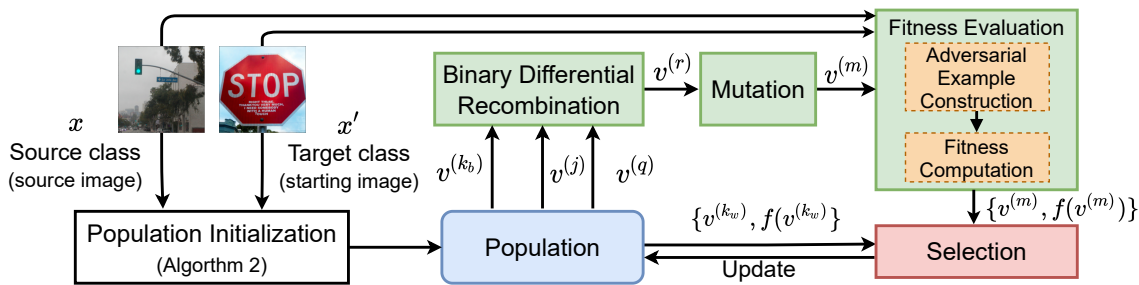


Figure 4.2. An illustration of SPARSEEVO algorithm. *Population Initialization* creates the first population generation. This population is evolved over iterations through *Binary Differential Recombination*, *Mutation*, *Fitness Evaluation* (*Adversarial Example Construction* and *Fitness Computation*) and *Selection* stages. The source and starting images (used in a targeted attack) are employed to create the initial candidate solutions —binary vector representations— at *Population Initialization* and to construct an adversarial example based on a candidate solution $v^{(m)}$ at *Fitness Evaluation* stage.

Fitness Evaluation. Prior to describing the other phases of the algorithm, we describe the *Fitness Evaluation* employed for determining the goodness of a candidate solution necessary for the *Population Initialization* and the *Fitness Evaluation* stages.

Adversarial Example Construction. Since a candidate solution—a binary vector v —is used to construct an adversarial example, its fitness is measured by computing an optimization objective for its corresponding adversarial example. Therefore, we first yield an adversarial example corresponding to v based on the following with c, n, m

Algorithm 4.1: SPARSEEVO

Input: source image x , starting image x' , source label y , target label y^* , model f
population size p , initialization rate α mutation rate μ , query limit T

- 1 $t \leftarrow 0$; $\mathbb{V}, \mathbf{G} \leftarrow \text{INITIALISEPOPULATION}(x, x', f, p, \alpha)$
- 2 $k_w \leftarrow \arg \max_k(\mathbf{G}), k_b \leftarrow \arg \min_k(\mathbf{G})$ // Find best and worst individuals
- 3 **for** $t = 1, \dots, T$ **do**
- 4 Uniformly select $v^{(j)}, v^{(q)} \in \mathbb{V} \setminus v_{k_b}$ at random
- 5 Yield $v^{(r)}$ using Equation 4.5 and $v^{(k_b)}, v^{(j)}, v^{(q)}$
 // Recombination
- 6 Yield $v^{(m)}$ by uniformly altering a fraction μ of all 1-bits of $v^{(r)}$ at random
 // Mutation
- 7 Construct \tilde{x} using Equation 4.2, with x, x' and $v^{(m)}$
- 8 Calculate $g(\tilde{x})$ using Equation 4.3 and $f(\tilde{x})$ // Fitness
 computation
- 9 **if** $g(\tilde{x}) < G_{k_w}$ **then** // Selection
- 10 $G_{k_w} \leftarrow g(\tilde{x})$
- 11 $v_{k_w} \leftarrow v^{(m)}$
- 12 $k_w \leftarrow \arg \max_k(\mathbf{G}), k_b \leftarrow \arg \min_k(\mathbf{G})$
- 13 **end for**
- 14 Construct \tilde{x} using Equation 4.2 with x, x' and $v^{(k_b)}$ // Build adversarial
example
- 15 **return** \tilde{x}

representing a channel and two coordinates of a pixel.

$$\tilde{x}_{c,n,m} \leftarrow (1 - v_i)x_{c,n,m} + v_i x'_{c,n,m}. \quad (4.2)$$

The Fitness Function Formulation. A fitness function should reflect the optimization objective. In the score-based setting, the objective is to optimize loss such that a given input can be misclassified, a reasonable choice for the fitness function is based on output scores as in (Alzantot et al., 2019; Qiu, Custode and Iacca, 2021). However, in our problem, the objective to minimize l_0 distortion directly results in an NP-hard problem. To alleviate this computational burden, (Modas, Moosavi-Dezfooli and Frossard, 2019) relaxed l_0 to l_1 norm to construct the white-box attack, SparseFool and had access to the output scores, unlike in a decision-based setting. Nonetheless, in the

4.3.2 SparseEvo Attack Algorithm

decision-based setting, we find that optimizing l_2 norm provides a better alternative than l_1 . Therefore, in this research, we formulate our fitness function g (for the targeted attack) as:

$$g(\tilde{x}) \leftarrow \begin{cases} \|x - \tilde{x}\|_2, & \text{if } f(\tilde{x}) = \tilde{y} \\ \infty, & \text{otherwise} \end{cases}, \quad (4.3)$$

Where \tilde{x} is an image constructed using Equation 4.2 and \tilde{y} is a target class. A similar fitness function for the untargeted attack can be formulated as Equation 4.3 but the constraint is now $f(\tilde{x}) \neq y$.

Algorithm 4.2 presents pseudo-code for our Population Initialization approach as presented in Section 4.3.2.

Algorithm 4.2: INITIALISEPOPULATION

Input: source image x , starting image x' , source label y , target label y^* , model f
population size p , initialization rate α

- 1 $\mathbb{V} \leftarrow \emptyset, \mathbf{G} \leftarrow \infty$
- 2 $n \leftarrow \lfloor \alpha WH \rfloor$ // W, H are image width and height
- 3
- 4 Generate a binary vector v using Equation 4.4
- 5 **for** $t = 1, 2, \dots, p$ **do**
- 6 **while** *True* **do**
- 7 Generate $v^{(0)}$ by uniformly altering n of all 1-bits of v at random
- 8 Construct \tilde{x} using Equation 4.2 with x, x' and $v^{(0)}$
- 9 Calculate $g(\tilde{x})$ using Equation 4.3 and $f(\tilde{x})$ // Calculate Fitness
Score
- 10
- 11 **if** $g(\tilde{x}) < \mathbf{G}_t$ **then**
- 12 $\mathbf{G}_t \leftarrow g(\tilde{x})$
- 13 $\mathbb{V} \leftarrow \mathbb{V} \cup \{v^{(0)}\}$
- 14 **end while**
- 15 **end for**
- 16 **return** \mathbb{V}, \mathbf{G}

Population Initialization. Recall, our search objective is to discover a minimum perturbation represented by a binary vector—*candidate solution*. Hence, we initialize

a population of p different candidate solutions from an *initialized vector* $v^{(0)}$ formulated as following with C channel number.

$$v_i^{(0)} \leftarrow \begin{cases} 0, & \text{if } x_{c,n,m} = x'_{c,n,m} \forall c \in \{1, \dots, C\} \\ 1, & \text{otherwise} \end{cases} \quad (4.4)$$

Every candidate is generated by only randomly altering d 1-bits of $v^{(0)}$, where $d = \lfloor \alpha WH \rfloor$, α is an initialization rate. A candidate solution is successfully added to the population if its fitness score is not ∞ ; we explain our fitness function in Equation 4.3. Otherwise, another d 1-bits are randomly flipped to generate another candidate solution. This process is repeated until all p successful candidates are found and stored in a population set \mathbb{V} . The corresponding fitness score of each candidate solution is stored in a fitness score matrix G . The pseudocode of the Population Initialization phase is detailed in Algorithm 4.2.

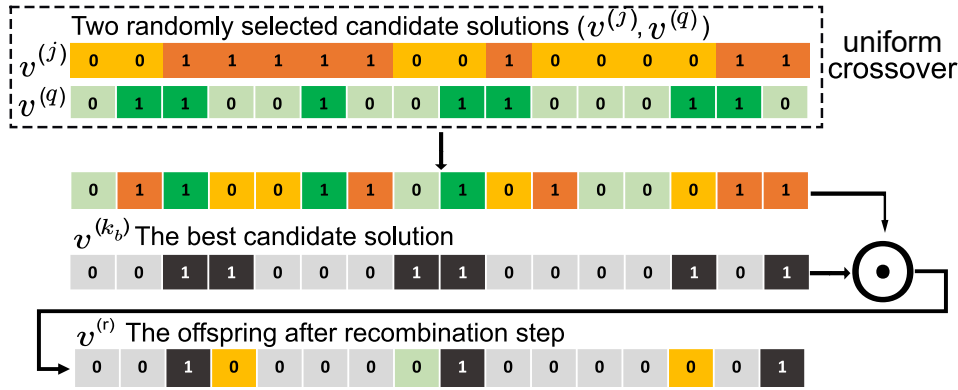


Figure 4.3. The Binary Differential Recombination is shown in Algorithm 4.1 (line 6) and Equation 4.5. \odot is an element-wise product, $v^{(k_b)}$, $v^{(j)}$, $v^{(q)}$ are the best and two randomly selected candidate solutions from a population respectively.

Binary Differential Recombination. In some recombination methods used in genetic algorithms (GA) e.g. k-point or uniform crossover, a couple of parents are mated to produce an *offspring* for the next generation. However, after the Population Initialization stage, all first-generation parents are slightly different from each other since all of them are generated from an initialized vector $v^{(0)}$. Consequently, these crossover variants lead to sub-par solutions and low query efficiency. To address this problem, we increase diversity in a population. Inspired by the differential evolutionary (DE) algorithms (Storn and Price, 1997), we create the next generation by mutating and combining multiple existing parents. Nonetheless, applying DE naively

4.3.2 SparseEvo Attack Algorithm

is impractical since the mutation operation of the DE algorithm adds the weighted difference of multiple selected parents to another parent to yield offspring. These individuals are vectors in real coordinate space so the offspring can benefit from weighted real-valued difference but it cannot be gained in our proposed search space in which all candidate solutions are binary vectors. Therefore, we propose a Binary Differential Recombination scheme—a hybrid method based on the uniform crossover in GA and the notion of mutation in DE.

There are different mutation schemes that can influence the overall performance (Georgioudakis and Plevris, 2020). In the problem of decision-based attacks, through our empirical results shown in Appendix C.12, we observe that the approach of recombining the best and two selected candidate solutions outperform others. Hence, we first select two candidate solutions $v^{(j)}, v^{(q)}$ uniformly at random from the population. We then employ *uniform crossover* for selecting each bit from either selected candidate solutions with equal probability to yield a new candidate solution. Subsequently, the best individual $v^{(k_b)}$ in the population is recombined with the new candidate solution by altering all 1-bits of $v^{(k_b)}$ whose corresponding bits in the new candidate solution are 0-bits. Formally, the Binary Differential Recombination can be formulated as:

$$v^{(r)} \leftarrow v^{(k_b)} \odot \text{UniformCrossover}(v^{(j)}, v^{(q)}) \quad (4.5)$$

where \odot is an element-wise product. This operation is visualized in Figure 4.3. As a consequence of gaining from the difference between individuals, our method is capable of boosting evolutionary progress as shown in Section 4.4.

Mutation. Diversity in the population is a key factor that enables exploration in the search space to obtain better individuals. As a result, mutation operation aiming to promote this population diversity is a crucial component of our method and every offspring after the recombination step can be subject to mutation. In practice, we uniformly select a fraction μ of all 1-bits of the offspring v_o at random and set these bits to zero. We do not select 0-bits for altering because it hinders the optimization progress and requires more iteration to search for the optimum.

Selection. Our simple intuition is that individuals with better fitness values should lead to survival over future generations. In problem 4.1, a smaller fitness value is better and represents a more imperceptible adversarial example. To this end, if the worst individual in the population has a higher fitness value than the offspring's, it will be discarded and the new offspring is then chosen to take its place.

4.4 Experiments and Evaluations

This section evaluates the robustness and query efficiency of SPARSEEVO and compares it with PGD₀—white-box adapted l_0 attack—and Pointwise—the state-of-the-art sparse attack in decision-based settings. These attacks are evaluated on two standard vision tasks CIFAR10 (Krizhevsky, Nair and Hinton, n.d.) and ImageNet (Deng et al., 2009).

4.4.1 Experiment Settings

Attacks and Datasets. For a comprehensive evaluation of the effectiveness of SPARSEEVO, we employ two standard computer vision tasks with different dimensions: CIFAR10 (Krizhevsky, Nair and Hinton, n.d.) and ImageNet (Deng et al., 2009). We compare with the state-of-the-art sparse attack algorithm in Pointwise (Schott et al., 2019) and use the white-box sparse attack PGD₀ (Croce and Hein, 2019) to benchmark against the black-box decision-based counterparts. For the evaluation sets, we select a balanced sample set. We randomly draw 1,000 and 200 *correctly* classified test images from CIFAR10 and ImageNet, respectively. These selected images are evenly distributed among the 10 (CIFAR10) and 200 randomly selected (ImageNet) classes. In the *targeted* setting, while each image from CIFAR10 is attacked to flip its ground-truth label to 9 target classes, a set of five target classes are randomly selected for each image from ImageNet to reduce the computational burden of the evaluation tasks. All of the parameter settings are summaries in Appendix C.11.

Models. For convolution-based models, we use a state-of-the-art architecture—ResNet—(He et al., 2016), particularly, ResNet18 for CIFAR10, achieving 95.28% test accuracy, and a pre-trained ResNet-50 provided by torchvision (Marcel and Rodriguez, 2010) for ImageNet with a 76.15% Top-1 label test accuracy. For attention-based models, we selected a pre-trained ViT-B/16 model obtaining 77.91% Top-1 label test accuracy (Dosovitskiy et al., 2021). Notably, this model was trained by Google on the large scale and high resolution ImageNet dataset.

Evaluation Measures. To evaluate the performance of methods, we define a normalised *sparsity measure* as l_0 -norm distortion divided by the total number of pixels of an image and then compute the *median* of sparsity over an evaluation set—since it is not sensitive to outliers. A measure used to evaluate the robustness of a model is *Attack Success Rate* (ASR). A generated perturbation is successful if it can yield an

4.4.2 Experimental Regime

adversarial example with a sparsity *below a given sparsity threshold*, then ASR is defined as *the number of successful attacks over the entire evaluation set*. In black-box settings, ASR can be calculated at different sparsity thresholds after the assessment of the evaluation set with a given query budget. Notably, there is no query constraint for PGD₀. We run PGD₀ with different perturbation budgets and ASR is calculated based on the best achieved results.

Attack initialization (targeted and untargeted). We need a starting image x' to initialize an attack. For *targeted attacks*, we consider a randomly chosen correctly classified image from the dataset. For *untargeted attacks*, we may perturb the source image by adding a *uniform, Gaussian* (Cheng et al., 2020; Chen, Jordan and Wainwright, 2020) or *salt and pepper* noise (Schott et al., 2019) until it is misclassified. In practice, we observe that employing salt and pepper noise for our untargeted attack is more effective.

4.4.2 Experimental Regime

This section summarizes all extensive experiments conducted on CIFAR-10 and ImageNet datasets with different sparse attacks.

- *Sparse Attacks against Deep Learning Model.* Section 4.4.3, 4.4.4 and 4.4.5 evaluate the robustness of sparse attacks against different Deep Learning models in decision-based settings on different datasets.
- *Robustness of CNNs and ViT under Sparse Attacks.* Section 4.4.6 compares the robustness of the ViT model with the CNN model against sparse attacks under the decision-based scenario.
- *Sparse Attacks against a Defended Model.* Section 4.4.7 examines the robustness of SPARSEEVO and other sparse attacks against an adversarially trained model.
- *Impact of Hyper-parameters, Recombination and Mutation.* Section C.11 and C.12 study the impact of Hyper-parameters, Recombination and Mutation schemes on the performance of SPARSEEVO.
- *Analysis and Comparison with Other Baselines* Appendix B.4 and B.5 discuss and compare SPARSEEVO with Pointwise and improved PointWise. Appendix B.6 compares SPARSEEVO with dense attacks adapted to a sparse setting.

- *Adversarial Example Demonstration.* Appendix C.16 illustrates some adversarial example crafted by SPARSEEVO.

4.4.3 Attacks Against Convolutional Deep Neural Networks

This section evaluates the performance of different sparse attacks against a convolutional-based model on a high-resolution dataset ImageNet.

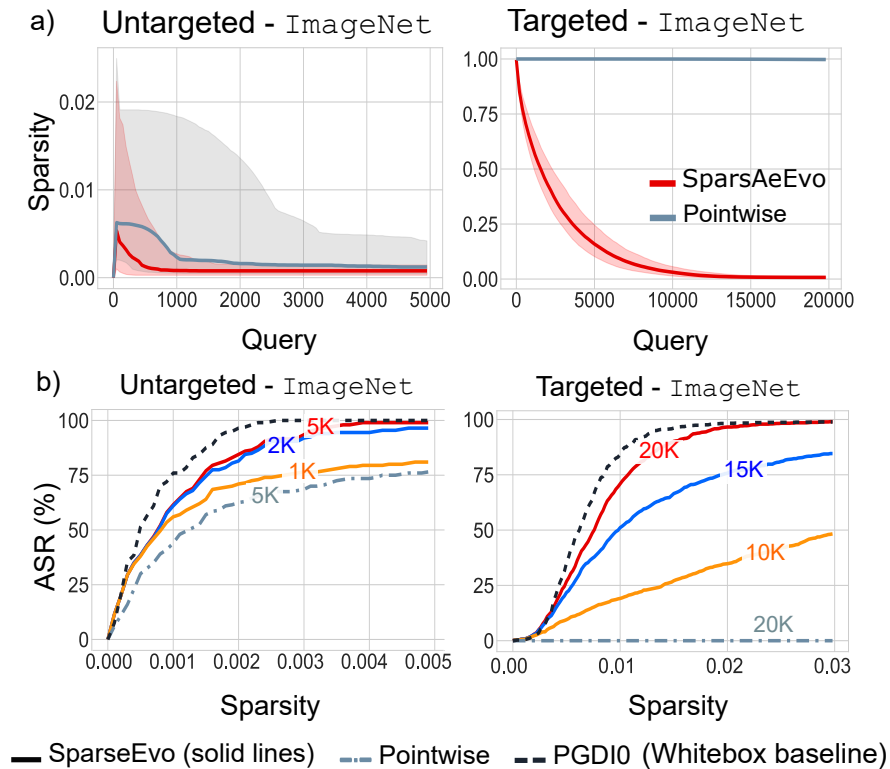


Figure 4.4. Evaluation set from ImageNet using the ResNet50 model with image size ($W \times H$): 224×224 . a) Median sparsity with the first and third quartiles used as lower and upper error bars versus the number of model queries; and b) attack success rate versus sparsity thresholds.

Query Efficiency Evaluation. Figure 4.4a shows the median sparsity against model query budgets on the ImageNet task. Our attack consistently outperforms the Pointwise method in terms of queries and sparsity. In the untargeted setting, SPARSEEVO achieves a lower sparsity than the Pointwise attack under various query budgets. In the targeted setting, our attack is able to craft adversarial images with extremely sparse perturbation within 20,000 queries for most images from ImageNet but Pointwise does not perform well in this task.

4.4.4 Attacks Against a Vision Transformer

Attack Success Rate. Figure 4.4b illustrates ASR against different sparsity thresholds at different query budgets for SPARSEEVO on the ImageNet task and we compare with the best achievement of PGD₀ (ideal, whitebox attack) and Pointwise (decision-based sparse attack). In the untargeted setting, we observe that SPARSEEVO achieves a higher ASR than Pointwise, employing a 5,000 query budget with a small budget of 1,000 queries. In the targeted setting, our attack with a 10,000 query budget demonstrates significantly better ASR than Pointwise employing 20,000 queries. Interestingly, a small query budget of 5,000 queries is adequate to achieve the same ASR as the white-box setting in the PGD₀ attack in the untargeted setting, while around 20,000 queries achieve comparable performance to the ideal white-box setting for a targeted attack. This is significant for decision-based attacks since adversaries are given very limited access to a model. Summary of results at different query budgets and attack settings on the ImageNet vision task in Table 4.1.

4.4.4 Attacks Against a Vision Transformer

This section evaluates the performance of different sparse attacks against a Transformer-based model on a high-resolution dataset ImageNet.

Query Efficiency Evaluation. Figure 4.5a shows the median sparsity against the queries. With a limited number of queries, SPARSEEVO is able to achieve significantly lower sparsity than Pointwise in both targeted and untargeted settings. While our attack is able to converge to an extremely high sparsity after 3,000 and 15,000 queries for untargeted and targeted settings, respectively. Pointwise fails to converge to lower values in both settings.

Attack Success Rate. Figure 4.5b illustrates that with only 1000 queries, SPARSEEVO outperforms Pointwise with a 5,000 query budget across all different sparsity thresholds. Notably, in the untargeted setting, SPARSEEVO with a query budget of 5,000 is able to achieve slightly higher ASR than the ideal white-box PGD₀ from a sparsity threshold of 0.002. In the harder, targeted setting—SPARSEEVO with only 15,000 queries is able to obtain marginally lower ASR than PGD₀, whereas, with a 20,000 query budget, our attack is as robust as PGD₀ when the sparsity threshold is larger than 0.01.

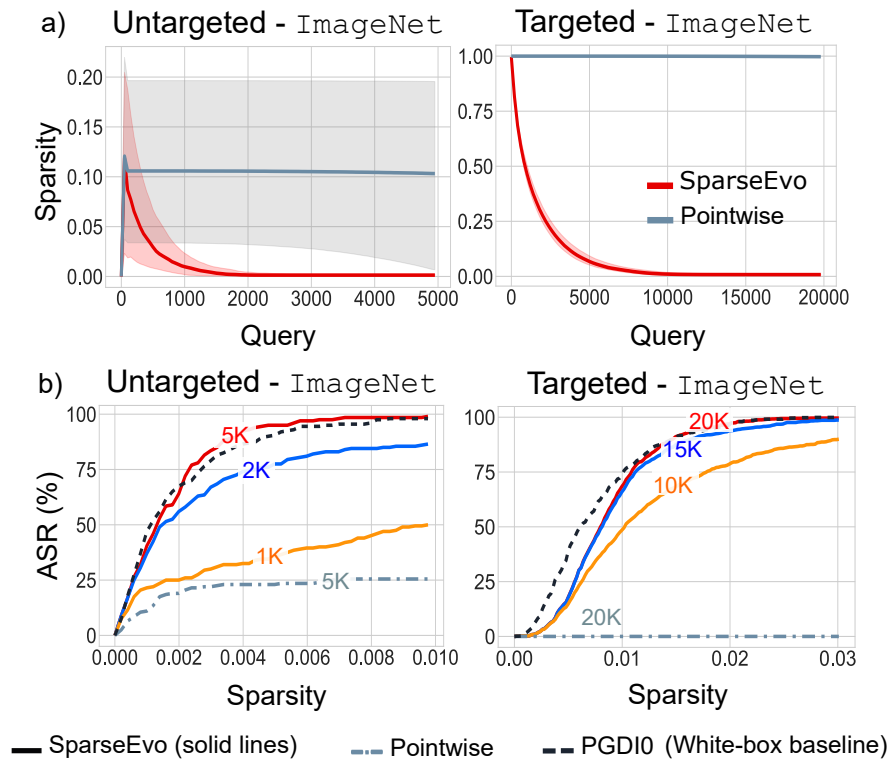


Figure 4.5. Evaluation set from ImageNet using the ViT model with image size ($W \times H$): 224×224 . a) Median sparsity with the first and third quartiles used as lower and upper error bars versus the number of model queries; and b) attack success rate versus sparsity thresholds.

4.4.5 Attacks Against a CNN Model on the CIFAR10

Figure 4.6a shows the median sparsity against the queries as well as the first and third quartiles used as lower and upper error bars. The figure provides a comprehensive comparison of different attacks on the evaluation set from CIFAR10 in both untargeted and targeted settings. Our attack consistently outperforms the Pointwise attack in terms of queries and sparsity. Particularly, in the untargeted setting, our attack is able to craft adversarial images by perturbing an extremely low number of pixels, on average within 2,000 queries for most images on CIFAR10; while Pointwise only obtains a sparsity of 0.75 for this evaluation set. In the targeted setting, SPARSEEVO converges to a lower sparsity than the Pointwise attack with a given query budget.

Attack Success Rate (ASR). Figure 4.6b illustrates ASR against different sparsity threshold at different query budgets for SPARSEEVO on the evaluation set from CIFAR10 and also compare with the best achievement of PGD₀ (ideal, white-box baseline) and Pointwise (state-of-the-art black-box sparse attack). In the untargeted setting, we observe that SPARSEEVO using 200 queries or more achieves higher success

4.4.6 Compare The Robustness of the Transformer and the CNN

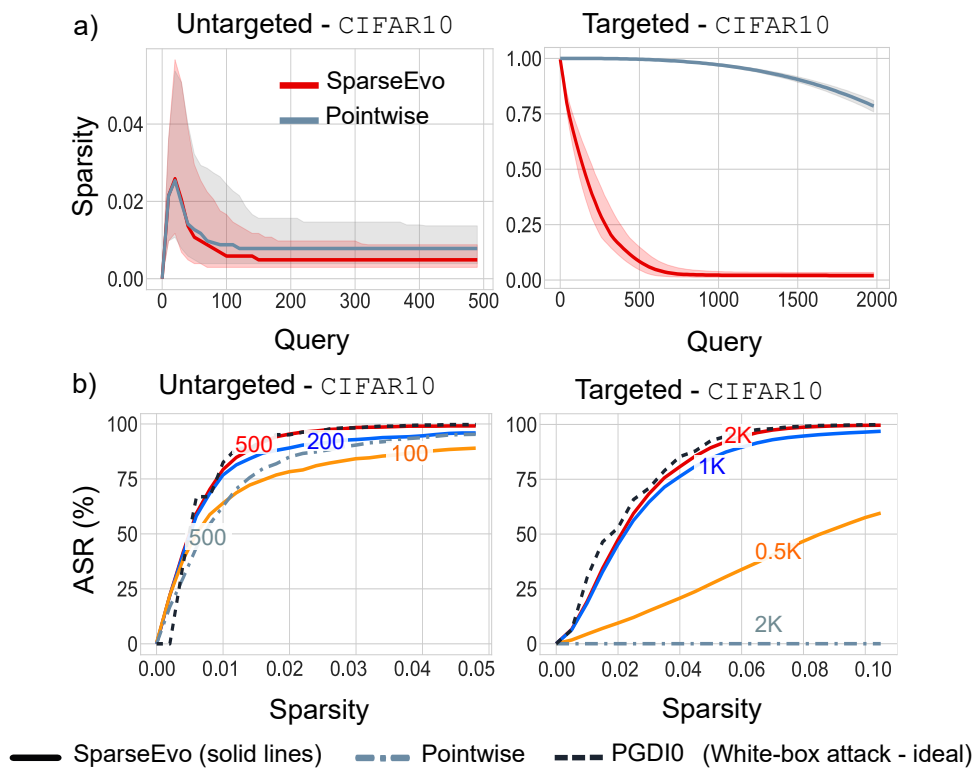


Figure 4.6. Evaluation set from CIFAR10 using a ResNet18 model. a) Median sparsity with the first and third quartiles used as lower and upper error bars versus a number of model queries; and b) attack success rate (ASR) versus sparsity thresholds.

rates than Pointwise using 500 queries. Notably, our black-box sparse attack can achieve comparable ASR to PGD₀ with a small query budget of 500 queries. In the targeted setting, with only 500 queries our attack demonstrates significantly better ASR than Pointwise across all sparsity thresholds, while SPARSEEVO achieves marginally lower ASR than PGD₀ (ideal, white-box baseline) with a query budget of 2,000. Summary of results at different query budgets and attack settings on the CIFAR-10 vision task in Table 4.1.

4.4.6 Compare The Robustness of the Transformer and the CNN

In this section, we compare the robustness of ViT and ResNet50 models to sparse perturbation in untargeted and targeted settings. Figure 4.7 reports the accuracy of these models over adversarial examples of an evaluation set of 100 images from ImageNet. We summarise results at query budgets and attack settings in Table 4.2 in the Appendix. Overall, we find that the performance of ViT degrades as expected,

Table 4.1: Median sparsity and ASR at different query budgets. A comprehensive comparison among different attacks (PGD₀, Pointwise and SPARSEEVO) on small and large scale balance datasets.

Setting	Query budget	Methods	CIFAR10		Query budget	ImageNet	
			Median	ASR		Median	ASR
Untargeted		PGD ₀	0.0059	99.8%		0.0005	100%
	200	Pointwise	0.0078	88.0%	2000	0.0016	68.0%
		SPARSEEVO	0.0049	96.5%		0.0008	96.5%
	500	Pointwise	0.0078	96.2%	5000	0.0012	77.0%
		SPARSEEVO	0.0049	99.2%		0.0008	99.0%
	Targeted		PGD ₀	0.0703	99.8%		0.0061
1000		Pointwise	0.9612	0.0%	10000	0.9997	0.0%
		SPARSEEVO	0.0311	96.5%		0.0511	48.5%
2000		Pointwise	0.7863	0.0%	20000	0.9975	0.0%
		SPARSEEVO	0.0251	99.6%		0.0076	99.1%

Table 4.2: Accuracy of ResNet50 and ViT under attacks at different query budgets and sparsity thresholds. A comprehensive comparison among different attacks (PGD₀ and SPARSEEVO) on small and large scale balanced evaluation sets from ImageNet

Setting	Methods	Query Budget	ResNet50		ViT	
Sparsity			0.002	0.004	0.002	0.004
Untargeted	PGD ₀	na	5%	0.0%	31%	14%
	SPARSEEVO	2000	20%	5%	45%	25%
		5000	17%	0.0%	35%	7%
Sparsity			0.02	0.03	0.02	0.03
Targeted	PGD ₀	na	2.0%	1.2%	4.4%	0.2%
	SPARSEEVO	10000	66.8%	52.8%	20%	9.0%
		20000	2.2%	0.6%	2.4%	0.2%

4.4.6 Compare The Robustness of the Transformer and the CNN

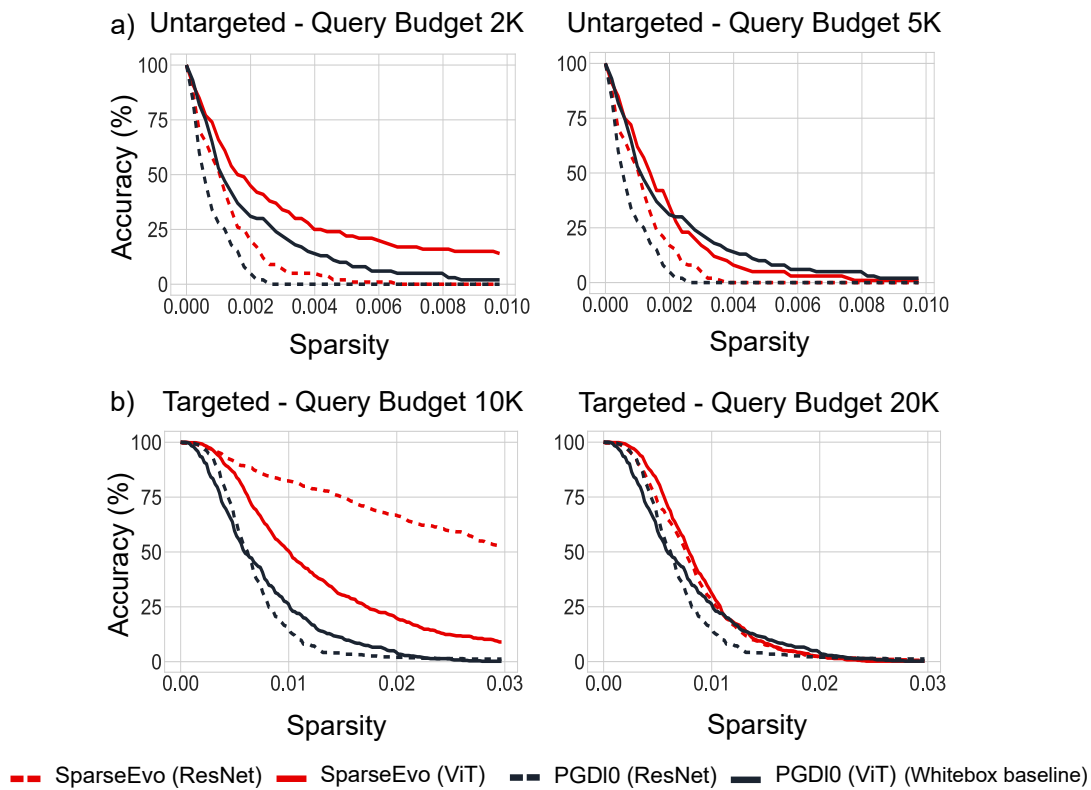


Figure 4.7. Attack success rate versus sparsity thresholds at different query budgets for the evaluation set from ImageNet with ViT vs ResNet. PGD₀ is a white-box attack (ideal).

but it appears to be less susceptible than the ResNet50 model. Particularly, in the untargeted setting, the accuracy of ViT across different sparsity thresholds is higher than the ResNet50 model under both SPARSEEVO and PGD₀. Interestingly, SPARSEEVO only needs a *small query budget of 2,000* to degrade the accuracy of ResNet50 that is similar to white-box PGD₀, while up to 5,000 queries are needed to make SPARSEEVO attack on ViT worse than PGD₀. In the targeted scenario, we observe that at a low query budget e.g. 10,000, ResNet50 is much more robust than ViT under SPARSEEVO whereas, at 20,000 queries, the accuracy of both ResNet50 and ViT models is almost analogous and drops to approximately zero when the sparse perturbation is larger than 0.02. Notably, SPARSEEVO with a sufficient query limit e.g. 20,000 is able to maintain its attack effectiveness against both ViT and ResNet50 while the attack effectiveness of PGD₀ is reduced—demonstrated by lower accuracy scores—when attacking ViT.

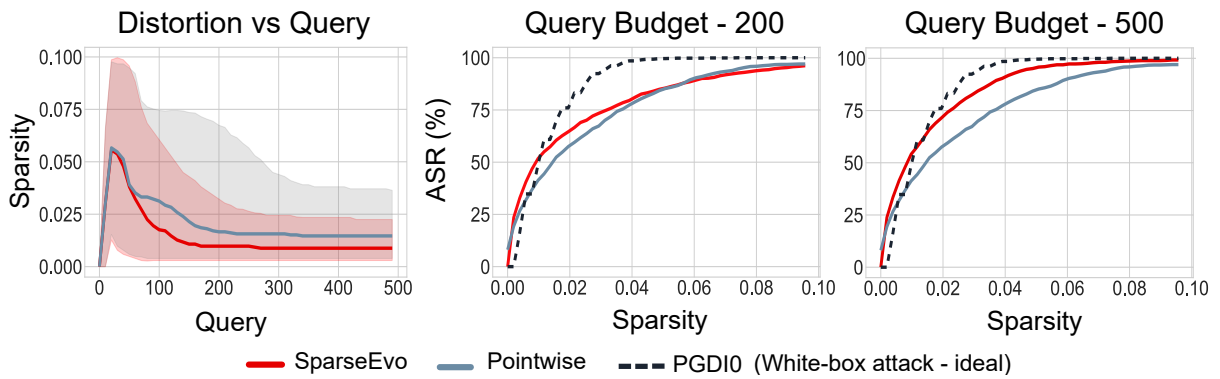


Figure 4.8. Different sparse attacks against an adversarially trained model on the CIFAR10 task. We show sparsity versus queries and ASR versus sparsity two different query budgets: 200 and 500.

4.4.7 Sparse Attacks Against an Adversarially Trained Model

This section studies the robustness of different sparse attacks against adversarially trained ResNet-18 network on the CIFAR10 task using l_∞ perturbations—one of the most effective defense mechanisms against adversarial attacks (Athalye, Carlini and Wagner, 2018). The accuracy of this adversarially trained network is 83.87%. We choose PGD₀ (Croce and Hein, 2019), a state-of-the-art *white-box attack* as a baseline for comparison. The adversarial training based models used in this experiment is trained with Projected Gradient Descent (PGD) proposed by (Madry et al., 2018).

The experiment is conducted on a balance evaluation set withdrawn from CIFAR10 randomly (we describe the dataset in Section 4.4.1). Median sparsity against the number of queries is shown in Figure 4.8. The results indicate that SPARSEEVO converges faster than the Pointwise attack. Figure 4.8 also shows the ASR at different distortion levels and query limits for different attack methods against the adversarially trained model. We observe that our attacks are able to obtain a comparable performance with the ideal white-box PGD₀ baseline attacks with a very limited query budget of merely 500 queries. Meanwhile SPARSEEVO is comparable with Pointwise with a given query budget of 200, and outperforms it with a query budget of 500.

4.5 Discussion and Conclusion

In this work, the study proposes a new algorithm for a sparse attack—SPARSEEVO—under a decision-based scenario. The comprehensive results

4.5 Discussion and Conclusion

demonstrate SPARSEEVO to outperform the state-of-the-art black-box attack in terms of sparsity and ASR within a given query budget. More importantly, in a high-resolution and large-scale dataset, SPARSEEVO illustrates significant query efficiency and remarkably lower sparsity when compared with the existing sparse attacks in the black-box setting. Most notably, the proposed black-box attack achieves comparable success under small query budgets to the state-of-the-art white-box attack—PGD₀. Notably, whilst an extensive set of results is presented in the main chapter, additional results to support the study are in Appendix B.

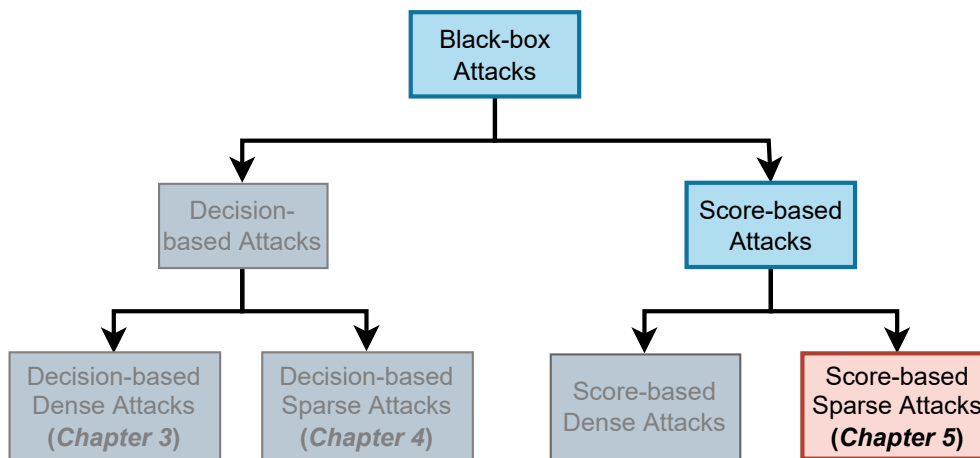


Figure 4.9. Upcoming chapter sneak peek.

The efficient sparse attacks in decision-based settings require the model decision (*i.e.* a predicted label) to mislead DNN models. A natural question is what if adversaries can access the confidence score other than the model decision. To this end, the next chapter, as depicted in Figure 4.9, will investigate a potential threat—a sparse attack in score-based settings. In particular, the next chapter will aim to answer how efficiently a score-based sparse attack can search for a small set of pixels to deceive a DNN model. Notably, sparse attacks have not been extensively studied and thus our knowledge about model weaknesses to such attacks is limited. The chapter will delve into the challenges faced, such as the non-differentiable search space and the NP-hard problem, and review existing sparse attack techniques. Subsequently, a new sparse attack method in a score-based scenario will be proposed, and its query efficiency will be empirically demonstrated through extensive experiments.

