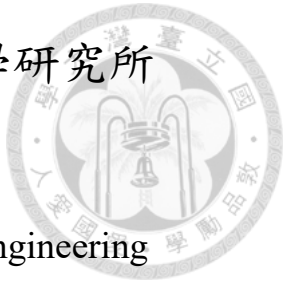


國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

透過多樣化的權重修剪提升目標攻擊的可轉移性

Enhancing Targeted Attack Transferability via Diversified Weight Pruning

王竑睿

Hung-Jui Wang

指導教授: 陳尚澤 博士

Advisor: Shang-Tse Chen Ph.D.

中華民國 112 年 6 月

June, 2023

doi:10.6342/NTU202301130

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

透過多樣化的權重修剪提升目標攻擊的可轉移性

Enhancing Targeted Attack Transferability via Diversified Weight Pruning

本論文係王竑睿君（學號 R10922061）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 112 年 6 月 12 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 12 June 2023 have examined a Master's thesis entitled above presented by HUNG-JUI,WANG (student ID: R10922061) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

陳尚澤 曹昱

（指導教授 Advisor）

陳祝嵩

系主任/所長 Director: 洪士灝

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

Acknowledgements

We would like to express our sincere gratitude to all those who have contributed to the completion of our paper.

First and foremost, we extend our deepest appreciation to our supervisor, Dr. Shang-Tse Chen, for his invaluable guidance, support, and expertise throughout the entire research process. His insightful feedback and continuous encouragement have been instrumental in shaping the direction and quality of this work.

We would also like to acknowledge the support and resources provided by Artificial Intelligence Security Lab of National Taiwan University. The access to computational infrastructure and the research facilities have significantly facilitated our experiments and analysis.

We would also like to thank our colleagues and fellow researchers who have provided valuable discussions, suggestions, and feedback, enhancing the quality of our work. Their intellectual contributions have greatly influenced the ideas and concepts presented in this paper.

Lastly, we express our deep appreciation to the entire research community of adversarial machine learning. The body of knowledge, open-source tools, and datasets that have been made available by researchers and practitioners worldwide have been crucial

doi:10.6342/NTU202301130

in shaping our understanding and enabling advancements in the field.

While efforts have been made to acknowledge everyone involved, we apologize if any individual or group has been unintentionally omitted. We sincerely thank all those who have contributed directly or indirectly to this research project.

doi:10.6342/NTU202301130

摘要

惡意攻擊者可以透過添加微小的擾動生成目標性對抗例，促使神經網路產生特定的錯誤輸出。在跨模型的遷移性下，即使無法直接存取神經網路模型的參數，模型一樣可能受到對抗例的攻擊。現今研究提出以集成式方法來生成對抗例以增加遷移性。為了更加提升遷移性，模型增強法會增加參與集成式方法的模型數量。然而，現存的模型增強法只在無目標性攻擊的設定上進行。在本作中，我們提出多樣化權重修剪，一個新穎的模型增強法，來產生目標性攻擊。相較於以往的研究，多樣化權重修剪會保護模型中必要的權重，並同時確保修剪後模型的多樣性。我們將在實驗中呈現此對目標性攻擊的重要性。我們在 ImageNet 兼容資料集上，提供了更具挑戰性的設定下的實驗結果：分別是遷移到經過對抗訓練的模型、非捲積神經網路模型以及 Google 雲端電腦視覺服務。結果顯示我們提出的多樣化權重修剪分別能在三個設定下，基於最新方法，提升目標性攻擊成功率 10.1%, 6.6%, 以及 7.0%。我們將會在投稿接受後開放程式碼。

關鍵字：對抗式攻擊、神經網路修剪、電腦視覺與圖型識別

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

Abstract

Malicious attackers can generate targeted adversarial examples by imposing tiny noises, forcing neural networks to produce specific incorrect outputs. With cross-model transferability, network models remain vulnerable even in black-box settings. Recent studies have shown the effectiveness of ensemble-based methods in generating transferable adversarial examples. To further enhance transferability, model augmentation methods aim to produce more networks participating in the ensemble. However, existing model augmentation methods are only proven effective in untargeted attacks. In this work, we propose Diversified Weight Pruning (DWP), a novel model augmentation technique for generating transferable targeted attacks. DWP leverages the weight pruning method commonly used in model compression. Compared with prior work, DWP protects necessary connections and ensures the diversity of the pruned models simultaneously, which we show are crucial for targeted transferability. Experiments on the ImageNet-compatible dataset under various and more challenging scenarios confirm the effectiveness: transfer-

doi:10.6342/NTU202301130

ring to adversarially trained models, Non-CNN architectures, and Google Cloud Vision. The results show that our proposed DWP improves the targeted attack success rates with up to 10.1%, 6.6%, and 7.0% on the combination of state-of-the-art methods, respectively. The source code will be made available after acceptance.

Keywords: Adversarial Attack, Network Pruning, Computer Vision and Pattern Recognition

doi:10.6342/NTU202301130

Contents

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

List of Figures

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

List of Tables

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

Denotation

Res-50	50-layer Residual Convolutional Network
Den-121	121-layer Densely Connected Convolutional Network
VGG16	Visual Geometry Group 16-layer Network
Inc-v3	Inception Network Version-3
ViT-S-16-224	Vision Transformer Small patch size 16* 16 image size 224* 224
ViT-B-16-224	Vision Transformer Base patch size 16* 16 image size 224* 224
Swin-S-224	Swin Transformer Small image size 224* 224
Swin-B-224	Swin Transformer Base image size 224* 224
MLP-Mixer	Multi-Layer Perceptrons Mixer
ResMLP	Residual Multi-Layer Perceptrons
gMLP	Gated Multi-Layer Perceptrons
NI	Nesterov Iterative Attack

doi:10.6342/NTU202301130

SI	Scale Invariant Attack
TI	Translation Invariant Attack
DI	Input Diversity Attack
GN	Ghost Network
DSNE	Dual Stage Network Erosion
DWP	Diversified Weight Pruning
IG	Integrated Gradients
SVRE	Stochastic Variance Reduced Ensemble
VT	Variance Tuning
GradCAMs	Gradient-weighted Class Activation Mapping

doi:10.6342/NTU202301130

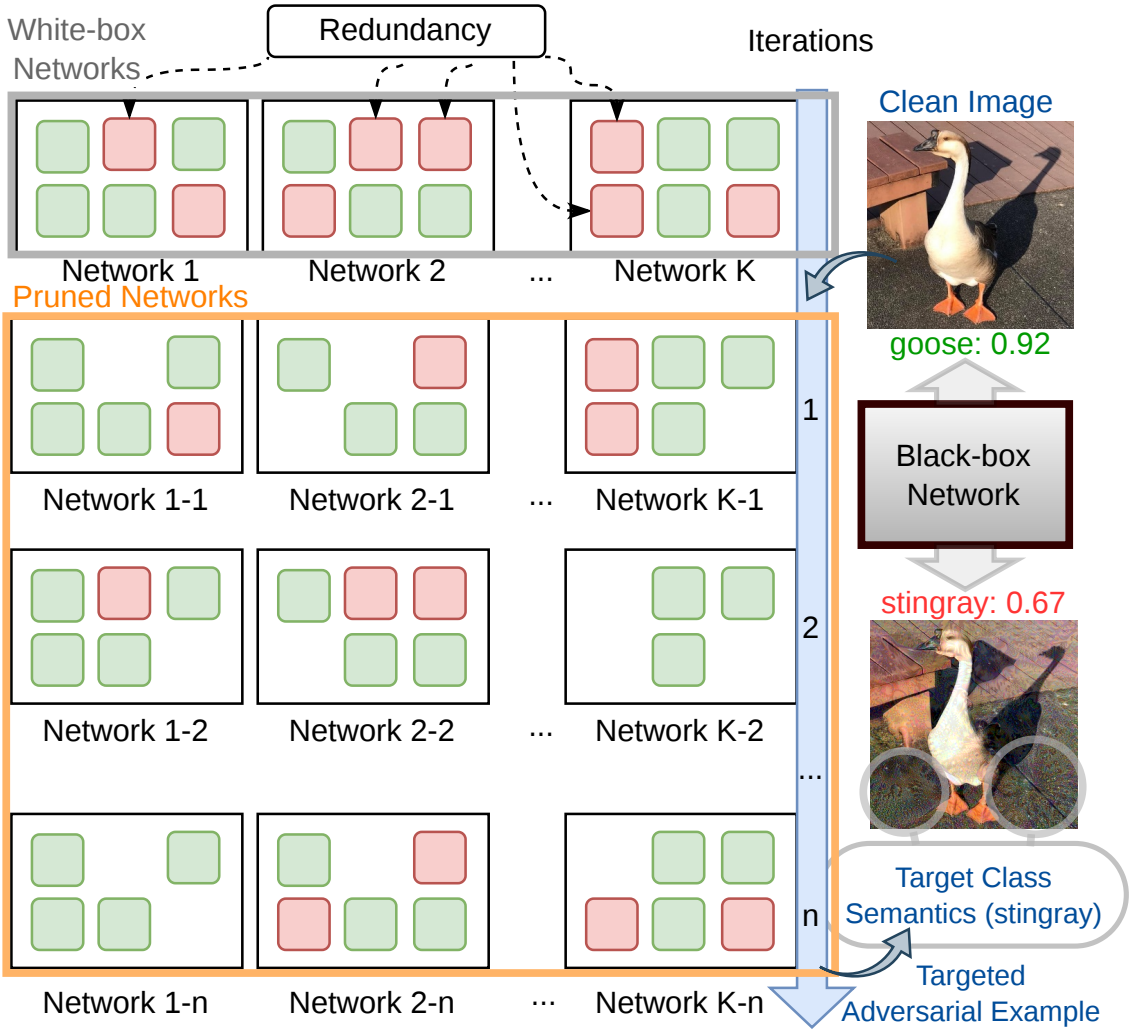


Figure 1.1: **The big picture of DWP.** Based on the over-parameterized property of neural networks, we leverage weight pruning to produce additional diversified pruned models from existing white-box networks at each iteration. By protecting necessary weight connections in each network, the quality of models is well-preserved. These additional pruned models can better impose the semantics of the target class onto adversarial images, yielding higher targeted transferability.

Chapter 1 Introduction

While deep learning continues to achieve breakthroughs in various domains, recent

doi:10.6342/NTU202301130

studies have shown vulnerabilities of deep neural networks to adversarial attacks, causing severe threats in safety-critical applications. For example, in image classification, an attacker can add human-imperceptible perturbations to benign images at testing time. These adversarial examples can fool a well-trained neural network to yield arbitrary classification results. Several attacks have been proposed to improve and evaluate the robustness of CNNs [2, 21, 42].

In the white-box settings, with complete information on the victim model, the attacker can generate adversarial examples effectively and efficiently. As for black-box settings, where the attacker only has limited information about the victim model, it is still possible to create cross-model attacks using a substitute model with white-box adversarial attack methods. This kind of black-box attack depends on the transferability of adversarial attacks.

