

Universal Adversarial Directions

Ching Lam Choi*, Farzan Farnia†

Abstract

Despite their great success in image recognition tasks, deep neural networks (DNNs) have been observed to be susceptible to universal adversarial perturbations (UAPs) which perturb all input samples with a single perturbation vector. However, UAPs often struggle in transferring across DNN architectures and lead to challenging optimization problems. In this work, we study the transferability of UAPs by analyzing equilibrium in the *universal adversarial example game* between the classifier and UAP adversary players. We show that under mild assumptions the universal adversarial example game lacks a pure Nash equilibrium, indicating UAPs’ suboptimal transferability across DNN classifiers. To address this issue, we propose *Universal Adversarial Directions (UADs)* which only fix a universal direction for adversarial perturbations and allow the perturbations’ magnitude to be chosen freely across samples. We prove that the UAD adversarial example game can possess a Nash equilibrium with a pure UAD strategy, implying the potential transferability of UADs. We also connect the UAD optimization problem to the well-known principal component analysis (PCA) and develop an efficient PCA-based algorithm for optimizing UADs. We evaluate UADs over multiple benchmark image datasets. Our numerical results show the superior transferability of UADs over standard gradient-based UAPs.

1 Introduction

Deep neural networks (DNNs) have achieved great success in many supervised learning problems from computer vision [1], speech recognition [2], natural language processing [3], and computational biology [4]. Their performance, however, has been observed to be highly susceptible to small perturbations to the neural network’s input data widely recognized as *adversarial attacks* [5, 6, 7]. A typical adversarial attack scheme assigns a norm-bounded perturbation to an input feature vector, where the designed perturbation is intended to fool either a known DNN (white-box adversarial attacks) or an unknown DNN (black-box adversarial attacks) to predict a wrong label. Over the recent years, adversarial attack and robust training schemes have received enormous attention in the machine learning community.

An adversarial attack scheme typically designs different perturbation vectors for different input data. This property allows the attack algorithm to tailor the designed perturbation to every specific input sample and further empowers the adversary to attain higher success rates in misleading a DNN machine. On the other hand, the influential study by [8] has empirically shown the existence of a *universal adversarial perturbation (UAP)* that can change the target classifier’s predictions over a significant fraction of input samples. As demonstrated in this work and several other papers on universal perturbations [9, 10, 11, 12], while UAPs are highly constrained across different input data, they still manage to achieve a fair success rate on unseen test data.

While UAPs can successfully attack a target DNN machine, the recent papers [13, 14] have reported that UAPs generated by standard gradient-based methods could weakly transfer to unobserved DNN classifiers different from the source DNN used for their construction. Specifically, the reported results suggest that gradient-based UAPs perform noticeably weaker than standard PGD adversarial perturbations in transferring to an unseen DNN architecture. Such observations motivate the following question:

Why do gradient-based UAPs perform suboptimally in transferring to different DNN classifiers?

*Department of Computer Science and Engineering, The Chinese University of Hong Kong, clchoi1@cse.cuhk.edu.hk

†Department of Computer Science and Engineering, The Chinese University of Hong Kong, farnia@cse.cuhk.edu.hk

The answer to the above question will play a key role in understanding and improving the generalization and transferability properties of universal perturbations. In this work, we focus on the above question and apply a max-min approach to examine the transferability features of UAPs. The applied max-min framework extends the adversarial example game introduced by [15] for generating transferable adversarial examples to the setting of universal adversarial perturbations. According to the adversarial example game, the adversary attempts to find an attack strategy for generating adversarial examples that achieves the maximum success rate against the most robust classifier from a given function space. We show that under some mild assumptions on a DNN architecture, every universal attack scheme can be completely thwarted by the classifier player. From a game-theoretic perspective, the universal adversarial example game possesses no pure Nash equilibria where the players’ deterministic strategies are simultaneously optimal. Consequently, while a gradient-based UAP can significantly drop the performance of a fixed target DNN, the same UAP may have limited impact on a modified DNN function.

To study and address the transferability suboptimality of gradient-based UAPs, we introduce a variant of universal adversarial perturbations which we call *Universal Adversarial Directions (UADs)*. According to the UAD approach, the adversary is only constrained to generate the perturbations along a unique direction in the sample space. Therefore, the UAD perturbations are no longer required to share the same magnitude and could be chosen differently for different input samples. In particular, unlike gradient-based UAPs, the UAD adversary has the freedom of choosing not to perturb an input sample, which sounds a sensible option in the evaluation of a universal adversarial attack scheme.

In order to find an effective UAD, we introduce a bilevel max-max optimization problem and propose a projected gradient-based algorithm to find a stationary solution in its optimization landscape. Moreover, we develop an efficient principal component analysis (PCA)-based approach to approximate the solution to the UAD optimization problem. The PCA-based method indeed finds the top principle component of the matrix of unnormalized fast gradient method (FGM) perturbations to training data. We provide a stochastic optimization algorithm for computing the solution to the PCA-based optimization problem that is suitable for large-scale machine learning problems.

We perform theoretical analysis of the equilibrium properties of the UAD adversarial example game. We show that under the assumption that the Fast Gradient Method (FGM)-perturbation matrix has a unique top principal component, the PCA-based approximate UAD game will possess a Nash equilibrium with a pure strategy for the universal adversary. This result indicates the existence of a single UAD with the maximum impact on the most robust classifier. In the general case, our analysis suggests an extension of the UAD framework to rank-constrained adversarial attacks where the designed perturbations are restricted to a low-rank linear subspace. We show that the rank-constrained adversarial example game will generally possess a Nash equilibrium with a pure adversarial attack strategy.