Chapter 5

BruSLeAttack: A Sparse Attack Under Score-base Settings

THIS chapter continues the study of the lesser understood problem of generating *sparse adversarial attacks* but under *score-based* replies to *model queries*. As outlined in the previous chapter, sparse attacks aim to discover a small number of pixels—the l_0 bounded—inputs in order to craft adversarial examples and *deceive* deep learning models. However, constructing sparse adversarial attacks, even with *output score* to queries in a *score-based* setting, is non-trivial, because such an attack leads to: i) an NP-hard problem; and ii) a non-differentiable search space. An intuitive approach to these challenges is to adapt decision-based sparse attacks, as investigated in Chapter 4, for score-based settings. Nevertheless, these attacks cannot achieve high query efficiency due to the lack of direct optimization and exploitation from the output scores. To remedy these problems, this chapter introduces BRUSLEATTACK—a *new* algorithm built upon a Bayesian framework for the problem and evaluates the algorithm against Convolutional Neural Networks, *Vision Transformers*, recent *stylized ImageNet* models, *defense methods* and machine learning as a service (MLaaS) (*i.e.* **Google Cloud Vision**). The proposed attack scales to achieve *state-of-the-art attack success rates* and *query efficiency* on standard computer vision tasks such as ImageNet. Importantly, the attack algorithm proposed here raises questions regarding the safety, security and reliability of deployed systems.

5.1 Motivation and Contribution

A large body of research has investigated malicious capability to deceive deep learning models (*i.e.* exploiting model decisions as discussed in Chapter 3 and 4). Since confidence scores expose more information compared to model decisions, we can expect fewer queries to elicit effective attacks. Consequently, the potential for developing *attacks at scale* under *score-based* settings is higher.

Additionally, while dense attacks have been widely explored, the success of *sparse attacks*, especially under *score-based* settings, has drawn much less attention and remains less well understood (Croce et al., 2022). This leads to a lack of knowledge regarding model vulnerabilities to sparse perturbation regimes under a score-based threat model. To this end, this chapter explores the fragility of deep learning models against *sparse attacks* in *score-based* settings.

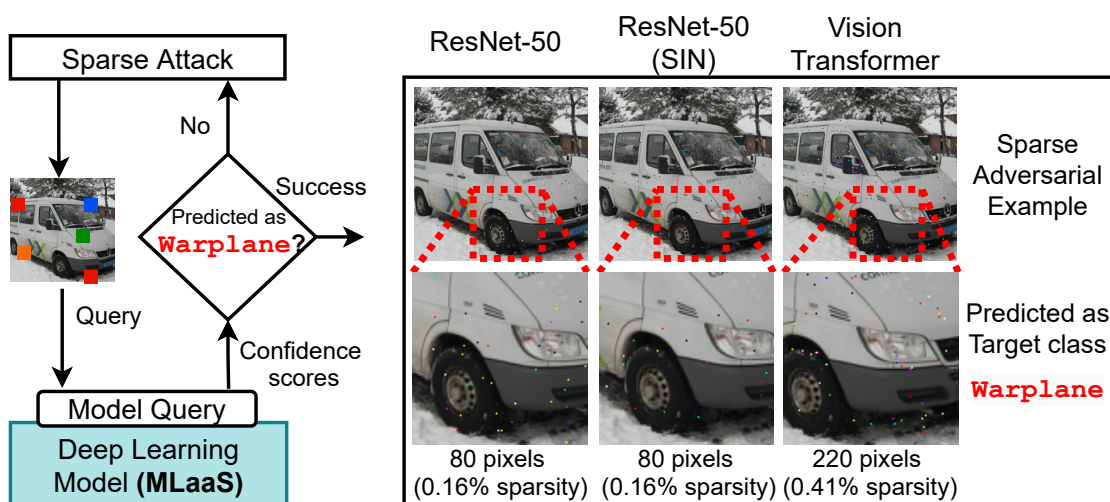


Figure 5.1. Targeted Attack. Malicious instances are generated by BRUSLEATTACK with different perturbation budgets against three deep learning models on ImageNet. An image with ground-truth label Minibus is misclassified as a Warplane. Interestingly, in contrast to needing 220 pixels to mislead the vision transformer, BRUSLEATTACK requires only 80 perturbed pixels to fool ResNet-based models (more visuals in Appendix C.16). Visualizations and evaluation against Google Cloud Vision is in Section 5.5.6 and Appendix C.15.

As discussed in Chapter 4, constructing sparse attacks is incredibly difficult as minimizing l_0 norm leads to an NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020) even with scores and a non-differentiable search spaces that are mixed (discrete and continuous) (Carlini and Wagner, 2017). Now, for a given l_0 constraint or number of pixels, it is necessary to search for the optimal

set of pixels to perturb in a source image, as well as the pixel colors—floats in $[0, 1]$. Solutions are harder if aiming to achieve both *query efficiency* and *high attack success rate* (ASR) for high-resolution vision tasks such as ImageNet. To deal with these challenges and realize a sparse attack in score-based settings, an intuitive approach is to adapt decision-based attacks for score-based settings (*i.e.* SPARSEEVO—introduced in Chapter 4). Specifically, we reformulate the score-based settings as decision-based settings by exploiting the top-1 label corresponding to the top-1 output score and equivalent to the model decision in decision-based settings. However, exploiting the top-1 label hinders direct optimization from the output scores. Therefore, decision-based sparse attacks cannot result in query efficiency in score-based settings, as shown in Section 5.5.4.

Due to the limitation of adapting decision-based attacks to sparse settings, a better approach is to design a sparse attack directly exploiting score information for searching an adversarial example. The only *scalable* attempt to deal with these challenges, SPARSE-RS (Croce et al., 2022), is to apply a stochastic search method. However, SPARSE-RS still lacks query efficiency on high-resolution data sets. Thus, the study in this chapter will consider a *new formulation* and propose a new search method—BRUSLEATTACK—for a sparse adversarial example over an effective, lower-dimensional search space. In contrast to the prior stochastic search and pixel selection methods, the search direction is guided by the knowledge learned from incorporating historical information of pixel manipulations (past experience) and the informed selection of pixel-level perturbations from a lower-dimensional search space.

To explore the fragility of deep learning models against *sparse attacks* in score-based settings, this chapter will focus on both convolutional-based and attention-based architectures. While convolutional-based architectures are used in a plethora of applications, attention-based architectures such as ViT (Dosovitskiy et al., 2021) or *Data-Efficient Image Transformers* (Touvron et al., 2021) recently produced performance breakthroughs and are generating increasing interest. Moreover, only a few studies (Modas, Moosavi-Dezfooli and Frossard, 2019; Croce and Hein, 2019; Fan et al., 2020; Dong et al., 2020) have considered robustness to sparse perturbation regimes. This raises an important problem regarding the reliable deployment of real-world applications that employ these architectures. To demonstrate the practical feasibility of sparse attacks, we also consider the Google Cloud Vision. Figure 5.1 demonstrates examples of our attack against different deep learning models on ImageNet.

5.1.1 Chapter Overview

This chapter explores the vulnerability of different deep learning models to sparse adversarial attacks in score-based settings and examines the query efficiency of these sparse attacks. As such, the study in this chapter seeks to address the following research questions (RQ).

RQ1: How can adversaries effectively construct query-efficient score-based attacks to yield highly sparse adversarial perturbations in high dimensional spaces? This question will be addressed in Section 5.4.

RQ2: How successful are score-based sparse attacks against convolutional neural networks (CNNs), vision transformers and defended models? How do vision transformers compare with CNNs in terms of robustness? This question will be explored in Section 5.5.

Contributions. This study aims to increase our understanding of lesser understood, *hard*, score-based attacks to generate *sparse* adversarial examples, the main contributions to which are threefold:

- We formulate a new sparse attack—BRUSLEATTACK—in the score-based setting. The algorithm exploits the knowledge of model output scores and our intuitions on: *i*) learning influential pixel information from historical pixel manipulations; and *ii*) informed selection of pixel perturbations based on pixel dissimilarity between our search space prior and a source image to accelerate the search for a *sparse* adversarial example.
- As a *first* step, investigate the robustness of ViT and compare its relative robustness with ResNet models on the high-resolution dataset Imagenet under score-based sparse settings.
- We demonstrate the significant query efficiency of our algorithm over the state-of-the-art counterpart in different data sets against various deep learning models, as well as defense mechanisms, Google Cloud Vision in terms of ASR and sparsity under 10K query budgets.

5.1.1 Chapter Overview

Section 5.2 presents the related work on score-based attacks; Section 5.4 introduces the problem formulation and details the proposed attack algorithm; Section 5.5 evaluates

the performance of different sparse attacks across different data sets and demonstrates a possible threat against a real-world system—Google Cloud Vision. Section 5.6 summarizes the study’s findings and concludes this chapter.

5.2 Related Work

This section provides a discussion on existing non-sparse and sparse adversarial attacks, first describing non-sparse attacks under different scenarios and then presenting sparse attack methods under decision-based and score-based scenarios.

Non-Sparse (Dense) Attacks (l_2, l_∞). Past research has extensively examined dense attacks in white-box settings (Goodfellow, Shlens and Szegedy, 2014; Madry et al., 2018; Carlini and Wagner, 2017; Dong et al., 2018; Wong, Schmidt and Kolter, 2019; Xu et al., 2020) and black-box settings (Chen et al., 2017; Tu et al., 2019; Liu et al., 2019b; Ilyas, Engstrom and Madry, 2019; Andriushchenko et al., 2020; Shukla et al., 2021). Due to the non-differentiable, high-dimensional and mixed (continuous and discrete) nature of search spaces encountered in sparse settings, adopting these methods is non-trivial (see our analysis in **Appendix C.4**). Recent work has explored sparse attacks in white-box settings (Papernot et al., 2016a; Modas, Moosavi-Dezfooli and Frossard, 2019; Croce and Hein, 2019; Fan et al., 2020; Dong et al., 2020; Zhu, Chen and Wang, 2021). Here we mainly review *sparse* attacks in *black-box* settings but compare these with a *white-box sparse attack* for interest in Section 5.5.4.

Decision-based Sparse Attacks (l_0). Only a few recent studies, such as that of POINTWISE (Schott et al., 2019) and SPARSEEVO (Chapter 4), have tackled the difficult problem of sparse attacks in decision-based settings. The fundamental difference between decision-based and score-based settings is the output information (labels versus scores) and the *need* for a target class image sample in decision-based algorithms. The label information hinders direct optimization from output information. As such, decision-based sparse attacks rely on an image from a target class (targeted attacks) and gradient-free methods. This leads to a different set of problem formulations. We study and demonstrate that sparse attacks formulated for decision-based settings do not lead to query-efficient attacks in score-based settings in Section 5.5.4.

Score-based Sparse Attacks (l_0). A score-based setting seemingly provides more information than a decision-based setting. However, the formulations of score-based

5.3 Notation Table

attacks (Narodytska and Kasiviswanathan, 2017; Zhao et al., 2019; Croce and Hein, 2019) suffer from prohibitive computational costs (low query efficiency) and do not scale to high-resolution data sets *i.e.* ImageNet. The recent SPARSE-RS algorithm in (Croce et al., 2022) reports the state-of-the-art, query-efficient, sparse attack and is a significant advance. But large query budgets are still required to achieve low sparsity on high-resolution tasks such as ImageNet in the more difficult targeted attacks.

5.3 Notation Table

Prior to delving into details, this section provides a list of notations in Table 5.1 to aid the description of the proposed approach in Section 5.4.

Table 5.1: Table of notation descriptions.

Notation	Description
x	Source image
\tilde{x}	Synthetic color image
y	Source class
y_{target}	Target class
$f(x)$	Softmax scores
$L(\cdot)$ or $\ell(\cdot)$	Loss function
B	A budget of perturbed pixels
b	A number of selected elements remaining unchanged
$u^{(t)}$	A binary matrix to determine perturbed and unperturbed pixels
$v^{(t)}$	A binary matrix to determine perturbed pixels remaining unchanged
$q^{(t)}$	A binary matrix to determine new pixels to be perturbed
α^{prior}	An initial concentration parameter
$\alpha^{\text{posterior}}$	An updated concentration parameter
θ	Parameter of Categorical distribution
$\text{Dir}(\alpha)$	Dirichlet distribution
$\text{Cat}(\theta)$	Categorical distribution
λ_0	An initial changing rate
m_1	A power decay parameter
m_2	A step decay parameter
M	Dissimilarity Map
w, h, c	Width, height and number of channels of an image

5.4 Proposed Method

In this study, we have focused on exploring adversarial attacks in the context of score-based and sparse settings. First, we present the general problem formulation for sparse adversarial attacks. Let $x \in [0, 1]^{c \times w \times h}$ be a normalized source image, where c is the number of channels, w , h are the width and height of the image and y is its ground truth label—the *source class*. Let $f(x)$ denote a vector of all class probabilities—softmax scores—from a victim model and $f(r|x)$ denote the probability of class r . An adversary aims to search for an adversarial example $\tilde{x} \in [0, 1]^{c \times w \times h}$ such that \tilde{x} can be misclassified by the victim model (untargeted setting) or classified as a target class y_{target} (targeted setting). Formally, in a targeted setting, for a given x , a sparse attack aiming to search for the best adversarial example x^* can be formulated as a constrained combinatorial optimization problem:

$$x^* = \arg \min_{\tilde{x}} L(f(\tilde{x}), y_{\text{target}}) \text{ s.t. } \|x - \tilde{x}\|_0 \leq B, \quad (5.1)$$

where $\|\cdot\|_0$ is the l_0 norm denoting the number of perturbed pixels, B denotes a budget of perturbed pixels and L denotes the loss function of the victim model f 's predictions. This loss may be different from the training loss and remains unknown to the attacker. In practice, we adopt the loss functions in (Croce et al., 2022), particularly *cross-entropy loss* in targeted settings and *margin loss* in untargeted settings. The problem with Equation 5.1 is the large search space, given that we need to search for colors—float numbers in $[0, 1]$ —for perturbing a group of pixels in the source images x .

5.4.1 New Problem Formulation to Facilitate a Solution

Sparse attacks aim to search for the *positions* and *color values* of perturbed pixels; for a normalized image, the color value of each channel of a pixel—RGB color value—can be a float number in $[0, 1]$. Consequently, the search space is enormous. Instead of searching in the mixed (discrete and continuous), high-dimensional search space, we consider turning the mixed search space problem into a lower-dimensional, discrete search space problem. Subsequently, we propose a formulation that will aid the development of a new solution to the combinatorial search problem.

Proposed Lower Dimensional Search Space. We introduce a simple but effective perturbation scheme. We uniformly sample, at random, a color image $x' \in \{0, 1\}^{c \times w \times h}$ —which we call the *synthetic color image*—to define the color of perturbed

5.4.2 A Probabilistic Framework for the l_0 Constrained Combinatorial Search

pixels in the source image x . In this manner, each pixel is allowed to attain arbitrary values in $[0, 1]$ for each color channel, but the dimensionality of the space is reduced to a discrete space of size $w \times h$. The resulting search space is eight times smaller than would be the case were we using the perturbation scheme in SPARSE-RS (Croce et al., 2022) (see an analysis in Appendix C.7). Surprisingly, our proposal is shown to be incredibly effective, particularly in high-resolution images such as ImageNet (we provide a comparative analysis with alternatives in Appendix C.8).

Search Problem Over the Lower Dimensional Space. Despite the lower-dimensional nature of the search space, a combinatorial search problem persists. As a remedy, we propose changing the problem of finding \tilde{x} to finding a binary matrix u for selecting pixels in x to construct an adversarial instance. To that end, we consider choosing a set of pixels in the given image x to be replaced by pixels from the synthetic color image $x' \in \{0, 1\}^{c \times w \times h}$. These pixels are determined by a binary matrix $u \in \{0, 1\}^{w \times h}$ where $u_{i,j} = 1$ indicates a pixel to be replaced. The adversarial image is then constructed as $\tilde{x} = ux' + (\mathbf{1} - u)x$ where $\mathbf{1}$ denotes the matrix of all ones with dimensions of u , and each element of u corresponds to one pixel of x with c channels.

Consequently, manipulating each pixel of \tilde{x} corresponds to manipulating an element in u . Therefore, rather than solving Equation 5.1, we consider the equivalent alternative (proof is shown in Appendix C.6):

$$u^* = \arg \min_u \ell(u) \quad \text{s.t. } \|u\|_0 \leq B, \quad (5.2)$$

where $\ell(u) := L(f(ux' + (\mathbf{1} - u)x), y_{\text{target}})$. Although the problem in 5.2 is combinatorial in nature and does not have a polynomial time solution, the formulation facilitates the use of two simple intuitions to iteratively generate better solutions—sparse adversarial samples.

5.4.2 A Probabilistic Framework for the l_0 Constrained Combinatorial Search

It is clear that some pixels impart a more significant impact on the model decision than others. As such, given a binary matrix u with a set of selected elements—a candidate solution, we can expect some of these elements, if altered, to be more likely to result in an increase in the loss $\ell(u)$. Then, our assumption is that some selected elements must be *hard to manipulate* to reduce the loss, and as such, should be unaltered. Retaining

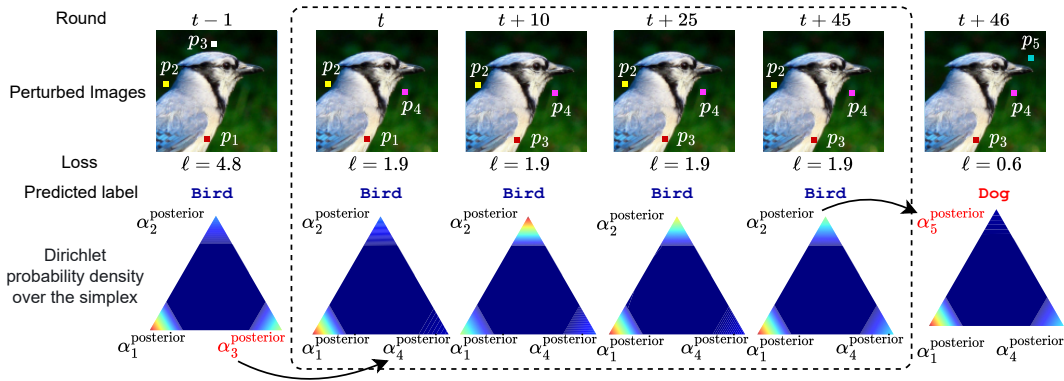


Figure 5.2. A **Sampling and Update** illustration. The attack aims to mislead a model into misclassifying a Bird image as Dog. Assuming that in round $t - 1$, an adversarial instance is classified as Bird and loss $\ell = 4.8$. We visualize *three elements* of $\alpha^{\text{posterior}}$ for simplicity. Let $\{p_1, p_2, p_3\}$ denote three perturbed pixels with corresponding posterior parameters $\{\alpha_1^{\text{posterior}}, \alpha_2^{\text{posterior}}, \alpha_3^{\text{posterior}}\}$. Assume that in round t , two pixels p_1, p_2 remain while p_3 is replaced by p_4 because a loss reduction is observed from 4.8 to 1.9. All $\{\alpha_1^{\text{posterior}}, \alpha_2^{\text{posterior}}, \alpha_3^{\text{posterior}}, \alpha_4^{\text{posterior}}\}$ are updated using Equation 5.6 but we visualize $\{\alpha_1^{\text{posterior}}, \alpha_2^{\text{posterior}}, \alpha_4^{\text{posterior}}\}$. Since $\alpha_4^{\text{posterior}}$ is new and has never been selected before, it is small in value (and represented using colder colors). From t to $t + 45$, while sampling and learning to find a better group of perturbed pixels, $\alpha^{\text{posterior}}$ is updated. Because p_1 has a high influence on the model’s prediction (represented using warmer colors), it is more likely to remain, while p_2, p_4 are more likely to be selected for a replacement due to their lower impact on the model decision. In round $t + 46$, pixel p_2 is replaced by p_5 because a loss reduction is observed from 1.9 to 0.6. Now, the predicted label is flipped from Bird to Dog.

these selected elements is more likely to circumvent a bad solution successfully. In other words, these selected elements may significantly influence the model’s decision and are worth keeping. In contrast to a stochastic search for influential pixels, we consider learning the influence of each element based on historical information about pixel manipulations.

The influence of these elements can be modeled probabilistically, with the more influential elements attaining higher probabilities. To this end, we consider a categorical distribution parameterized by θ , because we aim to select multiple elements and this is equivalent to multiple draws of one of many possible categories. It then follows to consider a Bayesian formulation to learn θ recursively. We adopt a general Bayesian framework and *design the new components and approximations* needed to learn θ . Intuitively, we can expect a new solution, u^t , generated according to θ to more likely outweigh the current solution and guide the future candidate

5.4.2 A Probabilistic Framework for the l_0 Constrained Combinatorial Search

solution towards more effectively minimizing the loss $\ell(\mathbf{u})$. Next, we describe these components and defer the algorithm we have designed, incorporating its components in Section 5.4.3.

Prior. In Bayesian statistics, the conjugate prior distribution of the categorical distribution is the Dirichlet distribution. Thus, we give θ a prior distribution defined by a Dirichlet distribution with the concentration parameter α as $P(\theta; \alpha) := \text{Dir}(\alpha)$.

Sampling $\mathbf{u}^{(t)}$. For $t > 0$, given a solution—binary matrix $\mathbf{u}^{(t-1)}$ —and $\theta^{(t)}$, we aim to: i) select and preserve highly influential selected elements (Equation 5.3); and ii) draw new elements from unselected elements (Equation 5.4), conditioned upon $\mathbf{u}^{(t-1)} = \mathbf{1}$ and $\mathbf{u}^{(t-1)} = \mathbf{0}$, respectively, to jointly yield a new solution $\mathbf{u}^{(t)}$ (Equation 5.5). Concretely, we can express this process as follows:

$$\mathbf{v}_1^{(t)} \dots, \mathbf{v}_b^{(t)} \sim \text{Cat}(\mathbf{v} \mid \theta^{(t)}, \mathbf{u}^{(t-1)} = \mathbf{1}), \quad (5.3)$$

$$\mathbf{q}_1^{(t)}, \dots, \mathbf{q}_{B-b}^{(t)} \sim \text{Cat}(\mathbf{q} \mid \theta^{(t)}, \mathbf{u}^{(t-1)} = \mathbf{0}), \quad (5.4)$$

$$\mathbf{u}^{(t)} = [\bigvee_{k=1}^b \mathbf{v}_k^{(t)}] \vee [\bigvee_{r=1}^{B-b} \mathbf{q}_r^{(t)}]. \quad (5.5)$$

Here $\mathbf{v}_k^{(t)}, \mathbf{q}_r^{(t)} \in \{0, 1\}^{w \times h}$, B denotes a total number of selected elements (the perturbation budget), b represents the number of selected elements that remain unchanged, and \vee denotes logical OR operator.

Updating $\theta^{(t)}$ (Using Our Proposed Likelihood). Finding the exact solution for the underlying parameters $\theta^{(t)}$ of the categorical distribution in Equation 5.3 and Equation 5.4 to increase the likelihood of yielding a better solution for $\mathbf{u}^{(t)}$ in Equation 5.5 is often intractable. Our approach is to find an estimate of $\theta^{(t)}$ by obtaining the expectation of the posterior distribution of the parameter, which is learned and updated over time through Bayesian inference. We note that since the prior distribution of the parameter is a Dirichlet, which is the conjugate prior of the categorical (*i.e.* distribution of \mathbf{u}), the posterior of the parameter is also Dirichlet. Formally, at each step $t > 0$, updating the posterior and $\theta^{(t)}$ is formulated as follows:

$$\alpha_{i,j}^{\text{posterior}} = \alpha_{i,j}^{\text{prior}} + s_{i,j}^{(t)} \quad (5.6)$$

$$P(\theta \mid \alpha, \mathbf{u}^{(t-1)}, \ell^{(t-1)}) := \text{Dir}(\alpha^{\text{posterior}}) \quad (5.7)$$

$$\theta^{(t)} = \mathbb{E}_{\theta \sim P(\theta \mid \alpha, \mathbf{u}^{(t-1)}, \ell^{(t-1)})}[\theta], \quad (5.8)$$

where $\alpha^{\text{prior}} = \alpha^{(0)}$ is the initial concentration parameter, $\alpha^{\text{posterior}} = \alpha^{(t)}$ denotes the updated concentration parameter (illustration in Figure 5.2) and $s_{i,j}^{(t)} =$

Algorithm 5.1: BRUSLEATTACK

Input: source image \mathbf{x} , synthetic color image \mathbf{x}' , source label y , target label y_{target} , model f
query limit T , scheduler parameters m_1, m_2 , initial changing rate λ_0
perturbation budget B , a number of initial samples N , concentration parameters α^{Prior}

- 1 Create Dissimilarity Map \mathbf{M} using Equation 5.11
- 2 $\mathbf{u}^{(0)}, \ell^{(0)} \leftarrow \text{INITIALIZATION}(\mathbf{x}, \mathbf{x}', y, y_{\text{target}}, f, N, B)$
- 3 $t \leftarrow 1, \mathbf{a}^{(0)} \leftarrow \mathbf{0}, \mathbf{n}^{(0)} \leftarrow \mathbf{u}^{(0)}$
- 4 Calculate $\theta^{(0)}$ using α^{Prior} and Equation 5.8
- 5 **while** $t < T$ and $y^{(t)} \neq y_{\text{target}}$ **do**
- 6 $\lambda^{(t)} \leftarrow \lambda_0(t^{m_1} + m_2^t)$
- 7 */* Generate a new solution */*
- 8 $\mathbf{u}^{(t)} \leftarrow \text{GENERATION}(\theta^{(t)}, \mathbf{M}, \mathbf{u}^{(t-1)}, \lambda^{(t)})$
- 9 $\ell^{(t)} \leftarrow L(f(\mathbf{u}^{(t)}\mathbf{x}' + (\mathbf{1} - \mathbf{u}^{(t)})\mathbf{x}), y_{\text{target}})$
- 10 $y^{(t)} \leftarrow \arg \max_r f(r|\mathbf{u}^{(t)}\mathbf{x}' + (\mathbf{1} - \mathbf{u}^{(t)})\mathbf{x})$
- 11 */* Update θ and solution */*
- 12 $\mathbf{u}^{(t)}, \ell^{(t)}, \theta^{(t)}, \mathbf{a}^{(t)}, \mathbf{n}^{(t)} \leftarrow \text{UPDATE}(\mathbf{u}^{(t)}, \ell^{(t)}, \mathbf{u}^{(t-1)}, \ell^{(t-1)}, \mathbf{a}^{(t)}, \mathbf{n}^{(t)})$
- 13 $t \leftarrow t + 1$
- 14 **end while**
- 15 **return** $\mathbf{u}^{(t)}$

$((a_{i,j}^{(t)} + z)/(n_{i,j}^{(t)} + z)) - 1$. z is a small constant (*i.e.* 0.01) to ensure that both the nominator and denominator are always non-zero. This smoothing technique is applied since both the nominator and denominator can be zero when "never" manipulated pixels are selected. $a_{i,j}^{(t)}$ is the accumulation of altered pixel i, j (*i.e.* $u_{i,j}^{(t)} = 0$ and $u_{i,j}^{(t-1)} = 1$) when it leads to an increase in the loss, *i.e.* $\ell^{(t)} \geq \ell^{(t-1)}$ and $n_{i,j}^{(t)}$ is the accumulation of selected pixel i, j in the mask $\mathbf{u}^{(t)}$. Formally, $a_{i,j}^{(t)}$ and $n_{i,j}^{(t)}$ can be updated as the following:

5.4.3 Sparse Attack Algorithm

$$a_{i,j}^{(t)} = \begin{cases} a_{i,j}^{(t-1)} + 1 & \text{if } \ell^t \geq \ell^{(t-1)} \wedge u_{i,j}^{(t)} = 1 \wedge u_{i,j}^{(t-1)} = 0 \\ a_{i,j}^{(t-1)} & \text{otherwise} \end{cases} \quad (5.9)$$

$$n_{i,j}^{(t)} = \begin{cases} n_{i,j}^{(t-1)} + 1 & \text{if } u_{i,j}^{(t)} = 1 \vee u_{i,j}^{(t-1)} = 1 \\ n_{i,j}^{(t-1)} & \text{otherwise} \end{cases} \quad (5.10)$$

5.4.3 Sparse Attack Algorithm

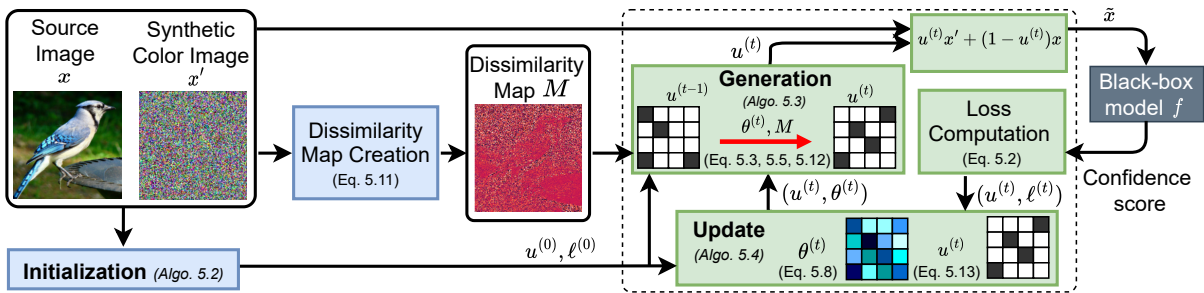


Figure 5.3. BRUSLEATTACK algorithm (detailed in Algo. 5.1). We aim to determine a set of pixels to replace in the source image x by corresponding pixels in a synthetic color image x' . In the solution, binary matrix $u^{(t)}$, white and black colors denote replaced and non-replaced pixels of the source image, respectively. **First**, our **intuition** is to retain useful elements in the solution $u^{(t)}$ by learning from historical pixel manipulations. We explore and **learn** the influence of selected elements by capturing it with θ using a general Bayesian framework—darker colors illustrate the higher influence of selected elements (Algo. 5.4). **Second**, we *generate* new pixel perturbations based on θ with the **intuition** that a larger pixel dissimilarity M between our search space x' and a source image can possibly move the adversarial to the decision boundary faster and accelerate the search (Algo. 5.3).

Using the probabilistic framework for l_0 constrained combinatorial search, we devise our sparse attack illustrated in Figure 5.3 and detailed in Algorithm 5.1. The attack phases are discussed in detail below:

Initialization (Algorithm 5.2). Given a perturbation budget B and a zero-initialized matrix, N first solutions are generated by uniformly altering B elements to 1 at random. The best solution which incurs the lowest loss $\ell^{(0)}$ is selected to be an initial solution $u^{(0)}$. $\theta^{(0)}$ is the expectation of α^{prior} .

Generation (Algorithm 5.3). It is necessary here to balance exploration versus exploitation, as in other optimization methods. Initially, to explore the search space,

Algorithm 5.2: INITIALIZATION

Input: source image x , synthetic color image x' source label y , target label y_{target}
number of initial samples N , perturbation budget B , victim model f_M

- 1 $\ell \leftarrow \infty$
- 2 **for** $i = 1$ **to** N **do**
- 3 Generate u' by uniformly enabling B bits of $\mathbf{0}$ at random
- 4 $\ell' \leftarrow L(f_M(g(u'; x, x')), y^*)$
- 5 **if** $\ell' < \ell$ **then**
- 6 $u \leftarrow u', \ell \leftarrow \ell'$
- 7 **end for**
- 8 **return** u, ℓ

we aim to manipulate a large number of selected elements. When approaching an optimal solution, we aim at exploitation to search for a solution in a region nearby the optimal solution and thus alter a small number of selected elements. Therefore, we use the combination of power and step decay schedulers to regulate a number of selected elements altered in round t . This scheduler is formulated as $\lambda_t = \lambda_0(t^{m_1} + m_2^t)$, where λ_0 is an initial changing rate, m_1, m_2 are power and step decay parameters respectively. Concretely, we define a number of selected elements remaining unchanged as $b = \lceil (1 - \lambda_t)B \rceil$.

Algorithm 5.3: GENERATION

Input: probability θ , bias map M , mask u , changing rate λ

- 1 $b \leftarrow \lceil (1 - \lambda)B \rceil$
- 2 $v_1 \dots, v_b \sim \text{Cat}(v \mid \theta, u = \mathbf{1})$
- 3 $q_1 \dots, q_{B-b} \sim \text{Cat}(q \mid \theta M, u = \mathbf{0})$
- 4 $u^{(t)} = [\vee_{k=1}^b v_k^{(t)}] \vee [\vee_{r=1}^{B-b} q_r^{(t)}]$
- 5 **return** u

Given a prior concentration parameter α^{prior} , to generate a new solution in round t , we first find $\alpha^{\text{posterior}}$ as in Equation 5.6 and estimate $\theta^{(t)}$ as in Equation 5.8. We then generate $v_k^{(t)}$ and $q_r^{(t)}$ as in Equation 5.3 and Equation 5.4 respectively. A new solution $u^{(t)}$ can be then formed as in Equation 5.5. Nonetheless, the naive approach of sampling $q_r^{(t)}$ as in Equation 5.4 is ineffective and achieves a low performance at low levels of sparsity as shown in Appendix C.10. Intuitively, when altering unselected

5.4.3 Sparse Attack Algorithm

Algorithm 5.4: UPDATE

Input: previous mask and loss $\mathbf{u}^{(t-1)}, \ell^{(t-1)}$, current mask and loss $\mathbf{u}^{(t)}, \ell^{(t)}$, small constant z , matrices $\mathbf{a}^{(t)}, \mathbf{n}^{(t)}$

- 1 $\mathbf{a} \leftarrow \mathbf{a}^{(t)}, \mathbf{n} \leftarrow \mathbf{n}^{(t)}$
- 2 $n_{i,j \in \{[i,j] | (u^{(t-1)} \vee u^{(t)})_{i,j} = 1\}}$ increase by 1
- 3 **if** $\ell^{(t)} < \ell^{(t-1)}$ **then**
- 4 $\mathbf{u} \leftarrow \mathbf{u}^{(t)}, \ell \leftarrow \ell^{(t)}$
- 5 **else**
- 6 $\mathbf{u} \leftarrow \mathbf{u}^{(t-1)}, \ell \leftarrow \ell^{(t-1)}$
- 7 $a_{i,j \in \{[i,j] | (u^{(t-1)} \oplus (u^{(t-1)} \wedge u^{(t)}))_{i,j} = 1\}}$ increase by 1
- 8 **end if**
- 9 $\mathbf{s} \leftarrow \frac{\mathbf{a} + \mathbf{z}}{\mathbf{n} + \mathbf{z}} - 1$
- 10 Update $\alpha^{\text{posterior}}$ using \mathbf{s} and Equation 5.6
- 11 Update θ using $\alpha^{\text{posterior}}$ and Equation 5.8
- 12 **return** $\mathbf{u}, \ell, \theta, \mathbf{a}, \mathbf{n}$

elements that are equivalent to replacing non-perturbed pixels in the source image with their corresponding pixels from the synthetic color image, the adversarial instance moves away from the source image by a distance. At a low sparsity level, since a small fraction of unselected elements are altered, the adversarial instance is able to take small steps toward the decision boundary between the source and target class. To mitigate this problem (taking inspiration from (Brunner et al., 2019)) we employ a prior knowledge of the *pixel dissimilarity* between the source image and the synthetic color image. Our intuition is that larger pixel dissimilarities lead to larger steps. As such, it is possible that altering unselected elements with a large pixel dissimilarity moves the adversarial instance to the decision boundary faster and accelerates optimization. The pixel dissimilarity is captured by a dissimilarity map \mathbf{M} as follows:

$$\mathbf{M} = \frac{\sum_{c=0}^2 |x_c - x'_c|}{3}, \quad (5.11)$$

where c denotes a channel of a pixel. In practice, to incorporate \mathbf{M} into the step of sampling $q_r^{(t)}$, Equation 5.4 is changed to the following:

$$q_1^{(t)}, \dots, q_{B-b}^{(t)} \sim \text{Cat}(q | \theta^{(t)} \mathbf{M}, \mathbf{u}^{(t-1)} = 0) \quad (5.12)$$

Update (Algorithm 5.4). The generated solution $\mathbf{u}^{(t)}$ is associated with a loss $\ell^{(t)}$ given by the loss function in Equation 5.2. This is then used to update $\alpha^{\text{posterior}}$ (Equation 5.6 and illustration in Figure 5.2) and the accepted solution as the following:

$$\mathbf{u}^{(t)} = \begin{cases} \mathbf{u}^{(t)} & \text{if } \ell^{(t)} < \ell^{(t-1)} \\ \mathbf{u}^{(t-1)} & \text{otherwise} \end{cases} \quad (5.13)$$

5.5 Experiments and Evaluations

This section evaluates the robustness and query efficiency of BRUSLEATTACK and compares it with SPARSE-RS—the state-of-the-art sparse attack in score-based settings, PGD₀—white-box adapted l_0 attack—and SPARSEEVO—the state-of-the-art sparse attack in decision-based settings. These attacks are evaluated on three standard vision tasks CIFAR10 (Krizhevsky, Nair and Hinton, n.d.), STL-10 (Coates, Lee and Ng, 2011) and ImageNet (Deng et al., 2009).

5.5.1 Experiment Settings

Attacks and Datasets. For a comprehensive evaluation of BRUSLEATTACK, we compose of evaluation sets from CIFAR-10 (Krizhevsky, Nair and Hinton, n.d.), STL-10 (Coates, Lee and Ng, 2011) and ImageNet (Deng et al., 2009). For CIFAR-10 and STL-10, we select 9,000 and 60,094 different pairs of the source image and target class respectively. For ImageNet, we randomly select 200 *correctly* classified test images evenly distributed among 200 random classes from ImageNet. To reduce the computational burden of the evaluation tasks in the *targeted* setting, five target classes are randomly chosen for each image. For attacks against defended models with adversarial training, we randomly select 500 *correctly* classified test images evenly distributed among 500 random classes from ImageNet. We compare with the state-of-the-art SPARSE-RS (Croce et al., 2022).

Models. For convolution-based networks, we use models based on a state-of-the-art architecture—ResNet—(He et al., 2016) including ResNet18 achieving 95.28% test accuracy on CIFAR-10, ResNet-9 obtaining 83.5% test accuracy on STL-10, pre-trained ResNet-50 (Marcel and Rodriguez, 2010) with a 76.15% Top-1 test accuracy, pre-trained stylized ImageNet ResNet-50—ResNet-50 (SIN)—with a 76.72% Top-1 test accuracy (Geirhos et al., 2019) on ImageNet. For the attention-based network, we use a

5.5.2 Experimental Regime

pre-trained ViT-B/16 model achieving 77.91% Top-1 test accuracy (Dosovitskiy et al., 2021). For robust ResNet-50 models⁷, we use adversarially pre-trained l_2/l_∞ models (l_2 -At and l_∞ -AT) (Logan et al., 2019) with 57.9% and 62.42% clean test accuracy respectively.

Evaluation Metrics. We define a *sparsity* metric as the number of perturbed pixels divided by the total pixels of an image. To evaluate the performance of an attack, we use *attack success rate* (ASR). A generated perturbation is successful if it can yield an adversarial example with sparsity *below a given sparsity threshold*, so ASR is defined as *the number of successful attacks over the entire evaluation set at different sparsity thresholds*. We measure the *robustness* of a model by the accuracy of that model under an attack at different query limits and sparsity levels.

5.5.2 Experimental Regime

This section summarizes all extensive experiments conducted on CIFAR-10, STL-10 and ImageNet datasets with different sparse attacks.

- *Sparse Attacks against Deep Learning Model.* Section 5.5.3, Appendix C.1, Appendix C.2 and Appendix C.3 evaluates the robustness of sparse attacks against different deep learning models in score-based settings across different datasets. This section also compares the robustness of the ViT model with the CNN model against sparse attacks.
- *BRUSLEATTACK versus Decision-Based sparse and l_0 -Adapted Attack Algorithms.* Section 5.5.4 and Appendix C.4 compares the performance of BRUSLEATTACK with decision-based sparse, Bayesian optimization-based and l_0 -adapted attack algorithms in targeted settings.
- *Sparse Attacks against a Defended Model.* Section 5.5.5 examines the robustness of BRUSLEATTACK and other sparse attacks against an adversarially trained model and a robust defense designed for black-box attacks.
- *Sparse Attacks against a Real-World System.* Section 5.5.6 demonstrates the practical threat of BRUSLEATTACK against a real-world system—Google Cloud Vision.

⁷<https://github.com/MadryLab/robustness>

- *Influence of synthetic image, prior knowledge and hyper-parameters.* Appendix C.8 analyzes the initialization of synthetic image. Appendix C.10 investigates the influence of prior knowledge with a pixel dissimilarity map. Appendix C.12 studies the impact of key hyper-parameters.

5.5.3 Attacking Transformers & Convolutional Nets

In our study, we carried out comprehensive experiments on the ImageNet task under the targeted setting to investigate sparse attacks against various deep learning models (standard ResNet-50, ResNet-50 (SIN) and ViT). The results for the targeted and untargeted settings are detailed in Appendix C.1. Additional results on STL-10 and CIFAR-10 are provided in Appendix C.2 and C.3 respectively.