Many methods have been proposed to increase the transferability for untargeted attacks, where the goal is to decrease the accuracy of the victim model. However, there is still room for improvement in creating transferable targeted attacks, where the attacker aims to mislead the victim model to produce a predefined specific outcome. Recent works use an ensemble-based approach to generate transferable targeted adversarial examples with multiple neural networks as substitute models simultaneously [26, 48]. To further enlarge the power of the ensemble, model augmentation creates additional networks by altering the existing ones [8, 23] and generates adversarial examples with these networks altogether.

However, we find that current methods of model augmentation rarely consider the importance of neurons and weight connections in networks. While Ghost Networks [23]

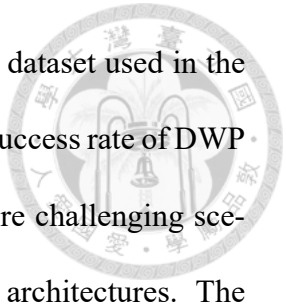
doi:10.6342/NTU202301130

inserts extra dropout layers and random skip connection mechanisms into networks to produce additional models, these dropout layers randomly drop neurons away without considering their significance. Authors in [8] introduce another uniform erosion on the remaining parameters after dropout and skip connection to increase diversity. However, there is still a lack of protection on necessary parameters. To avoid excessively destroying the performance of networks, these methods require heavy tuning on the hyperparameters like dropout rates, the amount of skip connection, the second erosion rates, and the locations of the inserted dropout layers.

When it comes to transferable targeted attacks, the quality of white-box substitute models plays a more crucial role. Rather than merely moving away from the original class, the semantics of targeted adversarial examples need to be close to the target class to acquire higher transferability [22, 30]. Dropping or disturbing the significant components in substitute networks can mislead targeted adversarial examples and yield worse transferability.

To overcome these problems, we learn from model compression and propose an improved model augmentation method named Diversified Weight Pruning (DWP). Model compression reduces the storage and computation overhead without substantial influence on model performances [9, 11, 20, 28]. With the over-parameterized property [4] of neural networks, weight pruning [11] can maintain the performance of a network by only removing redundant weight connections. Figure 1.1 summarizes our attack pipeline. To generate transferable targeted adversarial examples, we apply random weight pruning to each single CNN network accessible to form additional ones. These pruned networks remain stable since the significant weight connections are protected. We thus improve the ensemble-based approach with these extra diverse models.

doi:10.6342/NTU202301130



To evaluate DWP, we experiment with an ImageNet-compatible dataset used in the NIPS 2017 adversarial attack competition [18]. The average targeted success rate of DWP reaches 81.30% across CNNs. Furthermore, we test DWP in the more challenging scenarios of transferring to adversarially trained models and Non-CNN architectures. The results show that DWP improves the targeted success rate with up to 10.1% and 6.6% on average in these two settings. Finally, we demonstrate our DWP on the real-world Google Cloud Vision service and get 7.0% improvement.

In summary, our primary contributions are as follows:

- We propose DWP leveraging weight pruning to improve the existing model augmentation methods on transferable targeted attacks. Experiments show that our DWP enhances the combination of current state-of-the-art techniques.
- The experiment results show that DWP remains effective in more challenging settings like transferring to adversarially trained models, Non-CNN architectures, and even the real-world Google Cloud Vision service.
- We analyze the cosine similarities of adversarial perturbations between different pruned networks to verify that DWP increases the diversity of networks for generating adversarial perturbations.
- We provide intuitive experiments on explaining the success of targeted attacks with DWP.

doi:10.6342/NTU202301130

Chapter 2 Related Work

2.1 Transferable Attack

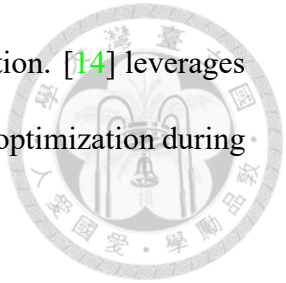
Throughout the work, we focus on simple transferable attacks [48], which require neither additional data nor model training for attacking compared to resource-intensive attacks [10, 16, 17, 44, 47]. Recent works aiming for simple transferable attacks mainly include four categories: gradient optimization, input transformation, advanced loss function, ensemble, and model augmentation.

2.1.1 Gradient Optimization

With iterative optimization-based methods [2, 19], one can get better solutions to an objective function for attacking through multiple times of optimization on adversarial examples and get stronger attacking results. Adjusting gradients used to update adversarial examples at each iteration appropriately has been shown beneficial for overcoming sub-optimal results in optimization. [5] combines momentum techniques with iterative attacks, accumulating gradients at each iteration to escape local optimum and stable the direction of updating. [24] applies Nesterov accelerated gradient for optimization, giving adversarial examples an anticipatory updating to yield faster convergence. [43] introduces variance

doi:10.6342/NTU202301130

tuning based momentum to reduce variance of gradients at each iteration. [14] leverages integrated gradients to include smoothing, attention modification and optimization during attacking.



2.1.2 Input Transformation

Motivated by Data Augmentation [35], several works suggest attacking transformed input to prevent adversarial examples from overfitting white-box models and failing to transfer to black-box ones. [45] uses random resizing and padding throughout the iterative attack. [6] enumerates several translated versions for each input image and fuses the gradients acquired on all of them. [24] leverages the scale-invariant property of CNNs and employs multiple scale copies from each input image.

2.1.3 Modern Loss Function

Cross entropy loss is widely used in image classification, also serving as the objective function for adversarial attacks. However, for targeted attacks, cross entropy is pointed out the saturation problem [22] as the output confidence of target class approaches to one. To this end, alternative loss functions attempt to provide more suitable gradients for optimization. [22] leverages Poincaré distance as the loss function, which amplifies the gradient magnitude as the confidence of the target class grows. [48] proposes a simple logit loss, which has constant gradient magnitude regardless of the output probability.

doi:10.6342/NTU202301130

2.1.4 Ensemble and Model Augmentation

Adversarial examples generated by ensembling multiple white-box networks are more likely to transfer to black-box networks [26]. Instead of simply fusing the output confidence of each white-box network, [46] suggests reducing the gradient variance of white-box models during attacking. To further improve ensemble-based approaches, Model Augmentation produces additional diverse models from the existing white-box networks. [23] acquires ghost networks for ensemble through perturbing dropout and skip connections of existing ones. [8] further improves the diversified ensemble via dual-stage erosion.

2.2 Network Pruning

The intensive cost of computation and storage hinders applications of neural networks, especially on embedding systems. Network Compression aims to reduce the scale of networks, making them more feasible for deployment. With the over-parameterized property [4], several works about removing redundancy in networks, known as Network Pruning, are proposed and become a branch of Network Compression. [20] uses the second-derivative information to find redundant weights in networks. [11] shows that neural networks can highly preserve performance even if trimming more than half of their connections. Retraining after pruning for better preservation of accuracy is also investigated [9, 28].

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

Chapter 3 Methodology

Given a neural network θ and a benign example x, we generate a targeted adversarial example x^{adv} for the target class y^{target} by solving the following constrained optimization problem:

$$\arg\min_{x^{\text{adv}}} J(x^{\text{adv}}, y^{\text{target}}; \theta) \quad \text{s.t.} \quad \left\|x^{\text{adv}} - x\right\|_\infty \leq \epsilon, \tag{3.1}$$

where J is the loss function for multiclass classification and ϵ is the perturbation budget. To circumvent the gradient saturation problem of cross-entropy, we use logit loss [48] as our loss function J.

3.1 Preliminary and Motivation

We start by establishing the roles of current state-of-the-art techniques in our iterative attack. Then, we demonstrate how we apply Weight Pruning to improve targeted transferability.

3.1.1 Momentum and Nesterov Iterative Method (NI)

Inspired by Nesterov Accelerated Gradient [31], Nesterov Iterative Method (NI) [24] modifies Momentum Iterative-FGSM [5] by adding the historical gradients to current ad-

doi:10.6342/NTU202301130

versarial examples x_n and gets x_n^{nes} in advance. Gradients at the ahead x_n^{nes} instead of the current x_n will be used for updating. The scheme helps accelerate convergence by avoiding the local optimum earlier:

$$x_n^{\text{nes}} = x_n + \alpha \cdot \mu \cdot g_{n-1} \tag{3.2}$$

$$g_n = \mu \cdot g_{n-1} + \nabla_x J(x_n^{\text{nes}}, y^{\text{target}}; \theta) \tag{3.3}$$

$$x_{n+1} = \text{Clip}_x^{\epsilon}(x_n - \alpha \cdot \text{sign}(g_n)). \tag{3.4}$$

Here μ is the decay factor of the historical gradients. The gradient computed encourages adversarial examples to increase confidence logit output by the white-box network model θ on the target class through gradient ascent with learning rate α. A clipping operation onto the ϵ-ball centered at the original input image x is at the end of each iteration. To preserve more information about the gradient for attacking [49], we don't include the L1 normalization.

3.1.2 Scale Invariant Method (SI)

Neural networks can preserve output even though the input image x goes through scale operations such as $S_m(x) = x/2^m$ [24]. With the scale-invariant property, each composite of white-box networks and scale operations becomes different functions. Adversarial examples can enjoy more diverse gradients:

$$g_n = \mu \cdot g_{n-1} + \frac{1}{M} \sum_{m=0}^{M-1} \nabla_x J(S_m(x_n^{\text{nes}}), y^{\text{target}}; \theta). \tag{3.5}$$

M is the number of scaled versions feeding into the network for each image.

doi:10.6342/NTU202301130

3.1.3 Diverse Input Patterns (DI)

Inspired by data augmentation techniques [35] used in network training, DI [45] imposes random resizing and padding on each image before it feeds into network models to avoid overfitting. Straightforward cooperation with NI and SI is as follows:

$$g_n = \mu \cdot g_{n-1} + \frac{1}{M} \sum_{m=0}^{M-1} \nabla_x J(S_m(T(x_n^{\text{nes}}, p_{\text{DI}})), y^{\text{target}}; \theta). \tag{3.6}$$

The introduced T decides whether to apply random resizing at each iteration with probability p_{DI}, which degenerates when $p_{\text{DI}} = 0$.

3.1.4 Translation Invariant Method (TI)

To deal with different discriminative regions [6] of various defense neural networks, TI produces several translated versions for the current image in advance and computes the gradient for each separately. These gradients will then be fused and used to attack the current image. [6] also shows that one can approximate the gradient fusion using convolution. The approximation prevents TI from enduring the costly computation on excessive translated versions for every single image, also yielding the further revised updating procedure:

$$g_n = \mu \cdot g_{n-1} + \boldsymbol{W} * \frac{1}{M} \sum_{m=0}^{M-1} \nabla_x J(S_m(T(x_n^{\text{nes}}, p_{\text{DI}})), y^{\text{target}}; \theta). \tag{3.7}$$

$\boldsymbol{W}$ is the convolution kernel matrix applied. Some typical options are linear, uniform, or Gaussian kernel.