Finally, we discuss the results of several numerical experiments comparing the performance of UAPs and UADs over standard image datasets and DNN architectures. Our experimental results support the better transferability and generalizability of UADs over gradient-base UAPs. In addition, the numerical results suggest that the designed UAD can be applied as a universal perturbation with a similar or even better performance than gradient-based UAP attack schemes. We can summarize the main contributions of our work as follows:

- Analyzing the transferability of universal adversarial perturbations through the max-min framework of adversarial example games
- Proposing universal adversarial directions (UADs) as an extension of universal adversarial perturbations
- Proving the existence of Nash equilibria with a pure UAD attack strategy in universal adversarial direction games
- Developing an efficient PCA-based algorithm for optimizing UADs
- Conducting an empirical study of the performance of UADs compared to gradient-based UAPs.

1.1 Related Work

Since their introduction by [8], universal adversarial attacks have been extensively studied in the machine learning literature. The related literature includes a large body of papers on generating universal perturbations [16, 17, 18, 19, 13, 20], black-box universal perturbations [21, 22], and on defense methods against universal adversarial attacks [23, 24, 25]. In our work, we focus on the gradient-based universal adversarial perturbations maximizing the perturbed loss function, which as discussed by [24] nicely connects to the bilevel optimization problem of universal adversarial example games. We note that the iterative deepfool-based approach in [8], the singular vector-based approach in [16], and generative model-based method in [13] are indeed different from our analyzed gradient-based UAPs which better match our formulation of UAD optimization problems.

In addition, the equilibrium and convergence properties of adversarial example games have been analyzed in several recent papers. The related works [15, 26, 27] focus on the max-min framework of designing transferable adversarial perturbations. Specifically, [26] prove that the standard adversarial example game generally lacks pure Nash equilibria and proposes finding the mixed Nash equilibria of the adversarial example game. Also, the analysis by [28] focuses on the adversarial training game with standard adversarial attacks. We note that the mentioned papers focus on the standard adversarial attack setting which does not directly apply to universal perturbations. On the other hand, the related papers [24, 29] focus on the sequential game of universal adversarial training where the classifier moves first followed by the universal adversary. While this sequence leads to robust classifiers against universal perturbations, it does not address the max-min game of transferable universal perturbations which we discuss in our work.

In another related work, [16] use the singular vectors of the DNN’s Jacobian matrices at different layers as universal perturbations. On the other hand, our approximate UAD framework chooses the top right-singular vector of the DNN loss’s gradient with respect to training data as a *universal adversarial direction* which allows optimizing the perturbations’ magnitudes unlike [16]’s proposed approach. Similarly, the SVD-based approach by [30] uses the singular vectors of normalized FGSM and PGD perturbations as UAPs and does not focus on the single-direction UAD attacks and its game-theoretic aspects. In addition, we theoretically analyze equilibrium in the UAD adversarial example game. Finally, we note that the SVD-based analyses in [8, 31] target the data matrix’s singular vectors which is different from our work’s PCA-based analysis of the loss’s gradient matrix.

2 Preliminaries

In this section, we review some standard definitions and tools regarding standard and universal adversarial attacks. Throughout the paper, we consider a supervised learning setting where the goal is to predict a label variable $Y \in \mathcal{Y}$ from the observation of a d -dimensional feature vector $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^d$. Given a loss function $\ell(y, y')$ for labels y and y' , the standard empirical risk minimization (ERM) learner aims to find a classifier function $f \in \mathcal{F}$ minimizing the expected prediction loss over a given function space \mathcal{F} .

However, the ERM learner has been observed to lack robustness against norm-bounded adversarial perturbations. To generate a standard norm-bounded adversarial perturbation for classifier f , input (\mathbf{x}, y) , attack norm $\|\cdot\|$, and attack power $\epsilon \geq 0$, the adversary finds an ϵ -norm-bounded perturbation $\delta \in \mathbb{R}^d$ maximizing the prediction loss for input \mathbf{x}, y :

$$\max_{\delta: \|\delta\| \leq \epsilon} \ell(f(\mathbf{x} + \delta), y). \quad (1)$$

In our theoretical analysis, we choose the attack norm function as the standard L_2 (Euclidean) norm. Note that the perturbation designed by solving (1) is a function of input data (\mathbf{x}, y) , which can result in different perturbations for different input samples.

On the other hand, a universal adversarial perturbation (UAP) adds the same perturbation to all input data points. Given n training samples $(\mathbf{x}_i, y_i)_{i=1}^n$, a standard approach to design a UAP is through the following optimization problem maximizing the averaged prediction loss for the universally-perturbed training

data:

$$\max_{\boldsymbol{\delta}: \|\boldsymbol{\delta}\| \leq \epsilon} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i). \quad (2)$$

Given an adversarial attack scheme, an adversarial training method trains the classifier using the generated adversarial examples. As a result, the standard adversarial training method [32] solves the following min-max optimization problem where the perturbations are generated separately for different input data:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\max_{\boldsymbol{\delta}_i: \|\boldsymbol{\delta}_i\| \leq \epsilon} \ell(f(\mathbf{x}_i + \boldsymbol{\delta}_i), y_i) \right] \equiv \min_{f \in \mathcal{F}} \max_{\substack{\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_n: \\ \forall i, \|\boldsymbol{\delta}_i\| \leq \epsilon}} \frac{1}{n} \sum_{i=1}^n [\ell(f(\mathbf{x}_i + \boldsymbol{\delta}_i), y_i)] \quad (3)$$