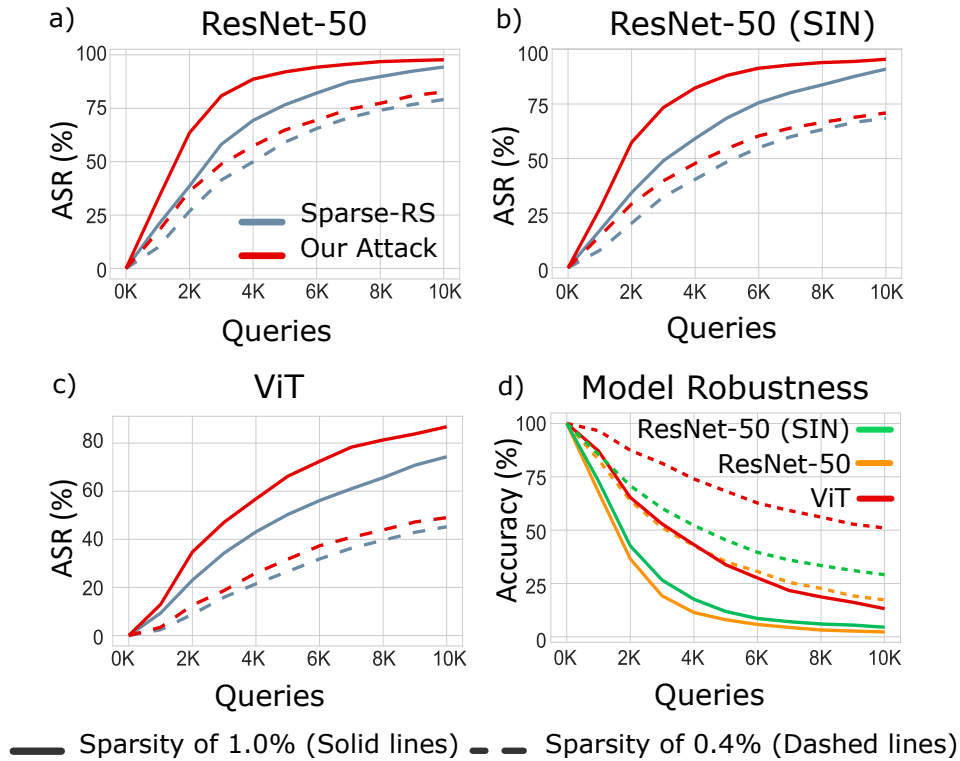


Figure 5.4. Targeted setting on ImageNet. a-c) ASR of BRUSLEATTACK and SPARSE-RS against different deep learning models at sparsity levels of 0.4% (solid lines) and 1.0% (dashed lines); d) Accuracy of different models against BRUSLEATTACK at sparsity levels of 0.4% (solid lines) and 1.0% (dashed lines). More results on ImageNet in targeted settings (sparsity between 0.4% and 1.0%) and untargeted settings in Appendix C.1.

Convolutional-based Models. Figure 5.4a and 5.4b show that, at sparsity 0.4% ($\approx \frac{200}{224 \times 224}$), BRUSLEATTACK achieves slightly higher ASR than SPARSE-RS while at

5.5.4 Comparing Performance with Prior Decision-Based and l_0 -Adapted Attack Algorithms

sparsity 1.0% ($\approx \frac{500}{224 \times 224}$), our attack significantly outweighs SPARSE-RS at different queries. Specifically, at queries from 2K to 6K, BRUSLEATTACK obtains about 10% higher ASR than SPARSE-RS. Notably, a small query budget of 6K queries is adequate for BRUSLEATTACK to achieve ASR higher than 90%.

Attention-based Model. Figure 5.4c demonstrates that at sparsity of 0.4% BRUSLEATTACK achieves a marginally higher ASR than SPARSE-RS, whereas at a sparsity of 1.0% our attack demonstrates a significantly better ASR than SPARSE-RS. At 1.0% sparsity and with query budgets above 2K, our method achieves roughly 10% higher ASR than SPARSE-RS. Overall, our method consistently outperforms the SPARSE-RS in terms of ASR across different query budgets and sparsity levels.

The Robustness of Transformer versus CNN

Figure 5.4d demonstrates the robustness of the ResNet-50, ResNet-50 (SIN) and ViT models to adversarially sparse perturbation in the targeted settings. We observe that the performance of all three models degrades as expected. Although ResNet-50 (SIN) is far more robust against several types of image corruptions than the standard ResNet-50, as shown in (Geirhos et al., 2019), it is equally vulnerable to sparse adversarial attacks. Notably, our results in Figure 5.4d illustrate that ViT is *much less susceptible* than the ResNet family to adversarially sparse perturbation. At sparsity levels of 0.4% and 1.0%, the accuracy of ViT is pragmatically higher than both ResNet models under our attack across different queries. Interestingly, BRUSLEATTACK merely requires a *small query budget of 4K* to degrade the accuracy of both ResNet models to the same accuracy of ViT at 10K queries. These findings can be explained by the fact that ViT’s receptive field spans over the whole image (Naseer et al., 2021) because some attention heads of ViT in the lower layers pay attention to the entire image (Paul and Chen, 2022). It is thus capable of enhancing relationships between various regions of the image and is more difficult to evade than convolutional-based models if a small subset of pixels is manipulated. Additional results in untargeted settings are shown in Appendix C.1.

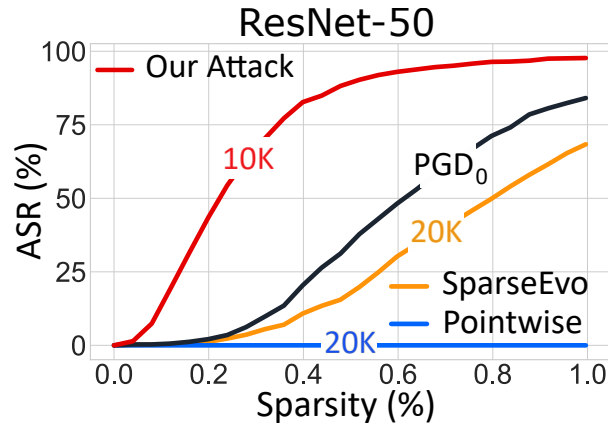


Figure 5.5. Targeted attacks on the ImageNet task against ResNet-50. ASR comparisons between BRUSLEATTACK and baselines: i) SPARSEEVO and POINTWISE (SOTA algorithms from decision-based settings); ii) PGD₀ (whitebox).

5.5.4 Comparing Performance with Prior Decision-Based and l_0 -Adapted Attack Algorithms

In this section, we compare our method (10K queries) with baselines—SPARSEEVO (Chapter 4), Pointwise (Schott et al., 2019) (both 20K queries) and PGD₀ (Croce and Hein, 2019; Croce et al., 2022) (white-box)—in targeted settings. Figure 5.5 demonstrates that BRUSLEATTACK significantly outperforms SPARSEEVO and PGD₀. This is as expected for SPARSEEVO and Pointwise, because decision-based attacks only have access to the hard label. For PGD₀, this outcome is predictable, since in this method the l_0 project step PGD₀ has to identify the minimum number of pixels required for projecting, such that the perturbed image remains adversarial, while there appears to be no effective projection method to identify the pixels that can satisfy this projection constraint. Notably, solving l_0 projection problem also leads to another NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020) and hinders the adoption of dense attack algorithms to the l_0 constraint. Moreover, the discrete nature of the l_0 ball impedes its amenability to continuous optimization (Croce et al., 2022). Additional results for l_0 adapted decision-based attacks on CIFAR-10 are presented in Appendix C.4.

Alternative Loss for SPARSEEVO. Chapter 4 points out an alternative fitness function based on output scores by replacing the objective to optimise distortion with an objective to optimise loss. Therefore, this section evaluates SPARSEEVO with an alternative fitness function in score-based settings. However, employing this

5.5.4 Comparing Performance with Prior Decision-Based and l_0 -Adapted Attack Algorithms

Table 5.2: ASR comparison between our proposal and SPARSEEVO (Alternative Loss) on CIFAR-10.

Sparsity	Our Proposal	SPARSEEVO (Alternative Loss)
1.0%	68.21%	54.78%
2.0%	90.24%	68.75%
2.9%	96.59%	74.0%
3.9%	98.48%	78.56%

alternative fitness function may not obtain a low sparsity level because minimising the loss does not result in a reduction in the number of pixels. Additionally, the Binary Differential Recombination (BDR) in Chapter 4 is designed for optimising l_0 distortion, not a loss objective (*i.e.* alters perturbed pixels to non-perturbed pixels which is equivalent to minimising distortion). Hence, naively adapting SPARSEEVO in Chapter 4 to score-based settings may not work well.

To demonstrate, we conducted an experiment on CIFAR-10 using the same experimental setup (same evaluation set of 9000 image pairs and a query budget of 500).

- First approach, we adapted the attack method in Chapter 4 to the score-based setting with an alternative fitness function for minimizing loss based on the output scores. We observed this attack always fails to yield an adversarial example with a sparsity level below 50%.
- Second approach, we adapted SPARSEEVO by employing the alternative fitness function, synthetic color image and slightly modifying BDR. Our results in Table 5.2 show that the adapted SPARSEEVO can create sparse adversarial examples but is unable to achieve a comparable performance to BRUSLEATTACK.

Overall, even with significant improvements, the sparse attack proposed in Chapter 4 with an alternative fitness function does not achieve as good performance as BRUSLEATTACK with a low query budget.

Differences Between BRUSLEATTACK and SPARSEEVO. Chapter 4 develops an algorithm for a sparse attack but assumes a decision-based setting. Although both works aim to propose sparse attacks, key differences exist, as expected; we explain these differences below:

Table 5.3: A comparison of ASR between our proposal (Synthetic Color Image) and employing a starting image as in Chapter 4 on CIFAR-10.

Sparsity	Our Proposal	Use starting image
1.0%	68.21%	62.68%
2.0%	90.24%	87.17%
2.9%	96.59%	94.37%
3.9%	98.48%	97.17%

- While both works discuss how they reduce dimensionality (a dimensionality reduction scheme) leading to a reduction in search space from $C \times H \times W$ to $H \times W$, Chapter 4 neither proposes a New Problem Formulation nor gives proof of showing the equivalent between the original problem in Equation (1) and the New Problem Formulation in Equation (2) as we did in Section 5.4.1 and Appendix C.6.
- Chapter 4 and Chapter 5 propose similar terms binary matrix u versus binary vector v as well as an interpolation between x and x' . However, a binary vector x in Chapter 4 evolves to reduce the number of 1-bits while a binary matrix u in our study maintains a number of 1-elements during searching for a solution.
- We can find a similar notion of employing a starting image (a pre-selected image from a target class) in Chapter 4 or synthetic color image (pre-defined by randomly generating) in our study. However, it is worth noting that applying a synthetic color image to Chapter 4 does not work in the targeted setting. For instance, to the best of our knowledge, there is no method can generate a synthetic color image that can be classified as a target class so the method in Chapter 4 is not able to employ a synthetic color image to initialize a targeted attack. In contrast, employing a starting image as used in Chapter 4 does not result in query-efficiency as shown in Table 5.3, especially at low sparsity levels.

5.5.5 Attacking Defended Models

BRUSLEATTACK *versus* SPARSE-RS. In this section, we investigate the robustness of sparse attacks (with a budget of 5K queries) against adversarial training-based models using projected gradient descent (PGD) as proposed by (Madry et al.,

5.5.6 Attacking a Real-World System

Table 5.4: A robustness comparison (lower \downarrow is stronger) between SPARSE-RS and BRUSLEATTACK against undefended and defended models employing l_∞ , l_2 adversarially trained models and RND on ImageNet. The robustness of the attacks is measured by the degraded accuracy of models under attacks at different sparsity levels.

Sparsity	Undefended Model		l_∞ -AT		l_2 -AT		RND	
	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK
0.04%	33.6%	24.0%	43.8%	42.2%	89.8%	88.4%	90.8%	85.0%
0.08%	13.2%	6.8%	26.8%	24.4%	81.2%	79.2%	82.2%	72.6%
0.12%	7.6%	2.6%	19.0%	18.4%	75.8%	73.8%	73.6%	61.0%
0.16%	5.2%	1.0%	16.6%	14.8%	71.4%	69.2%	64.8%	51.4%
0.2%	4.6%	1.0%	12.2%	11.8%	68.4%	66.4%	56.8%	42.6%

2018)—highly effective defense mechanisms against adversarial attacks (Athalye, Carlini and Wagner, 2018) and random noise defense (RND) (Qin et al., 2021)—a recent defense method designed for black-box attacks. The robustness of the attacks is measured by the degraded accuracy of models under attacks at different sparsity levels. The stronger an attack is, the lower the accuracy of the defended model. The results in Table 5.4 show that BRUSLEATTACK consistently outweighs SPARSE-RS across various defense mechanisms and different sparsity levels. Additional results on CIFAR-10 is provided in Appendix C.5.

Undefended and Defended Models. The results in Table 5.4 show the accuracy of undefended versus defended models against sparse attacks across different sparsity levels. In particular, under BRUSLEATTACK and sparsity of 0.2%, the accuracy of ResNet-50 drops to 1% while the l_∞ -AT model is able to obtain 11.8%. However, the l_2 -AT model and RND strongly resist adversarially sparse perturbations and remain accurate around 66.4% and 42.6 % respectively. Therefore, the l_2 -AT model and RND are more robust than the l_∞ -AT model in defending against sparse attacks.

5.5.6 Attacking a Real-World System

To illustrate the applicability and efficacy of BRUSLEATTACK against real-world systems, we attack the Google Cloud Vision (GCV) API provided by Google. Attacking GCV is challenging since 1) the classifier returns partial observations of predicted scores with a varied length based on the input and 2) the scores are neither probabilities (softmax scores) nor logits (Ilyas et al., 2018; Guo, Frank and Weinberger, 2019). To deal with these challenges, we employ the *marginal loss* between the top label and the target label and

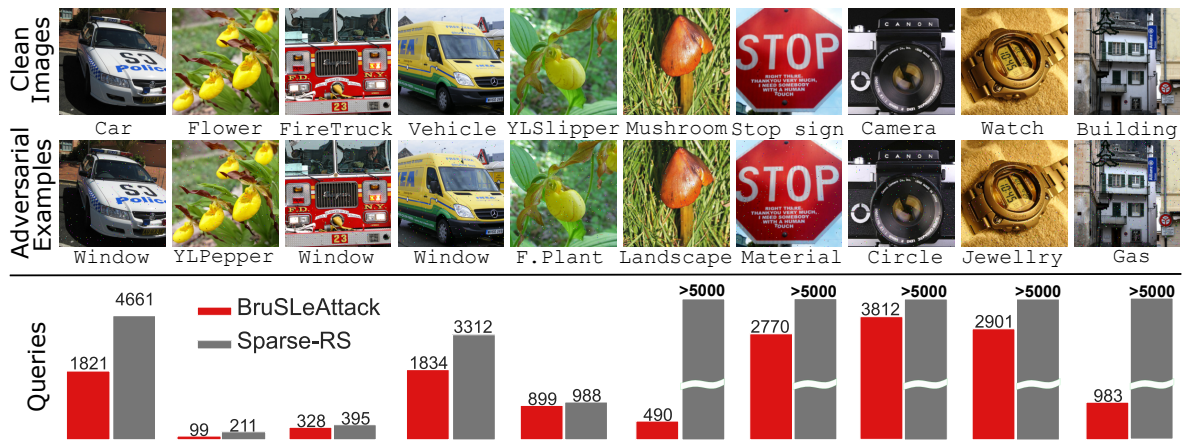


Figure 5.6. Demonstration of sparse attacks against GCV in targeted settings with a budget of 5K queries and sparsity of $0.5\% \approx \frac{250}{224 \times 224}$. BRUSLEATTACK is able to yield adversarial examples for all clean images with less queries than SPARSE-RS while SPARSE-RS fails to yield adversarial examples for Mushroom, Camera, Watch, & Building images. Illustration on **GCV API** (online platform) is shown in Appendix C.15.

successfully demonstrate our attack against GCV. With a budget of 5K queries and sparsity of 0.5%, BRUSLEATTACK is able to craft a sparse adversarial example of all given images to mislead GCV whereas SPARSE-RS fails to attack four of them as shown in Figure 5.6.

5.6 Discussion and Conclusion

The work in this chapter delves into the robustness of DNN models against sparse attacks in the score-based scenario and proposes a novel sparse attack—BRUSLEATTACK. This work demonstrates that when attacking different deep learning models, including undefended and defended models and in different datasets, BRUSLEATTACK consistently outperforms the state-of-the-art method in terms of ASR at different query budgets. Crucially, in a high-resolution data set, our comprehensive experiments show that BRUSLEATTACK is remarkably query-efficient and reaches higher ASR than the current state-of-the-art sparse attack. Notably, whilst an extensive set of results is presented in the main chapter, additional results to support the study are in Appendix C.

Until this point, this dissertation has concentrated on query-based black-box attacks in different settings and l_p norm constraints. Defending against these query-based black-box attacks is not trivial when the aim is to achieve the objectives of both high

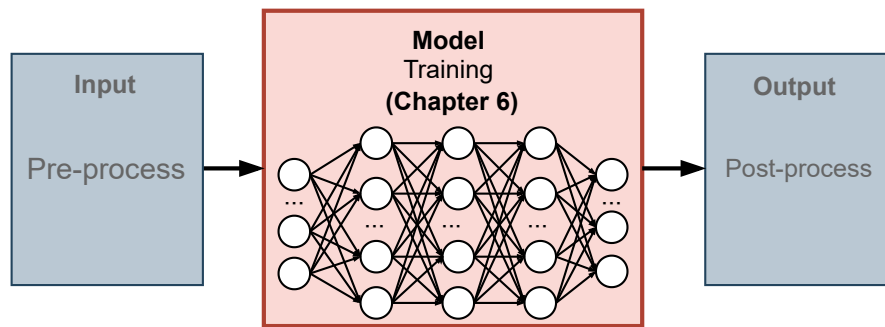


Figure 5.7. Upcoming chapter sneak peek.

robustness and clean accuracy. The following chapter, as illustrated in Figure 5.7, will study existing defense frameworks designed for query-based black-box attacks in order to understand their limitation in achieving robustness while maintaining clean accuracy. The chapter will then introduce a more effective defense method for achieving both strong robustness and high clean accuracy.

Model Diversity: A Defense Approach Against Query-Based Attacks

AS highlighted in Chapter 1 and investigated in Chapters 3, 4 and 5 the realization of query-based black-box adversarial attacks poses critical threats against the safety and security of deep learning models deployed in real-world systems. These safety and security concerns provide a reason to investigate mechanisms to defend against such black-box attacks. Although existing defense methods designed for white-box attacks can be leveraged, they sacrifice model performance for robustness. Therefore, this chapter aims to develop a *new defense framework* with the goal of achieving the twin objectives of robustness to black-box attacks and high model performance. The core idea is to introduce uncertainty into each query response by randomly selecting a well-trained model from a set for inference. To enhance this uncertainty, we aim to learn a set of well-performing models by promoting *model parameter diversity* during training using a *new* learning objective. By injecting uncertainty imparted, naturally, by the model parameter diversity, our proposed method is able to hinder the progress of query-based black-box attacks and make it significantly more difficult for models to be misled and hijacked. Although conceptually simple, the comprehensive empirical analysis in this chapter demonstrates the effectiveness of the proposed method against query-based black-box attacks.

6.1 Motivation and Contribution

The comprehensive analysis and extensive experiments outlined in Chapters 3, 4 and 5 facilitates the understanding and recognition of critical threats posed by black-box adversarial attacks to deep learning models deployed in real-world systems (*i.e.* Google Cloud Vision). These threats compel us to study mechanisms that can fortify deep learning models against such black-box attacks. One possible approach to these threats is to leverage existing defense methods (Goodfellow, Shlens and Szegedy, 2014; Liu et al., 2018a; Xie et al., 2019; Tramèr et al., 2018; Rakin, He and Fan, 2019; Sen, Ravindran and Raghunathan, 2020; Meng et al., 2021; Zhang et al., 2022) originally designed for white-box attacks. However, these methods improve the model’s robustness against white-box attacks at the cost of reducing standard accuracy (clean accuracy) (Tsipras et al., 2019; Yang et al., 2020b; Qin et al., 2021; Byun, Go and Kim, 2022). To illustrate, adversarial training (AT) is one of the most effective techniques against white-box attacks (Athalye, Carlini and Wagner, 2018; Tramer et al., 2020). However, its downsides include reduced clean accuracy, as shown in (Zhang et al., 2019; Zhang and Wang, 2019; Shafahi et al., 2019; Yang et al., 2020b; Doan et al., 2022a). The trade-off between robustness and clean accuracy thus presents a key challenge in developing effective methods capable of both high robustness and clean accuracy.

In contrast to their white-box counterparts, black-box attacks have access only to the output of deep learning models and not to gradient information. Consequently, black-box attacks interact with the model to obtain the response difference between *interactions* (*i.e.* query a single model on the input and observe the response from the model) to estimate gradients or seek search directions toward an adversarial example. To enhance the quality of gradient estimation or search directions, these black-box attacks necessitate a myriad of interactions (*i.e.* queries). This exposes the critical weakness of black-box attacks, many of which can be exploited by defenders. Recently, random noise defense (RND) has been proposed by (Qin et al., 2021) as a way of exploiting this inherent weakness and misleading black-box adversaries by introducing random noise into queried inputs during the inference phase. However, it is worth noting that adding excessive noise can reduce clean accuracy since the model becomes sensitive to noisy images (Qin et al., 2021), especially if the model has not been exposed to these noisy images in training (Cohen, Rosenfeld and Kolter, 2019). All existing defense mechanisms compromise accuracy to achieve robustness, particularly

at high distortion levels. Thus, achieving high robustness without compromising clean accuracy remains an open challenge.

This dissertation pursues a new avenue based on our insights into attack algorithms to hinder the progress of query-based black-box attacks while mitigating negative impacts on model performance. Since decision-based attacks appear to be significantly more difficult given that minimal information is exposed, we focus defending against score-based attacks. Because, the attack algorithms are significantly more query-efficient, as discussed in Chapter 5.

Our key insight is to inject uncertainty into the feedback exploited by an attacker, so as to misdirect black-box attack algorithms as they attempt to estimate gradients or seek directions toward adversarial examples. We achieve this while minimising negative impacts on model performance by considering the random selection of a model from a set of well-trained models for each prediction task request at test time. Our hypothesis is that the feedback returned from randomly selected models can introduce sufficient uncertainty to cause poor gradient estimates or randomise the search's attempts to hinder an attacker's progress. Such feedback could therefore make an adversary's attempts to compromise the model's robustness much more difficult. Notably, we expect the proposed approach to maintain high clean accuracy, since it: (a) does not inject noise into the input image at test time; and (b) relies on a set of well-trained models.

There now remains the question of how we can generate a set of well-trained models with sufficient diversity in outputs or feedback to the attacker to mislead query-based attack algorithms. It is intuitive to expect the defense to be more robust if the model outputs or feedback to sequences of model queries can be highly diverse. Consequently, we explore existing approaches to promoting diversity in model outputs in a set of well-trained models. Intuitively, we can expect that where a set of models learns different representations, this will result in model output diversity. Consequently, we explore methods that promote model diversity. In particular, we consider Bayesian deep learning methods with a theoretical basis for learning the distribution of models as a means to achieve a set of well-trained models capable of generating diverse model outputs. As such, the study in this chapter seeks to address the following research questions (RQ).

6.1.1 Chapter Overview

RQ1: How can model diversification approaches provide a robust defense against query-based black box attacks? This question will be addressed in Section 6.3.

RQ2: How robust are defense mechanisms against query-based black-box attacks? This question will be explored in Section 6.4.

Contributions. In summary, this chapter aims to develop a more effective defense mechanism against query-based black-box adversarial attacks; the main contributions are three-fold:

- We propose a conceptually simple but effective defense strategy that introduces uncertainty into models' responses for the purpose of misleading black-box adversarial attacks by randomly selecting a well-trained model that can make predictions at test time.
- We systematically study different approaches to promote diversity in model outputs to enhance uncertainty in responses generated to sequences of model queries by an attacker. We propose promoting diversity in model outputs by promoting model diversity during training whilst also achieving well-trained models based on a new learning objective.
- We conduct experiments to show that (along with the *newly proposed sample loss objective*), the Bayesian learning method, which employs an objective to push the parameters of each model apart using Stein variational gradient descent (SVGD), can encourage more diverse and well-performing models. This has the potential to increase diversity in model outputs, thereby thwarting query-based black-box attacks.

6.1.1 Chapter Overview

Section 6.2 presents a background to query-based black-box attacks and related work; Section 6.3 introduces the problem formulation and details the proposed defense algorithm; Section 6.4 evaluates the performance of different defense methods across different datasets. Section 6.5 concludes this chapter.

6.2 Related Work and Background

This section first presents existing defense methods against query-based black-box attacks. The section then discusses the ensemble approach and its robustness.

Defense against Black-Box Attacks. Although a considerable amount of research addresses the development defense mechanisms (Xie et al., 2019; Zhang and Wang, 2019; Liu et al., 2019b; Zhang et al., 2022; Wang and Wang, 2022) against white-box attacks, countermeasures aimed at black-box attacks have received less attention and have not been well studied. Recently, (Pang et al., 2020) has proposed AdvMind, an algorithm which leverages query history to detect malicious queries. Recently, (Qin et al., 2021) thoroughly investigated defense methods, dubbed “random noise defense” (RND), simply adding noise to the input. This method is well designed to disturb query-based black-box attacks and defend against such attacks with only a marginal drop in clean accuracy. However, our experimental results in Section 6.4 show that RND fails to obtain the same robustness level as our proposal. A similar approach is the regional-based classifier (RBC) (Cao and Gong, 2017), which adds noise to the input multiple times rather than once (as per the RND) and outputs an average confidence score.

Ensembles and Robustness. *Ensembles* is a widely studied approach in machine learning, the purpose of which is to construct *a set of models* and train each model in the set independently, with random initialization. In the inference phase, it combines the predictions of all models to achieve high generalization performance (Krogh and Vedelsby, 1994; Dietterich, 2000), resulting in high accuracy. (Zhang, Cheng and Hsieh, 2019) has pointed out that hijacking ensembles—*a set of models*—is more challenging than attacking a single model, as the attacker must deceive multiple models simultaneously. However, our empirical results in Section 6.4 show that launching black-box attacks against ensembles is *not hard*, and that ensembles are *slightly more robust* than their single counterparts if all the individual models of an ensemble jointly make predictions at test time.

Another line of works (Tramèr et al., 2018; Zhang, Cheng and Hsieh, 2019; Sen, Ravindran and Raghunathan, 2020; Zhang et al., 2022; Wang and Wang, 2022) has incorporated adversarial training with ensembles to investigate the resilience of ensemble adversarial training against white-box adversarial attacks. Although these proposed approaches have shown promising results in improving the ensemble’s robustness, they forfeit clean accuracy.

6.3 Proposed Method

In this section, we are interested in exploring and developing a defense method against black-box attacks based on a set of models. We hypothesize that:

- **Hypothesis 1.** A method of achieving uncertainty in the output of a set of models to a sequence of queried inputs at test-time through different functions (learned models) can lead to sufficient randomness in predictions and misguide a score-based black-box adversary. We expect the resulting uncertainty to complicate the task of estimating gradient direction or determining search directions towards an adversarial example for a black-box attacker.
- **Hypothesis 2.** Model parameters sampled from the posterior distribution obtained using the Bayesian formulation of deep learning methods can lead to diverse function representations. These functions, individually or in combination—while achieving high performance—can reduce the information available to a black-box attacker at test time with minimal to no compromise in model performance.

We will investigate the first hypothesis in Section 6.3.1 and the second in Section 6.3.2.

6.3.1 Achieving Model Output Uncertainty for Black-Box Attackers Through Randomness

To examine the *first* hypothesis, we recall that black-box attack algorithms need several interactions with a model (*i.e.* events in which the algorithms query a model on the input and observe the response from a model) to estimate gradients or search direction. If a defense mechanism can harden the gradient estimation or direction search, it is capable of decreasing the attack efficiency and enhancing the robustness of deep learning models. In this regard, an intuitive approach is to mislead attackers by injecting randomness into responses from models.

Conceptually simple but effective strategy for injecting randomness is to randomly select a well-trained model from a set of models for the purpose of servicing requests at test time. We can expect feedback from randomly selected models to misdirect gradient and search direction estimation algorithms. Now, the probability of finding

the sequence of gradients from randomly selected models leading to a strong attack are significantly smaller than that from a single model. Concretely, we can estimate it as $1/K^T$ where K denotes the number of individual models in a model set and T denotes the average number of interactions required to generate an adversarial perturbation with a single model. Formally, in this approach, the prediction at the inference (test) time is formulated as follows:

$$y^* = \arg \max_y p(y | \mathbf{x}), \tag{6.1}$$

$$p(y | \mathbf{x}) := \text{softmax}(f_k(\mathbf{x}; \boldsymbol{\theta}_k)), \quad f_k \sim \mathcal{F} \tag{6.2}$$

where $\boldsymbol{\theta}_k$ represents weights, $\mathcal{F} = \{f_1(\cdot, \boldsymbol{\theta}_1), f_2(\cdot, \boldsymbol{\theta}_2), \dots, f_K(\cdot, \boldsymbol{\theta}_K)\}$ denotes a set of models. However, it may be impracticable to train a large set of models; further, predicting with an ensemble is shown to lead to higher prediction accuracy (Krogh and Vedelsby, 1994; Dietterich, 2000). Consequently, rather than selecting one model, we uniformly select a subset of models at random with replacements. Naturally, this leads to an increase in the effective number of models presented to the attacker from the combination of models composed to make predictions. Formally, the prediction of a subset of models is formulated as follows:

$$p(y | \mathbf{x}) := \text{softmax}\left[\frac{1}{m} \sum_{k=1}^m f_k(\mathbf{x}; \boldsymbol{\theta}_k)\right], \quad f_k \sim \mathcal{F} \tag{6.3}$$

where m denotes the number of selected models of a subset. Overall, through randomness in model selection, the approach introduced in this section injects uncertainty into the model outputs extracted by a black-box attacker to evade gradient estimation and direction search methods. But, to enhance adversarial robustness, we aim to increase the uncertainty in exposed model outputs. Because we can expect the defense to be more robust if the model outputs or feedback to sequences of model queries is highly diverse or variable. We propose promoting model parameter diversity to promote model output variance with minimal to no impact on performance. Because, intuitively, we can expect a set of models learning different representations for a machine learning task to result in model output diversity or variance. With the goal of model diversity in mind, the next section investigates our second hypothesis by considering Bayesian learning approaches with a theoretical basis for learning the distribution of model parameters. Then, to achieve model output variance in a set of well-trained models, we propose a new learning objective in the Bayesian learning context. Subsequently, we explore alternative methods of promoting

6.3.2 Proposed Method for Achieving Model Diversity

model diversity in Section 6.3.3, comparing these with our proposed method in an extensive series of experiments in Section 6.4.

6.3.2 Proposed Method for Achieving Model Diversity

To explore the *second* hypothesis, we are interested in a framework for promoting model diversity. In general, we can train an ensemble of models in parallel such that their predictions are consistent while their parameters are diverse. Formally, the training objective of such a framework can be expressed as follows:

$$\min_{\Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\ell \left(\frac{1}{K} \sum_{k=1}^K f_k(x; \theta_k), y \right) \right], \quad \text{s.t. } \Delta(\mathcal{F}) \quad (6.4)$$

where \mathcal{D} denotes a training set, Δ is a set of constraints on the set of functions $\mathcal{F} = \{f_1(\cdot, \theta_1), f_2(\cdot, \theta_2), \dots, f_K(\cdot, \theta_K)\}$ to ensure diversity optimized over their parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$ and $\ell(\cdot, \cdot)$ is the loss (*i.e.* cross-entropy).

There are two questions that have to be answered in the formulation of (6.4): (1) what constraints best encourage the diversity of models (*i.e.* best Δ) leading to output variance; and (2) since we minimise the loss over the average logits of the set of functions to train these models, how can we ensure that the asymmetry between models promotes high average and individual model performance?

Parameter Diversity Approach

To promote model diversity, we consider adopting a training framework incorporating a Bayesian formulation of deep learning methods with Stein Variational Gradient Descent (SVGD) method (Liu and Wang, 2016; Wang and Liu, 2019) we refer to simply as the *Bayesian training framework*. The Bayesian training framework allows us to learn a posterior distribution of parameters and the model parameters sampled from that posterior distribution can result in diverse function representations, leading to model diversity. As the SVGD method is able to repulse models' parameters directly, it is capable of encouraging learning diversified parameters and provides an effective solution to question (1). Interestingly, this approach enables learning different representations (Doan et al., 2022a) and, consequently, leads to output variance without compromising clean accuracy. Formally, the learning diversity parameter

algorithm is formulated as follows:

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_k - \epsilon \phi^*(\boldsymbol{\theta}_k) \quad (6.5)$$

$$\phi^*(\boldsymbol{\theta}_k) = \sum_{k=1}^K \left[\kappa(\boldsymbol{\theta}_k, \boldsymbol{\theta}_i) \nabla_{\boldsymbol{\theta}_i} \ell(f_i(\mathbf{x}; \boldsymbol{\theta}_i), y) - \gamma \nabla_{\boldsymbol{\theta}_i} \kappa(\boldsymbol{\theta}_k, \boldsymbol{\theta}_i) \right], \quad (6.6)$$

where $\boldsymbol{\theta}_k$ denotes the weights of the k -th model, $\kappa(\cdot, \cdot)$ is a kernel function that encourages model diversity, and γ is a hyperparameter to control the trade-off between models' diversity and the loss $\ell(\cdot, \cdot)$ (*i.e.* cross-entropy).

Notably, while the SVGD method was employed for the purposes of adversarial defense in (Doan et al., 2022a, 2023), incorporating adversarial training and information gain, we do not adopt both of the adversarial training approaches due to the resulting clean accuracy drop. Additionally, the method proposed by (Doan et al., 2022a, 2023) does not consider the problem posed in question (2) and we propose a new formulation for the problem as discussed below.

New Training Objective

We observe that the training objective in Equation 6.4 is not able to satisfy question (2) as shown in Section 6.4.4. Simply, a naive adoption of the Bayesian training framework with SVGD does not yield models that perform well individually, despite the fact that the average performance of all models for a task is high. To address this problem, we propose a new training objective that encourages individual model learning.

To encourage individual model learning while training a set of models, we propose the incorporation of a *sample loss* training objective, $\ell(f_i(\mathbf{x}; \boldsymbol{\theta}_i), y)$, to formulate a new joint loss as follows:

$$\min_{\Theta} \mathbb{E}_{\mathcal{B} \sim \mathcal{D}, \boldsymbol{\theta}_i \sim \Theta} \left[\mathbb{E}_{(x,y) \sim \mathcal{B}} \left[\ell \left(\frac{1}{K} \sum_{k=1}^K f_k(\mathbf{x}; \boldsymbol{\theta}_k), y \right) + \ell \left(f_i(\mathbf{x}; \boldsymbol{\theta}_i), y \right) \right] \right], \quad (6.7)$$

where \mathcal{B} denotes a batch of data sampled from a training set \mathcal{D} . Notably, in this training framework, we aim to train all models simultaneously, and for each batch of data \mathcal{B} , we uniformly select $\boldsymbol{\theta}_i$ from Θ at random with replacement.

6.3.3 Alternative Approaches to Promote Model Diversity

We are interested in exploring alternative approaches which can improve the robustness of the model set by encouraging its diversity. Findings from

6.3.3 Alternative Approaches to Promote Model Diversity

(Lakshminarayanan, Pritzel and Blundell, 2017; Fort, Hu and Lakshminarayanan, 2020; Wen, Tran and Ba, 2020) show that employing random initializations and independent training strategies for ensembles can decorrelate networks' predictions and induce diversity. Thus, we will study and use ensembles (Lakshminarayanan, Pritzel and Blundell, 2017) as a baseline. Additionally, recent research has proposed two approaches—*gradient-based* (Teney et al., 2022) and *score-based* (Lee, Yao and Finn, 2023)—to promote model diversity. We will therefore investigate these two approaches and compare them with SVGD.

Ensembles employing Random Initialization Approaches. (Lakshminarayanan, Pritzel and Blundell, 2017) proposed to train a set of models—*Ensemble*—with random initializations independently. This training procedure can be formulated as follows:

$$\min_{\theta_k} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\ell(f_k(\mathbf{x}; \theta_k), y) \right], \quad (6.8)$$

where θ_i denotes the weights of the i -th model, and $\ell(\cdot, \cdot)$ is the loss (*i.e.* cross-entropy).

Gradient-based Approach. (Teney et al., 2022) introduced a method encouraging diversity over a set of models by quantifying the similarity of the gradient of the top predicted score of each model with respect to its features. This method aims to train a set of models to discover predictive patterns commonly missed by learning algorithms and promote diversity across the model set. Note that while their problem is to improve out-of-distribution robustness, our problem is to enhance adversarial robustness. In this study, we adopt their *Diversity Regularizer* (DivReg) to encourage the diversity of models and formulate the training objective as follows:

$$\min_{\Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sum_{k=1}^K \ell(f_k(\mathbf{x}; \theta_k), y) + \lambda_{\text{reg}} \sum_{i \neq j} \delta_{f_i, f_j} \right], \quad (6.9)$$

where $\delta_{f_i, f_j} = \nabla_{\mathbf{h}} f_i(\mathbf{h}_i) \cdot \nabla_{\mathbf{h}} f_j(\mathbf{h}_j)$, λ_{reg} controls the strength of the regularizer, $\nabla_{\mathbf{h}} f$ and $\nabla_{\mathbf{h}} f_j$ denote the gradient of the top predicted score of models f_i and f_j with respect to their own features \mathbf{h}_i and \mathbf{h}_j .

Score-based Approach. (Lee, Yao and Finn, 2023) proposed an approach to training a collection of diverse models by independently training each head pair to make predictions. We note that their method aims to enhance models' robustness in order to the shift between source and target data distribution whereas our problem improves models' robustness against adversarial attacks. In this study, we adopt their loss

function to encourage model diversity. The training objective is formulated as follows:

$$\min_{\Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sum_{k=1}^K \ell(f_k(\mathbf{x}; \boldsymbol{\theta}_k), y) + \lambda_{\text{MI}} \sum_{k \neq i} \mathcal{L}_{\text{MI}}(f_k(\mathbf{x}; \boldsymbol{\theta}_k), f_i(\mathbf{x}; \boldsymbol{\theta}_i)) \right], \quad (6.10)$$

$$\mathcal{L}_{\text{MI}}(f(\mathbf{x}; \boldsymbol{\theta}_k), f(\mathbf{x}; \boldsymbol{\theta}_i)) = D_{\text{KL}}(p(\hat{y}_k, \hat{y}_i) \parallel p(\hat{y}_k) \otimes p(\hat{y}_i)), \quad (6.11)$$

where $D_{\text{KL}}(\cdot \parallel \cdot)$ is the KL divergence and \hat{y}_i is the prediction $f_i(\mathbf{x}; \boldsymbol{\theta}_i)$, λ_{MI} controls the strength of mutual information loss \mathcal{L}_{MI} , $p(\hat{y}_k, \hat{y}_i)$ is the empirical estimate of the joint distribution and $p(\hat{y}_k)$, $p(\hat{y}_i)$ are the empirical estimates of the marginal distributions.

6.4 Experiments and Evaluations

This section evaluates the robustness and effectiveness of the so-called *randomness defenses* which inject randomness into the response of a deep learning model. These randomness defense methods. These defense mechanisms are evaluated on three standard vision tasks MNIST (Lecun et al., 1998), CIFAR10 (Krizhevsky, Nair and Hinton, n.d.), STL-10 (Coates, Lee and Ng, 2011).

Datasets, Network Architectures and Attack. In this study, we use three different standard datasets MNIST (Lecun et al., 1998), CIFAR-10 (Krizhevsky, Nair and Hinton, n.d.) and STL-10 (Coates, Lee and Ng, 2011). We use a network architecture from (Cheng et al., 2020) for MNIST, VGG-16 (Liu and Deng, 2015) for the CIFAR-10 task and ResNet18 (He et al., 2016) for the STL-10 task. The clean accuracy of these network architectures and defended models is presented in Appendix D.2. To evaluate the robustness of defense mechanism, we employ SQUAREATTACK (Andriushchenko et al., 2020) which is the state-of-the-art attack method in score-based settings and more effective than decision-based attacks as discussed in Chapter 5.