We abbreviate the attacking procedure so far to NI-SI-TI-DI with the combination of these techniques.

doi:10.6342/NTU202301130

3.2 Diversified Weight Pruning

We name the proposed approach Diversified Weight Pruning (DWP) due to the increased diversity of white-box models for ensemble via Weight Pruning. Following Weight Pruning, we sort the connections of each white-box network by the L1 norm of their weight values. With a predefined rate r, we only consider the lowest $(100 \cdot r)$% "prunable" since weights with small absolute values are shown unnecessary [11]. Networks can preserve accuracy after these connections are pruned away even without retraining [11].

For our pruning operation, we first identify the set of prunable weights. Let γ be the $(100 \cdot (1 - r))$-th percentile of weights in θ. We formulate the prunable set:

$$\Gamma(\theta, r) = \{w \in \theta | w < \gamma\} \subseteq \theta. \tag{3.8}$$

With $\Gamma(\theta, r)$ collecting all the prunable weights of θ, we introduce an indicator vector for it:

$$\Pi_{\Gamma(\theta,r)} = (\lambda_1, \lambda_2, ..., \lambda_\kappa), \tag{3.9}$$

where κ is the total number of weights in $\theta = \{w_1, w_2, ..., w_\kappa\}$ including non-prunable ones. λ_i is determined by whether its corresponding $w_i \in \theta$ is in the prunable subset $\Gamma(\theta, r)$:

$$\lambda_i = \begin{cases} 1, & \text{if } w_i \in \Gamma(\theta, r) \\ 0, & \text{otherwise} \end{cases}. \tag{3.10}$$

Supported by the indicator vector $\Pi_{\Gamma(\theta,r)}$, our pruning operation $P(\cdot)$ can protect the non-

doi:10.6342/NTU202301130

prunable weights by masking:

$$P(\theta, r) = (\mathbf{1}_\kappa - \Pi_{\Gamma(\theta,r)} \odot \mathbf{b}) \odot \theta, \tag{3.11}$$

where $\odot$ denotes the element-wise multiplication and $\mathbf{1}_\kappa = (1, 1, ..., 1) \in R^\kappa$ denotes an all-one vector. $\mathbf{b} = (b_1, b_2, ..., b_\kappa)$ is a vector with $b_i \overset{\text{i.i.d}}{\sim} \text{Bernoulli}(p_{\text{bern}})$, where p_{bern} is the probability for pruning each connection independently.

To be specific, $\Pi_{\Gamma(\theta,r)}$ and $\mathbf{b}$ both are binary masks with identical layout as θ. $\Pi_{\Gamma(\theta,r)}$ is responsible for protecting non-prunable weights, while $\mathbf{b}$ is for random pruning. Each binary element in $\Pi_{\Gamma(\theta,r)} \odot \mathbf{b}$ indicates whether to prune the corresponding weight value in θ. The main difference from Dropout [37] used in previous model augmentations [8, 23], is that DWP only considers prunable weights. The detailed comparison will be shown in the following sections.

Instead of producing all the pruned models beforehand, we acquire pruned models at each iteration right before gradient computing.

$$\begin{aligned} g_n &= \mu \cdot g_{n-1} + \\ &\frac{\boldsymbol{W}}{M} * \sum_{m=0}^{M-1} \nabla_x J(S_m(T(x_n^{\text{nes}}, p_{\text{DI}})), y^{\text{target}}; P(\theta, r)). \end{aligned} \tag{3.12}$$

With this longitudinal ensemble strategy [8, 23], the storage and computation overhead are almost identical to the original attack procedure. We summarize the time cost of DWP in Supplementary Material. With K white-box models, our final DWP attack procedure

doi:10.6342/NTU202301130

is shown as follows:

$$g_n = \mu \cdot g_{n-1} + \frac{\boldsymbol{W}}{M} * \sum_{m=0}^{M-1} \sum_{k=1}^{K} \beta_k \nabla_x J(S_m(T(x_n^{\text{nes}}, p_{\text{DI}})), y^{\text{target}}; P(\theta_k, r)), \tag{3.13}$$

where β_k are the ensemble weights, $\sum_{k=1}^{K} \beta_k = 1$.

Benefiting from no dependency on network retraining and extra data, our proposed DWP is simple and lightweight. As there is no further retraining, we select the L1 norm for pruning since it is better than L2 on preserving accuracy [11].

doi:10.6342/NTU202301130

Chapter 4 Experiments

In this section, we first describe the experiment settings and demonstrate the results of transferable targeted attacks under various scenarios. We then inspect the diversified property of pruned models in DWP from the view of geometry. Finally, we illustrate an intuitive explanation of the success of transferable targeted attacks.

4.1 Experimental Setup

4.1.1 Dataset

We use an ImageNet-compatible dataset[1] containing 1,000 images provided by the NIPS 2017 adversarial attack competition [18]. Each image in the dataset has an officially assigned target class for fair comparison.

4.1.2 Networks

We perform experiments on four naturally trained CNNs: Inception-v3 (Inc-v3) [39], ResNet-50 (Res-50) [12], VGGNet-16 (VGG-16) [36] and DenseNet-121 (Den-121) [13], four naturally trained Vision Transformers (ViTs): ViT-Small-Patch16-224 (ViT-S-16-

[1] https://github.com/cleverhans-lab/cleverhans/blob/11ea10/examples/nips17_adversarial_competition/dataset/dev_dataset.csv

doi:10.6342/NTU202301130

224), ViT-Base-Patch16-224 (ViT-B-16-224)[7], Swin-Small-Patch4-Window7-224 (Swin-S-224), Swin-Base-Patch4-Window7-224 (Swin-B-224)[27], three naturally trained Multi-Layer Perceptrons (MLPs): Mixer-Base-Patch16-224 (MLP-Mixer) [40], ResMLP-Layer24-224 (ResMLP) [41], gMLP-Small-Patch16-224 (gMLP) [25], and two adversarially trained CNNs: ens3-adv-Inception-v3 (Inc-v3ens3) and ens-adv-inception-resnet-v2 (IncRes-v2ens) [42]. All the networks are publicly accessible.

4.1.3 Hyper-parameters

Our method includes three input transformations TI, DI, and SI. Following [22], we set the probability p_{DI} of DI to be 0.7 and select a Gaussian kernel with a kernel length of 5 for $\boldsymbol{W}$ in TI. For SI, due to the limited computing resources, we set the number of scale copies $M = 3$. Following [5, 22, 24, 48], the momentum decay factor μ is set to 1. For all the iterative attacks in the experiments, we use 100 iterations with learning rate $\alpha = 2/255$ as in [48]. We use the perturbation budget $\epsilon = 16$ under L_∞ norm in all the experiments, complying with the rule in the NIPS 2017 competition. Last but not least, for our proposed DWP, the probability p_{bern} is 0.5 and the prunable rate r is 0.7. In other words, we prune 35% of the connections of each network in expectation at each iteration.

4.1.4 Baseline Methods

We compare the targeted transferability of DWP and the previous model augmentation methods, Ghost Networks (GN) and Dual-Stage Network Erosion (DSNE) [8, 23], in combination with the state-of-the-art techniques NI-SI-TI-DI. For non-residual networks like VGG-16 and Inc-v3, we insert dropout layers after each activation function.

doi:10.6342/NTU202301130

As for residual networks such as Res-50 and Den-121, we apply skip connection erosion on the blocks of each network. GN [23] drops activation outputs with a dropout rate $\Lambda_{\text{GN}}^{\text{drop}}$ and multiplies the skip connection by a factor sampled from the uniform distribution $U[1-\Lambda_{\text{GN}}^{\text{skip}}, 1+\Lambda_{\text{GN}}^{\text{skip}}]$. Based on GN, DSNE[8] not only uses a dropout rate $\Lambda_{\text{DSNE}}^{\text{drop}}$, but also scales the values passing dropout by an uniform random factor from $U[1-\Lambda_{\text{DSNE}}^{\text{scale}}, 1+\Lambda_{\text{DSNE}}^{\text{scale}}]$. DSNE also alters the skip connections like GN with $U[1-\Lambda_{\text{DSNE}}^{\text{skip}}, 1+\Lambda_{\text{DSNE}}^{\text{skip}}]$, and introduces an additional bias factor γ. We set $\Lambda_{\text{GN}}^{\text{drop}}=0.012, \Lambda_{\text{GN}}^{\text{skip}}=0.22, \Lambda_{\text{DSNE}}^{\text{drop}}=0.01, \Lambda_{\text{DSNE}}^{\text{scale}}=0.1, \Lambda_{\text{DSNE}}^{\text{skip}}=0.14$, and $\gamma=0.8$ following [8, 23]. We do not include VT [43], SVRE [46], and IG [14] due to their specific assumptions or implementation details. However, we compare the results with them in Supplementary Material.

4.2 Transferable Targeted Attack in Various Scenarios

We consider targeted transferability under four scenarios: transferring across CNNs, transferring to adversarially trained models, Non-CNN architectures, and the real-world Google Cloud Vision service. We prepare specified networks for each case. We generate adversarial examples on the ensemble of the white-box models and evaluate targeted success rates on the specified black-box model. No access to the black-box model is allowed during an attack. Note that for the ensemble, we use equal weights $\beta_k = 1/K$ for each of the K white-box models.

4.2.1 Transferring across Naturally Trained CNNs

As convolution neural networks are widely used, we first examine the cases between CNNs. We select four classic CNN networks following [48]: Res-50, VGG-16, Den-121,

doi:10.6342/NTU202301130

Attack	-Res-50	-Den-121	-VGG16	-Inc-v3	Avg
TI-DI	68.2	81.2	75.3	57.6	70.58
+GN	58.7	78.9	77.6	64.0	69.80
+DSNE	55.9	65.4	77.8	57.1	64.05
+DWP	69.0	82.1	81.3	61.4	**73.45**
NI-SI	29.0	40.1	30.3	34.6	33.5
+GN	49.2	63.0	67.5	40.9	55.15
+DSNE	44.8	58.7	66.8	41.4	52.93
+DWP	52.4	62.6	67.9	40.4	**55.83**
NI-SI-TI-DI	76.1	86.7	77.1	66.9	76.70
+GN	68.7	85.0	80.1	72.4	76.55
+DSNE	67.0	75.1	79.1	66.7	71.98
+DWP	77.7	89.4	87.2	70.9	**81.30**

Table 4.1: The targeted success rates of transferring across CNNs. The "-" prefix stands for the black-box network with the other three serving as the white-box ones for ensemble. "+" means the participation of a specific model augmentation method. DWP outperforms other leading model augmentations GN and DSNE.

and Inc-v3.