To perform universal adversarial training through UAPs, [24] introduce the following min-max optimization problem:

$$\min_{f \in \mathcal{F}} \max_{\boldsymbol{\delta}: \|\boldsymbol{\delta}\| \leq \epsilon} \frac{1}{n} \sum_{i=1}^n [\ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i)]. \quad (4)$$

Note that the standard and universal adversarial training problems have different maximization variables, where the maximization variable in the standard adversarial training problem (3) has a size dependent on the training set size n , while the the maximization variable in the universal adversarial training (4) is independent of the number of training examples.

3 Universal Adversarial Example Games

In this section, we aim to analyze the transferability features of UAPs. To do this, we extend the adversarial example game framework introduced by [15] to the setting of universal perturbations. This extension, which we call the *universal adversarial example game*, is based on the following max-min optimization problem searching for the most transferable norm-bounded UAP $\boldsymbol{\delta} \in \mathbb{R}^d$ against the most robust classifier over function space \mathcal{F} :

$$\max_{\boldsymbol{\delta}: \|\boldsymbol{\delta}\| \leq \epsilon} \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i). \quad (5)$$

Note that the above bilevel optimization problem represents a zero-sum game where the UAP player designing an ϵ -norm-bounded universal perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$ moves first followed by the classifier player $f \in \mathcal{F}$ predicting the label from the universally-perturbed feature vector. The solution to this max-min problem provides the most transferable universal perturbation with the highest worst-case impact on the classifiers in \mathcal{F} .

Also, we highlight the difference between the above max-min optimization problem and the min-max problem of universal adversarial training [24, 29]. Although these two problems only differ in the order of minimization and maximization, they do not necessarily share the same solution as the game could lack a pure Nash equilibrium where the deterministic minimization and maximization strategies are simultaneously optimal. In the following theorem, we indeed show that under a mild assumption on classifier space \mathcal{F} which applies to multi-layer DNN architectures, every universal perturbation achieves the same transferability score against the robust classifier, revealing that the min-max and max-min problems have different solutions.

Theorem 1. *Suppose that for every $f \in \mathcal{F}$ and bias vector $\mathbf{b} \in \mathbb{R}^d$ the function $f_{\mathbf{b}} : \mathbb{R}^d \rightarrow \mathbb{R}$ defined as $f_{\mathbf{b}}(\mathbf{x}) := f(\mathbf{x} + \mathbf{b})$ still belongs to \mathcal{F} . Then,*

- *The minimized objective function in (5) over function space \mathcal{F} takes the same value for every choice of $\boldsymbol{\delta} \in \mathbb{R}^d$.*
- *The universal adversarial example game has no Nash equilibria with a non-zero pure strategy $\boldsymbol{\delta}^* \neq \mathbf{0}$ for the UAP adversary.*

Proof. We defer the proof to the Appendix. □

Theorem 1 observes that if the classifier set \mathcal{F} is closed under the addition of an input bias vector, then the effect of a UAP can be reversed by subtracting the UAP using the bias vector. Hence, one can expect that a UAP could struggle in transferring to other DNN classifiers, since its effect is reversible. In next section, we discuss how to improve the performance of universal perturbations by addressing the reversibility of UAPs in the adversarial example game.

4 Universal Adversarial Directions

To address the lack of pure Nash equilibria in the universal adversarial example game, we propose a modified notion of universal perturbations. One of the main factors preventing the UAP game from reaching an equilibrium is the UAP adversary’s restriction to apply the same perturbation to all input data. As a result, a classifier that knows the UAP in advance can easily reverse the effect of the fixed perturbation. Based on this discussion, we propose considering a universal adversary capable of choosing between adding the universal perturbation or not adding the perturbation for every individual sample. Such a universal adversarial attack is therefore only constrained to generate all the perturbations along the same direction. The discussion motivates the definition of a *universal adversarial direction*.

Definition 1. We call a unit-norm δ a universal adversarial direction (UAD) if the adversarial perturbation $\delta(\mathbf{x}, y)$ designed for every input (\mathbf{x}, y) is aligned with δ , i.e. $\delta(\mathbf{x}, y) = \tau_{\mathbf{x}, y} \delta$ holds for a scalar $\tau_{\mathbf{x}, y} \in \mathbb{R}$.

To generate a powerful UAD with the maximum impact on a given classifier f , we propose solving the following optimization problem:

$$\max_{\delta: \|\delta\| \leq 1} \frac{1}{n} \sum_{i=1}^n \left[\max_{\tau_i \in \mathbb{R}: |\tau_i| \leq \epsilon} \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) \right] \equiv \max_{\substack{\delta, \tau_1, \dots, \tau_n: \\ \|\delta\| \leq 1, \forall i: |\tau_i| \leq \epsilon}} \frac{1}{n} \sum_{i=1}^n [\ell(f(\mathbf{x}_i + \tau_i \delta), y_i)] \quad (6)$$

In the above formulation, every scalar variable τ_i represents the magnitude of the additive perturbation $\tau_i \delta$ for the i th data point (\mathbf{x}_i, y_i) . Note that all the perturbations are constrained to be along the optimization variable δ . Taking a standard gradient-based approach to optimize the UAD-based objective function in (6), one can apply the projected gradient method (PGM). Here, the optimization variables $\delta, \tau_1, \dots, \tau_n$ are optimized using the projected gradient ascent algorithm. To derive a stochastic version of the optimization algorithm using a mini-batch of training data at every iteration, we propose Algorithm 1 applying stochastic projected gradient ascent to solve the UAD problem.