Defense Mechanisms. We compare our proposed method with RND (Qin et al., 2021), RBC (Cao and Gong, 2017), the dropout approach (Gal and Ghahramani, 2016; Srivastava et al., 2014), ENSEMBLES (Lakshminarayanan, Pritzel and Blundell, 2017), diversity regularizer (DIVREG(Adapted)) (Teney et al., 2022) and diversity disambiguity (DIVDIS(Adapted)) (Lee, Yao and Finn, 2023). Moreover, we demonstrate the robustness of a single model (no defense) and ensembles (no defense). In the case of ensembles (no defense), we simply conduct black-box attacks against ensembles when all individual models make predictions together at test time.

Evaluation Metrics. We note that employing randomness defense mechanisms causes a model to generate variant outputs (different confidence scores) for a benign or

6.4.1 Experimental Regime

adversarial input. This input can be correctly or wrongly predicted when fed into a defended model adopting randomness several times. Therefore, when an adversarial input yielded by an attack aims to fool a model defended by a randomness defense method, it could fail or succeed. The more frequently it fails, the more robust the randomness defense is. To this end, we define the robustness of a randomness defense method as follows:

$$\text{Robustness} = \mathbb{E}_{x_{\text{adv}} \sim \mathcal{D}_{\text{ADV}}} [\text{Acc}_r(x_{\text{adv}})], \quad (6.12)$$

where $\text{Acc}_r(x_{\text{adv}}) = \frac{n}{N}$, N represents the number of inferences of an adversarial example and n is the number of correct predictions of the adversarial example x_{adv} . \mathcal{D}_{ADV} is a set of adversarial examples generated by an attack algorithm.

Evaluation Protocol. Recall that when a benign input is fed into a defended model that incorporates randomness into its response, it can be correctly or incorrectly predicted or classified. The more frequently a benign input is misclassified by a defended model, the less reliable that benign input will be for the purpose of constructing an attack. For a fair and reliable comparison, we select benign inputs correctly inferred 1,000 times, dubbed *reliable benign inputs*. To reduce the computational burden of the evaluation tasks on four different datasets, we compose each evaluation set out of 500 *reliable benign inputs*. We employ SQUAREATTACK (Andriushchenko et al., 2020) with a 10,000 query budget to generate adversarial examples from *reliable benign inputs* for the purpose of evaluating the performance of each defense method.

6.4.1 Experimental Regime

This section summarizes the extensive experiments conducted on the MNIST, CIFAR-10 and STL-10 datasets and against SQUAREATTACK.

- *Effectiveness of the Proposed Method Against Black-box Attacks.* Section 6.4.2 evaluates the robustness of different defense mechanisms (prior and proposed methods using the new learning objective as well as other model diversification approaches) against SQUAREATTACK (a score-based attack algorithm) across different datasets.
- *Diversity Analysis.* Section 6.4.3 analyzes the degree of model diversity that is encouraged by different methods.

- *Effectiveness of New Joint Loss.* Section 6.4.4 demonstrates the effectiveness of the new training objective in encouraging individual model learning while promoting model diversity.

6.4.2 Robustness to Black-box Attacks

As discussed in Section 6.3.1 we can expect a more extensive set of models to yield a lower chance of estimating gradient or searching a proper attack direction as well as possibly obtaining better accuracy. However, due to the extensive computational resources required and the complexity of different datasets (*i.e.* high dimension data), we train a larger number of models for low-resolution data and a lower number of models for high-resolution datasets. In this section, we carry out:

- Extensive training and evaluation on a set of 40 models with MNIST and
- Training and evaluation of a set of 10 models with CIFAR-10 and STL-10.

Furthermore, to reduce the computational burden of attack execution times with large sample sets across multiple model diversification methods, we:

- Conduct extensive evaluations with MNIST and CIFAR-10 and
- Show *generalisation* of the results with a select model evaluation set using STL-10.

Evaluation on MNIST

In this section, we carry out extensive experiments to demonstrate the robustness of different defense methods on an evaluation set selected from MNIST. For our proposed approach, we train a set of 40 models using ENSEMBLES, DIVDIS(Adapted), DIVREG(Adapted) and our proposed method and select the best model set for each method based on test accuracy. To evaluate and compare robustness, we choose different settings with different sizes of model subsets (*i.e.* 1, 3, 5, 20 or 30). The results in Table 6.1 show that our proposed method is more robust than other diversity promotion methods across different distortion levels and settings. More results relating to training a set of 10 and 20 models are presented in Appendix D.1.

Additionally, we compare the performance of defense methods designed for black-box attacks and our simple approach employing random model selection with various

6.4.2 Robustness to Black-box Attacks

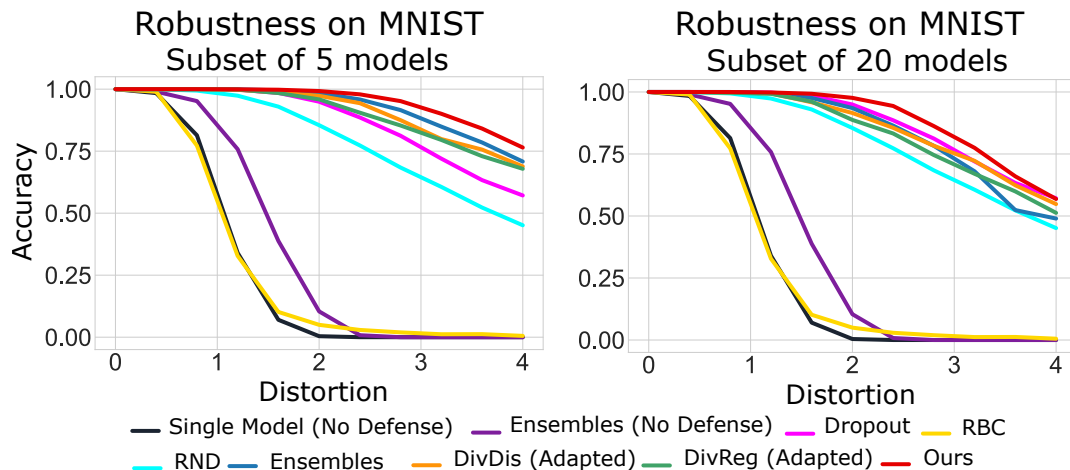


Figure 6.1. A robustness comparison (higher \uparrow is stronger) between our proposed method and other methods against `SQUAREATTACK` on MNIST. For different diversity promotion methods, we train a set of 40 models and randomly select a subset of 5 and 20 models. Defense robustness is measured by the average accuracy of models over an evaluation set under attacks at different distortion levels. Notably, a subset of one or three models is more robust than a subset of five models, which is more robust than other defense methods.

diversity promotion strategies. In this comparison, we choose a setting for a set of 40 models and randomly select a subset of three out of 40 models. The same result and performance can be illustrated with other settings. The results in Figure 6.1 demonstrate that our simple approach employing random model selection outweighs other baselines and state-of-the-art defense methods designed for black-box attacks.

Evaluation on CIFAR-10

In this section, we conduct extensive experiments to demonstrate the robustness of different defense methods on an evaluation set selected from CIFAR-10. Due to limitations in training resources, we train a set of 10 models using `ENSEMBLES`, `DIVDIS(Adapted)`, `DIVREG(Adapted)` and our proposed method and select the best model set for each method based on test accuracy. For robustness evaluation and comparison, we choose different settings with different sizes of model subsets (*i.e.* 1, 3, 5 and 8). The results in Table 6.2 show that our proposed method is more robust than other diversity promotion methods across different distortions and settings.

Additionally, we compare the performance of defense methods designed for black-box attacks and our simple approach employing random model selection with different diversity promotion strategies. In this comparison, we choose a setting for a set of

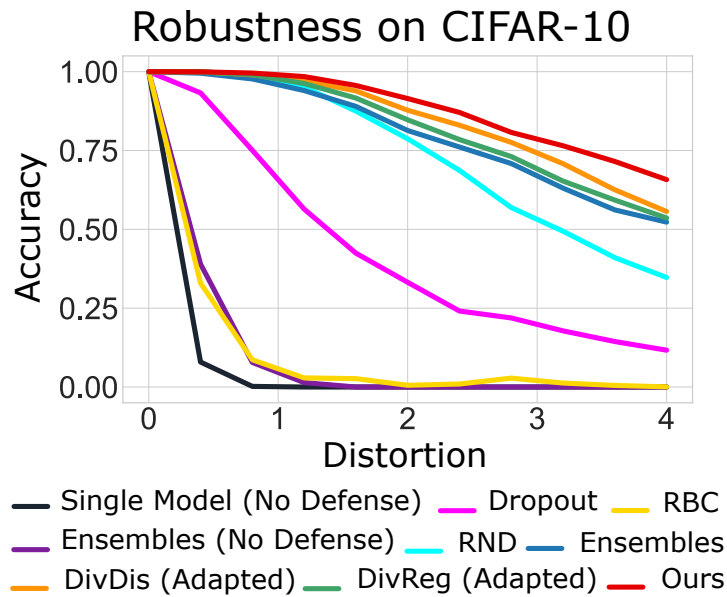


Figure 6.2. A comparison of robustness (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on CIFAR-10. For different diversity promotion methods, we train a set of 10 models and randomly select a subset of five models. The defense robustness is measured by the average accuracy of models over an evaluation set under attacks at different distortion levels.

10 models and randomly select a subset of five models. Likewise, the same result and performance can be illustrated with other settings. The results in Figure 6.2 demonstrate that our simple approach employing random model selection outweighs other baselines and state-of-the-art defense methods designed for black-box attacks.

Generalisation with Evaluation on STL-10

In this section, we conduct extensive experiments to demonstrate the robustness of different defense methods on an evaluation set selected from STL-10. Due to limitations in training resources, we train a set of 10 models using ENSEMBLES, DIVDIS(Adapted), DIVREG(Adapted) and our proposed method and select the best model set for each method based on test accuracy. To evaluate and compare robustness, we choose a subset of five model subsets. The results in Table 6.3 show that our proposed method is more robust than other diversity promotion methods across different distortions and settings.

Additionally, we compare the performance of methods designed to defend against black-box attacks and our straightforward approach employing random model

6.4.3 Diversity Analysis

selection with different diversity promotion strategies. In this comparison, we choose a setting for a set of 10 models and randomly select a subset of five models. The same result and performance can also be illustrated with other settings. The results in Figure 6.3 demonstrate that our simple approach employing random model selection outweighs other baselines and state-of-the-art defense methods designed for black-box attacks.

6.4.3 Diversity Analysis

As presented in Section 6.3.2, more diversity among individual models results in more output uncertainty for black-box attackers. Therefore, in this section, we use Jensen–Shannon divergence as a metric to show model diversity promoted by different methods. We measure model diversity on CIFAR-10 and STL-10 by calculating the Jensen–Shannon divergence between the average softmax outputs of all models versus the softmax output of each particle. We compute it over the testset of CIFAR-10 and STL-10. The result in Figure 6.4 shows that our proposed method is able to achieve

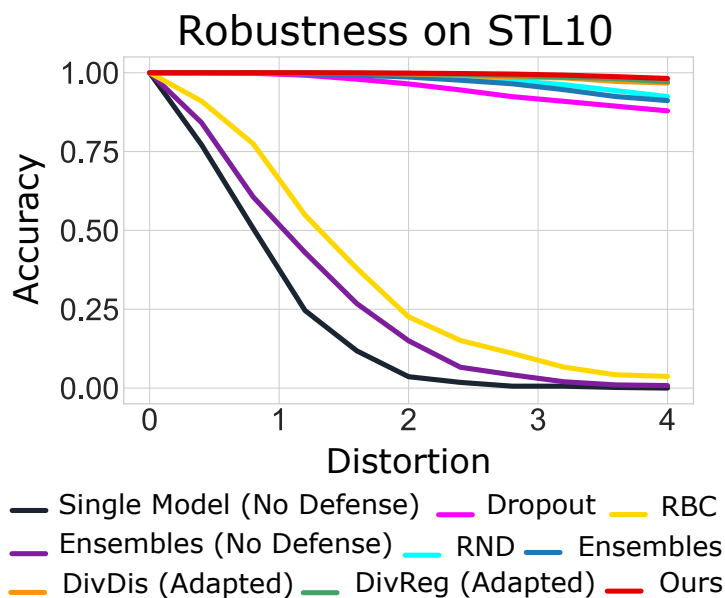


Figure 6.3. A comparison of robustness (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on STL-10. For different diversity promotion methods, we train a set of 10 models and randomly select a subset of five models. Defense robustness is defined as the average accuracy of models over an evaluation set under attacks at different distortion levels.

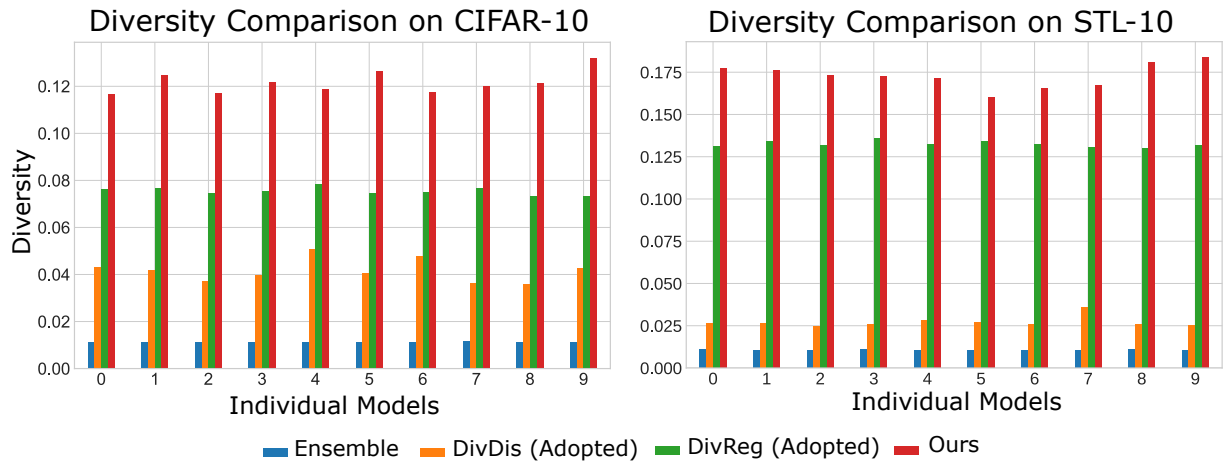


Figure 6.4. A model diversity comparison among ENSEMBLES, DIVDIS(Adapted), DIVREG(Adapted) and our proposed method using Jensen–Shannon divergence on CIFAR-10 and STL-10.

greater diversity among individual models, while ensembles obtain the least diversity. It is expected because ensembles do not have any mechanism to encourage diversity.

6.4.4 Effectiveness of the Proposed Learning Objective (Sample Loss)

As mentioned in Section 6.3.2, incorporating sample loss as a training objective can encourage individual learning and helps each individual model obtain high performance. These well-trained models lead to the success of our proposed approach. Therefore, in this section, we aim to show the effectiveness and insights of the new training objective with and without sample loss. We employ the SVGD method to train a set of 10 models simultaneously with and without sample loss on two datasets, MNIST and CIFAR-10. We train up to 1,000 epochs and select the best model set based on test accuracy. The results in Table 6.4 show that each individual model in a model set trained with the sample loss achieves high performance on both datasets. As a result, randomly selecting five individual models is able to obtain high accuracy (92.4%) that is slightly lower than the accuracy achieved by the set of all models (93.2%). In contrast, without the sample loss, most models exceed 50% accuracy, and the random selection of five models does not result in high accuracy (79%).

6.5 Conclusion

The study in this chapter proposes a novel defense mechanism against query-based attacks which exploit output model scores. Results from extensive experiments demonstrate that the proposed defense mechanism consistently achieves better robustness than state-of-the-art defense methods designed for query-based attacks across different datasets. Interestingly, after we had encouraged a diverse set of models employing the SVGD method and the proposed learning objective, our results significantly outperformed those of other model diversity promotion schemes in terms of achieving the twin objectives of high robustness and clean accuracy. While an extensive set of results is presented in the main chapter, additional results in support of the study can be found in Appendix D.

Up to this juncture, this dissertation has concentrated on exploring the vulnerability of deep learning models to black-box query-based attacks and investigating defense mechanisms against these attacks. The forthcoming chapter will provide a concise overview of the challenges addressed and contributions made in this dissertation towards building robust deep neural networks. The chapter will then outline promising avenues for future research that have emerged as a result of the studies presented in this dissertation.

Table 6.1: A comparison of robustness (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on MNIST. For the evaluation of different diversity promotion methods, we train a set of 40 models and randomly select a subset of a different number of models.

Random	Methods	Distortion = 0	0.8	1.6	2.4	3.2	4.0
1	ENSEMBLES	100%	99.6%	97.3%	93.4%	89.0%	80.2%
	DIVDIS(Adapted)	100%	99.6%	97.4%	93.9%	88.1%	82.6%
	DIVREG(Adapted)	100%	99.2%	96.2%	91.8%	84.3%	77.4%
	Ours	100%	99.7%	98.9%	97.2%	93.5%	88.2%
3	ENSEMBLES	100%	100%	99.4%	94.2%	85.2%	74.6%
	DIVDIS(Adapted)	100%	100%	98.6%	93.8%	83.7%	73.1%
	DIVREG(Adapted)	100%	100%	99.0%	93.0%	79.7%	67.6%
	Ours	100%	100%	99.8%	98.0%	91.4%	77.8%
5	ENSEMBLES	100%	100%	99.5%	95.8%	84.9%	70.8%
	DIVDIS(Adapted)	100%	100%	98.6%	94.3%	79.9%	68.9%
	DIVREG(Adapted)	100%	100%	98.4%	90.5%	79.5%	67.9%
	Ours	100%	100%	99.8%	97.9%	90.1%	76.5%
20	ENSEMBLES	100%	100%	97.6%	86.4%	68.0%	49.0%
	DIVDIS(Adapted)	100%	99.8%	95.9%	85.5%	72.2%	54.8%
	DIVREG(Adapted)	100%	99.7%	96.1%	83.3%	67.0%	51.3%
	Ours	100%	100%	99.3%	94.4%	77.5%	56.8%
30	ENSEMBLES	100%	99.9%	96.8%	81.2%	60.6%	40.0%
	DIVDIS(Adapted)	100%	99.9%	95.9%	80.0%	64.9%	46.9%
	DIVREG(Adapted)	100%	99.5%	93.9%	77.3%	59.7%	43.2%
	Ours	100%	100%	98.6%	91.9%	70.4%	52.2%

6.5 Conclusion

Table 6.2: A robustness comparison (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on CIFAR-10. For the evaluation of different diversity promotion methods, we train a set of 10 models and randomly select a subset of a different number of models.

Random	Methods	Distortion = 0	0.8	1.6	2.4	3.2	4.0
1	ENSEMBLES	100%	90.0%	83.6%	75.4%	64.2%	55.1%
	DIVDIS(Adapted)	100%	95.1%	90.1%	82.6%	72.0%	59.1%
	DIVREG(Adapted)	100%	90.6%	86.2%	79.5%	69.6%	59.5%
	Ours	100%	90.2%	86.9%	82.2%	75.2%	67.6%
3	ENSEMBLES	100%	97.1%	88.3%	78.6%	67.4%	55.4%
	DIVDIS(Adapted)	100%	99.2%	96.1%	86.2%	75.83%	62.1%
	DIVREG(Adapted)	100%	99.6%	93.5%	84.3%	72.1%	60.3%
	Ours	100%	99.8%	96.7%	90.0%	82.2%	72.6%
5	ENSEMBLES	100%	97.7%	89.0%	76.1%	63.1%	52.3%
	DIVDIS(Adapted)	100%	99.0%	93.9%	83.0%	70.8%	55.7%
	DIVREG(Adapted)	100%	99.0%	91.6%	78.5%	65.3%	53.6%
	Ours	100%	99.6%	95.6%	87.1%	76.5%	65.8%
8	ENSEMBLES	100%	98.2%	87.9%	76.9%	63.5%	52.2%
	DIVDIS(Adapted)	100%	99.1%	94.1%	82.7%	70.4%	56.4%
	DIVREG(Adapted)	100%	99.2%	90.9%	76.3%	64.6%	51.6%
	Ours	100%	99.7%	96.0%	86.2%	76.2%	66.0%

Table 6.3: A comparison of robustness (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on STL-10. For the evaluation of different diversity promotion methods, we train a set of 10 models and randomly select a subset of five models.

Methods	Distortion = 0	1.2	2.4	3.6	4.8	6.0
ENSEMBLES	100%	99.6%	97.6%	92.5%	89.3%	82.9%
DIVDIS(Adapted)	100%	99.9%	99.2%	97.3%	94.9%	91.1%
DIVREG(Adapted)	100%	100%	99.5%	98.2%	95.0%	91.3%
Ours	100%	100%	99.7%	98.8%	96.6%	92.8%

Table 6.4: Clean accuracy of a set of 10 models trained simultaneously with and without sample loss on MNIST and CIFAR-10.

Dataset Training Objective	MNIST		CIFAR-10	
	Without Sample Loss	With Sample Loss	Without Sample Loss	With Sample Loss
All Models	99.6%	99.7%	89.8%	93.2%
Random 8 Models	87.3%	99.6%	59.7%	92.8%
Random 5 Models	79.4%	99.6%	39.8%	92.4%
Random 3 Models	69.9%	99.6%	31.0%	91.4%
Model 1	50.7%	99.3%	15.1%	88.5%
Model 2	36.6%	99.5%	13.8%	88.3%
Model 3	22.8%	99.3%	13.5%	88.5%
Model 4	42.2%	99.2%	10.0%	88.1%
Model 5	32.7%	99.5%	9.3%	88.9%
Model 6	35.4%	99.4%	12.4%	86.9%
Model 7	35.6%	99.4%	11.3%	88.4%
Model 8	32.0%	99.4%	12.4%	89.7%
Model 9	55.6%	99.3%	10.2%	87.7%
Model 10	99.3%	99.3%	80.8%	88.4%

Chapter 7

Conclusion

THIS chapter concludes the dissertation and suggests directions for future work.

7.1 Thesis Overview and Summary

This dissertation undertook an in-depth investigation into emerging research on the vulnerability of deep learning models to black-box attacks, ranging from the *Score-based* to *Decision-based* scenarios. Through investigations into these problems, the dissertation advances knowledge in the field and paves the way for more secure and resilient real-world systems and applications employing such models.

To gain insights into the susceptibility of deep learning models, this dissertation first introduced three novel attack methods in score-based and decision-based settings. These methods exhibit state-of-the-art attack success rates against models built from the two widely used deep neural network (DNN) architectures—*Convolutional Networks* and *Transformers*. A comprehensive examination of these attack methods was presented in Chapter 3, 4 and 5. In summary:

- The study in Chapter 3 first revealed the presence of *hard* cases and hypothesized that these *hard* cases arise due to entrapment in local minima when gradient estimation fails to guide the attack. To overcome the entrapment problem, the study devised a novel search method—RAMBOATTACK—drawing inspiration from randomized block coordinate descent (BLOCKDESCENT). Unlike existing dense attacks, RAMBOATTACK focuses on altering local regions of the input aligned with the filter sizes used in DNNs to generate low-distortion adversarial examples in *hard* cases. This mechanism aims to exploit the model’s dependence on salient features of the target class for classification to discover potential adversarial perturbations. Lastly, the study employed a visual explanation tool to provide a representation of the connections between the introduced perturbations and salient regions in images associated with the target class.
- While the study in Chapter 3 shed light on the vulnerability of deep neural networks to dense attacks, the study in Chapter 4 further explored a new threat—sparse attacks in decision-based scenarios—that revealed new weaknesses in machine learning models. However, conducting sparse attacks is non-trivial due to the NP-hard problem, and it is more challenging in decision-based settings. To remedy this problem, the study in this chapter introduced a novel approach—SPARSEEVO—an evolution-based algorithm. Furthermore, for the first time, this study explored the susceptibility of the transformer to decision-based sparse attacks on the standard computer vision

task ImageNet and established its relative robustness to convolution-based models. Importantly, extensive experiments in this study demonstrated that SPARSEEVO surpasses the performance of a state-of-the-art sparse attack and achieves comparable attack success rates to the leading *white-box* attack—PGD₀—with limited query budgets.

- Similar to Chapter 4, Chapter 5 undertook an extensive investigation into the vulnerability of deep learning models to sparse attacks. However, the sparse attacks examined in this chapter require output score information rather than the model’s decision. Even where score information is available, sparse attacks still come up against the NP-hard problem. To mitigate this challenging problem, the research in Chapter 5 introduces a novel sparse attack method—BRUSLEATTACK. This method aims to learn influential pixel characteristics from historical information on pixel manipulations. It then integrates a pixel selection mechanism based on the dissimilarity of pixels between a search space prior and a source image. Extensive and comprehensive experiments in the chapter demonstrated that BRUSLEATTACK outperforms state-of-the-art methods in terms of both attack success rate and sparsity across various datasets (*i.e.* CIFAR-10, STL10, ImageNet), deep learning models (*i.e.* Convolutional-based, Transformer-based), and defense mechanisms (*i.e.* Adversarial Training and RND) within limited query budgets. More interestingly, the research exhibited successful sparse attacks against a real-world system—Google Cloud Vision.

To pave the way for more secure deep learning models employable in real-world systems and applications under black-box scenarios, this dissertation explored a wide range of approaches to mechanisms for defense against black-box attacks. Nonetheless, these approaches face the challenge of obtaining two objectives: a marginal drop in clean accuracy and high robustness. To address these challenge, this dissertation proposed a novel defense method. The approach was examined through comprehensive experiments in Chapter 6. In summary:

- The study in Chapter 6 devised a novel countermeasure that introduces uncertainty into model outputs by randomly selecting a subset of well-trained models when making predictions. This uncertainty allows the proposed method to deceive black-box attacks that rely on gradient estimation or random search

7.2 Future Work

methods exploiting model outputs. Moreover, this study examines various approaches to encourage a diverse set of models, thereby increasing the variance in model outputs and further enhancing defense capability. Through extensive experiments, the study showed that a Bayesian learning approach utilizing Stein variational gradient descent (SVGD) alongside a novel sample loss objective is capable of encouraging greater diversity in a model set regarding its respective outputs. Overall, the rigorous empirical study in this chapter demonstrated that incorporating randomness and promoting model diversity significantly impedes the progress of black-box attacks and strengthens the resilience of the models against these attacks.

7.2 Future Work

From a security perspective, black-box attacks always pose critical threats to the safety of applications and systems employing deep learning models. This dissertation has mainly focused on black-box attacks against machine learning as a service that performs image classification tasks in the digital and vision domains. However, these black-box attacks may endanger deep learning models in video learning tasks (*i.e.* video classification, recognition or object detection) and in the physical, text, audio or video domains. However, deceiving deep learning models employed in the physical domain or in video learning tasks is even more challenging due to several environmental and hardware factors. This results in greater complexity in input manipulation, as well as a higher need for computing resources. In regard to this, understanding both the problems and challenges of black-box attacks in the digital domain presented in this dissertation will reveal some insights and possibilities to deal with these challenges. On the other hand, from a defense standpoint, withstanding black-box attacks in speech or text domains is still challenging due to domain differences. To this end, this section will discuss numerous opportunities and possibilities that merit future studies on new black-box threats and countermeasures to enhance further the resilience and robustness of deep learning models in various learning tasks.

- **How can we realise black-box attacks developed in Chapter 4 and Chapter 5 against deep learning models for video learning tasks?** While exploring the safety and reliability of models used in the context of classification as a service

deployed in real-world systems, video classification services are increasingly offered by providers such as Google Cloud Intelligence⁸ or Amazon Rekognition Video⁹. While a large body of work has been conducted examining the vulnerability of deep learning models used for the image classification task, there is a handful of research (Wei et al., 2020; Li et al., 2021b; Zhan et al., 2023) about the robustness of deep learning models for video learning tasks (*i.e.* video classification (Diba et al., 2018) or recognition tasks (Feichtenhofer et al., 2019)) against black-box attacks. However, these attack algorithms mainly focus on dense settings. Therefore, safety and security concerns relating to the employment of models for video learning tasks in real-world systems under sparse black-box scenarios are less well-studied. To this end, it is crucial for future research to explore sparse black-box attacks against deep learning models trained for video learning tasks.

- **How can we extend sparse attacks to the physical domain?** In the physical domain, a (physical) adversarial example that is generated, printed and captured by a camera can fool a real-world system employing a deep learning model that requires interactions with the physical world (*i.e.* self-driving cars or drones) (Jan et al., 2019; Wang et al., 2022). A handful of works have studied physical adversarial attacks that generate adversarial examples (*i.e.* sticker, patch, laser or projector light) (Eykholt et al., 2018; Thys, Ranst and Goedeme, 2019; Liu et al., 2019a; Lovisotto et al., 2020; Yang et al., 2020a; Nguyen et al., 2020; Jia et al., 2022; Chen et al., 2022b; Huang and Ling, 2022; Doan et al., 2022b). These visible adversarial examples are possibly detected by defense systems adopting abnormal input detection methods (Xiang and Mittal, 2021). To evade the detection method employed by defense systems, a generated perturbation needs to be less visible or sparse—quantified by l_0 constraint. With regard to this, the process of deceiving a deep learning model in the physical domain is less well-studied. Therefore, future research endeavors should focus on developing techniques to yield sparse perturbation in the physical domain.
- **How can we apply the defense concept of uncertainty and model diversity introduced in Chapter 6 in other domains?** The defense method introduced in Chapter 6 mainly focuses on black-box attacks in the vision domain. However,

⁸<https://cloud.google.com/video-intelligence/docs/streaming/video-classification>

⁹<https://aws.amazon.com/rekognition/video-features/>

black-box attacks have been developed in other domains such as text (Lee et al., 2022) and audio (Taori et al., 2019). To withstand these black-box attacks, it would be insufficient to naively adopting the defense method introduced in Chapter 6 across domains. Therefore, the adoption of the proposed concept of uncertainty and model diversity to defend against black-box attacks across domains represents a promising research direction.

- **How can we improve the training efficiency of SVGD?** Although training a robust model using the SVGD method can achieve state-of-the-art robustness against black-box attacks, as shown in Chapter 6, this method is time-consuming and computationally intensive. The robust training method based on SVGD can take several weeks to complete, particularly for a large model set and for large-scale and high-resolution datasets (*i.e.* ImageNet). As a consequence, it hinders the employment of the SVGD method in real-world applications that require the training of large-scale and high-resolution datasets or model owners who have limited computation resources. Future research should thus address these concerns by adopting pre-trained models and the transfer learning method. This approach can allow SVGD to push model weight apart at some last layer so as to achieve diversity. Exploring this promising research direction will ideally serve to reduce training time and computation resources, making training with the SVGD method more scalable and employable to all model owners.

Appendix A

Chapter 3 Appendix

A.1 Targeted Attacks on Balanced and Non-hard Sets

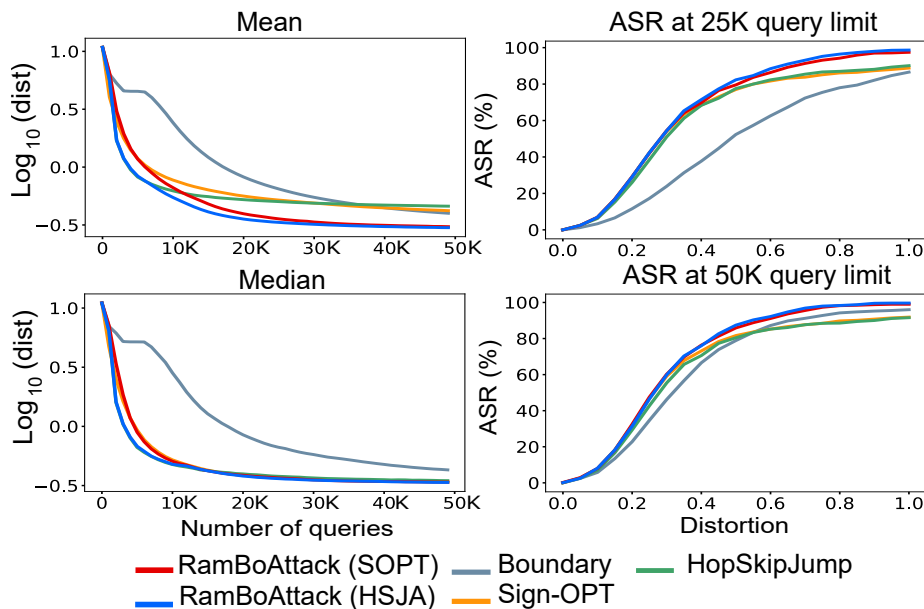


Figure A.1. A comparison between three current state-of-the-art attacks and RAMBOATTACK on a balance set selected from CIFAR10.

Balanced Set with CIFAR10. It is simple to carry out a comprehensive evaluation over all classes, so we choose $N=10$, $n=10$ and $m=9$. In addition, to demonstrate the query efficiency and effectiveness of each attack, we employ a query budget of 25,000 and 50,000 across all experiments. RAMBOATTACK obtain slightly better median and mean distortion than HopSkipJump and Sign-OPT at 25K and 50K, as shown in Figure A.1 and Table A.1. On the standard deviation metric used to measure distortion variance across an evaluation set, our RAMBOATTACK outperform Boundary, HopSkipJump and Sign-OPT at query limit of 25K and 50K. In order words, our attack performs robustly across the evaluation set.

A.1 Targeted Attacks on Balanced and Non-hard Sets

Table A.1: Comparison among attacks with RAMBOATTACK on small and large scale balance datasets.

Query budget	Methods	CIFAR10				ImageNet			
		Mean	Std	Median	ASR($\epsilon=0.3$)	Mean	Std	Median	ASR
25K	Boundary	0.674	0.654	0.499	22.6%	31.80	18.43	32.88	5.5%
	HopSkipJump	0.507	0.748	0.296	50.8%	11.91	8.39	10.87	51.4%
	Sign-OPT	0.526	0.754	0.286	53.6%	14.21	11.52	9.81	46.3%
	RamBo. (HSJA)	0.336	0.218	0.283	54.0%	11.33	8.0	8.62	53.1%
	RamBo. (SOPT)	0.363	0.359	0.282	54.1%	11.25	9.47	9.62	57.5%
50K	Boundary	0.399	0.404	0.319	45.2%	23.73	15.65	20.71	16.6%
	HopSkipJump	0.460	0.683	0.273	55.3%	7.09	5.11	4.87	82.0%
	Sign-OPT	0.420	0.562	0.267	59.1%	7.79	7.84	5.87	73.3%
	RamBo. (HSJA)	0.300	0.178	0.260	59.9%	4.80	3.70	3.92	93.1%
	RamBo. (SOPT)	0.306	0.193	0.261	60.11%	5.02	4.57	3.84	92.3%

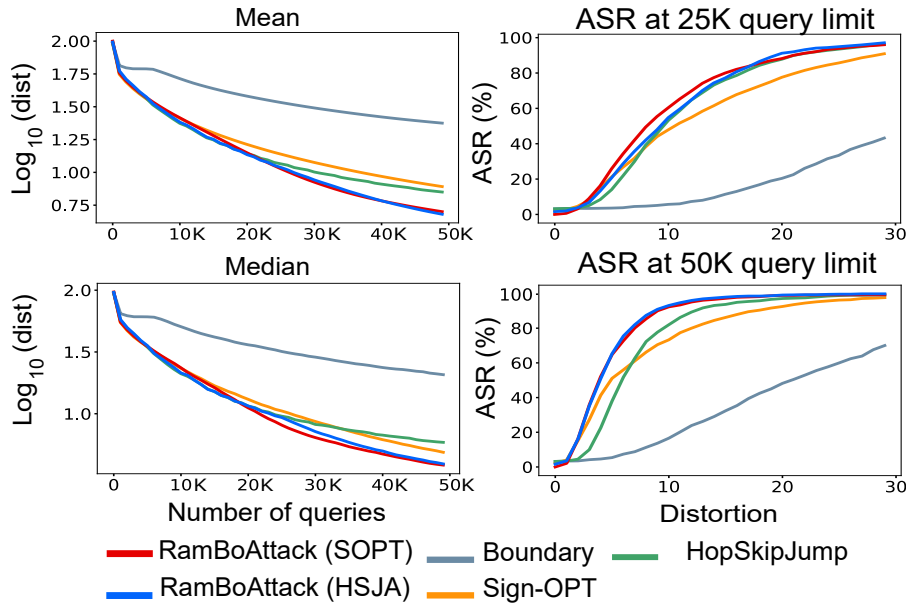


Figure A.2. A comparison between three current state-of-the-art attacks and RAMBOATTACK on a large scale balance set selected from ImageNet.

Balanced Set with ImageNet. ImageNet has 1000 distinct classes, hence carrying out a comprehensive evaluation like on CIFAR10 requires huge computing resources and time. Therefore, we choose $N=200$, $n=1$, $m=5$ and limit the query budget to 25,000 and 50,000. The average distortion (on a \log_{10} scale) against the queries and attack success rate (ASR) at 25K and 50K query budgets achieved by RAMBOATTACK is better than Boundary, Sign-OPT and HopSkipJump attacks as seen in shown in Figure A.2.

As shown in Table A.1, on average distortion metric, RAMBOATTACKS obtain better results and achieve a significantly smaller standard deviation of distortion overall.

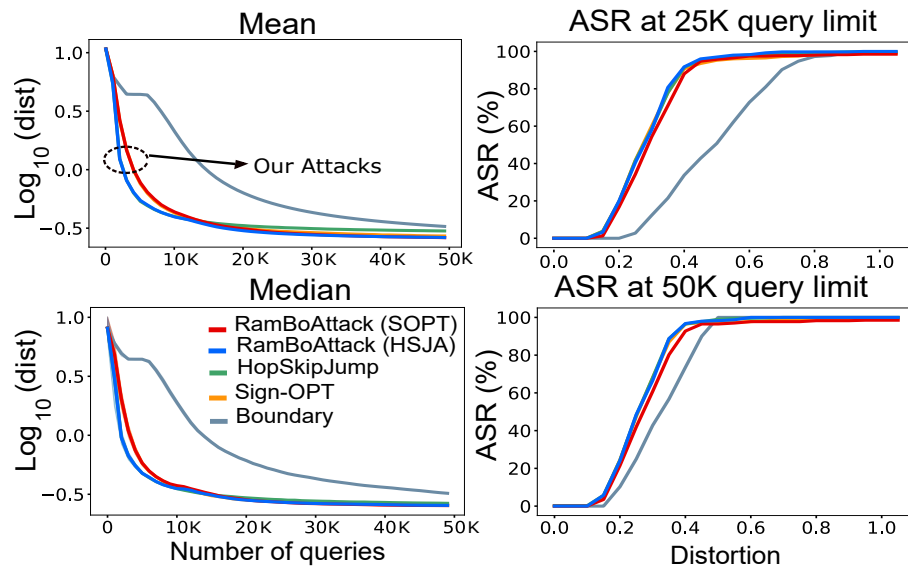


Figure A.3. A comparison between three current state-of-the-art attacks and RAMBOATTACK on a *non-hard* set C selected from CIFAR10. In *non-hard* cases, we perform comparably.