Table 4.1 shows the results of transferable targeted attacks between CNNs. DWP boosts the attack methods and outperforms GN and DSNE. As the four CNNs possess designs such as Residual, Dense, and Inception blocks, the results demonstrate the benefits of the diversified ensemble in attacking black-box CNNs with different mechanisms from the white-box substitutes. Protecting necessary connections is also shown to be advantageous. We also provide untargeted results and transferring from a single model in Supplementary Material.

4.2.2 Transferring to Adversarially Trained Models

Adversarial training [29, 42] is one of the primary techniques for defending against malicious attacks. It brings robustness to models by training them with adversarial exam-

doi:10.6342/NTU202301130

ples. Under the scenario of transferring to adversarially trained models, we ensemble only the four naturally trained networks (Res-50, Den-121, VGG16, and Inc-v3) as white-box models to simulate the situation where attackers have few details about defense. The two adversarially trained networks (Inc-v3ens3 and IncRes-v2ens) will act as our black-box model separately. We also include the results of multi-step adversarially trained networks [33] in Supplementary Material.

Table 4.2 summarizes the results of transferring to adversarially trained networks. Targeted success rates under this scenario are lower due to the robustness of adversarially trained models. Under such a challenging scenario, DWP still helps alleviate the discrepancy between white-box naturally trained and black-box adversarially trained networks, bringing about up to 10.1% improvement on average. The power of the diversified ensemble under the premise of protecting necessary connections is highlighted again, especially for black-box networks with significant differences from the white-box ones.

4.2.3 Transferring to Non-CNN Architectures

In practice, information about the networks used by defenders remains unknown to attackers. A targeted attack method is more practical if adversarial examples can transfer to black-box architectures different from the white-box ones accessible by attackers. Beyond CNNs, recent works attempt to solve computer vision tasks using Vision Transformers (ViTs) [7, 27] and Multi-Layer Perceptrons (MLPs) [25, 40, 41]. To be more comprehensive, we evaluate the targeted transferability from CNNs to these architectures.

We generate targeted adversarial images on the ensemble of the four naturally trained CNNs. NI-SI-TI-DI comes with the three model augmentation methods, respectively,

doi:10.6342/NTU202301130

Attack Method	NI-SI-TI-DI	+GN	+DSNE	+DWP
Inc-v3ens3	50.0	51.6	49.7	**65.3**
IncRes-v2ens	19.4	29.8	34.5	**39.0**
Average	34.7	40.7	42.1	**52.15**

Table 4.2: The targeted success rates of transferring to adversarially trained models. Our DWP outperforms GN and DSNE over 10%.

Attack Method	NI-SI-TI-DI	+GN	+DSNE	+DWP
ViT-S-16-224	25.9	31.5	31.1	**37.3**
ViT-B-16-224	24.8	29.9	28.4	**37.4**
Swin-S-224	26.7	29.1	26.5	**36.7**
Swin-B-224	23.9	27.1	23.9	**32.9**
MLP-Mixer	21.7	24.2	27.0	**30.9**
ResMLP	51.3	56.5	54.4	**64.1**
gMLP	20.4	25.3	26.9	**30.4**
Average	27.81	31.94	31.17	**38.53**

Table 4.3: The targeted success rates of transferring to Non-CNN architectures. Our DWP maintains higher success rates stably.

including our DWP. From Table 4.3, model augmentations are effective even though the black-box networks have no convolution operations other than the input projection. Our DWP improves the results on both ViTs and MLPs, outperforming all the other methods.

4.2.4 Transferring to Google Cloud Vision

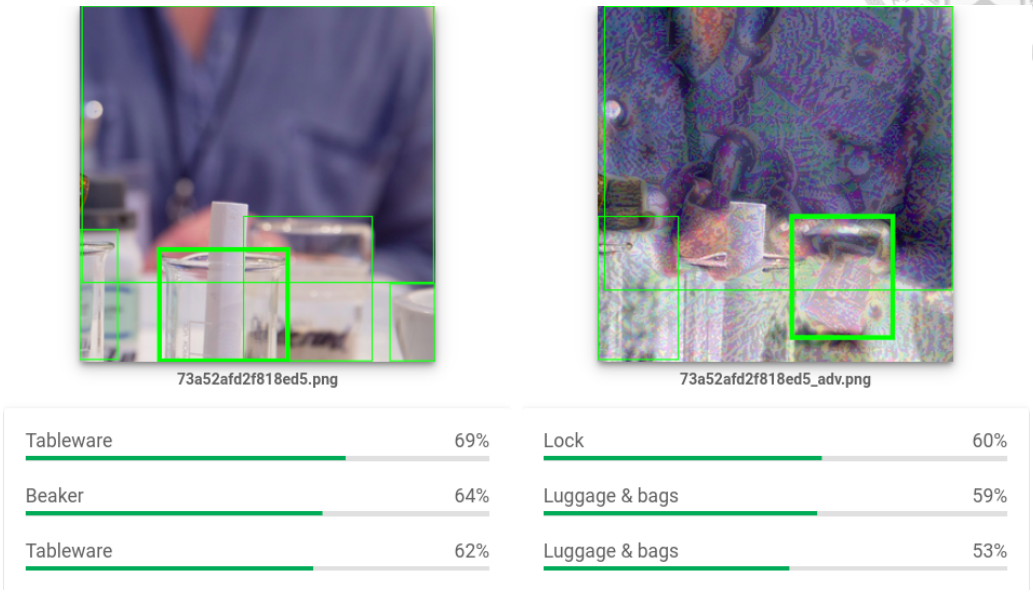
For a more practical scenario, we use Google Cloud Vision to evaluate our targeted adversarial examples. Google Cloud Vision predicts a list of labels with their corresponding confidence scores and only returns label annotations with confidence above 50%. The scenario is completely black-box since no information about gradients and parameters of the underlying system is accessible. Previous works leverage query-based attacks [1, 3, 15] or black-box transferability [26, 48]. However, query-based methods often require large numbers of queries, and the existing transferable attacks still have substantial room for improvement.

In this experiment, we randomly select 100 images correctly labeled by Google Cloud

doi:10.6342/NTU202301130

NI-SI-TI-DI	+GN	+DSNE	+DWP
27	43	42	**50**

Table 4.4: Targeted success rates (%) on Google Cloud Vision



(a) Clean image (b) NI-SI-TI-DI + DWP

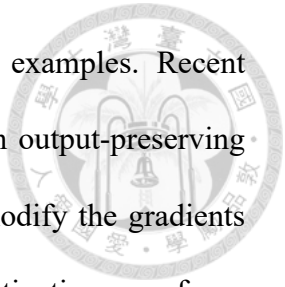
Figure 4.1: A demo of our DWP attack on Google Cloud Vision. The attacked image with the ground truth label of "Beakers" is recognized as the target class "Padlocks" assigned by the NIPS 2017 Imagenet-compatible dataset.

Vision from the Imagenet-compatible dataset. Similarly, we use the four naturally trained CNNs to generate adversarial examples. We identified an image as a successful attack if at least one of the labels returned by Google Cloud Vision is semantically close to its corresponding target class. We summarize the attack results in Table 4.4. DWP outperforms the previous model augmentation methods by 7%. Figure 4.1 demonstrates an attack on Google Cloud Vision. We refer readers to Supplementary Material for more demos.

4.3 Perturbations from Different Pruned Models

To investigate whether our method can promote diversity of the ensemble, we examine the relationship between the generated adversarial perturbations instead of between model outputs. The reason is that compared to the logit outputs by models, perturba-

doi:10.6342/NTU202301130



tion vectors are more direct in determining the update of adversarial examples. Recent works [6, 24] have proposed methods improving transferability with output-preserving operations. Although these operations retain model outputs, they modify the gradients and enrich the directions of adversarial perturbations. With these motivations, we focus on the diversity between perturbations computed from pruned models.

Liu et al. [26] first studied the effectiveness of ensemble in enhancing transferability. They demonstrate the diversity of the ensemble by showing near-zero cosine similarities between perturbations from different white-box networks. Following [26], we calculate cosine similarities between perturbations generated from the additional pruned models produced by DWP. From each of our four naturally trained CNNs, we acquire five pruned models with different connections pruned. We term the cosine similarity between perturbations of pruned models from an identical CNN as intra-CNN similarity. The case from different CNNs is termed as inter-CNN similarity. To avoid cherry-picking, both intra-CNN and inter-CNN similarities come from the average of the first ten images in the ImageNet-compatible dataset. Futhermore, we only use NI in combination with DWP to produce perturbations in this experiment to prevent the influence of factors other than pruning.

Figure 4.2 is a symmetric matrix containing 16 (4×4) blocks. The diagonal blocks summarize ten (C_2^5) intra-CNN similarities while the non-diagonal blocks summarize 25 (5×5) inter-CNN similarities in cells. The diagonal cells are all 1.0 since they are all from two identical perturbation vectors. As for the non-diagonal cells, we find the cell values in diagonal blocks (intra-CNN) slightly higher than in non-diagonal blocks (inter-CNN). However, these values are still close to zero, appearing dark red. The results show that whether two pruned models come from the same CNN, the generated perturbations

doi:10.6342/NTU202301130

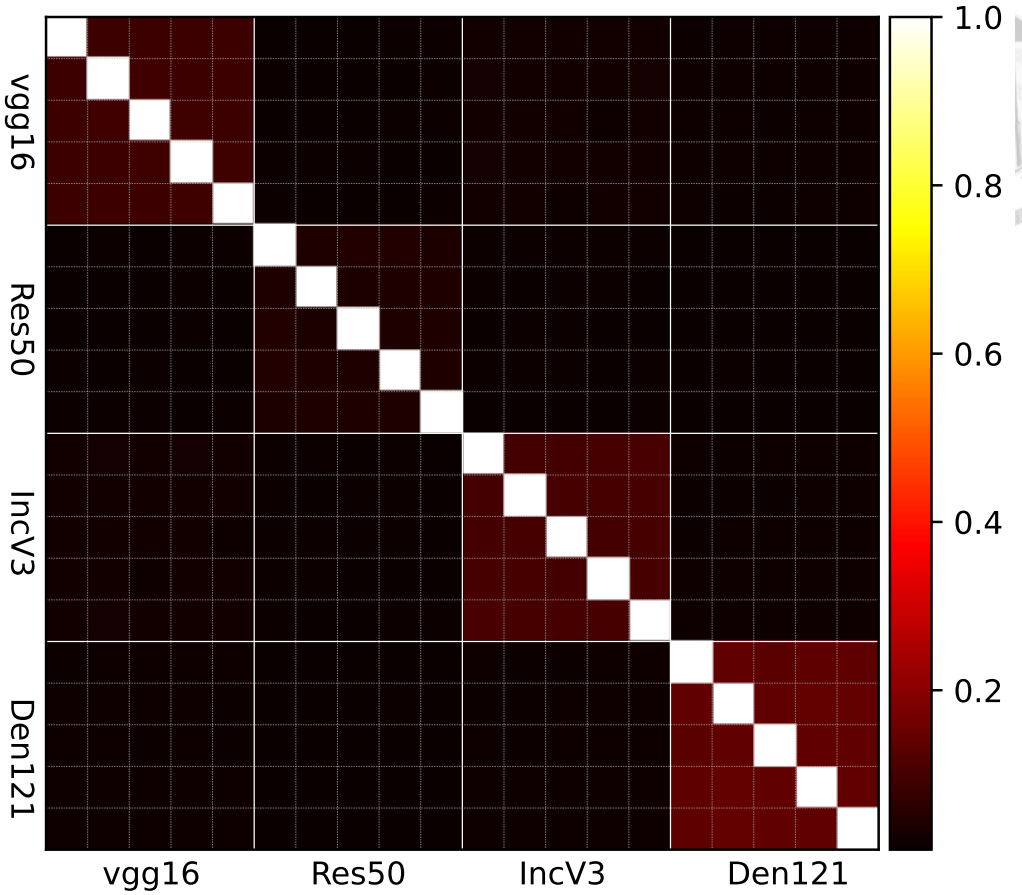


Figure 4.2: Perturbation cosine similarities between pruned models. Each diagonal block summarizes 10 (C_2^5) intra-CNN similarity cells. Each non-diagonal block summarizes 25 (5×5) inter-CNN similarity cells. The pairwise cosine similarity matrix is symmetric and shows orthogonality between perturbations.

generated are always nearly orthogonal. These observations on orthogonality support our claim that pruned models obtained via DWP provide more diversity for attacking.