5 A PCA-based Approach to Universal Adversarial Directions

In the previous section, we defined UADs and introduced a gradient-based algorithm for solving the UAD optimization problem. However, since the underlying UAD optimization task maximizes a non-concave objective function, the algorithm is only guaranteed to find a first-order stationary solution under regularity assumptions. In this section, we use a Taylor series-based approximation of the UAD optimization objective to relate the optimal UAD to the top principal component of the fast gradient method (FGM) perturbation matrix. This connection results in an analytically tractable optimization problem for approximating the optimal UAD, which facilitates the analysis of UADs.

To build the connection, we focus on the following Lagrangian version of the UAD optimization problem for a coefficient $\lambda > 0$ replacing the role of attack power ϵ in the UAD problem:

$$\max_{\delta: \|\delta\| \leq 1} \frac{1}{n} \sum_{i=1}^n \left[\max_{\tau_i \in \mathbb{R}} \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) - \frac{\lambda}{2} \tau_i^2 \right]. \quad (7)$$

Algorithm 1 UAD-Projected Gradient Ascent

Initialize perturbation δ^0 , stepsizes η_1, η_2 , batch-size B , number of inner updates K

for $t = 0, \dots, T - 1$ **do**

Draw a mini-batch of samples $(\mathbf{x}_{t_i}, y_{t_i})_{i=1}^B$

Initialize magnitudes $\tau_1^0, \dots, \tau_B^0$

for $k = 0, \dots, K - 1$ **do**

$$\forall i : \tau_i^{k+1} = \tau_i^k + \eta_2 \frac{d\ell(f(\mathbf{x}_{t_i} + \tau_i^k \delta^t), y_{t_i})}{d\tau}$$

$$\forall i : \tau_i^{k+1} = \min\{\max\{\tau_i^{k+1}, -\epsilon\}, \epsilon\}$$

end

$$\delta^{t+1} = \delta^t + \frac{\eta_1}{B} \sum_{i=1}^B \nabla_{\delta} \ell(f(\mathbf{x}_{t_i} + \tau_i^K \delta^t), y_{t_i})$$

$$\delta^{t+1} = \frac{\delta^{t+1}}{\max\{1, \|\delta^{t+1}\|\}}$$

end

Output $\delta = \delta^T$

Proposition 1. Suppose that $\ell \circ f$ is a ρ -smooth differentiable function of the input feature vector \mathbf{x} , i.e. for every $\mathbf{x}, \mathbf{x}', y$ we have $\|\nabla_{\mathbf{x}} \ell(f(\mathbf{x}), y) - \nabla_{\mathbf{x}} \ell(f(\mathbf{x}'), y)\| \leq \rho \|\mathbf{x} - \mathbf{x}'\|$. Assuming that $\|\delta\|^2 \leq B$ holds with probability 1 and $\lambda > B\rho$, the following inequalities hold for every sample (\mathbf{x}_i, y_i) :

$$\begin{aligned} \frac{1}{2(\lambda + B\rho)} (\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i))^2 &\leq \max_{\tau_i \in \mathbb{R}} \left\{ \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) - \frac{\lambda}{2} \tau_i^2 \right\} - \ell(f(\mathbf{x}_i), y_i) \\ &\leq \frac{1}{2(\lambda - B\rho)} (\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i))^2 \end{aligned} \quad (8)$$

Proof. We defer the proof to the Appendix. □

The above proposition suggests optimizing the above upper-bound on the UAD optimization objective function approximating the objective function within an error factor of $\frac{\lambda + \rho}{\lambda - \rho}$:

$$\begin{aligned} &\max_{\delta: \|\delta\| \leq 1} \frac{1}{n} \sum_{i=1}^n \left[\ell(f(\mathbf{x}_i), y_i) + \frac{(\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i))^2}{2(\lambda - \rho)} \right] \\ &\equiv \frac{1}{n} \sum_{i=1}^n [\ell(f(\mathbf{x}_i), y_i)] + \frac{1}{2(\lambda - \rho)} \max_{\delta: \|\delta\| \leq 1} \left\{ \delta^\top \left(\frac{1}{n} \sum_{i=1}^n \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i)^\top \right) \delta \right\}. \end{aligned} \quad (9)$$

We observe that the solution to the above optimization problem is indeed the top principal component, i.e. the top right-singular vector, of the following matrix $G_S(f)$ including the loss's gradient for classifier f with respect to training samples in dataset $S = \{(\mathbf{x}_i, y_i)_{i=1}^n\}$:

$$G_S(f) := \frac{1}{\sqrt{n}} \begin{bmatrix} \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_1), y_1) \\ \vdots \\ \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_n), y_n) \end{bmatrix}_{n \times d} \quad (10)$$