On *non-hard* sets. In this section, we evaluate the performance of SignOPT, HopSkipJump and our RAMBOATTACKS on both CIFAR10 and ImageNet *non-hard* set. The common *non-hard* set C drawn from CIFAR10 for all methods is composed of 400 *non-hard* sample pairs. They are selected such that a distortion between a

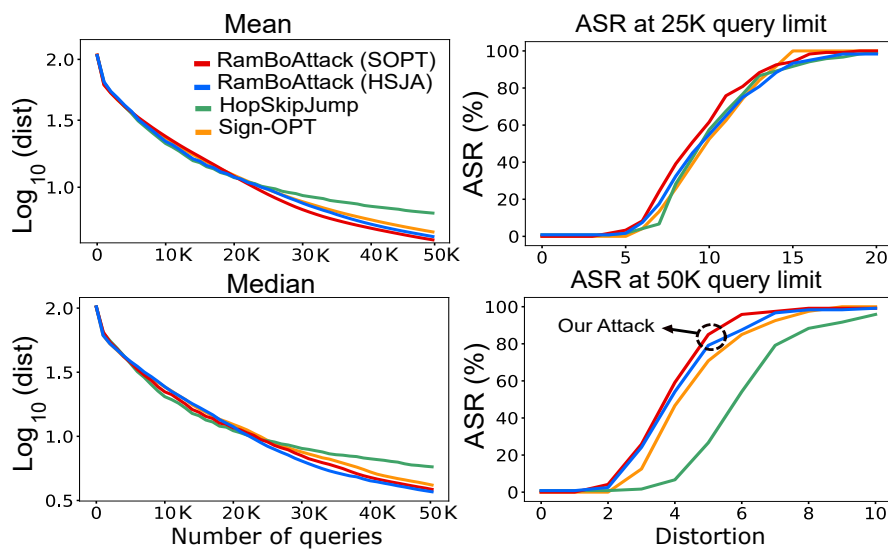


Figure A.4. A comparison between three current state-of-the-art attacks and RAMBOATTACK on a *non-hard* set selected from ImageNet. In *non-hard* cases, RAMBOATTACKS improve attack performance by yielding more effective adversarial examples notable in ASR results.

A.2 Untargeted Attack Validation

source image and its adversarial example found after 50,000 is smaller or equal 0.6. Likewise, a *non-hard set* from ImageNet is composed of 120 *non-hard* sample pairs and the distortion threshold to select these is 7. Figure A.3 and A.4 show that our attack has comparable performance to SignOPT and HopSkipJump on CIFAR10 *non-hard* subsets whilst demonstrating improved attack performance by yielding more effective adversarial examples, especially with a 50K query budget, as seen in the higher attack success rates obtained by RAMBOATTACKS.

A.2 Untargeted Attack Validation

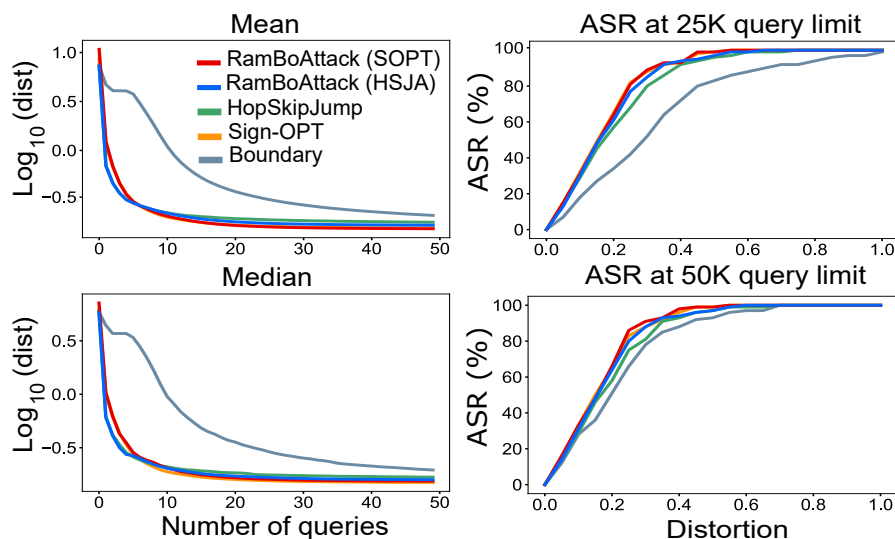


Figure A.5. Comparing between three current state-of-the-art attacks and RAMBOATTACK on the balance set selected from CIFAR10 under untargeted setting.

Here, we evaluate our RAMBOATTACK and other state-of-the-art attacks on two different balanced sets from CIFAR10 and ImageNet as described in Appendix A.1 under an untargeted scenario for completeness. First, on the balance set from CIFAR10, our attacks can achieve comparable performance with Sign-OPT and HopSkipJump and obtain approximately 97% success rate at a distortion of 0.5 on a 25K query budget (see Figure A.5); however, our attack method outperforms Boundary attack. In contrast, on the balance set selected from ImageNet, we observe that our methods can achieve comparable performance with Sign-OPT but outperform HopSkipJump and Boundary attack as shown in Figure A.6.

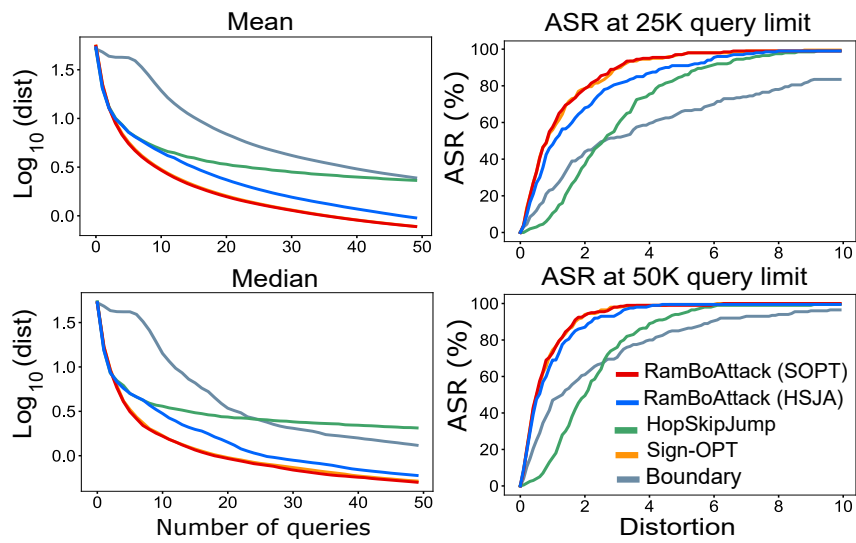


Figure A.6. Comparing between three current state-of-the-art attacks and RAMBOATTACK on the balance set from ImageNet under untargeted setting.

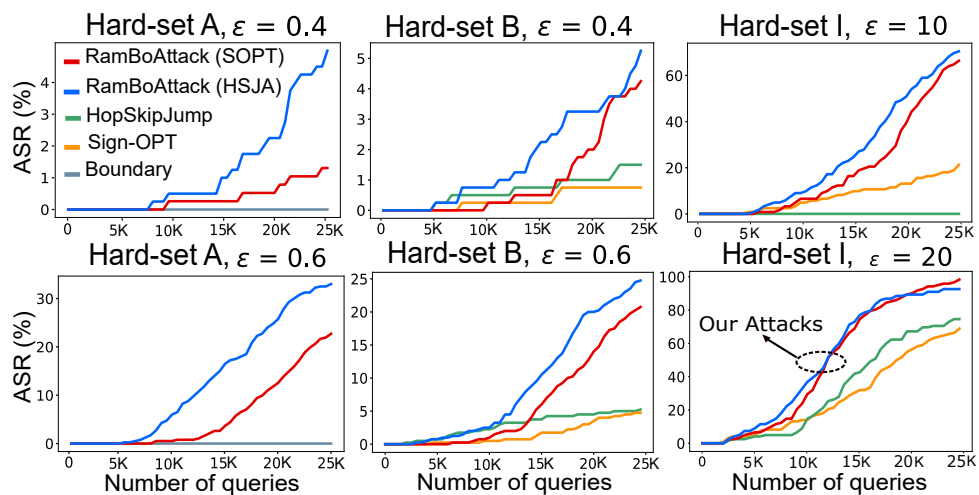


Figure A.7. The first and second columns illustrates ASR vs. queries for our RAMBOATTACKS with respect to Boundary attack on *hard-set A* and with respect to HopSkipJump and Sign-OPT on *hard-set B*. For a given query budget, as expected, our RAMBOATTACKS yield similar ASR to Sign-OPT and HSJA with very low query budgets and significantly higher ASR with budgets above 10K queries, where gradient estimation methods do not appear to improve the adversarial example found with increasing numbers of queries. Similarly, the third column illustrates ASR vs. queries for our RAMBOATTACKS with respect to HopSkipJump and Sign-OPT on the *hard-set I*. RAMBOATTACKS are more query efficient and are able to yield significantly higher ASR under low distortion settings.

A.3 Attack Success Rates vs Query Budgets

In this section, we show results at three different perturbation budgets $\epsilon = 0.4$ and 0.6 for *hard-sets* A and B from CIFAR10 and $\epsilon = 10$ and 20 for the *hard-set* I selected from ImageNet. The results in A.7 demonstrate that our attack is significantly more robust than other attacks within 4-11K query budgets. From 11K, RAMBOATTACKS outperforms others. The reason is that, around this region, the gradient estimation method switches to BLOCKDESCENT, resulting in much higher attack success rates compared to the baselines. Notably, on the high-resolution benchmark task ImageNet, RAMBOATTACKS achieve *significantly* better results compared to the baselines.

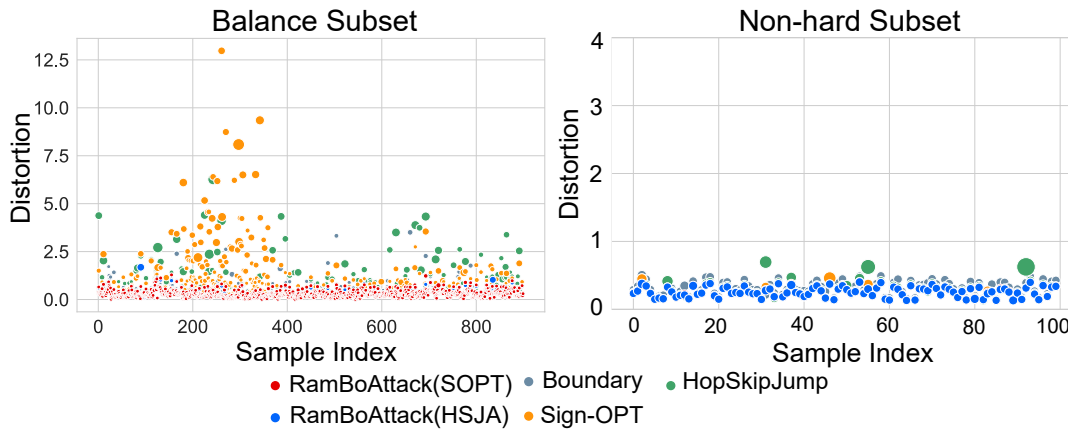


Figure A.8. An illustration of the sensitivity of different attacks to various chosen starting images. The size of each circle denotes the standard deviation and y-axis indicates the mean distortion. The results are from the CIFAR10 balance set and a *non-hard* subset from *non-hard* set C. Compared with Boundary, Sign-OPT and HopSkipJump attacks, our RAMBOATTACKS are **much less sensitive to the choice of starting image** in general. In *non-hard* cases, all of the attacks can achieve comparable results. Hence our attack is demonstrably more robust.

A.4 Impact of Starting Images Balance & Non-hard subset

In this section, we first compose a *non-hard* subset with 100 random *non-hard* sample pairs selected from *non-hard* set C. We also compose a balanced subset from the balance set described in Appendix A.1. We then evaluate our RAMBOATTACK, Sign-OPT, HopSkipJump and Boundary attack on these subsets. To conduct this experiment, for every source image and each of its target classes, we randomly select 10 different

starting images and these attacks are executed with a query budget of 50K. We calculate the mean and standard deviation of distortion for each sample to measure the robustness of each attack to yield adversarial examples for each source image and target class pair.

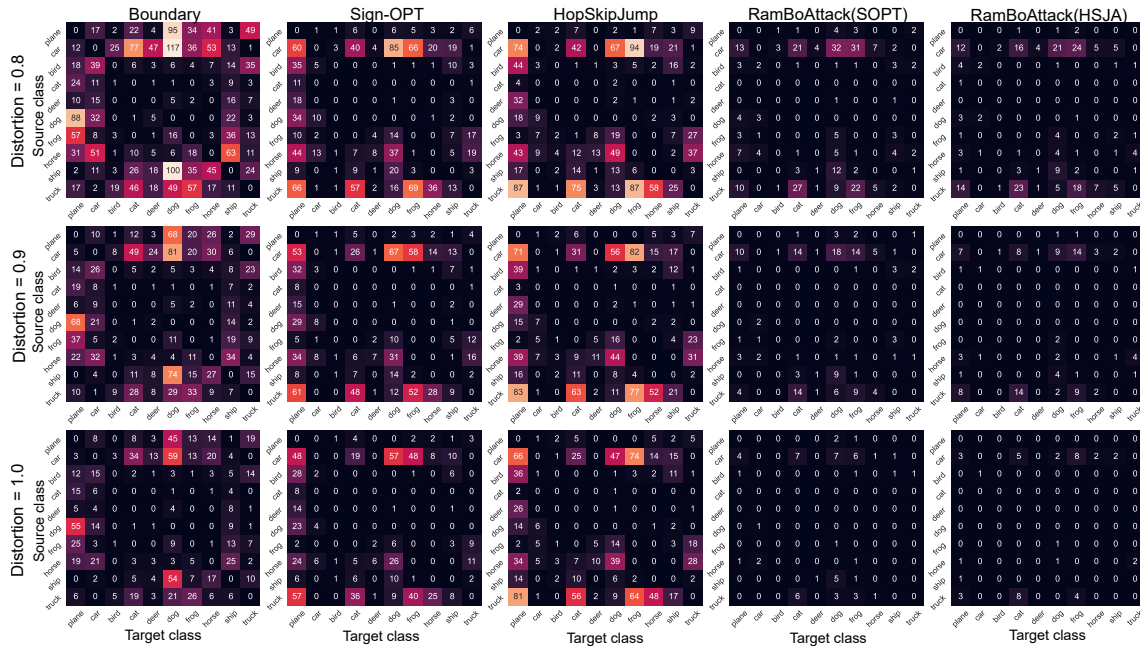


Figure A.9. The number of *hard* cases on CIFAR10 obtained from different attack methods categorized by pairs of source and target classes (at distortion threshold = 0.8, 0.9 and 1.0). RAMBOATTACKS are seen to nearly overcome all of the *hard* cases encountered by other decision-based black-box attack methods; thus, demonstrating the robustness of our proposed attack.

In Figure A.8, the size of each bubble denotes the standard deviation while the y-axis indicates the mean distortion value. We can see that, on the *non-hard* subset, the RAMBOATTACKS are able to achieve comparable results to all of the state-of-the-art methods. On the *balance* subset, our RAMBOATTACKS can achieve significantly less variance (smaller bubbles) at lower distortions while most results achieved by Sign-OPT, HopSkipJump and Boundary indicate larger variance (larger bubbles) and higher distortions. Consequently, our RAMBOATTACKS are more robust than Sign-OPT and HopSkipJump and less sensitive to the chosen starting image.

A.5 Robustness of RamBoAttack

Figure A.9 provides further detailed results on *hard* cases encountered by different attack methods at distortion thresholds of 0.8, 0.9 and 1.0. Compared to Boundary,

A.6 Perturbation Regions and Attack Insights

Sign-OPT and HopSkipJump attacks, our RAMBOATTACKS achieve a much lower number of *hard* cases at all distortion thresholds.

A.6 Perturbation Regions and Attack Insights

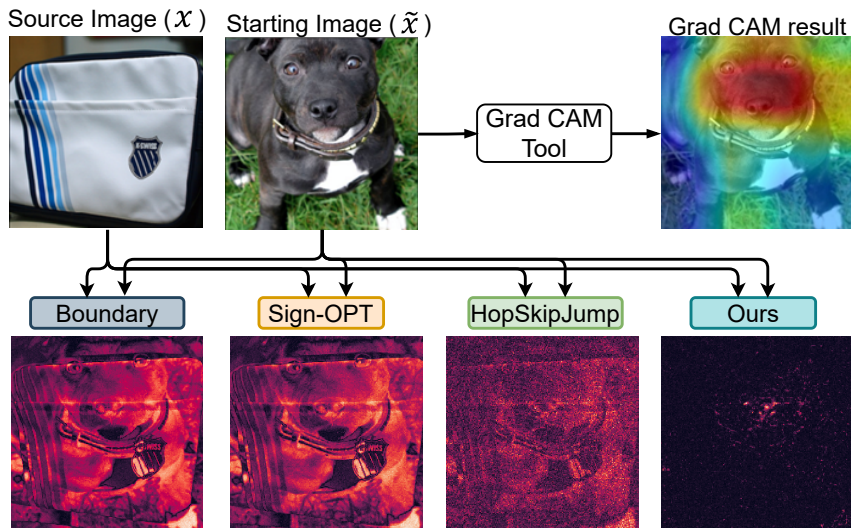


Figure A.10. Grad-CAM tool visualizes salient area of the starting image Staffordshire bull terrier. A perturbation heat map (PHM) visualizes the normalized perturbation magnitude at each pixel. It shows that the perturbation yielded by RAMBOATTACK is able to concentrate on salient areas illustrated by GRAD-CAM even though RAMBOATTACK does not exploit the knowledge of salient regions to perturb.

In this section, we provide additional results on the connection between the adversarial perturbations yielded by RAMBOATTACK and salient regions visualized by the Grad-CAM tool. Effectively, all of the attack methods embedded the target features within the source image where the changes are effectively unnoticeable. However, Figure A.10 illustrates that a high density of adversarial perturbations yielded by our attack concentrates on a region that is matched to the salient features visualized by the Grad-CAM tool. This is possible because our attack methods employ localized changes to search for adversarial examples and are able to effectively find perturbations targeting salient features of the target class to apply to the input source class image to fool the classifier to classify the source image as the target class.

Further, to help visualize different levels of l_2 distortions, we include Figure A.11. We illustrate two examples where we showcase the sample adversarial examples crafted by RAMBOATTACK during the progression of the attack.

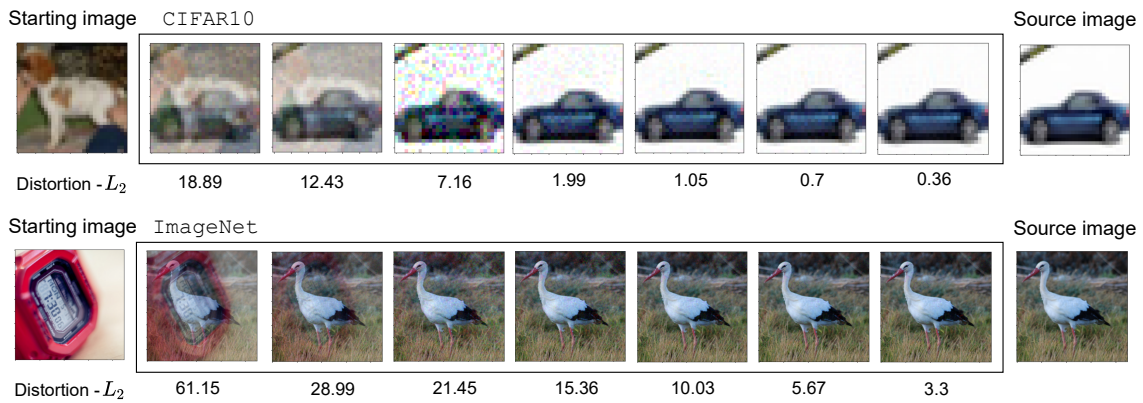


Figure A.11. An illustration of different distortion levels produced by RAMBOATTACK. The first row demonstrates an example from CIFAR10 with a starting image of a dog gradually perturbed until it is similar to the source image car—the adversarial example. The bottom row demonstrates an example from ImageNet with a starting image of a digital watch gradually perturbed until it is similar to the source image white stork—the adversarial example.

A.7 Computation Time of Experiments

A.7.1 Hyper-parameters and Impacts

Gradient Estimation: The main hyper-parameter n_t used in the gradient estimation method is to control when the first component terminates and switches to BLOCKDESCENT. In practice, we keep track of query numbers executed and distortion between the source image and a crafted sample per iteration. This information is then used to determine distortion reduction rate Δ over T queries. On CIFAR10, if applying HopSkipJump or Sign-OPT to the first component, $T = 500$ or 400 , respectively while on ImageNet, $T = 2000$ or 1000 , respectively.

BLOCKDESCENT: The hyper-parameters used are $n = 1$, initial $\delta = P_i(|x - x_s|)$, $m = 1$, $\lambda = 1.2$, $\epsilon_r = 0.01$, $\epsilon_s = 0.01$ for GradEstimation, $\epsilon_s = 0.01$ for BLOCKDESCENT, $T = 500$ and $P_i = P_{100}$. For the *larger* dataset, ImageNet, the changes are: $m = 16$, $\lambda = 2$, $\epsilon_r = 0.1$, $\epsilon_s = 1$ for GradEstimation, $\epsilon_s = 0.1$ for BLOCKDESCENT, $T = 1000$ and $P_i = P_{50}$.

A.7.2 The impact of parameter λ :

The key parameter that may influence BLOCKDESCENT is λ because it controls the step size (or perturbation magnitude δ) for each cycle (see line 28 in Algorithm 3.3). For

A.7.2 The impact of parameter λ :

Table A.2: Summary of computation time for each experiment

Experiments	Duration
Robustness of RAMBOATTACK (Sec. 3.4.4)	627 hrs
Benchmark on <i>Hard</i> & <i>Non-hard</i> sets (Sec. 3.4.5)	275 hrs
Impact of the Starting Images (Sec. 3.4.6)	38 hrs
Visual Explanation (Sec. 3.4.7)	5 hrs
Attack against a Defended Models (Sec. 3.4.8)	281 hrs
Hyper-Parameters and Impacts (App. A.7.1)	60 hrs
Validation on Balance Datasets (App. A.1)	414 hrs
Untargeted Attack Validation (App. A.2)	126 hrs
Total	1826 hrs

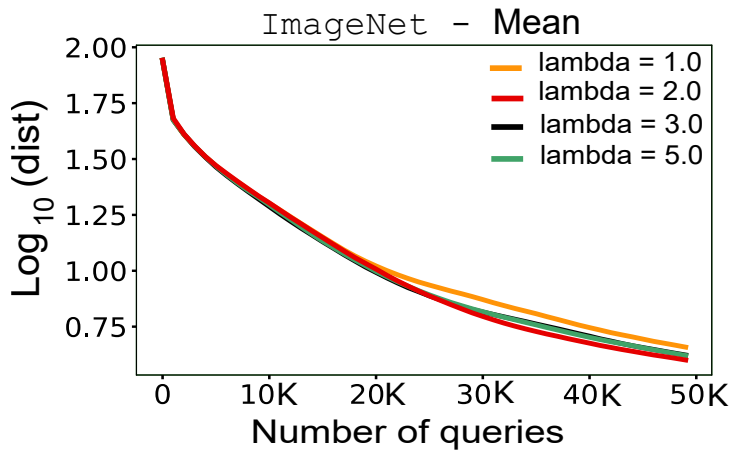


Figure A.12. A comparison between RAMBOATTACK with different values of λ on 100 source and target class sample pairs selected from ImageNet.

example, λ is used to determine the step from $x^{(4)}$ to $x^{(5)}$ in Figure 3.6. If λ is small, δ reduces slightly and thus remains relatively large after each cycle. Consequently BLOCKDESCENT takes large movements that are likely to yield large-magnitude adversarial examples and/or miss the optimal solution. Alternatively, it may cross the decision boundary into an undesired class (source image class in a targeted attack).

In contrast, if λ is large, BLOCKDESCENT takes finer steps to yield adversarial samples whilst moving towards the source image and likely stay in the desired class (target class in a targeted attack). Nevertheless, the empirical result with 100 pairs of source and target class images on ImageNet shown in Figure A.12 illustrates that the

overall performance of RAMBOATTACK is not greatly affected by λ and at $\lambda = 2$, RAMBOATTACK achieves the best performance.

A.8 C&W Attack Configuration and Results Collection

For clarity, here we describe the configuration used for the C&W attack, the C&W execution strategy, and results collection for the C&W attack and black-box attacks.

For the C&W attack, we adopt the PyTorch implementation of the C&W method used in (Cheng et al., 2019b, 2020). In their implementation, they use a learning rate of 0.1 and 1000 iterations for all evaluations (see [GitHub](#)). To search for an adversarial example for an image, the method performs a binary search step to find a relevant constant c within a range from 0.01 to 1000 *until a successful attack is achieved*. With this configuration, the C&W attack is run once to always yield an adversarial example for every instance. We record the distortion of the adversarial example found.

C&W Results Collection. To construct ASR vs. distortion results, at different distortion thresholds: i) we compute the number of source images in the evaluation set meeting a given distortion threshold (along the x-axis); ii) then divide this by the total number of images in the evaluation set to compute the ASR at each distortion value.

Blackbox Attack Results Collection. For the black-box attacks, we perform a black-box attack for each evaluation-set source image, using the set query budgets: 5K, 10K, and 25K. We record the distortion achieved by each source image with a set query budget. To construct ASR vs. distortion, at different distortion thresholds with a given query budget: i) we compute the number of source images in the evaluation set meeting a given distortion threshold (along the x-axis); and ii) then divide this by the total number of images in the evaluation set to compute the ASR at each distortion value.

Appendix B

Chapter 4 Appendix

B.1 Hyper-parameters

We list in Table B.1 the key hyper-parameters used for SPARSEEVO on the two different evaluation sets across CIFAR10 and ImageNet. This hyperparameter set can be applicable for attacking against ViT-B/16 on a large scale and high-resolution dataset—ImageNet. Notably, we only needed to adjust the mutation rate when moving from the high resolution to the low resolution CIFAR10 task; thus, our method provides a robust algorithm that can be easily adapted for different vision tasks. The image size used in all our ImageNet experimental tasks (including experiments on ResNet50 and ViT models) is (3 channels) \times 224 (W) \times 224 (H). This is the standard input size for the pre-trained model (PyTorch) on the ImageNet dataset we used.

B.2 Investigate Hyper-Parameters, Recombination and Mutation

In this section, we conduct comprehensive experiments to study the impacts of hyper-parameters used in our algorithm and the different recombination and mutation schemes we considered. These experiments are carried out on 1,000 randomly selected images from CIFAR10 in an untargeted setting. For the hyper-parameter study, we tune

Table B.1: Hyper-parameters setting in our experiments

Parameters	CIFAR10		ImageNet	
	Untargeted	Targeted	Untargeted	Targeted
Population size (p)	10	10	10	10
Initialization rate (α)	0.004	0.004	0.004	0.004
Mutation rate (μ)	0.04	0.01	0.004	0.001

B.2 Investigate Hyper-Parameters, Recombination and Mutation

population size or mutation rate at a time while using the scheme of recombining the best and two randomly selected candidates from the population as well as the scheme of mutating only 1-bit binary values.

Figure C.4a shows that with different population sizes and a mutation rate of 0.04, even a small population size of 10 is adequate for SPARSEEVO to converge rapidly. Our method with a larger population size almost converges to as low sparsity as the population size of 10 after 200 queries. So population size has a small impact on the overall performance of SPARSEEVO. With a mutation rate of 0.04 and a fixed population size of 10, the algorithm performs well and converges fastest to a low sparsity compared to other mutation rates as shown in Figure C.4b. Consequently, our attack method is more influenced by the mutation rate but this is not unexpected.

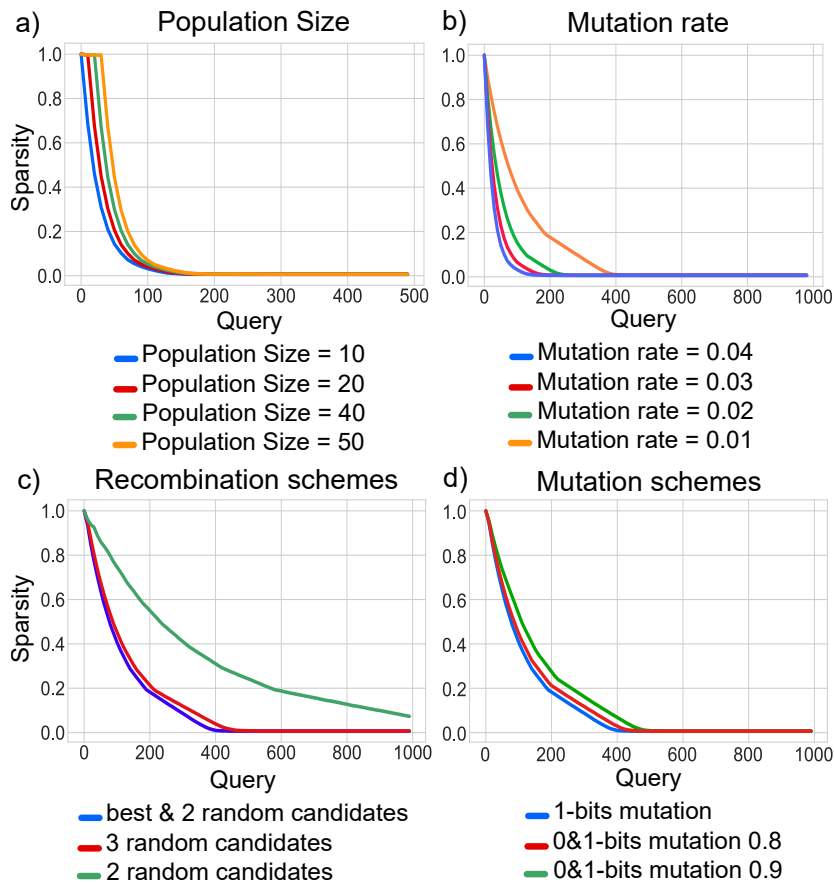


Figure B.1. Sparsity versus number of model queries on CIFAR10 with ResNet18 to show the impacts of different hyper-parameters on SPARSEEVO.

To evaluate how different schemes of recombination and mutation steps affect our method, we use a population size of 10 and mutation rate of 0.01 and change

the recombination or mutation scheme, one at a time. Figure C.4c illustrates that recombining three randomly selected individuals does not achieve as high query efficiency as the scheme of recombining the best and the other two randomly selected from the population. For mutation schemes, we intend to mutate merely 1-bits—*1-bits mutation*—or both 0-bits and 1-bits—*0 & 1-bits mutation*—of a binary vector at a time. For 1-bits mutation scheme, we randomly alter a factor μ of all 1-bits of a selected binary vector. For schemes mutating both 0-bits and 1-bits, we randomly flipped n 1-bits and $\frac{n(1-\beta)}{\beta}$ 0-bits where $n = \mu\beta$. We find that the scheme of mutating only 1-bits performs marginally better than other schemes with $\beta = 0.8$ and $\beta = 0.9$ because mutating both 0 and 1-bits possibly slows down the convergent speed as illustrated in Figure C.4d.

B.3 A Comparison with the Whitebox Baseline

Notably, PGD_0 is an adapted-to- l_0 version of the PGD attack with a projection. PGD_0 simply projects the adversarial example generated by PGD attack onto the l_0 -ball (the process is described in Appendix B.6 earlier regarding adopting non-sparse decision-based attacks). This projection does not guarantee that a projected solution yields the best gradient descent direction for the following iteration of PGD to find an adversarial example that minimises l_0 . Hence, even with full access to the model, PGD_0 may not always yield the optimal solution but rather an approximation and is not always an upper bound for the attack performance, particularly in the untargeted setting on ImageNet as shown in Figure 4.5b and the second plot of Figure 4.7.

B.4 Algorithmic Comparison with PointWise

In this section, we discuss why SPARSEEVO is capable of searching for a desirable solution (an adversarial example with a smaller number of perturbed pixels) with much fewer queries.

1. *Greedy vs. Evolutionary approach.* Pointwise chooses to greedily minimize the number of perturbed pixels by randomly selecting and altering one dimension (i.e. single colour channel) of a randomly selected pixel position ij of the starting image $x' \in R^{C \times W \times H}$ at a time (i.e per query). If the alternation successfully

B.5 Comparison with an Improved PointWise Algorithm

- fools the model, it will be retained; otherwise, the change will be discarded. In contrast, SparseEvo evaluates candidate proposals to alter several pixels at a time and all dimensions of a pixel simultaneously to yield new candidate solutions for the next evolution; so it is able to converge faster and with fewer queries.
2. *Smaller search space.* Pointwise formulation leads to a search space with a size of $C \times W \times H$ where C is the three RGB channels, W is image width and H is image height. We reduce this search space to $W \times H$ because SparseEvo solely searches for pixel positions but does not try to search for different colors for each pixel (see “Defining a Dimensionality Reduced Search Space” in Section 4.3.2 and Appendix B.5).
 3. *Better scalability to large image sizes.* Given that PointWise only changes one dimension at a time (i.e a pixel), to reduce the number of starting image (target class) pixel values different from the source image (to minimize l_0), the random selection method needs to select: i) the same pixel position i, j ; and ii) a different colour channel for the same pixel position i, j in subsequent iterations to move a given pixel value i, j in a starting image (target class image) to be the same as the source image. While this is more likely in a small image task (with smaller W and H values) like CIFAR10, it is far less likely, even within the 20,000 query budget used with large input images in the ImageNet task where mean sparsity values for the 1000 test image pairs remain nearly 1.
 4. *Iterative improvements to “good” solutions.* Importantly, our approach formulates a search for a solution with the minimum number of perturbed pixels through an iterative process of improving upon good solutions from previous iterations informed by our objective function. In contrast, Pointwise employs a purely random method to select the pixel dimension and position i, j to alter.

B.5 Comparison with an Improved PointWise Algorithm

PointWise randomly selects and alters one dimension (a colour channel) of a randomly selected pixel position i, j of an image $x' \in R^{C \times W \times H}$ at a time (i.e. per query). Therefore, the Pointwise formulation leads to a search space with a size of $C \times W \times H$ where C is the three RGB channels, W is image width and H is image height. Consequently, it is not scalable to large image sizes, for example, ImageNet with a size of 224×224 ; this can be observed in Figure 5.4 and 4.5.

Table B.2: Mean sparsity measure at different queries (lower is better) for a targeted attack setting. A comparison between SparseEvo and improved Pointwise on a set of 100 image pairs on ImageNet (here PW- n_p denotes PointWise with the number of selections set to n_p and italicised fonts indicate the best results for PW.)

Query Budgets	1	500	1000	2000	4000	8000	12000	16000	20000
PW(published version)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PW-4	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.93	0.88
PW-8	1.00	<i>1.00</i>	<i>1.00</i>	<i>1.00</i>	<i>0.99</i>	<i>0.94</i>	<i>0.81</i>	<i>0.58</i>	<i>0.35</i>
PW-16	1.00	1.00	1.00	0.99	0.94	0.64	0.45	0.42	0.40
PW-32	1.00	1.00	0.99	0.95	0.71	0.54	0.50	0.46	0.42
PW-64	1.00	1.00	0.95	0.78	0.67	0.62	0.56	0.51	0.46
PW-128	1.00	0.96	0.84	0.77	0.74	0.67	0.61	0.56	0.52
SparseEvo	1.00	0.76	0.63	0.46	0.26	0.08	0.03	0.01	0.01

In this section, we attempted to make PointWise more query efficient on ImageNet by modifying PointWise to perform multiple selections at a time (*i.e.* per query) and perform a series of experiments using different selection parameters n_p . Table B.2 shows the mean sparsity obtained by our improved Pointwise method with different selection parameter values; $n_p = 4, 8, 16, 32, 64, 128$. The results show that the *best performance* of the modified Pointwise algorithm—PW-8—is much better than the original implementation but it is still far behind our method. SparseEvo still outperforms our improved Pointwise algorithms across various query budgets.

B.6 Comparison with Adapted l_0 Attacks

We are motivated to investigate if decision-based dense attacks (l_2 and l_∞ constrained) such as BA (Brendel, Rauber and Bethge, 2018), HSJA (Chen, Jordan and Wainwright, 2020), QEBA (Li et al., 2020), NLBA (Li et al., 2021a), PSBA (Zhang et al., 2021b), SignOPT (Cheng et al., 2020) or RayS (Chen and Gu, 2020) can be adapted to a sparse setting by a projection to l_0 -ball. This idea is promising because PGD can be successfully adapted to a sparse setting to provide a sparse attack algorithm in a white-box setting. In this section, we conduct a study to evaluate this idea by modifying the HSJA method because it is shown to be a query-efficient decision-based dense attack (l_2 and l_∞ constraint), to an l_0 constraint algorithm called l_0 -HSJA.

B.6 Comparison with Adapted l_0 Attacks

Table B.3: Mean sparsity measure at different queries (lower is better) for a targeted setting. A comparison between l_0 -HSJA and SparseEvo on a set of 100 image pairs on CIFAR10

Queries	1	500	1000	2000	4000	8000	12000	16000	20000
l_0 -HSJA	1.00	0.82	0.95	0.92	0.92	0.95	0.95	0.94	0.94
SPARSEEVO	1.00	0.36	0.027	0.025	0.025	0.025	0.025	0.025	0.025

Notably, the same could be done for other methods e.g. QEBA, NLBA, PSBA, SignOPT, or RayS.

Importantly, the authors of HSJA proposed two different ways of gradient estimation purposely formulated for l_2 and l_∞ scenarios. However, the l_0 distance metric is non-differentiable and therefore is ill-suited for standard gradient descent (Carlini and Wagner, 2017; Fan et al., 2020) so we leverage l_2 to estimate the gradient. The difference between the l_0 -HSJA algorithm and published HSJA is the projection step. Instead of performing l_2 and l_∞ projection steps as in HSJA, l_0 -HSJA performs an l_0 projection as in the PGD10 method. To search for the minimum number of pixels to perturb, we adopt a binary search to minimise l_0 . At each iteration (with the discovered adversarial sample from HSJA), we perform the following projection procedure:

1. l_0 -HSJA sorts pixel differences between the sample adversarial crafted by HSJA and the source image.
2. l_0 -HSJA then performs a binary search for k denoting the minimum number of (perturbed) pixels to retain from the sample adversarial crafted by HSJA. Here, $k = \frac{ur+lr}{2}$ where lr and ur are lower and upper ranges, initialized with 0 and N , respectively. N is the total number of pixels in an image.
3. Subsequently, we create a sparse adversarial example by keeping only the top- k pixels of the HSJA crafted adversarial sample and replacing the rest of the pixels of the crafted sample with their corresponding pixel in the source image we plan to fool. These top- k pixels have the least difference from their corresponding pixels. This yields the projected image x_p for evaluation. If the projected sample can mislead a victim model successfully, ur is updated with k (to search for a lower number of perturbed pixels). Otherwise, lr is updated with k .
4. This step is repeated until the ur and lr difference is less than or equal to 1.

For the following iteration of l_0 -HSJA, we use the projected image x_p to craft a new adversarial example x'_p to attempt to improve upon the projected adversarial example from the current iteration.

The results we obtained, shown in Table B.3, illustrate the average sparsity for a set of 100 image pairs on CIFAR-10. Our evaluations show that applying l_0 projection to dense attacks (formulated for l_2 and l_∞ methods) does not yield a query-efficient sparse attack aiming to minimize the number of perturbed pixels. We can understand this result, because, at each projection step, the modified l_0 -HSJA algorithm still requires a large number of queries to determine a projection that minimises l_0 (in other words, to determine the minimum number of pixels to retain where the crafted sample is still adversarial).

To the best of our knowledge, there is no efficient method in a black-box decision-based setting to determine how many pixels and which pixels can be selected to be projected such that the perturbed image does not cross the unknown decision boundary of the DNN model. Additionally, the problem of minimizing the number of selected pixels to be projected leads to an NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020). Although we use the projected image with the minimum number of perturbed pixels, l_2 and l_∞ decision-based attacks require perturbing a whole image in the following iteration, thus the next iteration does not necessarily move the input towards the objective of minimizing the number of perturbed pixels. Thus, l_0 -HSJA and other dense methods do not provide an efficient algorithm for sparse attacks.

B.7 Illustration of Sparse Adversarial Examples

This section provides more illustration of sparse adversarial examples crafted by SPARSEEVO.