4.4 Semantics of the Target Class

Prior work has shown that targeted adversarial examples semantically close to the target class tend to be more transferable [16, 17, 30]. To provide more insight into the success of DWP, we explore the patterns in targeted adversarial examples.

GradCAM [34] uses the mean gradient values of a specific class output confidence with respect to each intermediate feature map to be its corresponding coefficient. The

doi:10.6342/NTU202301130

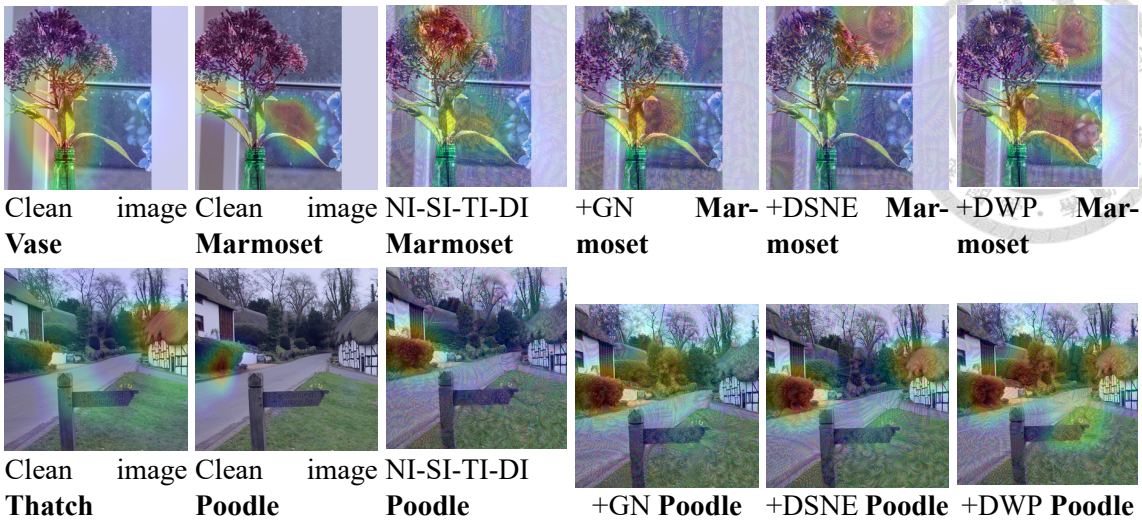


Figure 4.3: The GradCAMs of clean and targeted adversarial images on naturally trained ResNet-50. The two leftmost columns show GradCAMs of clean images regarding the ground truth and target class. The other columns provide the GradCAMs of adversarial images generated by different methods. Targeted perturbations guide the highlighted area and impose semantics of the target class on images.

weighted average of feature maps using these coefficients provides an explanation of a particular decision made by the model. In Figure 4.3, we draw GradCAMs on naturally trained ResNet-50 to provide some explainable observations on adversarial images generated by different methods.

In the two leftmost columns, we show the GradCAMs of clean images with their ground truth and target class, respectively. GradCAMs correctly highlight regions about the ground truth class on the clean images. On the other hand, without adversarial perturbations, there is no evident relation between the target class and the corresponding highlighted areas. The other four columns show the GradCAMs with the target class of adversarial images generated by NI-SI-TI-DI and NI-SI-TI-DI plus GN, DSNE, and DWP, respectively. The adversarial perturbations produce target class-specific patterns and guide the highlighted region of GradCAMs. Note that the perturbation budget is limited to $l_\infty \leq 16$ to ensure the attacks are quasi-imperceptible. For more results of GradCAMs, we refer readers to Supplementary Material.

doi:10.6342/NTU202301130

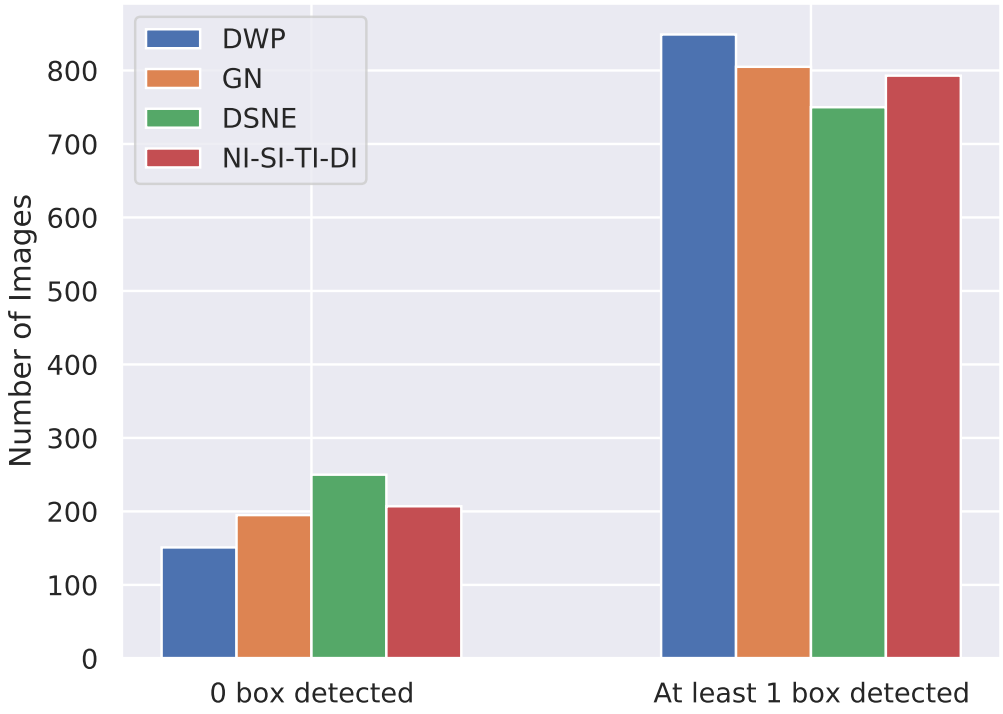


Figure 4.4: The comparison of object detection results of the target class. Compared to GN and DSNE, adversarial images generated by our DWP are more likely to contain at least one object detected as the target class.

For quantitative comparison, we leverage an object detector[2] to detect target-class patterns in the targeted adversarial images. We set the threshold to 0.1, which is lower than usual, to capture more potential patterns in images. The bars on the right side of Figure 4.4 summarize the number of images with at least one bounding box detected. Compared to other methods, DWP is the method most likely to generate adversarial examples with at least one target-class pattern detected. Notice that under the limit of the perturbation budget, even though we have a lower threshold, there are still about 150 images without any target-class object detected according to the bars on the left side of Figure 4.4. Since we do not integrate object detectors into our attack procedure, the results support that our DWP is better at producing target-class-specific information.

[2] https://github.com/ibaiGorordo/ONNX-ImageNet-1K-Object-Detector

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

Chapter 5 Conclusion

In this paper, we propose Diversified Weight Pruning (DWP) leveraging network compression to improve the targeted transferability of adversarial attacks. DWP produces additional pruned models for ensemble via weight pruning. Due to the over-parameterized property, the quality of pruned models introduced by DWP is well-preserved. Experiments show that by protecting the necessary weight connections of networks, targeted adversarial examples are more likely to acquire semantics of the target class. By evaluating DWP on ImageNet, we show that DWP improves the state-of-the-art model augmentation methods on transferable targeted attacks, especially for challenging scenarios such as transferring to adversarially trained models and Non-CNN architectures. We hope that our work can serve as a bridge between network compression and transferable attack, inspiring more collaboration.

doi:10.6342/NTU202301130

doi:10.6342/NTU202301130

References

[1] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[2] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, pages 39–57. IEEE Computer Society, 2017.

[3] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In S. Capkun and F. Roesner, editors, 29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020, pages 2667–2684. USENIX Association, 2020.

[4] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas. Predicting parameters in deep learning. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings

doi:10.6342/NTU202301130

of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2148–2156, 2013.

[5] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 9185–9193. Computer Vision Foundation / IEEE Computer Society, 2018.

[6] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 4312–4321. Computer Vision Foundation / IEEE, 2019.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021.

[8] Y. Duan, J. Zou, X. Zhou, W. Zhang, J. Zhang, and Z. Pan. Adversarial attack via dual-stage network erosion, 2022.

[9] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[10] M. Gubri, M. Cordy, M. Papadakis, Y. L. Traon, and K. Sen. LGV: boosting adversarial example transferability from large geometric vicinity. In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, Computer Vision -

doi:10.6342/NTU202301130

ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV, volume 13664 of Lecture Notes in Computer Science, pages 603–618. Springer, 2022.

[11] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 1135–1143, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778. IEEE Computer Society, 2016.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2261–2269. IEEE Computer Society, 2017.

[14] Y. Huang and A. W. Kong. Transferable adversarial attack based on integrated gradients. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022.

[15] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In J. G. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, ICML

doi:10.6342/NTU202301130

2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pages 2142–2151. PMLR, 2018.

[16] N. Inkawhich, K. J. Liang, L. Carin, and Y. Chen. Transferable perturbations of deep feature distributions. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[17] N. Inkawhich, K. J. Liang, B. Wang, M. Inkawhich, L. Carin, and Y. Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[18] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe. Adversarial attacks and defences competition, 2018.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net, 2017.

[20] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In D. S. Touretzky, editor, Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989], pages 598–605. Morgan Kaufmann, 1989.

doi:10.6342/NTU202301130

[21] C. Li, S. Gao, C. Deng, D. Xie, and W. Liu. Cross-modal learning with adversarial samples. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 10791–10801, 2019.