Algorithm 2 UAD-Principal Component Analysis

Initialize perturbation δ^0 , stepsize η , batch-size B

for $t = 0, \dots, T - 1$ **do**

Draw a mini-batch of samples $(\mathbf{x}_{t_i}, y_{t_i})_{i=1}^B$

$$\delta^{t+1} = \delta^t + \frac{\eta}{B} \sum_{i=1}^B \left[\left(\delta^{t \top} \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_{t_i}), y_{t_i}) \right) \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_{t_i}), y_{t_i}) \right]$$

$$\delta^{t+1} = \frac{\delta^{t+1}}{\max\{1, \|\delta^{t+1}\|\}}$$

end

Output $\delta = \delta^T$

The above matrix contains the unnormalized fast gradient method (FGM) perturbations as its rows. Note that if we perform a similar first-order approximation analysis for the UAP optimization, the approximate solution will be the mean of the rows of the above matrix. Therefore, according to the above first-order analysis, the UAD framework approximately substitutes the average row of the loss's gradients used by the UAP approach with the gradient matrix's top principal component, which could better capture the existing structures in the FGM-perturbation matrix $G_S(f)$. We note that the tractability of the PCA-based approach is due to the choice of ℓ_2 -norm for the universal direction. For other ℓ_p -norm functions, the computation of the optimal universal direction could be intractable.

Inspired by several recent works applying stochastic optimization methods for computing the top singular vector [33, 34], we propose Algorithm 2 to compute the PCA-based approximation of the optimal UAD. In particular, the stochastic nature of Algorithm 2 suits large-scale machine learning problems where a direct application of the singular value decomposition (SVD) algorithm could be computationally difficult.

6 Nash Equilibria in UAD Games

We previously discussed that UAPs suffer from the lack of equilibria in the universal adversarial example game. In this section, our aim is to show that a similar zero-sum game adapted for our proposed UADs will indeed possess a non-trivial Nash equilibrium, where a fixed non-zero direction is the most effective UAD against any classifier in function space \mathcal{F} the most. To prove such a guarantee, we first define the *universal adversarial direction game* played between an adversary player searching for the most effective direction $\delta \in \mathbb{R}^d$, along which the designed perturbations can mislead the classifier player $f \in \mathcal{F}$. Mathematically, we use the following max-min optimization problem for universal adversarial direction games:

$$\max_{\|\delta\| \leq 1} \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\max_{\tau_i \in \mathbb{R}: |\tau_i| \leq \epsilon} \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) \right]. \quad (11)$$

Following the PCA-based approximation of the UAD optimization problem and defining $L_S(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i), y_i)$ as the averaged prediction loss over unperturbed training data, we can apply Proposition 1 to formulate the approximate universal adversarial direction game with the following optimization problem with parameter $\eta > 0$:

$$\max_{\|\delta\| \leq 1} \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\ell(f(\mathbf{x}_i), y_i) + \eta (\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i))^2 \right] = \max_{\|\delta\| \leq 1} \min_{f \in \mathcal{F}} L_S(f) + \eta \delta^\top G_S(f) G_S(f)^\top \delta. \quad (12)$$

Therefore, the min-max problem corresponding to the above approximate UAD optimization task reduces to the following one-level optimization problem where $\|\cdot\|_2$ denotes the L_2 operator norm:

$$\min_{f \in \mathcal{F}} \max_{\|\delta\| \leq 1} L_S(f) + \eta \delta^\top G_S(f) G_S(f)^\top \delta \equiv \min_{f \in \mathcal{F}} L_S(f) + \eta \|G_S(f)\|_2^2. \quad (13)$$

The following theorem proves that if for every minimizer $f^* \in \mathcal{F}$ of the above objective function, the matrix $G_S(f^*)$ has a unique top singular value, then the approximate universal adversarial example game will possess a Nash equilibrium with a pure strategy for the UAD adversary.

Theorem 2. *Define the approximate UAD objective function $\mathcal{V}(f) := L_S(f) + \eta \|G_S(f)\|_2^2$. Suppose that the matrix set $\{L_S(f)I_d + \eta G_S(f)G_S(f)^\top : f \in \mathcal{F}\}$, where I_d denotes the identity matrix, is convex and compact. Then,*

- *If for every minimizer $f^* \in \mathcal{F}$ of $\mathcal{V}(f)$ the matrix $G_S(f^*)$ has a unique top right-singular vector, there exists a Nash equilibrium to the approximate universal adversarial example game with a pure strategy $\delta^* \in \mathbb{R}^d$ for the UAD adversary.*
- *If for every minimizer $f^* \in \mathcal{F}$ of $\mathcal{V}(f)$ the matrix $G_S(f^*)$ has the top singular value with multiplicity at most r , the approximate universal adversarial example game has a mixed Nash equilibrium where the UAD player always chooses the adversarial direction from a universal r -dimensional space $\Delta_r \in \mathbb{R}^{r \times d}$ spanned by a group of r universal vectors $\{\delta_1^*, \dots, \delta_r^*\}$.*

Proof. We defer the proof to the Appendix. □

The above theorem shows that unlike the UAP adversarial example game, the UAD-based game can indeed possess Nash equilibria with a pure or in general rank-constrained strategy for the universal adversary player. Hence, Theorem 2 indicates that the UAD adversary can apply a non-trivial pure strategy with the maximum impact on the classifier.

7 Numerical Results

We performed several numerical experiments to evaluate the UAD perturbations’ generalizability and transferability on benchmark image datasets including ImageNet & TinyImageNet [35], CIFAR-100 & CIFAR-10 [36], and MNIST [37]. Note that TinyImageNet is a reduced version of standard ImageNet dataset containing 100,000 images from 200 ImageNet classes, with 500 colored images for each class; CIFAR-100 and CIFAR-10 consist of 60,000 colored images from 100 and 10 classes, respectively; MNIST contains 70,000 greyscale handwritten digit images from 10 classes.