B.7 Illustration of Sparse Adversarial Examples

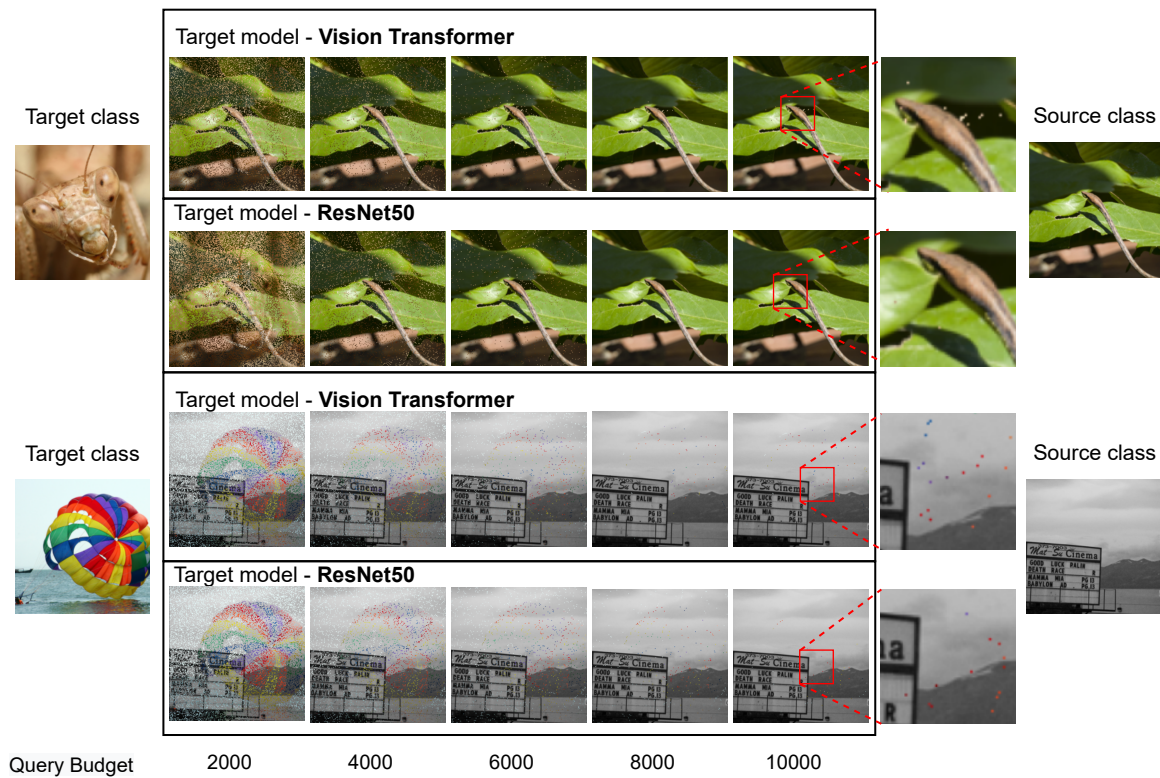


Figure B.2. Visualisations from a targeted attack Settings. Malicious instances generated for a sparse attack with different query budgets using our SPARSEEVO attack algorithm employed on black-box models built for the ImageNet task.

Appendix C

Chapter 5 Appendix

Overview of Materials in the Appendix

A brief overview of the extensive set of additional experimental results and findings in the Appendices that follow. Notably, given the significant computational resource required to mount black-box attacks against models and extensive additional experiments, CIFAR-10 is employed for the comparative studies. Importantly, empirical results have already established the generalizability of the proposed attack across CNN models, ViT models, three datasets and Google Cloud Vision.

1. Evaluation of score-based sparse attacks on ImageNet (targeted settings at sparsity levels between and including 0.4% and 1.0%; and untargeted settings) (Appendix C.1).
2. Evaluation of score-based sparse attacks on STL-10 to demonstrate generalization (Appendix C.2).
3. Evaluation of score-based sparse attacks on CIFAR-10 demonstrate generalization (Appendix C.3).
4. Additional evaluation of attack algorithms adopted for sparse attacks (l_0 attacks) (Appendix C.4)
5. Comparing BRUSLEATTACK and SPARSEEVO to supplement the results in Figure 5.5 (Appendix C.4.1)
6. A Discussion Between BRUSLEATTACK (Adversarial Attack) and B3D (Black-box Backdoor Detection) (Appendix C.4.5)
7. Additional evaluation of score-based sparse attacks against state-of-the-art robust models from Robustbench (Appendix C.5).

-
8. Proof of the optimization reformulation (Appendix C.6)
 9. An analysis of the search space reformulation and dimensionality reduction. (Appendix C.7).
 10. An analysis of different generation schemes for synthetic images we considered (Appendix C.8).
 11. Study of BRUSLEATTACK performance under different random seeds (Appendix C.9).
 12. An analysis of the effectiveness of the dissimilarity map employed in our proposed attack algorithm (Appendix C.10).
 13. Detailed information on the consistent set of hyper-parameters employed, initialization value for α^{prior} and computation resources used (Appendix C.11).
 14. The notable performance invariance to hyper-parameter choices studies with CIFAR-10 and ImageNet (Appendix C.12).
 15. Additional study of employing different schedulers (Appendix C.13).
 16. Detailed information on the evaluation protocols BRUSLEATTACK (Appendix C.14).
 17. Visualizations of sparse attack against Google Cloud Vision (Appendix C.15).
 18. Additional visualizations of dissimilarity maps and sparse adversarial examples (Appendix C.16).

C.1 Sparse Attack Evaluations On ImageNet

This section shows detailed results for evaluating the performance of sparse attacks in targeted and untargeted settings on ImageNet. The section then analyzes the relative robustness comparison among models.

Table C.1: ASR at different sparsity levels across different queries (higher is better). A comprehensive comparison among different attacks (SPARSE-RS and BRUSLEATTACK) against various Deep Learning models on ImageNet in the targeted setting.

Query	ResNet-50		ResNet-50(SIN)		ViT	
	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK
Sparsity = 0.4%						
4000	49.9%	57.3%	40.5%	47.8%	21.5%	26.0%
6000	65.5%	69.4%	55.0%	60.4%	31.8%	37.3%
8000	74.1%	77.3%	63.3%	66.6%	39.6%	43.9%
10000	79.1%	82.7%	68.5%	70.9%	45.2%	49.0%
Sparsity = 0.6%						
4000	59.6%	75.1%	49.7%	66.2%	30.8%	40.7%
6000	74.0%	86.3%	65.6%	77.8%	43.7%	52.0%
8000	85.0%	90.3%	77.6%	83.4%	52.2%	61.0%
10000	90.9%	93.0%	84.3%	87.0%	61.7%	67.3%
Sparsity = 0.8%						
4000	65.8%	84.3%	56.3%	76.7%	38.2%	49.4%
6000	79.2	90.6%	71.1%	87.0%	50.2%	63.4%
8000	87.9%	94.3%	81.9%	91.0%	60.0%	72.2%
10000	93.4%	96.4%	89.6%	92.4%	69.6%	79.0%
Sparsity = 1.0%						
4000	69.3%	88.6%	59.2%	82.4%	43.1%	56.8%
6000	82.1	94.2%	75.6%	91.4%	56.1%	72.4%
8000	89.8%	96.8%	83.8%	94.0%	65.6%	81.3%
10000	94.3%	97.7%	91.0%	95.5%	74.3%	86.8%

Targeted Settings. Table C.1 shows the detailed ASR results for sparse attacks on high-resolution dataset ImageNet in the targeted settings shown in Section 5.5.3. The results illustrate that the proposed method is consistently better than SPARSE-RS across different sparsity levels from 0.4 % to 1.0 %.

Untargeted Settings. In this section, we verify the performance of sparse attacks against different Deep Learning models including ResNet-50, ResNet-50 (SIN) and ViT models in the untargeted setting up to a 5K query budget. We use an evaluation set

C.1 Sparse Attack Evaluations On ImageNet

Table C.2: ASR at different sparsity levels across different queries (higher is better). A comprehensive comparison among different attacks (SPARSE-RS and BRUSLEATTACK) and various DL models on ImageNet in the untargeted setting.

Query	ResNet-50		ResNet-50(SIN)		ViT	
	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK
Sparsity = 0.04%						
1000	52.4%	58.8%	51.0%	55.4%	29.0%	31.2%
2000	58.4%	65.0%	59.2%	63.6%	36.2%	37.4%
3000	61.8%	68.4%	63.8%	67.0%	41.0%	41.2%
4000	65.4%	70.4%	65.8%	68.2%	44.2%	44.4%
5000	66.4%	72.4%	66.6%	69.2%	46.4%	46.7%
Sparsity = 0.08%						
1000	72.8%	77.4%	73.8%	75.8%	47.2%	50.6%
2000	81.2%	86.8%	80.4%	83.4%	57.6%	61.0%
3000	84.6%	89%	84.4%	87.0%	64.2%	67.8%
4000	85.6%	90.4%	86.6%	88.2%	69.6%	72.6%
5000	86.8%	90.8%	87.0%	88.6%	72.6%	74.6%
Sparsity = 0.16%						
1000	87.0%	89.4%	87.6%	88.0%	64.8%	68.6%
2000	90.8%	95.2%	92.0%	94.0%	78.4%	81.4%
3000	93.4	96.8%	94.8%	95.6%	85.0%	86.4%
4000	94.4%	97.6%	96.2%	97.0%	87.0%	89.2%
5000	94.8%	98.4%	96.8%	97.4%	89.8%	90.0%
Sparsity = 0.2%						
1000	88.6%	92.2%	90.2%	91.0%	71.2%	73.0%
2000	92.4%	96.6%	94.4%	95.0%	82.6%	84.4%
3000	94.4	97.8%	95.8%	96.4%	87.4%	89.8%
4000	95.2%	98.4%	97.2%	98.0%	90.8%	91.0%
5000	95.4%	98.6%	98.2%	98.4%	92.2%	92.6%

of 500 random pairs of an image and a target class to conduct this comprehensive experiment. Our results in Table C.2 and Table C.1a-c show that BRUSLEATTACK is marginally better than SPARSE-RS across different sparsity levels when attacking against ViT. For ResNet-50 and ResNet-50 (SIN), at lower sparsity or lower query limits, our proposed attack outperforms SPARSE-RS while at higher query budgets or higher sparsity levels, SPARSE-RS is able to obtain slightly lower ASR than our method. In general, BRUSLEATTACK consistently outperforms SPARSE-RS and only

needs 1K queries and sparsity of 0.2% (100 pixels) to achieve above 90% ASR against both ResNet-50 and ResNet-50 (SIN).

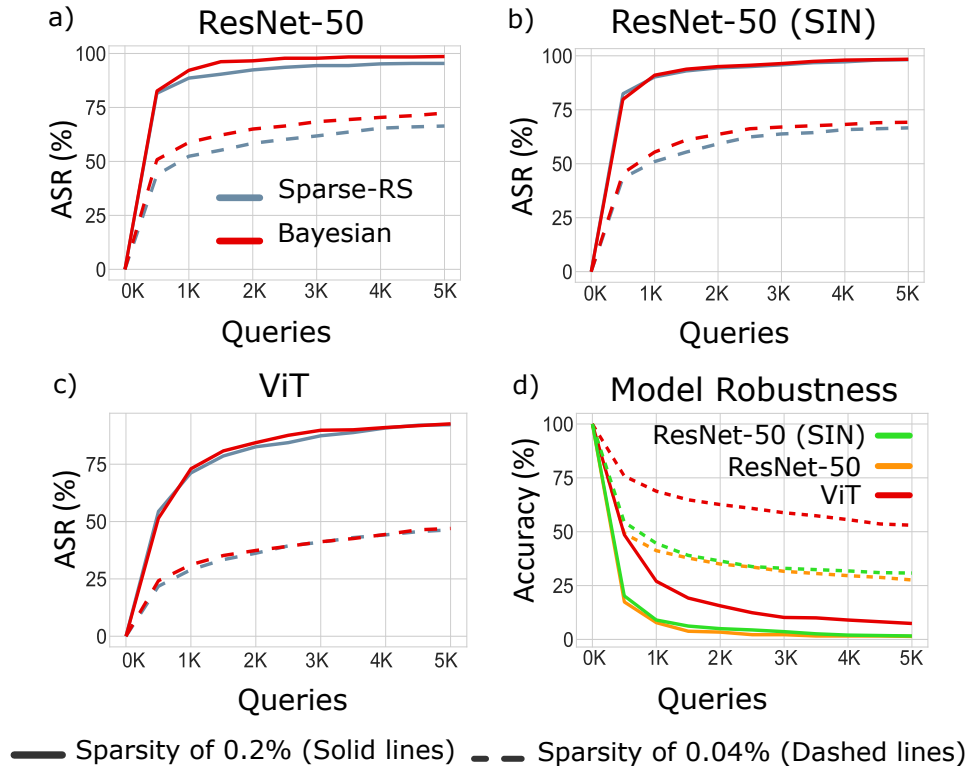


Figure C.1. a-c) **Untargetted Setting.** ASR versus the number of model queries against different Deep Learning models at sparsity levels of 0.4% (solid lines) and 1.0% (dashed lines); d) Accuracy versus the number of model queries for model robustness comparison against BRUSLEATTACK, in the untargeted setting and at sparsity levels ($0.04\% = \frac{40}{224 \times 224}$, $0.2\% = \frac{100}{224 \times 224}$).

Relative Robustness Comparison among Models. To compare the relative robustness of different models, we evaluate these models against our attack. Table C.2 and Figure C.1d confirm our observations about the relative robustness of ResNet-50 (SIN) to the standard ResNet-50 in the targeted setting (presented in Section 5.5.3). It turns out that ResNet-50 (SIN) is as vulnerable as the standard ResNet-50 even though it is robust against various types of image distortion. Interestingly, ViT is more robust than its convolutional counterparts under sparse attack. Particularly, at sparsity of 0.2% and 2K queries, while the accuracy of both ResNet-50 and ResNet-50 (SIN) is down to about 5%, ViT is still able to remain ASR around 15%.

C.2 Sparse Attack Evaluations on STL10 (Targeted Settings)

Table C.3: ASR (higher is better) at different sparsity levels in targeted settings. A comprehensive comparison between SPARSE-RS and BRUSLEATTACK against ResNet9 on a full evaluation set from STL-10.

Methods	Q=1000	Q=2000	Q=3000	Q=4000	Q=1000	Q=2000	Q=3000	Q=4000
	Sparsity = 0.22%				Sparsity = 0.44%			
SPARSE-RS	53.82%	61.65%	65.84%	68.0%	73.34%	81.47%	85.24%	87.49%
BRUSLEATTACK	57.69%	65.05%	68.8%	71.22%	78.21%	85.03%	88.31%	90.26%
	Sparsity = 0.33%				Sparsity = 0.54%			
SPARSE-RS	65.6%	74.0%	78.0%	80.65%	78.66%	86.31%	89.64%	91.61%
BRUSLEATTACK	70.27%	77.55%	81.16%	83.42%	83.29%	89.78%	92.55%	94.08%

We conduct more extensive experiments on STL-10 in the targeted setting with all correctly classified images of the evaluation set (60,094 sample pairs and image size 96×96). Table C.3 provides a comprehensive comparison of different attacks across different sparsity levels ranging from 0.11% (10 pixels) to 0.54% (50 pixels). Particularly, with only 50 pixels, BRUSLEATTACK needs solely 3000 queries to achieve ASR beyond 92% whereas SPARSE-RS only reaches ASR of 89.64%.

C.3 Sparse Attack Evaluations on CIFAR-10 (Targeted Settings)

In this section, we conduct extensive experiments in the targeted setting to investigate the robustness of sparse attacks on an evaluation set of 9,000 pairs of an image and a target class from CIFAR-10 (image size 32×32). Sparsity levels range from 1.0% (10 pixels) to 3.9% (40 pixels). Table C.4 provides a comprehensive comparison of different attacks in the targeted setting. Particularly, with only 20 pixels (sparsity of 2.0%), BRUSLEATTACK needs solely 500 queries to achieve ASR beyond 90% whereas SPARSE-RS only reaches ASR of 89.21%. Additionally, with only 300 queries, BRUSLEATTACK is able to reach above 95% of successfully crafting adversarial examples with solely 40 pixels. Overall, our attack consistently outperforms the SPARSE-RS in terms of ASR and this confirms our observations on STL-10 and ImageNet.

Table C.4: ASR (higher is better) at different sparsity thresholds in the targeted setting. A comprehensive comparison among different attacks (SPARSE-RS and BRUSLEATTACK) against ResNet18 on an evaluation set of 9,000 pairs of an image and a target class from CIFAR-10.

Methods	Q=100	Q=200	Q=300	Q=400	Q=500
Sparsity = 1.0%					
SPARSE-RS	36.22%	50.6%	58.17 %	62.59%	66.26%
BRUSLEATTACK	42.32%	54.73%	61.49%	65.33%	68.21%
Sparsity = 2.0%					
SPARSE-RS	60.51%	76.1%	83.13%	86.89%	89.21%
BRUSLEATTACK	66.01%	79.19%	84.84%	88.27%	90.24%
Sparsity = 2.9%					
SPARSE-RS	71.29%	85.67%	91.21%	94.28%	95.78%
BRUSLEATTACK	75.54%	88.22%	92.91%	95.2%	96.59%
Sparsity = 3.9%					
SPARSE-RS	75.91%	90.21%	94.78%	96.97%	97.98%
BRUSLEATTACK	80.44%	91.24%	95.43%	97.4%	98.48%

C.4 Comparing BruSLeAttack With Other Attacks Adapted for Score-Based Sparse Attacks For Additional Baselines

C.4.1 Additional Evaluations With Decision-Based Sparse Attack Methods

This section carries out a comprehensive experiment on CIFAR-10 in the targeted setting (more difficult attack) with 9000 different pairs of the source image and target classes (1000 images distributed evenly in 10 different classes against 9 target classes) to compare BRUSLEATTACK (500 queries) with SPARSEEVO (2k queries) introduced in Chapter 4. We compare ASR of different methods across different sparsity thresholds. The results in Figure C.2 demonstrate that our attack significantly outperforms SparseEvo. This is expected because SparseEvo is a decision-based attack and has only access to predicted labels.

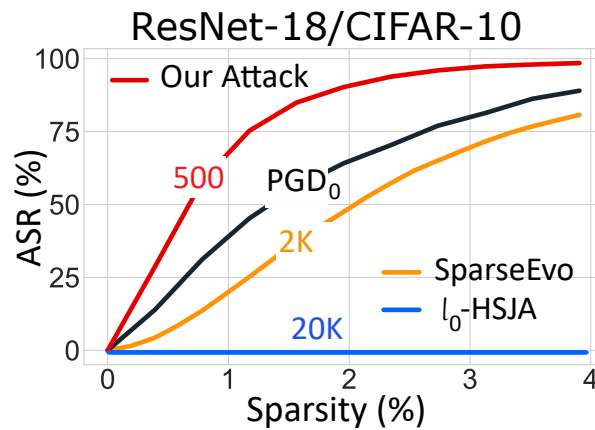


Figure C.2. Targeted attacks on CIFAR-10 against ResNet-18. ASR comparisons between BRUSLEATTACK and baselines i) SPARSE-RS and adapted l_0 -HSJA (decision-based settings); ii) PGD₀ (whitebox).

C.4.2 l_0 Adaptations of Dense Attacks

Adapted l_0 Attacks (White-box). For our interests, we explore a strong white-box l_0 attack which is adapted from PGD (Madry et al., 2018)—named PGD₀ (Croce and Hein, 2019). To this end, we compare BRUSLEATTACK with white-box adapted l_0 attack PGD₀ using the same evaluation set from CIFAR-10 as decision-based attacks. The results in Figure C.2 demonstrate that our attack significantly outperforms PGD₀ at low sparsity threshold and is comparable to PGD₀ at high level of sparsity. Surprisingly, our method outweighs white-box, adapted l_0 attack PGD₀. It is worth noting that there is no effective projection method to identify the pixels that can satisfy sparse constraint and solving l_0 projection problem also encounter NP-hard problem. Additionally, discrete nature of the l_0 ball impedes its amenability to continuous optimization (Croce et al., 2022).

Adapted l_0 Attacks (Decision-based). It is interesting to adapt l_2 attacks such as HSJA (Chen, Jordan and Wainwright, 2020), QEBA (Li et al., 2020), or CMA-ES (Dong et al., 2020) method for face recognition tasks to l_0 attacks. Consequently, we adopted the HSJA method to an l_0 constraint algorithm called l_0 -HSJA to conduct a study. For l_0 -HSJA, we follow the experiment settings and adapted l_0 -HSJA in Chapter 4 and refer to Chapter 4 for more details. Notably, the same approach could be adopted for QEBA (Li et al., 2020). The results in Table C.5 below illustrate the average sparsity for 100 randomly selected source images, where each image was used to construct a sparse adversarial sample for the 9 different target classes on CIFAR-10—hence we conducted 900 attacks or used 900 source-image-to-target-class pairs. The average sparsity across

Table C.5: Mean sparsity at different queries for a targeted setting. A sparsity comparison between l_0 -HSJA on a set of 100 image pairs on CIFAR-10.

Queries	4000	8000	12000	16000	20000
l_0 -HSJA	93.66%	94.73%	95.88%	96.74%	96.74%

different query budgets is higher than 90% even up to 20K queries. Therefore, the ASR is always 0% at low levels of sparsity (e.g. 4%) (shown in Figure C.2). These results confirm the findings in Chapter 4 and demonstrate that l_0 -HSJA (20K queries) is not able to achieve good sparsity (lower is better) when compared with our attack method. Consequently, applying an l_0 projection to decision-based dense attacks does not yield a strong sparse attack.

Similar to the problem of PGD_0 , adapted l_0 -HSJA has to determine a projection that minimizes l_0 (the minimum number of pixels) such that the projected instance is still adversarial. To the best of our knowledge, no method in a decision-based setting is able to effectively determine which pixels can be selected to be projected such that the perturbed image does not cross the unknown decision boundary of the DNN model. Solving this projection problem may also lead to another NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020) and hinder the adoption of these dense attack algorithms to the l_0 constraint. Consequently, any adapted method, such as HSJA or other dense attacks, is not capable of providing an efficient method to solve the combinatorial optimization problem faced in sparse settings.

C.4.3 Comparing BruSLeAttack With One-Pixel Attack

In this section, we conduct an experiment to compare BRUSLEATTACK with the One-Pixel Attack (Su, Vargas and Sakurai, 2019). We conduct an experiment with 1000 correctly classified images by ResNet18 on CIFAR10 in untargeted settings (notably the easier attack, compared to targeted settings) using ResNet18. These images are evenly distributed across 10 different classes. We compare ASR between our attack and One-Pixel at different budgets e.g. one, three and five perturbed pixels. For the One-Pixel attack¹⁰, we used the default setting with 1000 queries. To be fair, we set the same query limits for our attack. The results in Table C.6 show that our attack

¹⁰<https://github.com/Harry24k/adversarial-attacks-pytorch>

C.4.4 Bayesian Optimization

Table C.6: ASR comparison (higher \uparrow is stronger) between One-Pixel and BRUSLEATTACK against ResNet18 on CIFAR-10.

Perturbed Pixels	One-Pixel	BRUSLEATTACK
1 pixel	19.5%	27.9%
3 pixel	41.9%	69.9%
5 pixel	62.3%	86.4%

outperforms the One-Pixel attack across one, three and five perturbed pixels, even under the easier, untargeted attack setting.

C.4.4 Bayesian Optimization

We are interested in the application of Bayesian Optimization for high-dimensional, mixed search space. Recently, (Wan et al., 2021) has introduced CASMOPOLITAN, a Bayesian Optimization for categorical and mixed search spaces, demonstrating that this method is efficient and better than other Bayesian Optimization methods in searching for adversarial examples in score-based settings. Therefore, we study and compare our method with CASMOPOLITAN *in the vision domain and the application of seeking sparse adversarial examples*. We note that:

- CASMOPOLITAN solves problem 5.1 directly by searching for altered pixel positions and the colors for these pixels. In the meanwhile, our method aims to address problem 5.2, which is reformulated to reduce the dimensionality and complexity of the search space significantly. In general, CASMOPOLITAN aims to search for both color values and pixel positions, whilst BRUSLEATTACK only seeks pixel locations.
- To handle high dimensional search space in an image task, CASMOPOLITAN employs different downsampling/upsampling techniques. It first downscales the image and searches over a low-dimensional space, manipulates and then upscales the crafted examples. Unlike CASMOPOLITAN, our method—BRUSLEATTACK—does not reduce dimensionality by downsampling the original search space but only seeks pixels in an image (source image) and replaces them with corresponding pixels from a synthetic color image (a fixed and pre-defined image) (see Appendix C.7 for our analysis of dimensionality reduction).

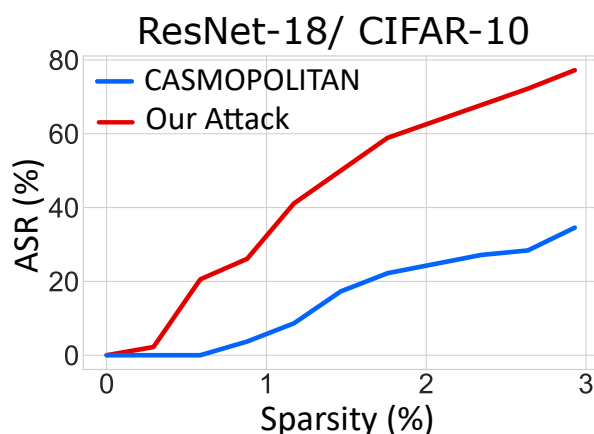


Figure C.3. Targeted attacks on CIFAR-10 with a query budget of 250. ASR comparisons between BRUSLEATTACK and CASMOPOLITAN (Bayesian Optimization).

- CASMOPOLITAN is not designed to learn the impact of pixels on the model decisions but treats all pixels equally, whereas BRUSLEATTACK aims to explore the influence of pixels through the historical information of pixel manipulation.

We use the code¹¹ provided in (Wan et al., 2021) and follow their default settings. We evaluate both BRUSLEATTACK and CASMOPOLITAN on an evaluation set of 900 pairs of a source image and a target class from CIFAR-10 (100 correctly classified images distributed evenly in 10 different classes versus the 9 other classes as target classes for each image) with a query budget of 250. The results in Figure C.3 show that BRUSLEATTACK consistently and pragmatically outperforms CASMOPOLITAN across different sparsity levels. This is because:

- The mixed search space in the vision domain, particularly in sparse adversarial attacks, is still extremely enormous even if downsampling to a lower dimensional search space. It is because CASMOPOLITAN still needs to search for a color value for each channel of each pixel from a large range of values (see Appendix C.7 for our analysis of dimensionality reduction).
- Searching in a low-dimensional search space and upscaling back to the original search space may not provide an effective way to yield a strong sparse adversarial perturbation. This is because manipulating pixels in a lower dimensional search space may not have the same influence on model decisions as manipulating pixels in the original search space. Additionally, some indirectly altered pixels

¹¹<https://github.com/xingchenwan/Casmopolitan>

C.4.5 A Discussion Between BruSLeAttack (Adversarial Attack) and B3D (Black-box Backdoor Detection)

stemming from upsampling techniques may not greatly impact the model decisions.

C.4.5 A Discussion Between BruSLeAttack (Adversarial Attack) and B3D (Black-box Backdoor Detection)

Natural Evolution Strategies (NES). A family of black-box optimization methods that learns a search distribution by employing an estimated gradient on its distribution parameters (Wierstra et al., 2008; Dong et al., 2021). NES was adopted for score-based dense (l_2 and l_∞ norms) attacks in (Ilyas et al., 2018) since they mainly adopted a Gaussian distribution for continuous variables. However, solving the problem posed in sparse attacks involving both discrete and continuous variables leads to an NP-hard problem (Modas, Moosavi-Dezfooli and Frossard, 2019; Dong et al., 2020). Therefore, naively adopting NES for sparse attacks is non-trivial.

The work B3D (Dong et al., 2021), in defense of a data poisoning attack or backdoor attack, proposed an algorithm to reverse-engineer the potential Trojan trigger used to activate the backdoor injected into a model. Although the method is motivated by NES and operates in a score-based setting involving both continuous and discrete variables, as with a sparse attack problem, they are designed for completely different threat models (backdoor attacks with data poisoning versus adversarial attacks). Therefore it is hard to make a direct comparison. However, more qualitatively, there are a number of key differences between our approach and those relevant elements in (Dong et al., 2021).

1. Method and Distribution differences: (Dong et al., 2021) learns a search distribution determined by its parameters by estimating the gradient on the parameters of this search distribution. In the meantime, our approach is to learn a search distribution through Bayesian learning. While (Dong et al., 2021) employed Bernoulli distribution for working with discrete variables, we used Categorical distribution to search discrete variables.
2. Search space (larger vs. smaller): B3D searches for a potential Trojan trigger in an enormous space as it requires to search for pixels' position and color. Our approach reduces the search space and only searches for pixels (pixels' position) to be altered so our search space is significantly lower than the search space used

in (Dong et al., 2021) if the trigger size is the same as the number of perturbed pixels.

3. Perturbation pattern (square shape vs. any set of pixel distribution): (Dong et al., 2021) aims to search for a trigger which usually has a size of 1×1 , 2×2 or 3×3 so the trigger shape is a small square. In contrast, our attack aims to search for a set of pixels that could be anywhere in an image and the number of pixels could be varied tremendously (determined by desired sparsity). Thus, the combinatorial solutions in a sparse attack problem can be larger than the one in (Dong et al., 2021) (even when we equate the trigger size to the number of perturbed pixels).
4. Query efficiency (is a primary objective vs. not an objective): Our approach aims to search for a solution in a query-efficiency manner while it is not clear how efficient the method is to reverse-engineer a trigger.

C.5 Evaluations Against l_2, l_∞ Robust Models From Robustbench and l_1 Robust Models

l_2, l_∞ *Robust Models*. To supplement our demonstration of sparse attacks (BRUSLEATTACK and SPARSE-RS) against defended models on ImageNet in Section 5.5.5, we consider evaluations against SoTA robust models from RobustBench¹² (Croce et al., 2020) on CIFAR-10. We evaluate the robustness of sparse attacks (BRUSLEATTACK and SPARSE-RS) against the undefended model ResNet-18 and two pre-trained robust models as follows:

- l_2 robust model: “Augustin2020Adversarial-34-10-extra”. This model is a top-7 robust model (over 20 robust models) in the leaderboard of robustbench.
- l_∞ robust model: “Gowal2021Improving-70-16-ddpm-100m”. This model is a top-5 robust model (over 67 robust models) in the leaderboard of robustbench.

We use 1000 samples correctly classified by the pre-trained robust models and evenly distributed across 10 classes on CIFAR-10. We use a query budget of 500. We compare the accuracy of different models (undefended and defended models) under sparse attacks across a range of Sparsity from 0.39% to 1.56%. Notably, defended models

¹²<https://github.com/RobustBench/robustbench>

C.5 Evaluations Against l_2, l_∞ Robust Models From Robustbench and l_1 Robust Models

Table C.7: A robustness comparison (lower \downarrow is stronger) between SPARSE-RS and BRUSLEATTACK against undefended and defended models employing l_∞, l_2 robust models on CIFAR-10. The attack robustness is measured by the degraded accuracy of models under attacks at different sparsity levels.

Sparsity	Undefended Model		l_∞ -Robust Model		l_2 -Robust Model	
	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK
0.39%	26.5%	24.2%	65.9%	65.0%	84.7%	84.2%
0.78%	7.8%	6.4%	48.1%	46.0%	70.6%	68.3%
1.17%	2.5%	2.0%	38.1%	35.1%	57.6%	54.3%
1.56%	0.6%	0.6%	28.8%	26.4%	44.4%	43.8%

are usually evaluated in the untargeted setting to show their robustness. The range of sparsity in the untargeted setting is usually smaller than the range of sparsity used in the targeted setting. Thus, in this experiment, we use a smaller range of sparsity than the one we used in the targeted setting. Our results in Table C.7 show that BRUSLEATTACK outperforms SPARSE-RS when attacking undefended and defended models. The results on CIFAR-10 also confirm our observations on ImageNet.

l_1 *Robust Models*. We also evaluate our attack method’s robustness against l_1 robust models. There are two methods AA-I1 (Croce and Hein, 2021) and Fast-EG-1 (Jiang et al., 2023) for training l_1 robust models. Although (Croce and Hein, 2021) and (Jiang et al., 2023) illustrated their robustness against l_1 attacks, Fast-EG-1 is the current state-of-the-art method (as shown in (Jiang et al., 2023)). Therefore, we chose the l_1 robust model trained by the Fast-EG-1 method for our experiment. In this experiment, we use 1000 images correctly classified by l_1 pre-trained model¹³ on CIFAR-10. These images are evenly distributed across ten classes. To keep consistency with the previous evaluation, we also use a query budget of 500 and compare the accuracy of the robust model under sparse attacks. The results in Table C.8 show that our attack outperforms BRUSLEATTACK across different sparsity levels. Interestingly, l_1 robust models are relatively more robust to sparse attacks than other adversarial training regimes in Table C.7, this could be because l_0 bounded perturbations are enclosed in the l_1 -norm ball.

¹³<https://github.com/IVRL/FastAdvL1>

Table C.8: A robustness comparison (lower \downarrow is stronger) between SPARSE-RS and BRUSLEATTACK against undefended and defended models employing l_1 robust models on CIFAR-10. The attack robustness is measured by the degraded accuracy of models under attacks at different sparsity levels.

Sparsity	Undefended Model		l_1 -Robust Model	
	SPARSE-RS	BRUSLEATTACK	SPARSE-RS	BRUSLEATTACK
0.39%	26.5%	24.2%	86.6%	85.8%
0.78%	7.8%	6.4%	75.8%	74.8%
1.17%	2.5%	2.0%	68.5%	64.8%
1.56%	0.6%	0.6%	59.4%	55.9%

C.6 Reformulate the Optimization Problem

Solving the problem in Equation 5.1 lead to an extremely large search space because of searching colors—float numbers in $[0, 1]$ —for perturbing some pixels. To cope with this problem, we i) reduce the search space by synthesizing a color image $x' \in \{0, 1\}^{c \times w \times h}$ —that is used to define the color for perturbed pixels in the source image (see Appendix C.7), ii) employ a binary matrix $u \in \{0, 1\}^{w \times h}$ to determine positions of perturbed pixels in x .

When selecting a pixel, the colors of all three pixel channels are selected together. Formally, an adversarial instance \tilde{x} can be constructed as follows:

$$\tilde{x} = (1 - u)x + ux' \quad (\text{C.1})$$

Proof of The Problem Reformulation. Given a source image $x \in [0, 1]^{c \times w \times h}$ and a synthetic color image $x' \in \{0, 1\}^{c \times w \times h}$. From Equation C.1, we have the following:

$$\begin{aligned} \tilde{x} &= (1 - u)x + ux' \\ \tilde{x} - (1 - u)x &= ux' \\ u\tilde{x} + (1 - u)\tilde{x} - (1 - u)x &= ux' \\ (1 - u)(\tilde{x} - x) &= u(x' - \tilde{x}) \end{aligned}$$

We consider two cases for each pixel here:

1. If $u_{i,j} = 0$: then $(1 - u_{i,j})(\tilde{x}_{i,j} - x_{i,j}) = 0$, thus $\tilde{x}_{i,j} = x_{i,j}$

2. If $u_{i,j} = 1$: then $u_{i,j}(x'_{i,j} - \tilde{x}_{i,j}) = 0$, thus $\tilde{x}_{i,j} = x'_{i,j}$

Therefore, manipulating binary vector \mathbf{u} is equivalent to manipulating $\tilde{\mathbf{x}}$ according to C.1. Hence, optimizing $L(f(\tilde{\mathbf{x}}), \mathbf{y}^*)$ is equivalent to optimizing $L(f((1 - \mathbf{u})\mathbf{x} + \mathbf{u}\mathbf{x}'), \mathbf{y}^*)$.

C.7 Analysis of Search Space Reformulation and Dimensionality Reduction

Intuitively, sparse attacks aim to search for the positions and color values of these perturbed pixels. For a normalized image, the color value of each channel of a pixel—RGB color value—can be a float number in $[0, 1]$ so the search space is enormous. The perturbation scheme proposed in (Croce et al., 2022) can be adapted to cope with this problem. This perturbation scheme limits the RGB values to a set $\{0, 1\}$ so a pixel has eight possible color codes $\{000, 001, 010, 011, 100, 101, 110, 111\}$ where each digit of a color code denotes a color value of a channel. This scheme may result in noticeable perturbations but does not alter the semantic content of the input. However, this perturbation scheme still results in a large search space because it grows rapidly with respect to the image size. To obtain a more compact search space, we introduce a simple but effective perturbation scheme. In this scheme, we uniformly sample at random a color image $\mathbf{x}' \in \{0, 1\}^{c \times w \times h}$ —*synthetic color image*—to define the color of perturbed pixels in the source image \mathbf{x} . Additionally, we use a binary matrix for selecting some perturbed pixels in \mathbf{x} and apply the matrix to \mathbf{x}' to extract color for these perturbed pixels as presented in Appendix C.6. Because \mathbf{x}' is generated once in advance for each attack and has the same size as \mathbf{x} , the search space is eight times smaller than using the perturbation scheme in (Croce et al., 2022). Surprisingly, our elegant proposal is shown to be incredibly effective, particularly in high-resolution images such as ImageNet.

Synthetic color image. Our attack method does not optimize but pre-specify a synthetic color image \mathbf{x}' by using our proposed random sampling strategy in our algorithm formulation. This synthetic image is generated once, dubbed a one-time synthetic color image, for each attack. We have chosen to generate it once rather than optimizing it because:

- We aim to reduce the dimensionality of the search space to find an adversarial example. Choosing to optimize the color image would lead to a difficult combinatorial optimization problem.
 - Consider what we presented in Section 5.4.1. To solve the combinatorial optimization problem in Equation 5.1, we might search a color value for each channel of each pixel—a float number in $[0,1]$ and this search space is enormous. For instance, if we need to perturb n pixels and the color scale is 2^m , the search space is equivalent to $C_{2^{m \times c \times w \times h}}^{c \times n}$.
 - To alleviate this problem, we reformulated the problem in Equation 5.1 and proposed a search over the subspace $\{0,1\}^{c \times w \times h}$. However, the size of this search space is still large.
 - To further reduce the search space, we construct a fixed search space—a pre-defined synthetic color image $x' \in \{0,1\}^{c \times w \times h}$ for each attack. The search space is now reduced to $C_{w \times h}^n$. It is generated by uniformly selecting the color value for each channel of each pixel from $\{0,1\}$ at random (as presented in Appendix C.7 and C.6).
- In addition, a pre-defined synthetic color image x' —a fixed search space—benefits our Bayesian algorithm. If we keep optimizing the synthetic color image x' , our Bayesian algorithm has to learn and explore a large number of parameters which is equivalent to $C_{2^{m \times c \times w \times h}}^{c \times n}$ and we might not learn useful information fast enough to make the attack progress.
- Perhaps, most interestingly, our attack demonstrates that a solution for the combinatorial optimization problem in Equation 5.1 can be found in a pre-defined and fixed subspace.

Searching for pixels' position and color concurrently. In general, changing the color of the pixels in searches led to significant increases in query budgets. In our approach, we aim to model the influence of each pixel bearing a specific color, probabilistically, and learn the probability model through the historical information collected from pixel manipulations. So, we chose not to first search for pixels' position and search for their color after knowing the position of pixels but we aim to do both simultaneously. In other words, the solution found by our method is a set of pixels with their specific colors.

C.8 Analysis of Synthetic Image Initialization

Table C.9: Target setting. ASR (higher is better) at different sparsity thresholds in the targeted setting. A comprehensive comparison among different strategies of synthetic color image generation to initialize BRUSLEATTACK attack against ResNet18 on CIFAR-10.

Methods	Q=100	Q=200	Q=300	Q=400	Q=500
Sparsity = 1.0%					
Uniform	32.18%	41.68%	48.09 %	52.38%	55.48%
Gaussian	21.29%	29.87%	35.0 %	38.72%	41.53%
Ours	42.32%	54.73%	61.49%	65.33%	68.21%
Sparsity = 2.0%					
Uniform	54.04%	69.08%	76.48%	80.91%	83.76%
Gaussian	40.02%	55.17%	63.2 %	68.58%	72.28%
Ours	66.01%	79.19%	84.84%	88.27%	90.24%
Sparsity = 2.9%					
Uniform	65.82%	80.62%	87.84%	91.39%	93.38%
Gaussian	52.4%	69.91%	78.42 %	83.24%	86.39%
Ours	75.54%	88.22%	92.91%	95.2%	96.59%
Sparsity = 3.9%					
Uniform	73.04%	86.32%	92.33%	95.02%	96.34%
Gaussian	61.0%	77.26%	84.88 %	89.63%	91.94%
Ours	80.44%	91.24%	95.43%	97.4%	98.48%

C.8 Analysis of Synthetic Image Initialization

In this section, we analyze the impact of different schemes including different random distributions, maximizing dissimilarity and low color search space.