[22] M. Li, C. Deng, T. Li, J. Yan, X. Gao, and H. Huang. Towards transferable targeted attack. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pages 638–646. Computer Vision Foundation / IEEE, 2020.

[23] Y. Li, S. Bai, Y. Zhou, C. Xie, Z. Zhang, and A. L. Yuille. Learning transferable adversarial examples via ghost networks. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 11458–11465. AAAI Press, 2020.

[24] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[25] H. Liu, Z. Dai, D. R. So, and Q. V. Le. Pay attention to mlps. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems 34: Annual Conference on

doi:10.6342/NTU202301130

Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 9204–9215, 2021.

[26] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In International Conference on Learning Representations, 2017.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pages 9992–10002. IEEE, 2021.

[28] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[29] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[30] M. Naseer, S. H. Khan, M. Hayat, F. S. Khan, and F. Porikli. On generating transferable targeted perturbations. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pages 7688–7697. IEEE, 2021.

[31] Y. NESTEROV. A method for unconstrained convex minimization problem with the rate of convergence o (1/k^ 2). In Doklady AN USSR, volume 269, pages 543–547, 1983.

doi:10.6342/NTU202301130

[32] Z. Qin, Y. Fan, Y. Liu, L. Shen, Y. Zhang, J. Wang, and B. Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, Advances in Neural Information Processing Systems, 2022.

[33] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 618–626. IEEE Computer Society, 2017.

[35] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. J. Big Data, 6:60, 2019.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[37] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, 2014.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet

doi:10.6342/NTU202301130

and the impact of residual connections on learning. In S. Singh and S. Markovitch, editors, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pages 4278–4284. AAAI Press, 2017.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 2818–2826. IEEE Computer Society, 2016.

[40] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 24261–24272, 2021.

[41] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. CoRR, abs/2105.03404, 2021.

[42] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[43] X. Wang and K. He. Enhancing the transferability of adversarial attacks through

doi:10.6342/NTU202301130

variance tuning. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pages 1924–1933. Computer Vision Foundation / IEEE, 2021.

[44] X. Wang, X. He, J. Wang, and K. He. Admix: Enhancing the transferability of adversarial attacks. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pages 16138–16147. IEEE, 2021.

[45] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 2730–2739. Computer Vision Foundation / IEEE, 2019.

[46] Y. Xiong, J. Lin, M. Zhang, J. E. Hopcroft, and K. He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14983–14992, June 2022.

[47] X. Yang, Y. Dong, T. Pang, H. Su, and J. Zhu. Boosting transferability of targeted adversarial examples via hierarchical generative networks. In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV, volume 13664 of Lecture Notes in Computer Science, pages 725–742. Springer, 2022.

[48] Z. Zhao, Z. Liu, and M. Larson. On success and simplicity: A second look at transferable targeted attacks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang,

doi:10.6342/NTU202301130

and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 6115–6128. Curran Associates, Inc., 2021.

[49] J. Zou, Y. Duan, B. Li, W. Zhang, Y. Pan, and Z. Pan. Making adversarial examples more transferable and indistinguishable. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 3662–3670. AAAI Press, 2022.

doi:10.6342/NTU202301130

Appendix A — Supplementary Material

A.1 Ablation Analysis on Prunable Rates

In the ablation analysis, we explore targeted attack success rates under different prunable rates r. As the prunable rate determines the number of connections possible to be pruned during attacking, white-box models can produce more diverse pruned models using higher prunable rates. However, with excessive connections pruned away, the quality of pruned networks will be unstable.

To find the sweet spot to the trade-off, we enumerate different prunable rates, conducting the attack experiments with all the other hyper-parameters as default. Figure A.1 shows the trade-off. We select $r = 0.7$ throughout our experiments as the curve of mean targeted success rates reaches its maximum. With our designated prunable rate $r = 0.7$, DWP prunes about 35% of weight connections. Figure A.2 shows the decline in the accuracy of the four CNNs with different rates of minor weight connections pruned.

A.2 Transferring across CNNs with Similar Architectures

Table A.1 summarizes the targeted attack success rates across Inception-v3 (Inc-v3), Inception-v4 (Inc-v4), Inception-Resnet-v2 (IncRes-v2) [38] and ResNet-101 (Res-101)

doi:10.6342/NTU202301130

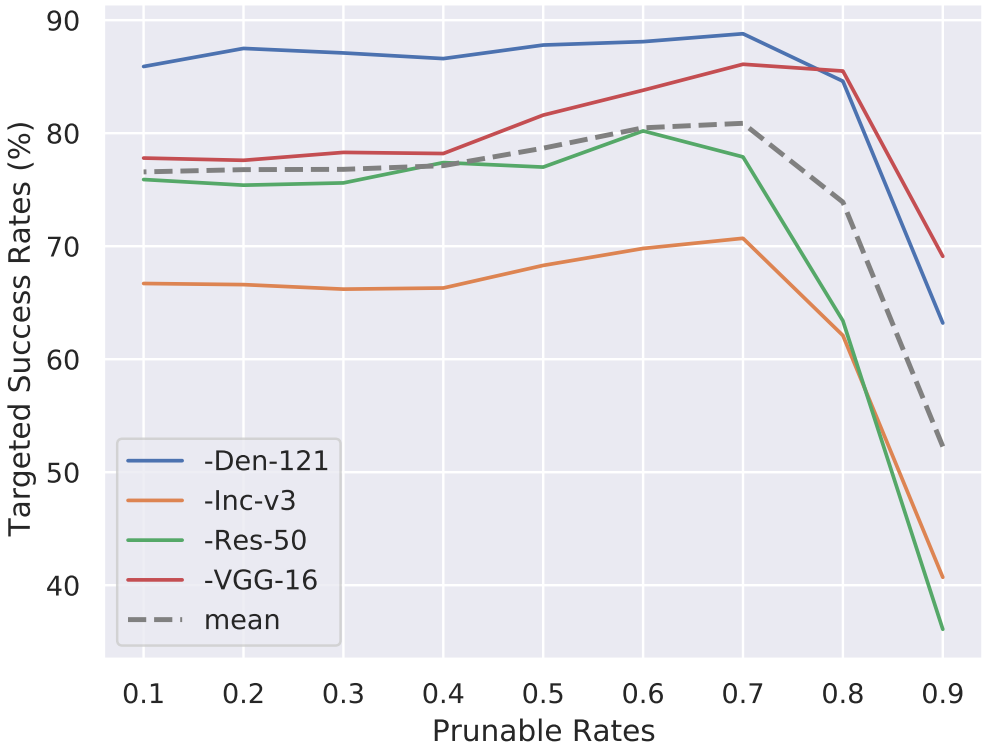


Figure A.1: **The targeted success rates under different prunable rates r on each black-box model.** Each curve shows the trade-off between the diversity and stability of pruned models. The curve for mean targeted success rates reaches its maximum at $r = 0.7$.

[12]. The group of CNNs was popular for evaluating attacks [5, 6, 22, 45, 46]. However, due to similar architectures between these CNNs, [48] suggests using a group of networks with relatively diverse designs. For completeness, we also include the targeted success rates of different model augmentation methods under this group of similar CNNs.

Attacks	-Inc-v3	-Inc-v4	-IncRes-v2	-Res-101	Avg
NI-SI-TI-DI	65.2	71.3	73.2	20.9	57.65
+GN	77.5	70.0	69.0	26.1	60.65
+DSNE	70.7	60.3	69.5	13.7	53.55
+DWP	**83.1**	**86.1**	**85.4**	**40.6**	**73.80**

Table A.1: The targeted success rates of transferring across similar CNN architectures. The "-" prefix stands for the black-box network with the other three serving as the white-box ones for the ensemble. "+" means the participation of a specific model augmentation method. DWP outperforms other leading model augmentation methods GN and DSNE.

doi:10.6342/NTU202301130

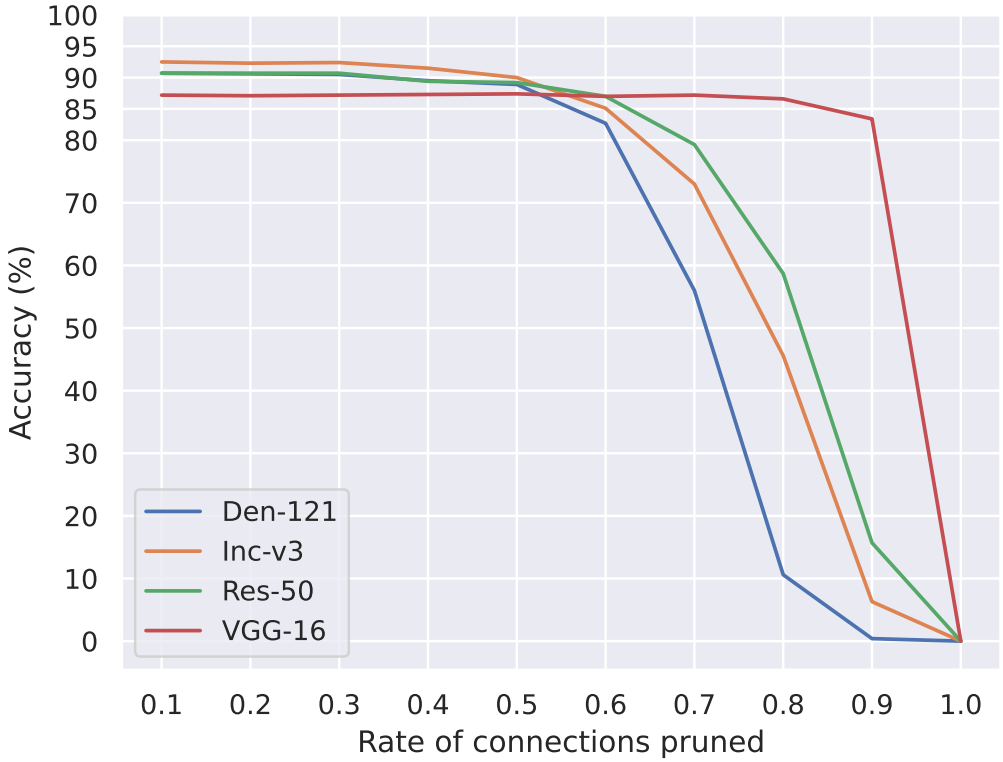


Figure A.2: **The decay on the accuracy of each network with respect to the rate of minor weight connections pruned.**

A.3 Transferring to Multi-Step Adversarially Trained Models

Authors in [42] propose "ensemble adversarial training", which trains the network with adversarial examples generated from external models. While the single-step attack in the procedure is less costly, the models fall short of resisting iterative attacks even in black-box scenarios. Thus, we explore the black-box targeted attack results on the models with multi-step adversarial training.