In our experiments, we target multiple widely-used DNN architectures including ResNet-18, ResNet-34 (ResNet-50 for ImageNet) [38], DenseNet-121 [39], AlexNet/CaffeNet [1], VGG-19 [40] and the recent EfficientNet-V2-S [41]. The neural network classifiers were trained for 100 epochs using the minibatch gradient descent optimization method with a batch-size of 32, learning rate of 3e-4, and weight decay of 1e-4, which were chosen using cross-validation as detailed in the Appendix.

We generated universal perturbations using the stochastic optimization methods for UAD, UAD-grad, gradient-based UAP, generative adversarial perturbation (GAP) [18], generalized universal adversarial perturbation (GUAP) [42] and cross-domain attack (CDA) [43], with bounded norms controlled by parameter $\epsilon = 0.1 \cdot \mathbb{E}_{\hat{P}}[\|\mathbf{X}\|_2]$ (fraction of the mean L_2 -norm of clean training samples).

Universal Perturbations’ Effectiveness. We performed attacks on the aforementioned DNN models with perturbations generated via UAD (PCA-based Algorithm 2), UAD-grad (gradient-based Algorithm 1), UAD-mag1, gradient-based UAP, GAP, CDA and GUAP. UAD-mag1 represents the performance of UADs when directly applied with a constant unit $\tau_i = 1$ magnitude onto test samples (therefore not optimized during inference time), which are again more effective than UAPs. In Table 1, we used an attack power of $\epsilon = 0.1 \cdot \mathbb{E}_{\hat{P}}[\|\mathbf{X}\|_2]$ to compare the strength of different adversarial attacks; we report the resultant test accuracies and fooling rates (% of labels flipped by the attack) across datasets.

Natural and Test Adversarial Accuracies (TA) & Fooling Rates (FR)													
Model		ResNet-18		ResNet-34 (50)		DenseNet-121		CaffeNet		VGG-19		EfficientNet-V2-S	
		TA ↓	FR ↑	TA ↓	FR ↑	TA ↓	FR ↑	TA ↓	FR ↑	TA ↓	FR ↑	TA ↓	FR ↑
CIFAR-10	Natural	92.4	0.0	91.9	0.0	92.2	0.0	85.7	0.0	87.9	0.0	86.9	0.0
	UAD	11.3	88.4	12.6	85.4	10.3	86.9	24.7	74.0	27.5	71.2	27.4	69.4
	UAD-mag1	15.5	84.7	15.5	83.9	12.1	87.8	25.9	73.0	27.6	71.6	35.5	62.2
	UAP	44.7	59.5	49.0	67.2	16.7	81.2	66.0	31.2	65.9	31.9	40.4	56.6
	GAP	38.5	61.1	39.1	59.9	39.2	60.2	61.8	35.2	40.2	57.6	51.7	46.4
	CDA	42.8	56.3	46.0	51.8	25.6	73.8	54.1	40.9	60.7	35.0	51.8	43.5
	GUAP	13.5	86.5	15.2	84.1	10.8	88.0	27.7	71.3	30.8	68.0	42.5	59.4
CIFAR-100	Natural	69.2	0.0	70.0	0.0	72.0	0.0	60.1	0.0	63.8	0.0	57.3	0.0
	UAD	4.0	91.9	8.2	89.3	4.6	97.5	4.0	95.3	5.9	87.0	8.9	84.8
	UAD-mag1	7.7	91.9	9.6	89.4	10.8	88.2	4.6	95.1	7.4	91.8	6.3	84.6
	UAP	15.9	82.7	29.8	67.9	14.6	81.3	13.2	80.7	31.0	64.8	19.8	70.0
	GAP	15.9	83.0	19.8	78.1	19.8	78.6	25.7	70.3	12.5	86.5	19.4	77.7
	CDA	24.5	72.6	28.4	68.1	16.3	82.0	32.3	58.1	38.4	56.3	19.8	77.0
	GUAP	4.8	90.9	8.0	91.3	6.1	93.8	5.2	94.3	6.6	86.3	7.2	86.9
Tiny-ImageNet	Natural	51.5	0.0	51.5	0.0	54.4	0.0	37.4	0.0	29.5	0.0	36.6	0.0
	UAD	1.2	98.2	1.0	98.6	1.6	98.3	3.3	98.1	0.4	99.4	1.3	97.4
	UAD-mag1	1.5	98.1	2.4	97.5	2.4	97.8	4.4	97.2	0.8	98.9	2.7	97.8
	UAP	3.2	93.9	5.2	95.3	6.3	95.1	4.0	89.8	3.1	96.9	2.3	98.3
	GAP	6.2	92.8	3.4	96.3	9.2	90.0	8.4	87.9	4.8	93.1	5.1	93.2
	CDA	4.8	94.5	3.9	95.3	7.7	91.2	9.4	85.8	4.3	93.9	3.3	97.0
	GUAP	1.6	97.9	0.9	98.0	1.8	98.3	4.1	96.5	0.8	99.3	2.6	97.2
ImageNet	Natural	69.8	0.0	76.1	0.0	74.4	0.0	56.5	0.0	74.2	0.0	84.2	0.0
	UAD	4.0	96.7	5.7	94.8	8.9	90.0	4.0	96.3	4.1	94.6	4.3	95.2
	UAD-mag1	4.9	93.6	6.6	92.2	11.7	87.2	6.1	92.6	7.2	92.2	6.1	92.7
	UAP	19.4	71.4	34.6	56.3	22.5	67.9	18.6	76.9	22.5	79.8	10.9	78.8
	GAP	14.4	73.5	19.9	69.0	20.7	69.3	12.0	81.5	19.2	84.0	17.7	71.4
	CDA	20.4	75.2	35.2	61.8	29.3	68.2	14.9	80.1	22.4	77.4	27.1	67.3
	GUAP	4.5	95.2	5.7	93.1	11.5	86.5	4.1	96.2	4.1	96.2	4.3	94.4