Different random distributions. Since the synthetic color images are randomly generated, we can leverage Uniform or Gaussian distribution and our method. Because the input must be within $[0, 1]$, we can sample x' from $\mathcal{U}[0, 1]$ or $\mathcal{N}(\mu, \sigma^2)$ where $\mu = 0.5, \sigma = 0.17$. For our method, we uniformly sample at random a color image $x' \in \{0, 1\}^{c \times w \times h}$. In other words, each channel of a pixel receives a binary value 0 or 1. The results in Table C.9 show that generating a synthetic color image from Uniform distribution is better than Gaussian distribution but it is worse than our simple method. The experiment illustrates that different schemes of generating the synthetic color image at random have different influences on the performance

of BRUSLEATTACK and our proposal outweighs other common approaches across different sparsity levels. Particularly at low query budgets (e.g. up to 300 queries) and low perturbation budgets (e.g. sparsity up to 3%), our proposal outperforms the other two by a large margin. Therefore, the empirical results show our proposed scheme is more effective in obtaining good performance. Most interestingly, as pointed out by HSJA authors (Chen, Jordan and Wainwright, 2020), the question of how best to select an initialization method or in their case initial target image remains an open-ended question worth investigating.

Maximizing dissimilarity. There may be different ways to implement your suggestion of generating a synthetic color image x' that maximize the dissimilarity between the original image x and x' . But to the best of our knowledge, no effective method can generate a random color image x' that maximize its dissimilarity with x .

Our approach to this suggestion is to find the inverted color values of x by creating an inverted image x_{invert} to explore color values different from x . We then find the frequency of these color values (in each R, G, B channel) in x_{invert} . Finally, we generate a synthetic color image x' such that the more frequent color values (in R, G, B channels) in x_{invert} will appear more frequently in x' . By employing the frequency information of color values in x , we can create a synthetic color image x' that is more dissimilar to x . In practice, our implementation is described as follows :

- Yield the inverted image $x_{\text{invert}} = 1 - x$. Note that $x \in [0, 1]^{c \times w \times h}$
- Create a histogram of pixel colors (for each R, G, B channel) to have their frequency in x_{invert} .
- Then we randomly generate a synthetic color image based on the frequency of color values that allows us to maximize the dissimilarity.

The results in Table C.10 show that an approach of maximizing the dissimilarity (using frequency information) yields better performance at low sparsity levels as we discussed in Appendix C.10. However, it does not result in better performance at high levels of sparsity if compared with our proposal.

Low color search space. Instead of reducing the space from 8 color codes to a fixed random one, we consider choosing between 2-4 random colors. That would allow us to search not only in the position space of the pixels but also in their color space without

C.9 BruSLeAttack under Different Random Seeds

Table C.10: ASR comparison between using a synthetic color image uniformly generated at random (our proposal) and maximizing dissimilarity on CIFAR-10.

Sparsity	Our Proposal	Maximizing Dissimilarity
1.0%	68.21%	70.16%
2.0%	90.24%	90.75%
2.9%	96.59%	95.78%
3.9%	98.48%	97.85%

Table C.11: ASR comparison between using a fixed random color search space (our proposal) and two or four random color search spaces on CIFAR-10.

Sparsity	Our Proposal	Two Random Colors	Four Random Colors
1.0%	68.21%	60.11%	57.9%
2.0%	90.24%	78.12%	78.1%
2.9%	96.59%	85.89%	90.67%
3.9%	98.48%	91.23%	95.28%

increasing search space significantly. The results in Table C.11 show that expanding color space leads to larger search space. Consequently, this approach may require more queries to search for a solution and results in low ASR, particularly with a small query budget.

C.9 BruSLeAttack under Different Random Seeds

It is possible that the initial generated by uniformly selecting the color value for each channel of each pixel from $\{0, 1\}$ at random (as presented in Appendix C.7 and Appendix C.6) could impact performance. We investigated this using Monte Carlo experiments. To analyze if our attack is sensitive to our proposed initialization scheme. We generated 10 different synthetic color images (x') for each source image and target class pair. We chose an evaluation set of 1000 source images (evenly distributed across 10 random classes) and used each one and our attack to flip the label to 9 different target classes. So we conducted $(1000 \times 9 \text{ source-image-to-target-class pairs}) \times 10$ (ten because we generated 10 different for each pair) attacks (90K attacks) against ResNet18 on CIFAR-10. We report the min, max, average and standard deviation ASR across the entire evaluation set at different sparsity levels. The results in Table C.12 show that

our method is invariant to the initialization of x' . Therefore, our initialization scheme does not affect the final performance of our attacks. Actually, the more complex task of optimizing x' and devising efficient algorithms to explore the high dimensional search space or the generation of better image synthesizing schemes (initialization schemes) to boost the attack performance leaves interesting works in the future.

Table C.12: ASR (Min, Mean, Max and Standard Deviation) of our attack methods across the entire evaluation set at different sparsity levels with a query budget of 500, with 10 times different initialization of synthetic color image for each attack on CIFAR-10.

Sparsity	ASR (Min)	ASR (Mean)	ASR (Max)	Standard Deviation
1%(10 pixels)	68.14 %	68.36 %	68.66%	0.35
2%(20 pixels)	90.24%	90.76%	91.38%	0.48
2.9%(30 pixels)	96.62%	96.71%	96.78%	0.11
3.9%(40 pixels)	98.17%	98.35%	98.49%	0.13

C.10 Effectiveness of Dissimilarity Map

In this section, we aim to investigate the advantage of using prior knowledge of pixel dissimilarity.

On CIFAR-10. Similarly, we conduct another experiment on an evaluation set which is composed of 1000 correctly classified images (from CIFAR-10) evenly distributed in 10 classes and 9 target classes per image. However, to reduce the burden of computation when studying hyper-parameters, we use a query budget of 500. The results in Table C.13 confirm our observation on ImageNet.

On ImageNet. We conduct an experiment on the same evaluation set of 500 samples from ImageNet used in Section 5.5 and in the targeted setting. The results in Table C.14 show that employing prior knowledge of pixel dissimilarity benefits our attack, particularly at a low percentage of sparsity rather. At a high percentage of sparsity, BRUSLEATTACK adopting prior knowledge only achieves a comparable performance to BRUSLEATTACK without prior knowledge. Notably, at a sparsity of 0.2%, BRUSLEATTACK is slightly worse than SPARSE-RS. Nonetheless, employing prior knowledge of pixel dissimilarity improve the performance of BRUSLEATTACK and makes it consistently outweigh SPARSE-RS.

C.11 Hyper-parameters, Initialization and Computation Resources

Table C.13: ASR comparison between with and without using Dissimilarity Map on CIFAR-10.

Sparsity	With Dissimilarity Map	Without Dissimilarity Map
1.0%	68.21%	67.16%
2.0%	90.24%	89.42%
2.9%	96.59%	95.96%
3.9%	98.48%	97.92%

Table C.14: ASR at different sparsity thresholds and queries (higher is better) for a targeted setting. A comparison between SPARSE-RS, BRUSLEATTACK and BRUSLEATTACK with prior on an evaluation set of 500 pairs of an image and a target class on ImageNet.

Methods	Q=2000	Q=4000	Q=6000	Q=8000	Q=10000
Sparsity = 0.2%					
SPARSE-RS	9.4%	20.6%	29.6 %	33.4%	38.4%
BRUSLEATTACK (without prior)	8.8%	19.6%	27.4 %	34.4%	38.2%
BRUSLEATTACK	12%	23.6%	31.6%	36.6%	40.4%
Sparsity = 0.4%					
SPARSE-RS	23.6%	48.4%	63.0%	72.6%	78.8%
BRUSLEATTACK (without prior)	30.2%	53.4%	64.4%	73.0%	78.6%
BRUSLEATTACK	33.2%	54.2%	66.8%	76%	82.4%
Sparsity = 0.6%					
SPARSE-RS	29.6%	57.6%	73.2%	85.8%	92.0%
BRUSLEATTACK (without prior)	43.6%	71.6%	85.0%	91.8%	94.6%
BRUSLEATTACK	45.4%	75.6%	87.4%	91.8%	94.6%

C.11 Hyper-parameters, Initialization and Computation Resources

All experiments in this study are performed on two RTX TITAN GPU ($2 \times 24\text{GB}$) and four RTX A6000 GPU ($4 \times 48\text{GB}$). We summarize all hyper-parameters used for BRUSLEATTACK on the evaluation sets from CIFAR-10, STL-10 and ImageNet as shown in Table C.15. Notably, only the initial changing rate λ_0 is adjusted for different resolution datasets e.g. STL-10 or ImageNet; thus, our method can be easily adopted for different vision tasks. Additionally, to realize an attack, we randomly synthesize

a color image x' for each attack. At the initialization step, BRUSLEATTACK randomly creates 10 candidate solutions and chooses the best.

Table C.15: Hyper-parameters setting in our experiments

Parameters	CIFAR-10		STL-10		ImageNet	
	Untargeted	Targeted	Untargeted	Targeted	Untargeted	Targeted
λ_0	0.3	0.15	0.3	0.15	0.3	0.05
α^{prior}	1	1	1	1	1	1
m_1	0.24	0.24	0.24	0.24	0.24	0.24
m_2	0.997	0.997	0.997	0.997	0.997	0.997

C.12 Hyper-Parameters Study

In this section, we conduct comprehensive experiments to study the impacts and the choice of hyper-parameters used in our algorithm. The experiments in this section are mainly conducted on CIFAR-10. For λ_0 , we conduct an additional experiment on ImageNet.

C.12.1 The Impact of m_1, m_2

In this experiment, we use the same evaluation set on CIFAR-10 mentioned above. To investigate the impact of m_1 , we set $m_2 = 0.997$ and change $m_1 = 0.2, 0.24, 0.28$. Likewise, we set $m_1 = 0.24$ and change $m_2 = 0.993, 0.997, 0.999$ to study m_2 . The results in Table C.16 show that BRUSLEATTACK achieves the best results with $m_1 = 0.24$ and $m_2 = 0.997$.

C.12.2 The Impact of λ_0

On CIFAR-10. Similarly, we conduct another experiment on the same evaluation set which is composed of 1000 correctly classified images (from CIFAR-10) as described above. We use the same query budget of 500. We use $m_1 = 0.24$ and $m_2 = 0.997$ and change $\lambda_0 = 0.15$ to study the impact of λ_0 . Our results in Table C.17 show that BRUSLEATTACK achieves the best results with $\lambda_0 = 0.15$.

On ImageNet. We use 500 random pairs of an image and a target class from ImageNet in a targeted setting. For the hyper-parameter study, we tune the initial changing rate at

C.12.2 The Impact of λ_0

Table C.16: ASR of BRUSLEATTACK with different values of m_1, m_2 on CIFAR-10.

Sparsity	Fixed $m_2 = 0.997$			Fixed $m_1 = 0.24$		
	$m_1 = 0.2$	$m_1 = 0.24$	$m_1 = 0.28$	$m_2 = 0.993$	$m_2 = 0.997$	$m_2 = 0.999$
1.0%	67.32%	68.21%	67.48%	67.34%	68.21%	67.21%
2.0%	88.67%	90.24%	88.94%	89.64%	90.24%	89.12%
2.9%	95.37%	96.59%	95.54%	96.25%	96.59%	95.82%
3.9%	97.24%	98.48%	97.68%	97.59%	98.48%	96.21%

Table C.17: ASR of BRUSLEATTACK with different values of λ_0 on CIFAR-10.

Sparsity	$\lambda_0 = 0.1$	$\lambda_0 = 0.15$	$\lambda_0 = 0.2$
1.0%	68.05%	68.21%	68.12%
2.0%	89.38%	90.24%	88.33%
2.9%	96.15%	96.59%	95.56%
3.9%	98.16%	98.48%	97.08%

Table C.18: ASR at different sparsity levels and queries (higher is better) in a targeted setting. A comparison between $\lambda_0 = 0.03$ and $\lambda_0 = 0.05$ on a set of 500 pairs of an image and a target class on ImageNet.

Initial changing rate	Q=2000	Q=4000	Q=6000	Q=8000	Q=10000
Sparsity = 0.2%					
$\lambda_0 = 0.03$	10.2%	22.6%	29.2 %	35.6%	41.4%
$\lambda_0 = 0.05$	12%	23.6%	31.6%	36.6%	40.4%
Sparsity = 0.4%					
$\lambda_0 = 0.03$	31%	53.6%	65.6%	74.2%	80%
$\lambda_0 = 0.05$	33.2%	54.2%	66.8%	76%	82.4%
Sparsity = 0.6%					
$\lambda_0 = 0.03$	45.4%	75.4%	84.6%	89.8%	92.8%
$\lambda_0 = 0.05$	45.4%	75.6%	87.4%	91.8%	94.6%

a time. Figure C.4 shows that with different initial changing rates λ_0 , BRUSLEATTACK obtains the best results when λ_0 is small such as 0.03 or 0.05. However, at a small sparsity budget, $\lambda_0 = 0.03$ often achieves lower ASR than $\lambda_0 = 0.05$ as shown in Table C.18 because it requires more queries to make changes and move towards a solution. Consequently, λ_0 should not be too small. If increasing λ_0 , BRUSLEATTACK reaches its highest ASR slower than using small λ_0 . Hence, the initial changing rate has an impact on the overall performance of BRUSLEATTACK.

C.12.3 The Choice of α^{prior}

In this section, we discuss the choice of α^{prior} and provide an analysis on the convergence time.

- $\alpha^{prior} = 1$ ($\alpha_i = 1$ where $i \in [1, k]$). In our proposal, we draw multiple pixels (equivalent to multiple elements in a binary matrix u) from the Categorical distribution (K categories) parameterized by $\theta = [\theta_1, \theta_2, \dots, \theta_K]$. When initializing an attack, we have no prior knowledge of the influence of each pixel that is higher or lower than other pixels on the model's decision so it is sensible to assume all pixels have a similar influence. Consequently, all pixels should have the same chance to be selected for perturbation (to be manipulated). To this end, the Categorical distribution where multiple pixels are drawn from should be a uniform distribution and $\theta_1 = \theta_2 = \dots = \theta_K = \frac{1}{K}$.

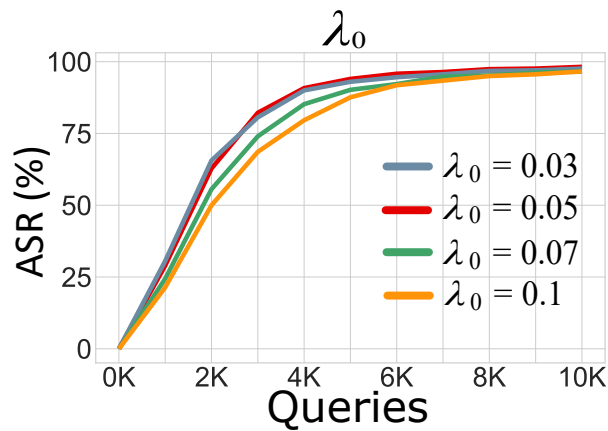


Figure C.4. ASR versus model queries on ImageNet. BRUSLEATTACK against ResNet-50 with sparsity of 1.0 % in a targeted setting to show the impacts of different hyper-parameters on BRUSLEATTACK.

C.13 BruSLeAttack With Different Schedulers

We note that Dirichlet distribution is the conjugate prior distribution of the Categorical distribution. If the Categorical distribution is a uniform distribution, the Dirichlet distribution is also a uniform distribution. In probability and statistics, Dirichlet distribution (parameterized by a concentration vector $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_K]$, each α_i represents the i -th element where K is the total number of elements) is equivalent to a uniform distribution over all of the elements when $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_K] = [1, 1, \dots, 1]$. In other words, there is no prior knowledge favoring one element over another. Therefore, we choose $\alpha^{prior} = \mathbf{1}$.

- $\alpha^{prior} < 1$ ($\alpha_i < 1$ where $i \in [1, k]$). We have $\alpha^{posterior} = \alpha^{prior} + s^{(t)}$ and $s^{(t)} = (a^{(t)} + z)/(n^{(t)} + z) - 1$. So we have $\alpha^{posterior} = \alpha^{prior} + (a^{(t)} + z)/(n^{(t)} + z) - 1$. Because $(a^{(t)} + z)/(n^{(t)} + z) \leq 1$, we cannot choose $\alpha^{prior} < 1$ to ensure that the parameters controlling the Dirichlet distribution are always positive ($\alpha^{posterior} > 0$).
- $\alpha^{prior} > 1$ ($\alpha_i > 1$ where $i \in [1, k]$). Since $\alpha^{posterior} = \alpha^{prior} + (a^{(t)} + z)/(n^{(t)} + z) - 1$ and $0 < (a^{(t)} + z)/(n^{(t)} + z) \leq 1$, if $\alpha^{prior} \gg 1$, in the first few iterations, $\alpha^{posterior}$ almost remains unchanged so the algorithm will not converge. If $\alpha^{prior} > 1$, the farther from 1 α^{prior} is, the more subtle the $\alpha^{posterior}$ changes. Now, the update $(a^{(t)} + z)/(n^{(t)} + z)$ needs more iterations (times) to significantly influence $\alpha^{posterior}$. In other words, the proposed method requires more time to learn the Dirichlet distribution (update $\alpha^{posterior}$). Thus, the convergence time will be longer. Consequently, the larger α_i is, the longer the convergence time is.

C.13 BruSLeAttack With Different Schedulers

We carry out a comprehensive experiment to examine the impact of different schedulers including cosine annealing and step decay. In this experiment, we use the same evaluation set with 1000 images from CIFAR-10 evenly distributed in 10 classes and 9 target classes per image and we use the same query budget (500 queries). The results in Table C.19 show the ASR at different sparsity levels. These results show that our proposed scheduler slightly outperforms all other schedulers. Based on the results, Step Decay or Cosine Annealing schedulers can be a good alternative.

Table C.19: ASR comparison between using a Power Step Decay (our proposal) and other schedulers on CIFAR-10.

Sparsity	Our Proposal	Step Decay	Cosine Annealing
1.0%	68.21%	68.11%	68.02%
2.0%	90.24%	89.34%	89.15%
2.9%	96.59%	96.12%	95.89%
3.9%	98.48%	98.26%	98.18%

C.14 Evaluation Protocol

In this section, we present the evaluation protocol used in this research.

1. In the targeted attack settings.

- SparseRS (Croce et al., 2022) evaluation with ImageNet: Selected 500 source images. But each source image class was flipped to only one random target class using the attack. So that is a total of 500 source-image-to-target class attacks. This evaluation protocol may select the same target class to attack in the 500 attacks conducted. Thus, this could lead to potential biases in the results because some classes may be easier to attack than others.
- To avoid the problem, in the targeted attack setting, we followed the evaluation protocol used in (Vo, Abbasnejad and Ranasinghe, 2022). Essentially, we flip the label of the source image to several targeted classes, this can help address potential biases caused by relatively easier classes getting selected multiple times for a target class.
- Our evaluation with ImageNet: We randomly selected 200 correctly classified source images evenly distributed among 200 random classes. But, in contrast to SparseRS, we selected 5 random target classes to attack for each source image. In total we did $200 \times 5 = 1000$ source-image-to-target class attacks on ImageNet for targeted attacks.

2. In the untargeted attack setting (attacks against defended models), we conducted 500 attacks (similar to SparseRS). We randomly selected 500 correctly classified test images from 500 different classes for attacks.

C.15 Attack Against Google Cloud Vision






3. Further, our unique and exhaustive testing with CIFAR-10 and STL-10 corroborates ImageNet results given the significant amount of resources it takes to attack the high-resolution ImageNet (224×224) models.
 - For STL-10 we conducted 60,093 attacks against each deep learning model (6,677 of all 10,000 images in the test set which are correctly classified versus 9 other classes as target classes for each source image). We used every single test set image in STL-10 (96×96) in our attacks to mount the exhaustive evaluation where no image from the test set was left out.
 - For CIFAR-10 (32×32) we conducted 9,000 attacks against each deep learning model (1000 random images correctly classified versus the 9 other classes as target classes for each source image).
4. For evaluations against a real-world system (GCV) in the significantly more difficult targeted setting (not the untargeted setting), we provide new benchmarks for attack demonstration because we provide a comparison between BRUSLEATTACK and the previous attack, SPARSE-RS. To make it clear, we provide a brief comparison as follows:
 - Other related past studies (dense attacks)([Ilyas et al., 2018](#); [Guo et al., 2019](#)), showcase an attack against a real-world system but uses 10 attacks. While ([Ilyas et al., 2018](#)) illustrated only one successful example when carrying out an attack against Google Cloud Vision.
 - Importantly, we did not simply use our method only, as in ([Ilyas et al., 2018](#); [Guo et al., 2019](#)) but demonstrated a comparison between BRUSLEATTACK and SPARSE-RS. In practice, we used 10 samples for each attack, so there are 20 attacks.

In general, our evaluation protocol is much stronger than the one used in previous studies. We evaluate on three different datasets CIFAR-10, STL-10 (not evaluated in prior attacks) and ImageNet with ResNet-50, ResNet-50 (SIN), Visitation Transformer (not evaluated in prior attacks).

C.15 Attack Against Google Cloud Vision

Table C.20 and Figure C.5, Table C.21 and Figure C.6 show our attack against real-world system Google Cloud Vision API.

Table C.20: Demonstration of sparse attacks against GCV in targeted settings. BRUSLEATTACK is able to successfully yield adversarial instances for all five examples with less queries than SPARSE-RS. Especially, for the example of Mushroom, SPARSE-RS fails to attack GCV within a budget of 5000 queries. Demonstration on GCV API (online platform) is shown in Figure C.5.

Examples					
No Attack	Car	Flower	Fire Truck	Vehicle	Mushroom
BRUSLEATTACK	Window (1.8K Queries)	Yellow Pepper (99 Queries)	Window (328 Queries)	Window (1.83K Queries)	Landscape (490 Queries)
SPARSE-RS	Window (4.66K Queries)	Yellow Pepper (211 Queries)	Window (395 Queries)	Window (3.3K Queries)	Mushroom (>5K Queries)

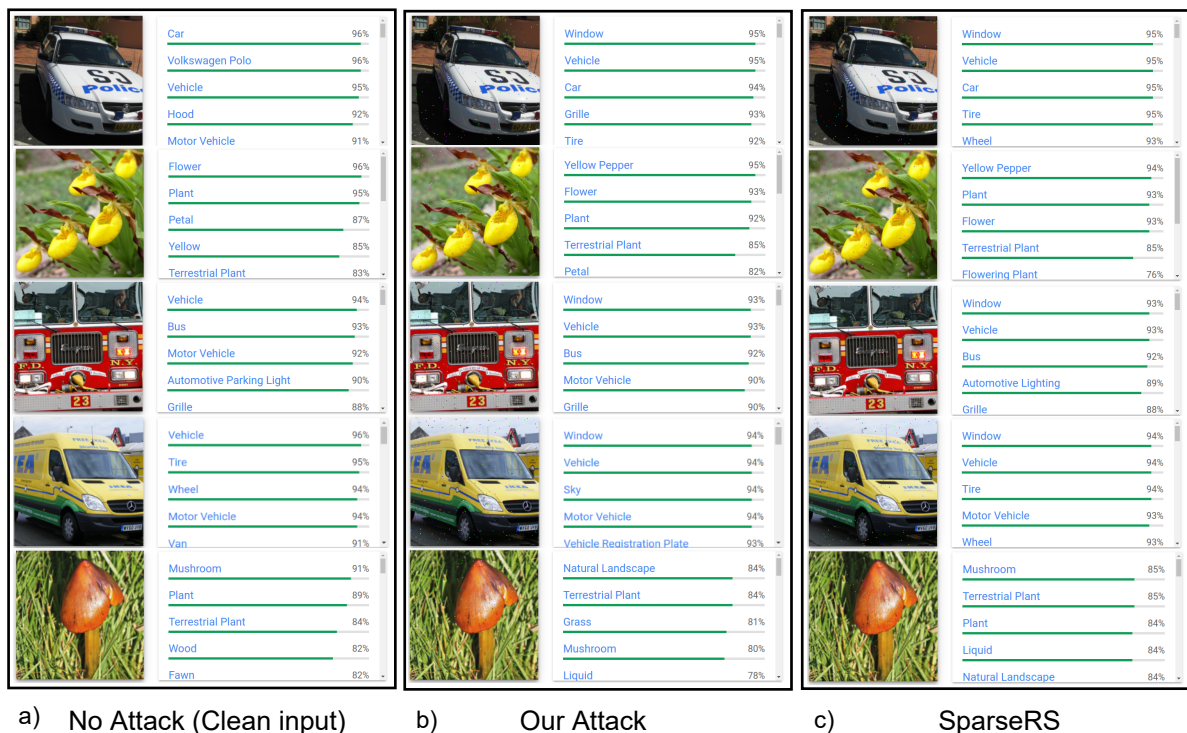





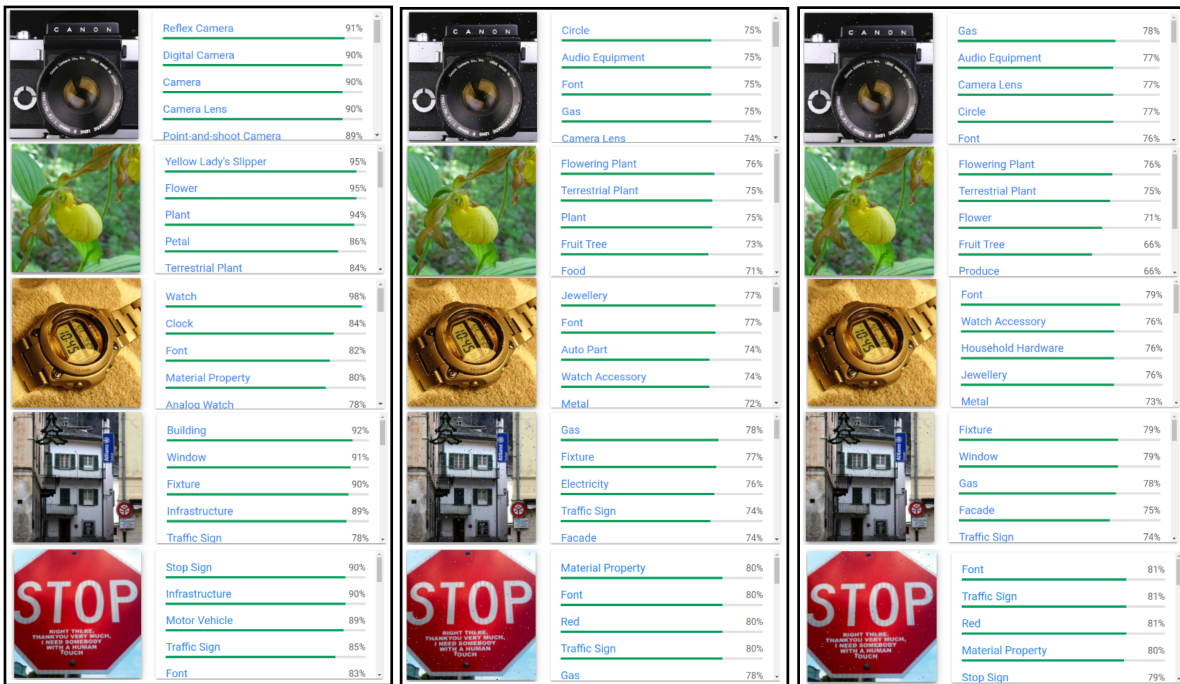


Figure C.5. a) demonstrates results for clean image (no attack) predicted by Google Cloud Vision (GCV). b) shows the predictions from GCV for adversarial examples crafted successfully by BRUSLEATTACK with less than 3,000 queries and sparsity of 0.05 %. c) shows the results from GCV for adversarial examples crafted by SPARSE-RS with the same sparsity. But SPARSE-RS needs more queries than BRUSLEATTACK to successfully yield adversarial images or fail to attack with query budget up to 5,000 as shown in Table C.20.

C.15 Attack Against Google Cloud Vision

Table C.21: Demonstration of sparse attacks against GCV in targeted settings. BRUSLEATTACK is able to successfully yield adversarial instances for all five examples with fewer queries than SPARSE-RS. Especially, for the example of Mushroom, SPARSE-RS fails to attack GCV within a budget of 5000 queries. Demonstration on GCV API (online platform) is shown in Figure C.5.

Examples					
No Attack	Reflex Camera	Y.L.Slipper	Watch	Building	Stop Sign
BRUSLEATTACK	Circle (3.8K Queries)	Flowering Plant (899 Queries)	Jewellery (2.9K Queries)	Gas (983 Queries)	Material P (2.77K Queries)
SPARSE-RS	Gas (>5K Queries)	Flowring Plant (988 Queries)	Font (>5K Queries)	Fixture (>5K Queries)	Font (>5K Queries)



a) No Attack (Clean input)

b) Our Attack

c) SparseRS

Figure C.6. a) demonstrates results for clean image (no attack) predicted by Google Cloud Vision (GCV). b) shows the predictions from GCV for adversarial examples crafted successfully by BRUSLEATTACK with less than 3,000 queries and sparsity of 0.05 %. c) shows the results from GCV for adversarial examples crafted by SPARSE-RS with the same sparsity. But SPARSE-RS needs more queries than BRUSLEATTACK to successfully yield adversarial images or fail to attack with query budget up to 5,000 as shown in Table C.21.

C.16 Visualizations of Dissimilarity Maps and Sparse Adversarial Examples

In this section, we illustrate:

- Sparse adversarial examples, sparse perturbation crafted by our methods versus salient region produced by GradCAM method (Selvaraju et al., 2017) or attention map produced by a ViT model (Dosovitskiy et al., 2021).
- Sparse adversarial examples crafted by BRUSLEATTACK when attacking ResNet-50, ResNet-50 (SIN) and Vision Transformer.
- Dissimilarity Map produced from a pair of a source and a synthetic color images.

Figure C.7 and C.8 illustrate sparse adversarial examples and sparse perturbation of images from ImageNet in targeted and untargeted settings. In targeted settings, we use a query budget of 10K, while in untargeted settings, we set a query limit of 5K. We use GradCAM and Attention Map from ViT to demonstrate salient and attention regions. The sparse perturbation δ is the absolute difference between source images and their sparse adversarial. Formally, sparse perturbations can be defined as $\delta = |x - \tilde{x}|$.

The results show that for ResNet-50, the solutions found do not need to perturb salient regions on an image to mislead the models (both targeted and untargeted attacks). Attacks with ViT models in untargeted settings also lead to a similar observation. Interestingly, for some images e.g. a snake or a goldfinch in Figure C.7, we observe that a set of perturbed pixels yielded by our method is more concentrated in the attention region of ViT. This seems to indicate some adversarial solutions achieve their objective by degrading the performance of a ViT. This is perhaps not an unexpected observation, given the importance of attention mechanisms to transformer models.

C.16 Visualizations of Dissimilarity Maps and Sparse Adversarial Examples

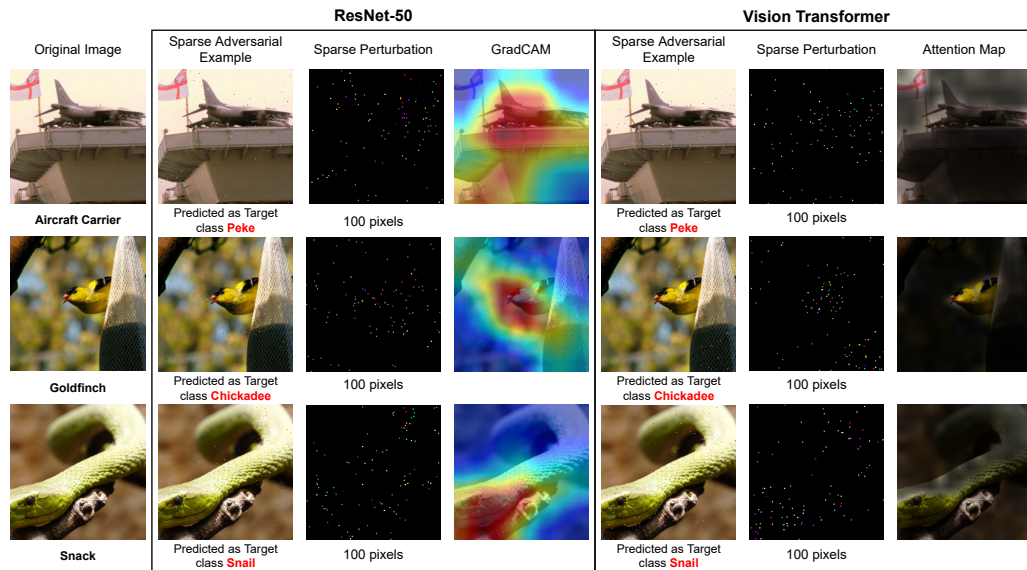


Figure C.7. Targeted Attack. Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 10K queries. In the image of sparse perturbation, each pixel is zoomed in 9 times ($9\times$) to make them more visible.

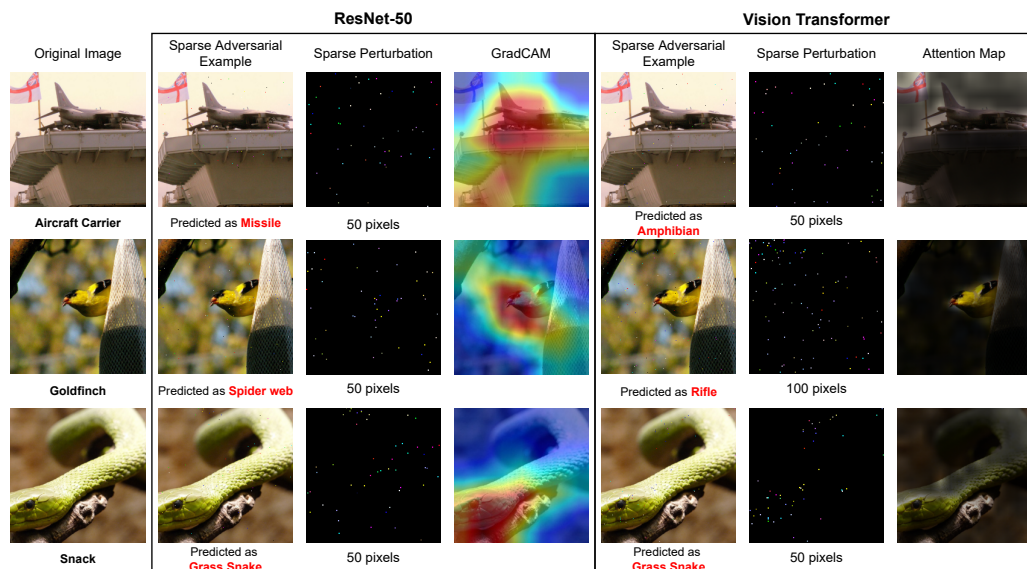


Figure C.8. Untargeted Attack. Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 5K queries. In the image of sparse perturbation, each pixel is zoomed in 9 times ($9\times$) to make them more visible.

Figure C.9 and C.10 demonstrate some examples of adversarial examples yielded by BRUSLEATTACK when attacking different deep learning models (ResNet-50, ResNet-50 (SIN) and Vision Transformer) in targeted settings produced using a 10K query budget.

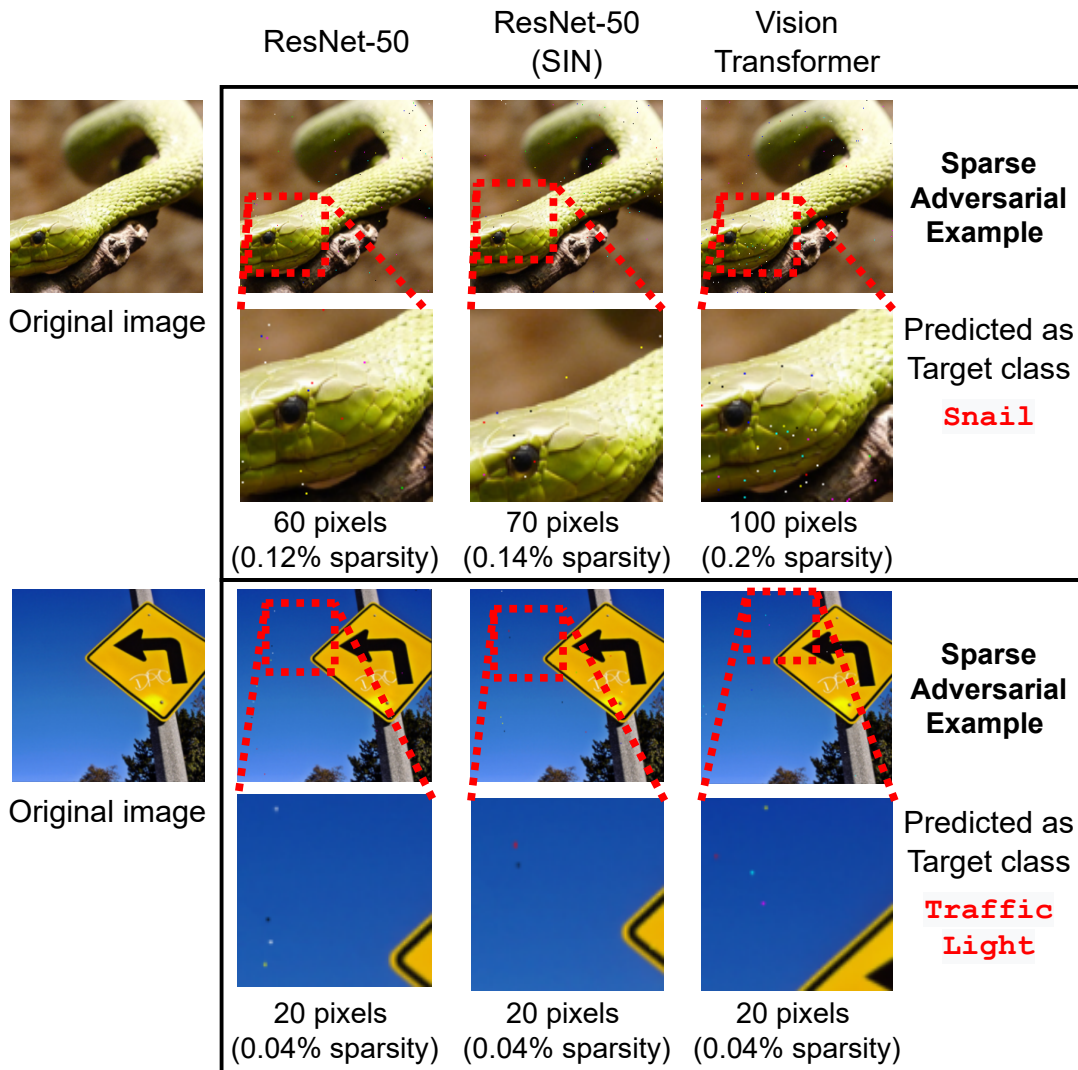


Figure C.9. Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 5000 queries.