Transferable targeted attacks from naturally trained CNNs to multi-step adversarially trained networks remain an open problem. Recent attacks only show the non-targeted results [32]. Even the resource-intensive attack [30] fails to achieve satisfied targeted

doi:10.6342/NTU202301130

success rates. We borrow the adversarially trained networks from [33] in the following experiments. Table A.2 shows the failure of transferring targeted attacks from the four naturally trained CNNs to various three-step adversarially trained models.

Despite the frustrating results, there is a different story when we generate adversarial examples on multi-step adversarially trained networks. Even if the victim network undergoing multi-step adversarial training has a different architecture, it remains vulnerable to these attacks. Table A.3 summarizes the targeted attack results of the ensemble composed of Res-18 ($|\epsilon|_\infty = 2$), Res-50 ($|\epsilon|_\infty = 2$) and WideRes-50-2 ($|\epsilon|_\infty = 2$). The two upper groups in Table A.3 report the targeted success rates on different CNN architectures and the norm of ϵ used in adversarial training. We also provide the results on naturally trained CNNs and the ones with ensemble adversarial training. Our DWP stably benefits the results.

Attack Method	NI-SI-TI-DI	+GN	+DSNE	+DWP
Res-18 ($\|\epsilon\|_\infty = 1$)	0.2	0.2	0.5	0.2
Res-50 ($\|\epsilon\|_\infty = 1$)	0.0	0.6	0.8	0.3
WideRes-50-2 ($\|\epsilon\|_\infty = 1$)	0.0	0.2	0.4	0.1
Res-18 ($\|\epsilon\|_2 = 3$)	0.0	0.1	0.1	0.0
Den-121 ($\|\epsilon\|_2 = 3$)	0.0	0.0	0.0	0.0
VGG16 ($\|\epsilon\|_2 = 3$)	0.0	0.0	0.0	0.0
Resnext-50 ($\|\epsilon\|_2 = 3$)	0.0	0.0	0.1	0.0

Table A.2: The targeted success rates of transferring to three-step adversarially trained networks from naturally trained CNNs.

A.4 Untargeted and Single-Model Results

Although our work mainly focuses on the targeted attack setting and ensemble strategy, we provide some untargeted results in Table A.4 and attacks without ensemble in Table A.5.

doi:10.6342/NTU202301130

Attack Method	NI-SI-TI-DI	+GN	+DSNE	+DWP
Res-18 ($\|\epsilon\|_\infty = 1$)	33.2	33.6	21.7	**37.0**
Res-50 ($\|\epsilon\|_\infty = 1$)	40.5	39.4	21.0	**41.4**
WideRes-50-2 ($\|\epsilon\|_\infty = 1$)	37.8	35.4	18.2	**39.5**
Res-18 ($\|\epsilon\|_2 = 3$)	12.6	12.6	12.6	**15.2**
Den-121 ($\|\epsilon\|_2 = 3$)	17.4	18.0	11.3	**19.2**
VGG16 ($\|\epsilon\|_2 = 3$)	12.5	13.3	9.60	**15.5**
Resnext-50 ($\|\epsilon\|_2 = 3$)	19.1	19.2	11.5	**21.0**
Res-50	21.9	16.8	12.6	**22.3**
Den-121	27.6	29.0	15.9	**39.0**
VGG16	8.60	8.80	6.20	**18.6**
Inc-v3	17.4	17.9	8.70	**26.7**
Inc-v3ens3	22.4	23.3	9.30	**30.5**
IncRes-v2ens	22.3	22.6	22.4	**30.0**

Table A.3: The targeted success rates of transferring to three-step adversarially trained networks from the ones with different architectures and ϵ.

A.5 Time Cost of DWP

As Adversarial attacks can be more practical with less overhead, we provide the time cost in Table A.6. The results are from 16 images as a batch and 100 attack iterations. Each cell is the average of five different runs on a single RTX A5000 GPU. With the same number of forwards, DWP incurs little overhead to NI-SI-TI-DI. For reference, Table A.7 summarizes the time cost after adding another CNN into the ensemble, which is higher than applying DWP due to additional forwards.

A.6 Compare with VT, SVRE and IG

VT and SVRE [43, 46] improve the naive logit-averaging ensemble in iterative attacks by reducing the variance of gradients between different iterations and models, respectively. Since the two methods assume the white-box substitute models remain un-

doi:10.6342/NTU202301130

Attack	NI-SI-TI-DI	+DSNE	+GN	+DWP
Inc-v3ens3	80.3	84.1	83.9	**88.0**
IncRes-v2ens	52.7	66.0	62.0	**67.5**
ViT-S-16-224	48.1	**57.7**	52.0	55.0
ViT-B-16-224	52.5	61.4	58.7	**64.8**
Swin-S-224	57.6	65.1	63.3	**66.5**
Swin-B-224	53.9	**62.9**	61.2	62.1
MLP-Mixer	50.1	57.7	54.9	**59.1**
ResMLP	72.7	78.5	77.7	**80.6**
gMLP	44.3	**55.5**	51.0	54.4

Table A.4: Untargeted success rates on adversarially trained models and Non-CNN architectures.

	white-box: Res-50			white-box: VGG16		
	→VGG16	→Den-121	→Inc-v3	→Res-50	→Den-121	→Inc-v3
NI-SI-TI-DI	51.7	75.3	31.6	23.8	26.8	15.0
+DWP	63.0	81.1	42.5	25.3	30.3	16.4
	white-box: Den-121			white-box: Inc-v3		
	→Res-50	→VGG16	→Inc-v3	→Res-50	→VGG16	→Den-121
NI-SI-TI-DI	38.6	24.7	14.4	3.6	3.7	6.4
+DWP	59.2	45.2	31.6	11.6	14.2	18.3

Table A.5: Targeted success rates of transferring to naturally trained CNNs without the ensemble strategy. The "→" prefix stands for the black-box network.

changed throughout attacking, it may not be trivially compatible with the model augmentation methods altering networks at each iteration. Thus, rather than including VT and SVRE, we compare the targeted success rates with them. Moreover, IG [14] uses integrated gradients instead of the traditional ones during attacking. Since the implementation of IG is highly similar to SI, we replace SI in NI-SI-TI-DI with IG and report the results.

We generate the adversarial examples using the four naturally trained CNNs and evaluate the attacks on IncRes-v2ens and ViT-B-16-224. Figure A.3a and Figure A.3b show the comparison between model augmentation methods, VT, SVRE, and IG. Since SVRE and VT have different numbers of gradient calculations than the naive ensemble at each iteration, we take the number of gradient calculations as the horizontal axis following [46]. Model augmentations have higher targeted success rates with additional models in-

doi:10.6342/NTU202301130

Time (sec.)	Res-50	Den-121	VGG16	Inc-v3
NI-SI-TI-DI	10.50	12.26	17.64	13.19
+DWP	10.86	15.87	18.72	15.62

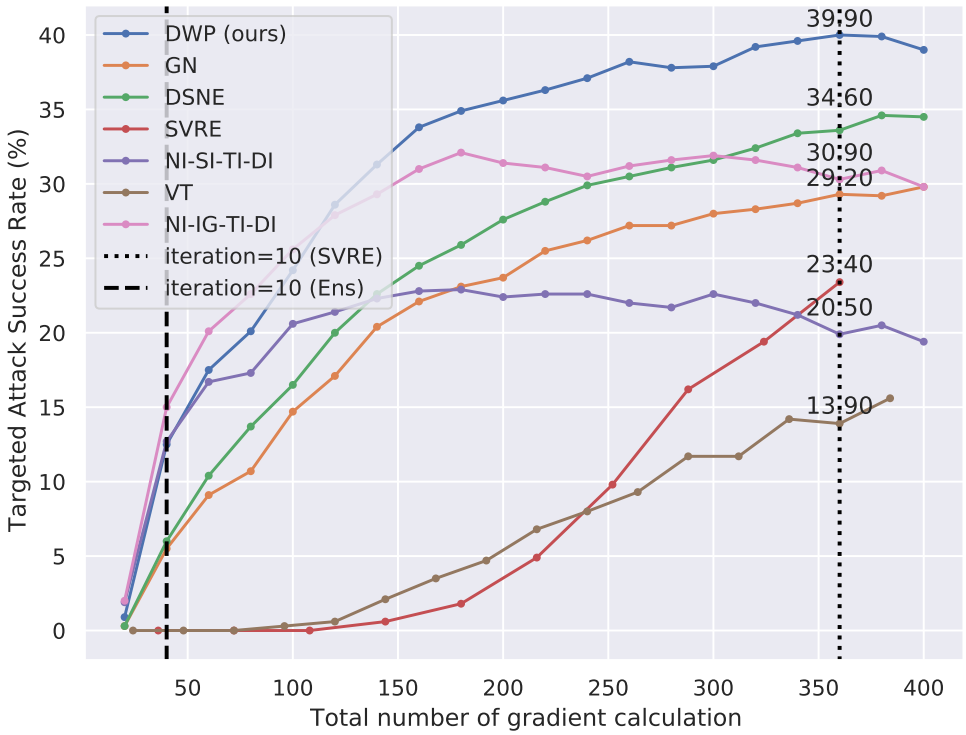
Table A.6: Time cost of NI-SI-TI-DI and DWP on a single CNN.

Time (sec.)	Res-50+VGG16	Res-50+Den-121	Res-50+Inc-v3
NI-SI-TI-DI	24.89	19.63	21.18
Time (sec.)	VGG16+Den-121	VGG16+Inc-v3	Den-121+Inc-v3
NI-SI-TI-DI	26.75	28.07	22.74

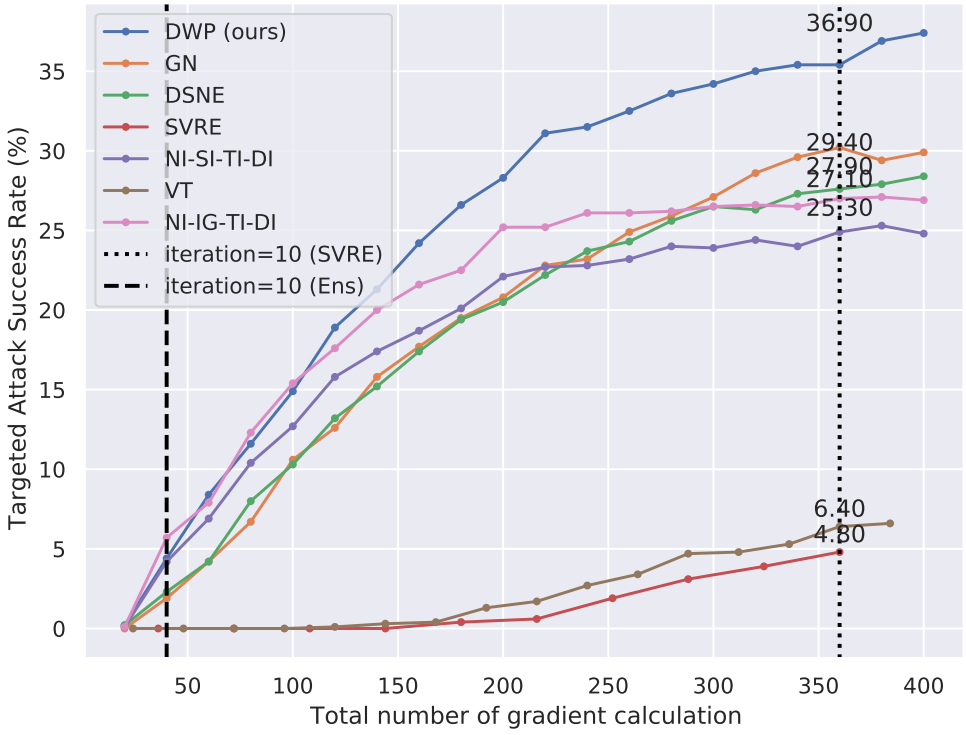
Table A.7: Time cost of NI-SI-TI-DI on the ensemble of two CNNs.