Table 1: UAD, UAD-mag1, UAP, GAP, CDA & GUAP perturbations’ effectiveness and fooling rates. The numbers show adversarial test accuracy (the lower the more effective) and fooling rate (the higher the better).

We further report the fooling rates for UAD-PCA, UAD-grad and UAP perturbations with $\epsilon = .1, .05, .02, .01 \cdot \mathbb{E}_{\hat{p}}[\|\mathbf{X}\|_2]$ in the Appendix Tables 3, 4 and 5, for a more fine-grained comparison. In addition to the quantitative scores, we also visualized perturbations generated by the UAD (UAD-PCA), UAD-grad and gradient-based UAP adversaries in Figure 1, with $\epsilon = 0.1 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$ adversarial noise on backbone models; in Figure 4, where horizontal rows correspond to perturbations with decreasing powers (0.1, 0.05, 0.02, 0.01). We see that the UAD adversary creates more regular noise patterns with enhanced semantic locality than UAP.

Transferability. We further benchmarked the transferability (from the source DNN to other target DNNs) capabilities of UAD and UAP adversaries on ImageNet, TinyImageNet, CIFAR-100 and CIFAR-10. We compare UAD and UAP via Table 2a, which shows the cosine similarity scores between perturbations designed for different networks; and via Table 2b, which displays the accuracy-based transferred fooling rates (TFR in (22)) of perturbations when transferred from the source network (for which it was designed) and applied to attack the target network.

As observed from Table 2a, while gradient-based UAPs designed for different DNNs were almost orthogonal to one another, the UADs achieve higher cosine similarity scores across the DNN architectures. Corresponding cosine similarity and TFR results for CIFAR-10, CIFAR-100 and TinyImageNet are included in Tables 6a, 7a & 8a and 6b, 7b & 8b of the Appendix. Finally, in Figures 5, 6, 7 and 8 in the Appendix, we visualized the bar plot of the sorted singular values (descending order) for the attempted datasets and architectures. We observe that the loss’s gradient matrix $G_S(f)$ for the UAD perturbation has always a unique top singular value.

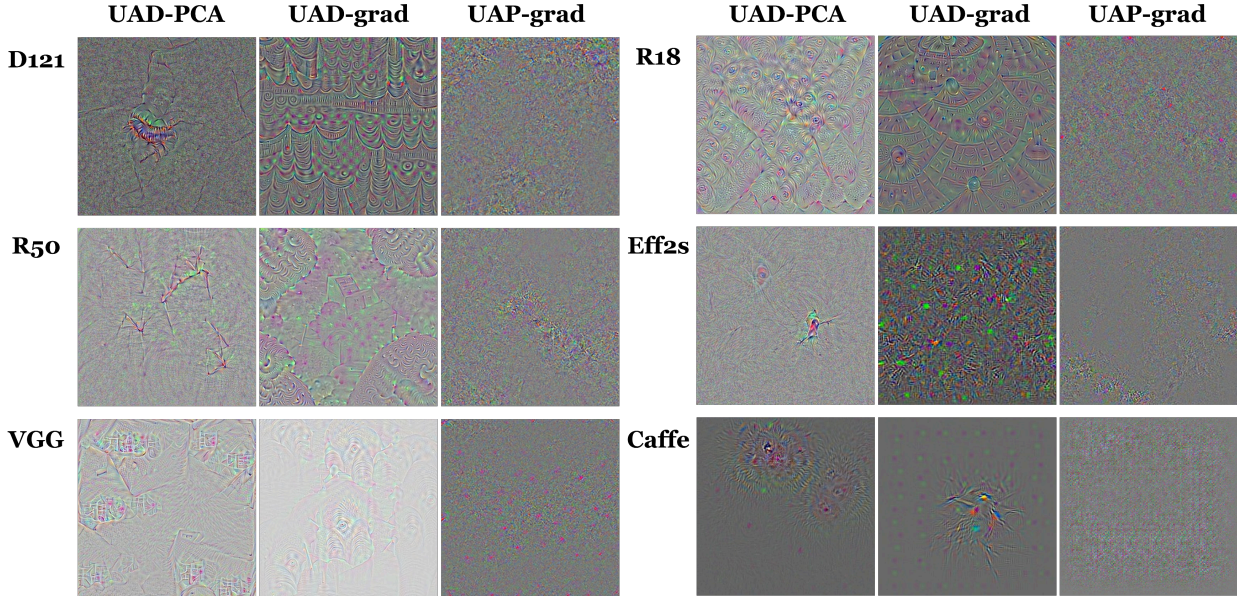


Figure 1: ImageNet visualizations of UAD-PCA, UAD-grad, UAP noise at $\epsilon = 0.1 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$