C.16 Visualizations of Dissimilarity Maps and Sparse Adversarial Examples

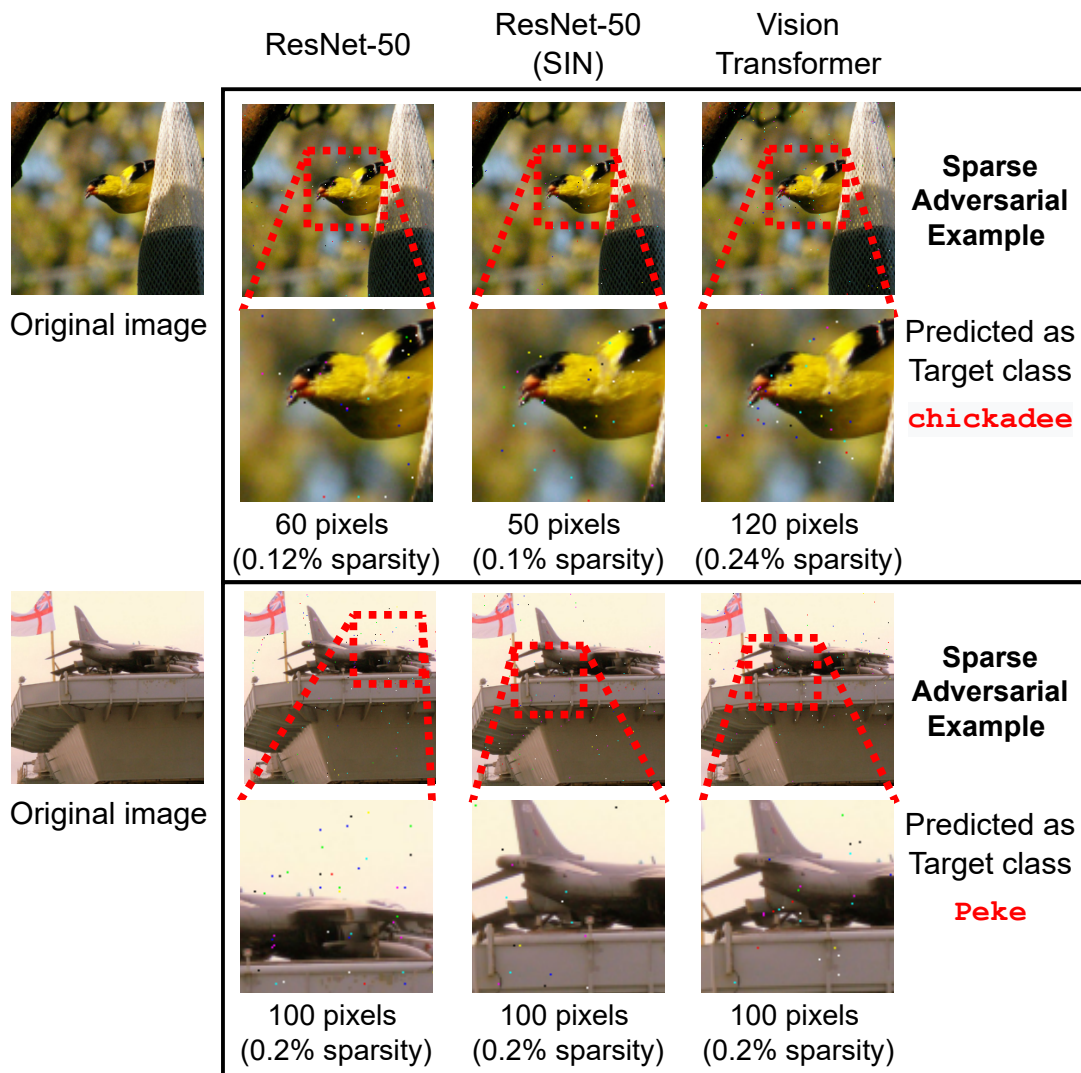


Figure C.10. Visualization of Adversarial examples crafted by BRUSLEATTACK with a budget of 5000 queries.

Figure C.11 illustrates some examples of Dissimilarity Map yielded by a source image and a synthetic color image.

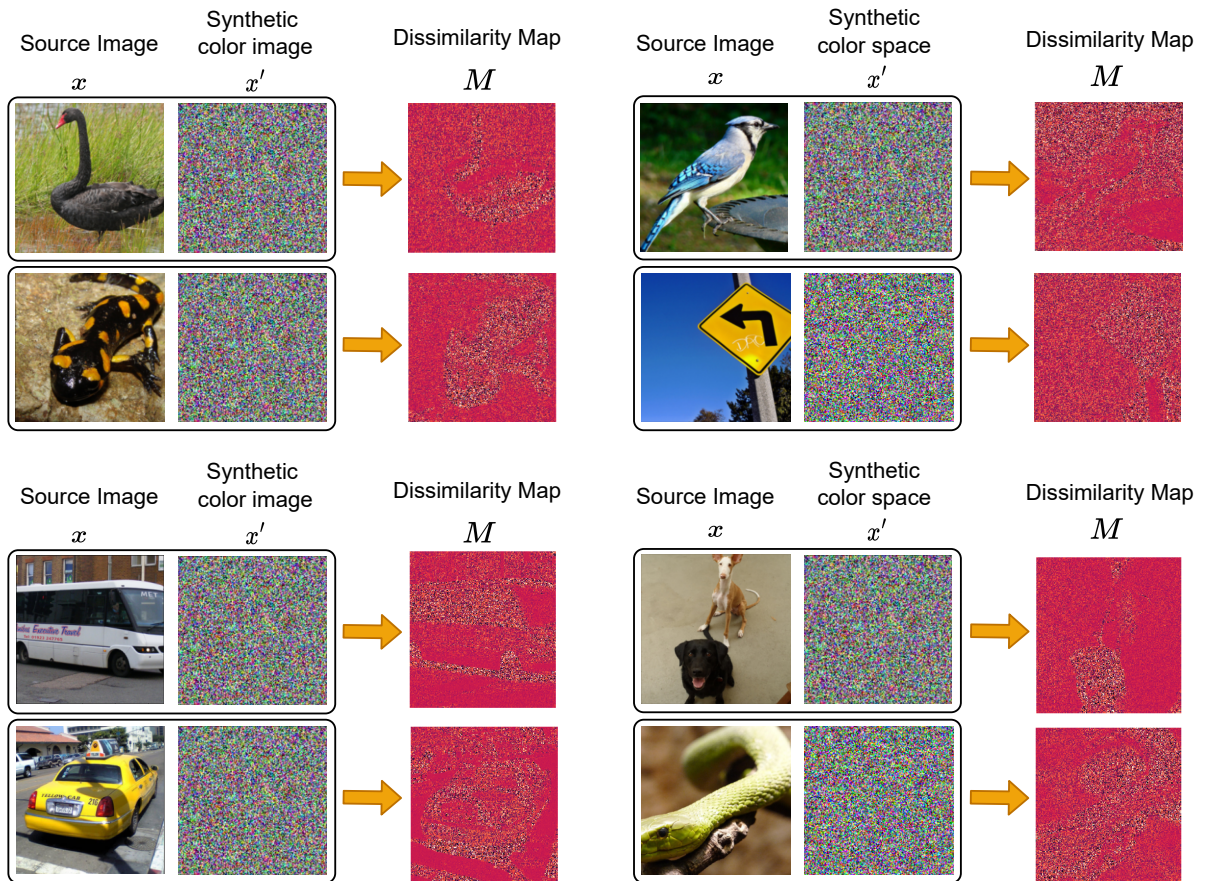


Figure C.11. Visualization of Dissimilarity Maps between a source image and a synthetic color image.

Appendix D

Chapter 6 Appendix

Table D.1: A robustness comparison (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on MNIST. For the evaluation of different diversity-promotion methods, we train a set of 10 models and randomly select a subset of a different number of models.

Random	Methods	Distortion = 0	0.8	1.6	2.4	3.2	4.0
1	ENSEMBLES	100%	98.4%	91.9%	87.1%	82.0%	74.6%
	DIVDIS(Adapted)	100%	98.3%	92.9%	86.6%	78.9%	70.9%
	DIVREG(Adapted)	100%	96.1%	88.5%	80.9%	72.5%	66.5%
	Ours	100%	98.7%	94.4%	89.3%	86.0%	80.0%
3	ENSEMBLES	100%	99.9%	98.7%	91.6%	78.7%	68.0%
	DIVDIS(Adapted)	100%	99.8%	97.6%	87.8%	76.9%	62.5%
	DIVREG(Adapted)	100%	99.8%	85.5%	88.1%	77.6%	69.1%
	Ours	100%	99.9%	99.6%	95.4%	83.3%	70.3%
5	ENSEMBLES	100%	100%	98.6%	90.5%	74.4%	55.8%
	DIVDIS(Adapted)	100%	99.8%	96.3%	85.0%	71.6%	57.0%
	DIVREG(Adapted)	100%	99.9%	94.4%	83.8%	72.1%	60.0%
	Ours	100%	100%	99.4%	95.3%	81.1%	63.8%
8	ENSEMBLES	100%	100%	98.5%	90.3%	73.5%	54.2%
	DIVDIS(Adapted)	100%	99.5%	94.3%	82.6%	67.1%	53.5%
	DIVREG(Adapted)	100%	99.9%	93.2%	80.7%	72.3%	59.7%
	Ours	100%	100%	99.4%	95.3%	81.1%	63.8%

D.1 Diverse Set of Models Against Black-box Attacks on MNIST

In this section, we demonstrate additional results for training a set of 10 and 20 models using ENSEMBLES, DIVDIS(Adapted), DIVREG(Adapted) and our proposed method.

D.2 Accuracy of Non-defense versus Defense Models

Table D.2: A robustness comparison (higher \uparrow is stronger) between our proposed method and other methods against SQUAREATTACK on MNIST. For the evaluation of different diversity-promotion methods, we train a set of 20 models and randomly select a subset of a different number of models.

Random	Methods	Distortion = 0	0.8	1.6	2.4	3.2	4.0
1	ENSEMBLES	100%	99.2%	96.1%	91.6%	85.1%	75.4%
	DIVDIS(Adapted)	100%	99.0%	95.5%	91.1%	82.7%	74.5%
	DIVREG(Adapted)	100%	96.5%	91.8%	83.7%	76.8%	68.6%
	Ours	100%	99.4%	97.3%	94.2%	90.2%	83.5%
3	ENSEMBLES	100%	100%	99.5%	95.4%	85.7%	70.8%
	DIVDIS(Adapted)	100%	100%	97.8%	89.9%	79.3%	66.5%
	DIVREG(Adapted)	100%	100%	97.0%	91.1%	83.9%	76.5%
	Ours	100%	100%	99.8%	98.1%	90.5%	78.8%
5	ENSEMBLES	100%	100%	99.3%	93.2%	77.5%	62.8%
	DIVDIS(Adapted)	100%	99.8%	97.5%	90.6%	76.8%	60.4%
	DIVREG(Adapted)	100%	99.9%	97.8%	90.6%	76.8%	60.4%
	Ours	100%	100%	99.4%	94.8%	85.1%	70.0%
10	ENSEMBLES	100%	99.9%	98.2%	86.5%	67.1%	46.0%
	DIVDIS(Adapted)	100%	99.7%	93.1%	79.1%	65.0%	50.1%
	DIVREG(Adapted)	100%	99.9%	94.0%	83.0%	72.9%	62.6%
	Ours	100%	100%	99.5%	95.0%	76.8%	60.0%

Table D.4 and Table D.3 show clean accuracy under different model training and random selection strategies. For robustness evaluation and comparison, we choose different settings with different sizes of model subsets (*i.e.* 1, 3, 5, 8 or 10) and show the results in Tables D.1 and D.2—these results provide further evidence to demonstrate that our proposed method is more robust than other diversity promotion methods across different distortions and settings.

D.2 Accuracy of Non-defense versus Defense Models

In this section, we show the clean accuracy achieved by different models and defended models on different datasets as shown in Table D.4. For models employing diversity-promotion methods, we demonstrate the accuracy obtained by these models in Table D.3.

Table D.3: Clean accuracy achieved by different defended models employing diversity-promotion techniques on different datasets with a different random number of models.

MNIST					
Quantity	Random	Ensembles	DivDis	DivReg	Ours
10	1	99.5%	99.5%	97.5%	99.4%
	3	99.6%	99.6%	99.2%	99.6%
	5	99.7%	99.6%	99.4%	99.6%
	8	99.6%	99.6%	99.5%	99.6%
20	1	99.5%	99.5%	97.6%	99.0%
	3	99.6%	99.6%	99.3%	99.5%
	5	99.6%	99.6%	99.4%	99.5%
	10	99.7%	99.6%	99.6%	99.6%
40	1	99.3%	99.5%	98.6%	98.7%
	3	99.5%	99.5%	99.4%	99.2%
	5	99.5%	99.6%	99.5%	99.4%
	20	99.6%	99.7%	99.6%	99.6%
	30	99.6%	99.7%	99.6%	99.6%
CIFAR-10					
Quantity	Random	Ensembles	DivDis	DivReg	Ours
10	1	92.2%	90.5%	91.8%	87.9%
	3	93.8%	92.5%	93.9%	91.1%
	5	94.0%	93.3%	94.3%	92.3%
	8	94.4%	93.5%	94.5%	92.5%
STL-10					
Quantity	Random	Ensembles	DivDis	DivReg	Ours
10	5	91.6%	90.2%	89.7%	88.2%

D.2 Accuracy of Non-defense versus Defense Models

Table D.4: Clean accuracy achieved by non-defense models and defended models on different datasets. All methods make a prediction with a single model except ensembles (using all individual models for inference).

Dataset	Single Model	Ensembles	Dropout	RND	RBC
MNIST	99.64%	99.72%	99.49%	98.59%	99.62%
CIFAR-10	92.09%	94.76%	91.93%	87.63%	92.02%
STL-10	90.39%	92.15%	90.01%	86.38%	90.34%

Bibliography

- Aithal, M.B. and Li, X., 2022. Boundary defense against black-box adversarial attacks. *International conference on pattern recognition (icpr)*. (Cited on page 23.)
- Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.J. and Srivastava, M., 2019. Genattack: Practical black-box attacks with gradient-free optimization. *The genetic and evolutionary computation conference (gecco)*. (Cited on pages 19, 22, and 71.)
- Andrei, P. and Ion, N., 2015. Random coordinate descent methods for l_0 regularized convex optimization. *Ieee transactions on automatic control* [Online], 60(7), pp.1811–1824. Available from: <https://doi.org/10.1109/TAC.2015.2390551>. (Cited on page 66.)
- Andriushchenko, M., Croce, F., Flammarion, N. and Hein, M., 2020. Square Attack: a query-efficient black-box adversarial attack via random search. *European conference on computer vision (eccv)*. (Cited on pages 89, 119, and 120.)
- Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M. and Khan, M.K., 2018. Medical image analysis using convolutional neural networks: A review. *Journal of medical systems*, 42(11). (Cited on page 2.)
- Athalye, A., Carlini, N. and Wagner, D., 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International conference on machine learning (icml)*. (Cited on pages 7, 58, 64, 83, 106, and 110.)
- Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T. and Veit, A., 2021. Understanding robustness of transformers for image classification. *International conference on computer vision (iccv)*. (Cited on page 64.)
- Brendel, W., Rauber, J. and Bethge, M., 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *International conference on learning recognition(iclr)*. (Cited on pages 21, 31, 32, 33, 36, 44, 45, 48, 67, and 153.)
- Brunner, T., Diehl, F., Le, M.T. and Knoll, A., 2019. Guessing smart: Biased sampling for efficient black-box adversarial attacks. (Cited on pages 32, 48, and 98.)

BIBLIOGRAPHY

- Byun, J., Go, H. and Kim, C., 2022. On the effectiveness of small input noise for defending against query-based black-box attacks. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. (Cited on pages 7 and 110.)
- Cao, X. and Gong, N.Z., 2017. Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification. (Cited on pages 23, 58, 113, and 119.)
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S., 2020. End-to-end object detection with transformers. *European conference on computer vision (ECCV)*. (Cited on page 64.)
- Carlini, N. and Wagner, D., 2017. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*. (Cited on pages 19, 21, 22, 58, 86, 89, and 154.)
- Chen, C., Seff, A., Kornhauser, A. and Xiao, J., 2015. Deepdriving: Learning affordance for direct perception in autonomous driving [Online]. *2015 IEEE International Conference on Computer Vision (ICCV)*. pp.2722–2730. Available from: <https://doi.org/10.1109/ICCV.2015.312>. (Cited on page 2.)
- Chen, J. and Gu, Q., 2020. RayS: A Ray Searching Method for Hard-label Adversarial Attack. *Knowledge Discovery and Data Mining (KDD)*. (Cited on page 153.)
- Chen, J., Jordan, M.I. and Wainwright, M.J., 2020. Hopskipjumpattack: A query-efficient decision-based attack. *IEEE Symposium on Security and Privacy (SSP)*. (Cited on pages 6, 21, 28, 31, 33, 35, 36, 44, 45, 49, 61, 67, 76, 153, 164, and 175.)
- 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. *ACM Workshop on Artificial Intelligence and Security (AISec)*. pp.15–26. (Cited on pages 21, 32, and 89.)
- Chen, S., Huang, Z., Tao, Q., Wu, Y., Xie, C. and Huang, X., 2022a. Adversarial Attack on Attackers: Post-Process to Mitigate Black-Box Score-Based Query Attacks. (Cited on page 23.)
- Chen, Y., Zhijian, X., Zhanyuan, Y., Xiaoyu, J. and Wenyuan, X., 2022b. Rolling colors: Adversarial laser exploits against traffic light recognition. *31st USENIX Security Symposium (USENIX Security 22)*. (Cited on page 135.)

- Cheng, M., Le, T., Chen, P.Y., Yi, J., Zhang, H. and Hsieh, C.J., 2019a. Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. *International conference on learning recognition(iclr)*. (Cited on page 31.)
- Cheng, M., Singh, S., Chen, P., Chen, P.Y., Liu, S. and Hsieh, C.J., 2020. Sign-opt: A query-efficient hard-label adversarial attack. *International conference on learning recognition(iclr)*. (Cited on pages 19, 28, 32, 33, 36, 44, 45, 47, 48, 49, 58, 67, 76, 119, 147, and 153.)
- Cheng, S., Dong, Y., Pang, T., Su, H. and Zhu, J., 2019b. Improving Black-box Adversarial Attacks with a Transfer-based Prior. (Cited on pages 32, 33, 44, 47, 48, and 147.)
- Coates, A., Lee, H. and Ng, A.Y., 2011. Analysis of Single Layer Networks in Unsupervised Feature Learning. *International conference on artificial intelligence and statistics(aistats)*. (Cited on pages 25, 99, and 119.)
- Cohen, J., Rosenfeld, E. and Kolter, Z., 2019. Certified adversarial robustness via randomized smoothing. *International conference on machine learning (icml)*. (Cited on pages 7 and 110.)
- Cordonnier, J.B., Loukas, A. and Jaggi, M., 2020. On the relationship between self-attention and convolutional layers. *International conference on learning recognition(iclr)*. (Cited on page 64.)
- Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P. and Hein, M., 2020. Robustbench: a standardized adversarial robustness benchmark. *arxiv preprint arxiv:2010.09670*. (Cited on page 169.)
- Croce, F., Andriushchenko, M., Singh, N.D., Flammarion, N. and Hein, M., 2022. Sparse-RS: A Versatile Framework for Query-Efficient Sparse Black-Box Adversarial Attacks. *Association for the advancement of artificial intelligence (aaai)*. (Cited on pages 22, 86, 87, 90, 91, 92, 99, 103, 164, 172, and 183.)
- Croce, F. and Hein, M., 2019. Sparse and imperceivable adversarial attacks. *International conference on computer vision (iccv)*. (Cited on pages 22, 66, 75, 83, 87, 89, 90, 103, and 164.)
- Croce, F. and Hein, M., 2021. Mind the box: l1-apgd for sparse adversarial attacks on image classifiers. *International conference on machine learning*. (Cited on page 170.)

BIBLIOGRAPHY

- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database [Online]. *2009 ieee conference on computer vision and pattern recognition*. pp.248–255. Available from: <https://doi.org/10.1109/CVPR.2009.5206848>. (Cited on pages 25, 44, 75, and 99.)
- Dhillon, G.S., Azizzadenesheli, K., Lipton, Z.C., Bernstein, J., Kossaifi, J., Khanna, A. and Anandkumar, A., 2018. Stochastic activation pruning for robust adversarial defense. *International conference on learning recognition(iclr)*. (Cited on page 7.)
- Diba, A., Fayyaz, M., Sharma, V., Arzani, M.M., Yousefzadeh, R., Gall, J. and Gool, L.V., 2018. Spatio-temporal channel correlation networks for action classification. *European conference on computer vision*. (Cited on page 135.)
- Dietterich, T.G., 2000. Ensemble methods in machine learning. *Multiple classifier systems*. Springer Berlin Heidelberg. (Cited on pages 113 and 115.)
- Doan, B.G., Abbasnejad, E.M., Shi, J.Q. and Ranasinghe, D.C., 2022a. Bayesian learning with information gain provably bounds risk for a robust adversarial defense. *International conference on machine learning (icml)*. (Cited on pages 7, 110, 116, and 117.)
- Doan, B.G., Xue, M., Ma, S., Abbasnejad, E. and C. Ranasinghe, D., 2022b. TnT attacks! universal naturalistic adversarial patches against deep neural network systems. *Ieee transactions on information forensics and security*. (Cited on page 135.)
- Doan, B.G., Yang, S., Montague, P., De Vel, O., Abraham, T., Camtepe, S., Kanhere, S.S., Abbasnejad, E. and Ranasinghe, D.C., 2023. Feature-space bayesian adversarial learning improved malware detector robustness. (Cited on page 117.)
- Dong, X., Chen, D., Bao, J., Qin, C., Yuan, L., Zhang, W., Yu, N. and Chen, D., 2020. GreedyFool: Distortion-Aware Sparse Adversarial Attack. *Neural information processing systems (neurips)*. (Cited on pages 6, 65, 68, 86, 87, 89, 103, 155, 164, 165, and 168.)
- Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X. and Li, J., 2018. Boosting adversarial attacks with momentum. *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)*. (Cited on page 89.)
- Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T. and Zhu, J., 2019. Efficient Decision-based Black-box Adversarial Attacks on Face Recognition. *Computer vision and pattern recognition(cvpr)*. (Cited on page 67.)

- Dong, Y., Yang, X., Deng, Z., Pang, T., Xiao, Z., Su, H. and Zhu, J., 2021. Black-box detection of backdoor attacks with limited information and data. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Cited on pages 168 and 169.)
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR)*. (Cited on pages 18, 64, 75, 87, 100, and 187.)
- 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. *Proceedings of the 12th USENIX Conference on Offensive Technologies*. (Cited on page 135.)
- Fan, Y., Wu, B., Li, T., Yong, Z., Li, M., Li, Z. and Yang, Y., 2020. Sparse Adversarial Attack via Perturbation Factorization. *European Conference on Computer Vision (ECCV)*. (Cited on pages 87, 89, and 154.)
- Feichtenhofer, C., Fan, H., Malik, J. and He, K., 2019. Slowfast networks for video recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Cited on page 135.)
- Fort, S., Hu, H. and Lakshminarayanan, B., 2020. Deep ensembles: A loss landscape perspective [Online]. Available from: <https://arxiv.org/abs/1912.02757>. (Cited on page 118.)
- Gal, Y. and Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 36th International Conference on Machine Learning (ICML)*. (Cited on page 119.)
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. and Brendel, W., 2019. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*. (Cited on pages 99 and 102.)
- Georgioudakis, M.S. and Plevris, V., 2020. A comparative study of differential evolution variants in constrained structural optimization. *Frontiers in Built Environment* [Online], 6, p.102. Available from: <https://doi.org/10.3389/fbuilt.2020.00102>. (Cited on page 74.)

BIBLIOGRAPHY

- Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep Learning. *Mit press*. (Cited on page 17.)
- Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. *International conference on learning recognition(iclr)*. (Cited on pages 7, 19, 21, 58, 89, and 110.)
- Guo, C., Frank, J.S. and Weinberger, K.Q., 2019. Low Frequency Adversarial Perturbation. (Cited on page 106.)
- Guo, C., Gardner, J., You, Y., Wilson, A.G. and Weinberger, K., 2019. Simple Black-box Adversarial Attacks. *International conference on machine learning (icml)*. (Cited on page 184.)
- He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Computer vision and pattern recognition (cvpr)*. p.770–778. (Cited on pages 45, 75, 99, and 119.)
- Huang, B. and Ling, H., 2022. Spaa: Stealthy projector-based adversarial attacks on deep image classifiers. *2022 ieee on conference virtual reality and 3d user interfaces (vr)*. (Cited on page 135.)
- Ilyas, A., Engstrom, L., Athalye, A. and Lin, J., 2018. Black-box adversarial attacks with limited queries and information. *International conference on machine learning (icml)*. (Cited on pages 21, 22, 64, 106, 168, and 184.)
- Ilyas, A., Engstrom, L. and Madry, A., 2019. Prior convictions: Black-box adversarial attacks with bandits and priors. *International conference on learning recognition(iclr)*. (Cited on page 89.)
- Jan, S.T., Messou, J., Lin, Y.C., Huang, J.B. and Wang, G., 2019. Connecting the digital and physical world: Improving the robustness of adversarial attacks. *Proceedings of the aaai conference on artificial intelligence*. (Cited on page 135.)
- Jia, W., Lu, Z., Zhang, H., Liu, Z., Wang, J. and Qu, G., 2022. Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems. *The network and distributed system security symposium (ndss)*. (Cited on page 135.)

- Jiang, K., Chen, Z., Huang, T., Wang, J., Yang, D., Li, B., Wang, Y. and Zhang, W., 2023. Efficient decision-based black-box patch attacks on video recognition. *Arxiv*. (Cited on page 170.)
- Krizhevsky, A., Nair, V. and Hinton, G., n.d. *Cifar-10 (canadian institute for advanced research)* [Online]. Available from: <http://www.cs.toronto.edu/~kriz/cifar.html>. (Cited on pages 24, 44, 75, 99, and 119.)
- Krogh, A. and Vedelsby, J., 1994. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems (nips)*. (Cited on pages 113 and 115.)
- Lakshminarayanan, B., Pritzel, A. and Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Proceedings of the 31st international conference on neural information processing systems (nips)*. (Cited on pages 118 and 119.)
- LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep Learning. *Nature*. (Cited on page 17.)
- Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the ieee* [Online], 86(11), pp.2278–2324. Available from: <https://doi.org/10.1109/5.726791>. (Cited on pages 17, 24, and 119.)
- Lee, D., Moon, S., Lee, J. and Song, H.O., 2022. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. *International conference on machine learning (icml)*. (Cited on page 136.)
- Lee, Y., Yao, H. and Finn, C., 2023. Diversify and Disambiguate: Out-of-Distribution Robustness via Disagreement. *International conference on learning recognition(iclr)*. (Cited on pages 118 and 119.)
- Li, H., Li, L., Xu, X., Zhang, X., Yang, S. and Li, B., 2021a. Nonlinear Projection Based Gradient Estimation for Query Efficient Blackbox Attacks. *Artificial intelligence and statistics (aistats)*. (Cited on pages 22, 67, and 153.)
- Li, H., Xu, X., Zhang, X., Yang, S. and Li, B., 2020. QEBA: Query-Efficient Boundary-Based Blackbox Attack. *Computer vision and pattern recognition(cvpr)*. (Cited on pages 67, 153, and 164.)

BIBLIOGRAPHY

- Li, S., Aich, A., Zhu, S., Asif, M.S., Song, C., Roy-Chowdhury, A.K. and Krishnamurthy, S.V., 2021b. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. *Neural information processing systems*. (Cited on page 135.)
- Liu, A., Liu, X., Fan, J., Ma, Y., Zhang, A., Xie, H. and Tao, D., 2019a. Perceptual-sensitive gan for generating adversarial patches. (Cited on page 135.)
- Liu, Q. and Wang, D., 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Neural information processing systems*. (Cited on page 116.)
- Liu, S. and Deng, W., 2015. Very deep convolutional neural network based image classification using small training sample size. *2015 3rd iapr asian conference on pattern recognition (acpr)*. (Cited on page 119.)
- Liu, X., Cheng, M., Zhang, H. and Hsieh, C.J., 2018a. Towards robust neural networks via random self-ensemble. *Proceedings of the european conference on computer vision (eccv)*. Cham: Springer International Publishing, pp.381–397. (Cited on page 110.)
- Liu, X., Li, Y., Wu, C. and Hsieh, C.J., 2019b. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *International conference on learning recognition(iclr)*. (Cited on pages 89 and 113.)
- Liu, Z., Liu, Q., Liu, T., Wang, Y. and Wen, W., 2018b. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. *Ieee/cvf conference on computer vision and pattern recognition (cvpr)*. (Cited on page 23.)
- Logan, E., Andrew, I., Hadi, S., Shibani, S. and Dimitris, T., 2019. Robustness (python library) [Online]. Available from: <https://github.com/MadryLab/robustness>. (Cited on page 100.)
- Lovisotto, G., Turner, H., Sluganovic, I., Strohmeier, M. and Martinovic, I., 2020. Slap: Improving physical adversarial examples with short-lived adversarial perturbations. *Usenix security symposium*. (Cited on page 135.)
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. *International conference on learning recognition(iclr)*. (Cited on pages 19, 21, 58, 66, 83, 89, 105, and 164.)

- Marcel, S. and Rodriguez, Y., 2010. Torchvision the machine-vision package of torch. *Proceedings of the 18th acm international conference on multimedia* [Online], p.1485–1488. Available from: <https://doi.org/10.1145/1873951.1874254>. (Cited on pages 45, 75, and 99.)
- Meng, Y., Jianhai, S., Jason, O. and Pooyan, J., 2021. Ensembles of many diverse weak defenses can be strong: Defending deep neural networks against adversarial attacks. (Cited on page 110.)
- Modas, A., Moosavi-Dezfooli, S.M. and Frossard, P., 2019. Sparsefool: a few pixels make a big difference. *Computer vision and pattern recognition (cvpr)*. (Cited on pages 6, 22, 65, 66, 68, 71, 86, 87, 89, 103, 155, 165, and 168.)
- Narodytska, N. and Kasiviswanathan, S., 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. *Computer vision and pattern recognition (cvpr) workshop*. (Cited on page 90.)
- Naseer, M., Ranasinghe, K., Khan, S.H., Hayat, M., Khan, F.S. and Yang, M.H., 2021. Intriguing properties of vision transformers. *Neural information processing systems (neurips)*. (Cited on page 102.)
- Nguyen, D.L., Arora, S.S., Wu, Y. and Yang, H., 2020. Adversarial light projection attacks on face recognition systems: A feasibility study. *2020 ieee/cvf conference on computer vision and pattern recognition workshops (cvprw)*. (Cited on page 135.)
- Pang, R., Zhang, X., Ji, S., Luo, X. and Wang, T., 2020. Advmind: Inferring adversary intent of black-box attacks. *Proceedings of the 26th acm sigkdd international conference on knowledge discovery and data mining*. (Cited on pages 24 and 113.)
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A., 2017. Practical black-box attacks against machine learning. *Acm asia conference on computer and communications security (asia ccs)*. (Cited on pages 22 and 66.)
- Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B. and Swami, A., 2016a. The limitations of deep learning in adversarial settings. *Security and privacy, 2016 ieee european symposium*, pp.372–387. (Cited on pages 21 and 89.)
- Papernot, N., McDaniel, P., Wu, X., Jha, S. and Swami, A., 2016b. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. *2016 ieee symposium on security and privacy (sp)*. (Cited on page 58.)

BIBLIOGRAPHY

- Paul, S. and Chen, P.Y., 2022. Vision transformers are robust learners. *Association for the advancement of artificial intelligence (aaai)*. (Cited on page 102.)
- Qin, Z., Fan, Y., Zha, H. and Wu, B., 2021. Random noise defense against query-based black-box attacks. *Advances in neural information processing systems (nips)*. vol. 34, pp.7650–7663. (Cited on pages 7, 24, 106, 110, 113, and 119.)
- Qiu, H., Custode, L.L. and Iacca, G., 2021. Black-box adversarial attacks using evolution strategies. *The genetic and evolutionary computation conference (gecco)*. (Cited on page 71.)
- Rakin, A.S., He, Z. and Fan, D., 2019. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. *Computer vision and pattern recognition (cvpr)*. (Cited on pages 7 and 110.)
- Ramachandran, P., Parmar, N., Vaswani, A., Bello, I.B., Levskaya, A. and Shlens, J., 2019. Stand-alone self-attention in vision model. *Neural information processing systems (neurips)*. (Cited on page 64.)
- Schott, L., Rauber, J., Bethge, M. and Brendel, W., 2019. Towards the first adversarially robust neural network model on mnist. *International conference on learning recognition(iclr)*. (Cited on pages 22, 65, 67, 75, 76, 89, and 103.)
- Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. (Cited on pages 55 and 187.)
- Sen, S., Ravindran, B. and Raghunathan, A., 2020. Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks. *International conference on learning recognition(iclr)*. (Cited on pages 110 and 113.)
- Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G. and Goldstein, T., 2019. *Adversarial training for free!* (Cited on pages 7 and 110.)
- Shao, R., Shi, Z., Yi, J., Chen, P.Y. and Hsieh, C.J., 2021. On the adversarial robustness of visual transformers. *Transactions on machine learning research (tmlr)* [Online]. Available from: <https://arxiv.org/abs/2103.15670>. (Cited on page 64.)
- Shukla, S.N., Sahu, A.K., Willmott, D. and Kolter, J.Z., 2021. Simple and Efficient Hard Label Black-box Adversarial Attacks in Low Query Budget Regimes. *Knowledge discovery and data mining (kdd)*. (Cited on pages 64 and 89.)

- Srivastava, N., Geoffrey, E.H., Alex, K., Ilya, S. and Ruslan, R.S., 2014. Dropout: A simple way to prevent neural networks from overfitting. *Annual computer security applications conference (acsac)*. (Cited on page 119.)
- Storn, R. and Price, K., 1997. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of global optimization* [Online], p.341–359. Available from: <https://doi.org/10.1023/A:1008202821328>. (Cited on page 73.)
- Su, J., Vargas, D.V. and Sakurai, K., 2019. One pixel attack for fooling deep neural networks. *Ieee transactions on evolutionary computation* [Online], 23, pp.828–841. Available from: <https://doi.org/10.1109/TEVC.2019.2890858>. (Cited on pages 67, 69, and 165.)
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2014. Intriguing properties of neural networks. *International conference on learning recognition(iclr)*. (Cited on pages 19 and 20.)
- Taori, R., Kamsetty, A., Chu, B. and Vemuri, N., 2019. Targeted adversarial examples for black box audio systems. *2019 ieee security and privacy workshops (spw)*. (Cited on page 136.)
- Teney, D., Abbasnejad, E., Lucey, S. and Hengel, A.v.d., 2022. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. *Computer vision and pattern recognition (cvpr)*. (Cited on pages 118 and 119.)
- Thys, S., Ranst, W.V. and Goedeme, T., 2019. Fooling automated surveillance cameras: Adversarial patches to attack person detection. *2019 ieee/cvf conference on computer vision and pattern recognition workshops (cvprw)*. (Cited on page 135.)
- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. and Jegou, H., 2021. Training data-efficient image transformers & distillation through attention. *International conference on machine learning (icml)*. (Cited on pages 64 and 87.)
- Tramer, F., Carlini, N., Brendel, W. and Madry, A., 2020. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems (nips)*. (Cited on pages 7 and 110.)

BIBLIOGRAPHY

- Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D. and McDaniel, P., 2018. Ensemble adversarial training: Attacks and defenses. *International conference on learning recognition(iclr)*. (Cited on pages 23, 58, 110, and 113.)
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A. and Madry, A., 2019. Robustness may be at odds with accuracy. *International conference on learning representations, ICLR 2019*. (Cited on pages 7, 24, and 110.)
- Tu, C.C., Ting, P., Chen, P.Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.J. and Cheng, S.M., 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. (Cited on page 89.)
- Vo, Q.V., Abbasnejad, E. and Ranasinghe, D., 2022. Query efficient decision based sparse attacks against black-box deep learning models. *International conference on learning recognition(iclr)*. (Cited on page 183.)
- Wan, X., Nguyen, V., Ha, H., Ru, B., Lu, C. and Osborne, M.A., 2021. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. *Proceedings of the 38th international conference on machine learning*. (Cited on pages 166 and 167.)
- Wang, D. and Liu, Q., 2019. Nonlinear stein variational gradient descent for learning diversified mixture models. *Proceedings of the 36th international conference on machine learning (icml)*. (Cited on page 116.)
- Wang, D., Yao, W., Jiang, T., Tang, G. and Chen, X., 2022. A survey on physical adversarial attack in computer vision. *Arxiv*. (Cited on page 135.)
- Wang, H. and Wang, Y., 2022. Self-Ensemble Adversarial Training for Improved Robustness. *International conference on learning recognition(iclr)*. (Cited on pages 23 and 113.)
- Wei, Z., Chen, J., Wei, X., Jiang, L., Chua, T., Zhou, F. and Jiang, Y., 2020. Heuristic black-box adversarial attacks on video recognition models. *The thirty-fourth AAAI conference on artificial intelligence, AAAI*. (Cited on page 135.)
- Wen, Y., Tran, D. and Ba, J., 2020. Batchensemble: An alternative approach to efficient ensemble and lifelong learning. *International conference on learning recognition(iclr)*. (Cited on page 118.)

- Wierstra, D., Schaul, T., Peters, J. and Schmidhuber, J., 2008. Natural evolution strategies. *2008 ieee congress on evolutionary computation (ieee world congress on computational intelligence)*. (Cited on page 168.)
- Wong, E., Schmidt, F.R. and Kolter, J.Z., 2019. Wasserstein adversarial examples via projected sinkhorn iterations. *Proceedings of the 36th international conference on machine learning ICML*. (Cited on page 89.)
- Xiang, C. and Mittal, P., 2021. Detectorguard: Provably securing object detectors against localized patch hiding attacks. *Proceedings of the 2021 acm sigsac conference on computer and communications security*. (Cited on page 135.)
- Xie, C., Wang, J., Zhang, Z., Ren, Z. and Yuille, A., 2018. Mitigating adversarial effects through randomization. *International conference on learning recognition(iclr)*. (Cited on page 7.)
- Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A. and He, K., 2019. Feature Denoising for Improving Adversarial Robustness. *Computer vision and pattern recognition (cvpr)*. (Cited on pages 7, 23, 110, and 113.)
- Xu, K., Liu, S., Zhao, P., Chen, P.Y., Zhang, H., Fan, Q., Erdogmus, D., Wang, Y. and Lin, X., 2019. Structured Adversarial Attack: Towards General Implementation and Better Interpretability. *International conference on learning recognition(iclr)*. (Cited on page 21.)
- Xu, Q., Tao, G., Cheng, S., Tan, L. and Zhang, X., 2020. Towards feature space adversarial attack. (Cited on page 89.)
- Xu, W., Evans, D. and Qi, Y., 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. *The network and distributed system security symposium (ndss)*. (Cited on page 23.)
- Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*. (Cited on page 18.)
- Yang, C., Kortylewski, A., Xie, C., Cao, Y. and Yuille, A., 2020a. Patchattack: A black-box texture-based attack with reinforcement learning. *Computer vision – eccv 2020*. (Cited on page 135.)

BIBLIOGRAPHY

- Yang, Y.Y., Rashtchian, C., Zhang, H., Salakhutdinov, R. and Chaudhuri, K., 2020b. A closer look at accuracy vs. robustness. *Proceedings of the 34th international conference on neural information processing systems*. (Cited on pages 7 and 110.)
- Zhan, Y., Fu, Y., Huang, L., Guo, J., Shi, H., Song, H. and Hu, C., 2023. Cube-evo: A query-efficient black-box attack on video classification system. *Ieee transactions on reliability*. (Cited on page 135.)
- Zhang, A., Lipton, Z.C., Li, M. and Smola, A.J., 2021a. Dive into deep learning. *arxiv preprint arxiv:2106.11342*. (Cited on page 17.)
- Zhang, D., Zhang, H., Courville, A., Bengio, Y., Ravikumar, P. and Suggala, A.S., 2022. Building robust ensembles via margin boosting. *Proceedings of the 39th international conference on machine learning (icml)*. (Cited on pages 23, 110, and 113.)
- Zhang, H., Cheng, M. and Hsieh, C.J., 2019. Enhancing certifiable robustness via a deep model ensemble [Online]. 1910.14655, Available from: <https://arxiv.org/abs/1910.14655>. (Cited on page 113.)
- Zhang, H. and Wang, J., 2019. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in neural information processing systems (nips)*. (Cited on pages 7, 23, 110, and 113.)
- Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E. and Jordan, M.I., 2019. Theoretically principled trade-off between robustness and accuracy. *Proceedings of the 36th international conference on machine learning (icml)*. (Cited on pages 7 and 110.)
- Zhang, J., Li, H., Zhang, X., Yang, S. and Li, B., 2021b. Progressive-Scale Boundary Blackbox Attack via Projective Gradient Estimation. *International conference on machine learning (icml)*. (Cited on pages 22, 67, and 153.)
- Zhao, P., Liu, S., Chen, P., Hoang, N., Xu, K., Kailkhura, B. and Lin, X., 2019. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Cited on page 90.)
- Zhu, M., Chen, T. and Wang, Z., 2021. Sparse and Imperceptible Adversarial Attack via a Homotopy Algorithm. *International conference on machine learning (icml)*. (Cited on page 89.)

Biography

Viet Quoc Vo received his M.Sc. degree from the Royal Melbourne Institute of Technology in 2014. He was a senior process engineer at Intel Products Vietnam from 2014-2019. In 2019, he was awarded a Postgraduate Research Scholarship to pursue his doctoral degree at the School of Computer and Mathematical Sciences, the University of Adelaide under the supervision of Associate Professor Damith Chinthana Ranasinghe and Doctor Ehsan Abbasnejad. His research interests include Artificial Intelligence Security and Safety and Trustworthy and Reliable Machine Learning.



Viet Quoc Vo
quocviet.vo@adelaide.edu.au