troduced. Our DWP got the highest targeted transferability with protecting the necessary connections.

doi:10.6342/NTU202301130



(a) IncRes-v2ens

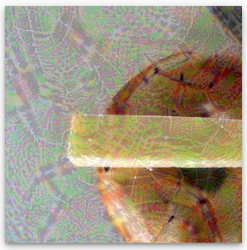


(b) ViT-B-16-224

Figure A.3: **Comparison of targeted transferability between model augmentation methods and VT, SVRE, and IG.**

doi:10.6342/NTU202301130

A.7 Results of DWP on Google Cloud Vision



Insect	91%
Arthropod	90%
Pest	76%
Parasite	72%
Terrestrial Plant	68%
Arachnid	61%

Bagel → Spider

Food	97%
Plum Tomato	87%
Ingredient	87%
Recipe	85%
Natural Foods	84%
Cuisine	82%

Toy Shop → Consomme

Water	97%
Boat	93%
Boats And Boating–Equipment And Supplies	86%
Lake	85%
Outdoor Recreation	85%
Paddle	84%

Mortarboard → Paddle

Bird	96%
Plant	90%
Beak	88%
Twig	83%
Wood	83%
Trunk	80%

Menu → Jay

Performance	59%
Visual Arts	58%
Stage	57%
Tree	57%
Rope	56%
Rock Concert	53%

Dog → Stage

Bird	92%
Phasianidae	88%
Beak	84%
Feather	80%
Chicken	80%
Wild Turkey	79%

Dowitcher → Cock

Plant	91%
Dog Breed	91%
Carnivore	89%
Organism	85%
Terrestrial Plant	84%
Fawn	82%

Butterfly → Dog

Brown	98%
Footwear	98%
Shoe	95%
Outdoor Shoe	87%
Durango Boot	85%
Walking Shoe	84%

Eagle → Geta

Paint	66%
Illustration	65%
Personal Protective Equipment	60%
Measuring Instrument	57%
Helmet	54%
Flesh	50%

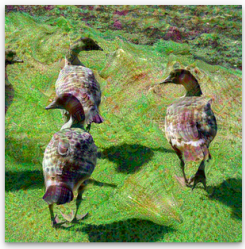
Beetle → Weight Machine

doi:10.6342/NTU202301130

03f90f7138f761e8.png

Plant	96%
Building	92%
Sky	92%
Fence	86%
Mesh	82%
Wire Fencing	80%

Monastery → Fence



7fd7ce3eb2bf9944.png

Snails And Slugs	72%
Wood	71%
Snail	67%
Molluscs	62%
Reptile	62%
Grassland	59%

Goose → Conch

4bb12984b41d0834.png

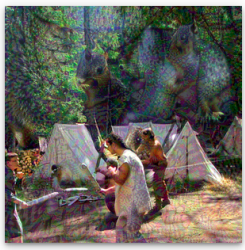
Chicken	84%
Feather	83%
Poultry	82%
Fowl	74%
Livestock	73%
Tail	70%

Turtle → Cock

7c754696067fdfa9.png

Car	95%
Vehicle	93%
Hood	92%
Motor Vehicle	91%
Automotive Lighting	91%
Automotive Design	84%

Rifle → Taxi



6ac94c9244f84aa3.png

Plant	87%
Mammal	85%
Adaptation	79%
Terrestrial Animal	78%
Grass	74%
Snout	73%

Fox → Squirrel

9a1c67f467b9beea.png

Bird	77%
Fish	74%
Tail	72%
Underwater	72%
Marine Biology	71%
Electric Blue	69%

Beetle → Cockatoo

df58f94361c6d105.png

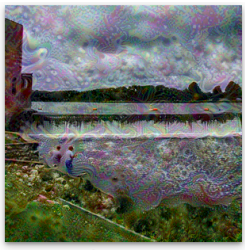
Car	95%
Vehicle	94%
Tire	94%
Wheel	91%
Motor Vehicle	88%
Bird	86%

Jeep → Linnet

296033c3a5910845.png

Event	68%
Grass	68%
Fictional Character	66%
Visual Arts	66%
Mask	66%
Artifact	60%

Otter → Mask



e7cafac174e90143.png

Painting	79%
Marine Biology	75%
Marine Invertebrates	75%
Reef	70%
Sky	70%
Landscape	70%

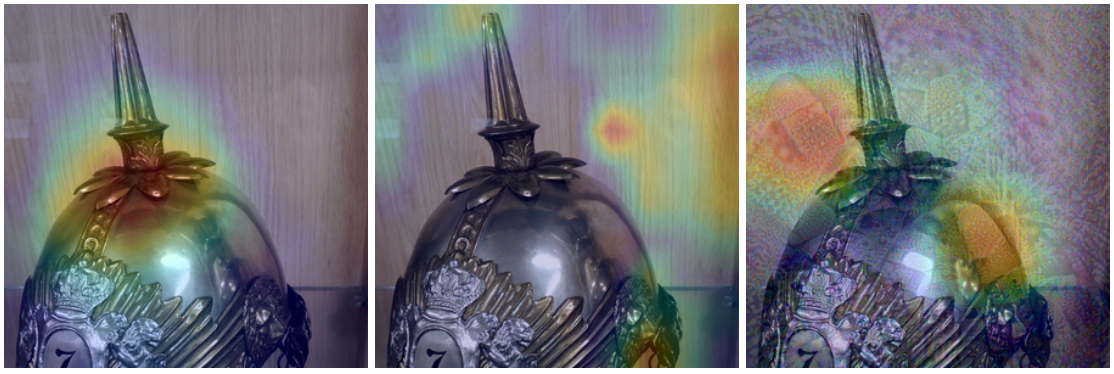
Dam → Sea Slug

doi:10.6342/NTU202301130

A.8 Samples of GradCAM

Clean **Tent** | Clean **Conch** | NI-SI-TI-DI **Conch**

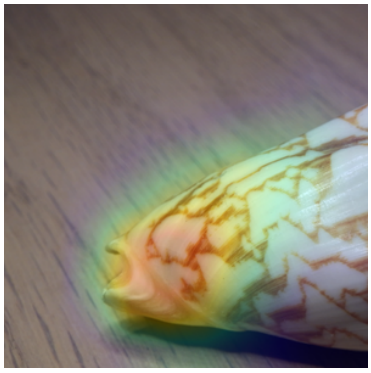
+GN **Conch** | +DSNE **Conch** | +DWP **Conch**



Clean **Pickelhaube** | Clean **Band Aid** | NI-SI-TI-DI **Band Aid**

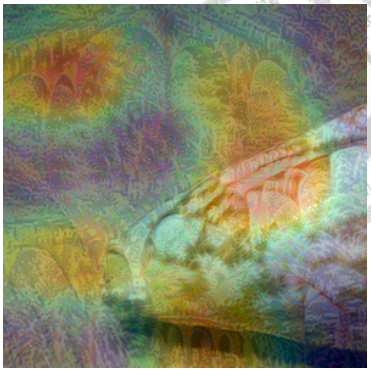
+GN **Band Aid** | +DSNE **Band Aid** | +DWP **Band Aid**

doi:10.6342/NTU202301130



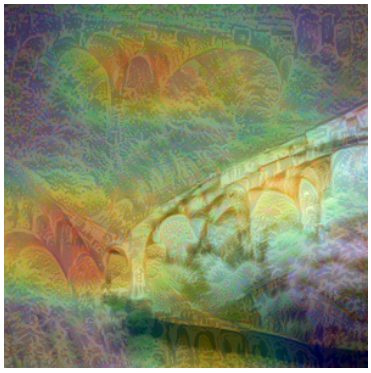
Clean **Conch**

Clean **Viaduct**

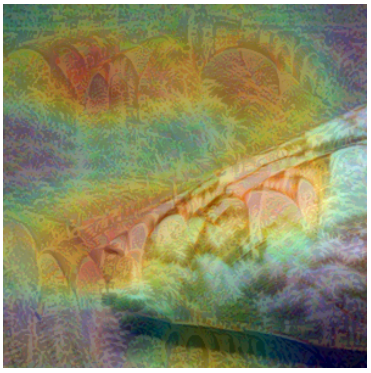


NI-SI-TI-DI **Viaduct**

+GN **Viaduct**



+DSNE **Viaduct**



+DWP **Viaduct**

Clean **Schooner**

Clean **Chainsaw**

NI-SI-TI-DI **Chainsaw**

+GN **Chainsaw**

+DSNE **Chainsaw**

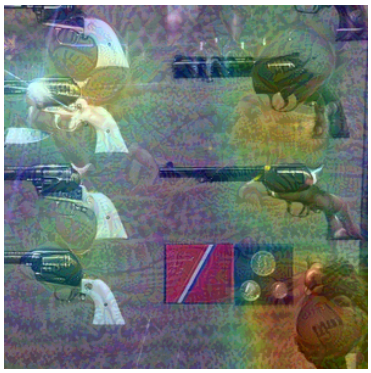
+DWP **Chainsaw**

doi:10.6342/NTU202301130

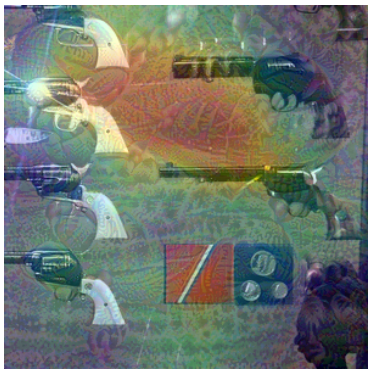
Clean **Revolver**

Clean **Rugby Ball**

NI-SI-TI-DI **Rugby Ball**



+GN **Rugby Ball**



+DSNE **Rugby Ball**

+DWP **Rugby Ball**

Clean **Racing Car**

Clean **Lab Coat**

NI-SI-TI-DI **Lab Coat**

+GN **Lab Coat**

+DSNE **Lab Coat**

+DWP **Lab Coat**

doi:10.6342/NTU202301130