UAD Cosine Similarities for ImageNet						
Tar / Src	R18	R50	D121	Alex	VGG	Eff2s
R18	1.00	-0.29	0.18	-0.29	0.21	0.25
R50	-0.29	1.00	0.19	-0.24	0.35	0.30
D121	0.18	0.19	1.00	0.17	-0.14	0.13
Alex	-0.29	-0.24	0.17	1.00	-0.25	0.32
VGG	0.21	0.35	-0.14	-0.25	1.00	0.31
Eff2s	0.25	0.30	0.13	0.32	0.31	1.00

UAP Cosine Similarities for ImageNet						
Tar / Src	R18	R50	D121	Alex	VGG	Eff2s
R18	1.00	0.00	-0.01	-0.01	-0.01	0.00
R50	0.00	1.00	0.01	0.00	-0.00	0.00
D121	-0.01	0.01	1.00	-0.01	0.00	0.00
Alex	-0.01	0.00	-0.01	1.00	0.00	-0.01
VGG	-0.01	-0.00	0.00	0.00	1.00	0.00
Eff2s	0.00	0.00	0.00	-0.01	0.00	1.00

UAD TFR for ImageNet						
Tar / Src	R18	R50	D121	Alex	VGG	Eff2s
R18	0.967	0.741	0.566	0.674	0.617	0.603
R50	0.804	0.948	0.512	0.693	0.585	0.770
D121	0.545	0.571	0.900	0.454	0.549	0.710
Alex	0.792	0.762	0.470	0.963	0.649	0.550
VGG	0.698	0.667	0.518	0.674	0.946	0.554
Eff2s	0.795	0.785	0.760	0.452	0.503	0.952

UAP TFR for ImageNet						
Tar / Src	R18	R50	D121	Alex	VGG	Eff2s
R18	0.714	0.620	0.360	0.548	0.511	0.423
R50	0.639	0.563	0.311	0.417	0.412	0.588
D121	0.318	0.352	0.679	0.313	0.350	0.423
Alex	0.631	0.614	0.105	0.769	0.427	0.338
VGG	0.540	0.484	0.222	0.540	0.798	0.252
Eff2s	0.439	0.529	0.523	0.213	0.241	0.788

(a) Cosine similarity scores for UAD & UAP on ImageNet. (b) Transferred fooling rates for UAD & UAP on ImageNet.

Table 2: UAD and UAP cross-network transferability comparison.

8 Conclusion

In this work, we introduced universal adversarial directions (UADs) as a new variant of universal attacks. We provided theoretical evidence that while the universal adversarial example game lacks pure Nash equilibria, the universal adversarial direction game can possess an equilibrium with a pure strategy for the universal adversary. In addition, our numerical results indicate the improved transferability of the UAD adversary in comparison to gradient-based universal perturbations. Our analysis further introduces a potential extension of the UAD framework to rank-constrained adversarial attack and training schemes. Another interesting future direction to our work is to apply the proposed game-theoretic framework to analyze existing generative

model-based adversarial perturbations. Furthermore, analyzing the challenging min-max optimization problem of UADs and their computational complexity is another potential extension of our proposed theoretical framework.

References

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. (Cited on pages 1 and 8.)
- [2] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8599–8603. IEEE, 2013. (Cited on page 1.)
- [3] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020. (Cited on page 1.)
- [4] Michael Wainberg, Daniele Merico, Andrew DeLong, and Brendan J Frey. Deep learning in biomedicine. *Nature biotechnology*, 36(9):829–838, 2018. (Cited on page 1.)
- [5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. (Cited on page 1.)
- [6] Battista Biggio, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. (Cited on page 1.)
- [7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. (Cited on page 1.)
- [8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. (Cited on pages 1 and 3.)
- [9] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 2755–2764, 2017. (Cited on page 1.)
- [10] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*, 2017. (Cited on page 1.)
- [11] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Fari-naz Koushanfar. Universal adversarial perturbations for speech recognition systems. *arXiv preprint arXiv:1905.03828*, 2019. (Cited on page 1.)
- [12] Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdiah Soleymani Baghshah, and Pascal Frossard. Universal adversarial attacks on text classifiers. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349. IEEE, 2019. (Cited on page 1.)
- [13] Atiye Sadat Hashemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt. Transferable universal adversarial perturbations using generative models. *arXiv preprint arXiv:2010.14919*, 2020. (Cited on pages 1 and 3.)

- [14] Sung Min Park, Kuo-An Wei, Kai Xiao, Jerry Li, and Aleksander Madry. On distinctive properties of universal perturbations. *arXiv preprint arXiv:2112.15329*, 2021. (Cited on page 1.)
- [15] Joey Bose, Gauthier Gidel, Hugo Berard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and Will Hamilton. Adversarial example games. *Advances in neural information processing systems*, 33:8921–8934, 2020. (Cited on pages 2, 3, and 4.)
- [16] Valentin Khruikov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8562–8570, 2018. (Cited on page 3.)
- [17] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018. (Cited on page 3.)
- [18] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. (Cited on pages 3 and 8.)
- [19] Hokuto Hirano and Kazuhiro Takemoto. Simple iterative method for generating targeted universal adversarial perturbations. *Algorithms*, 13(11):268, 2020. (Cited on page 3.)
- [20] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. *arXiv preprint arXiv:2102.06479*, 2021. (Cited on page 3.)
- [21] Yusuke Tsuzuku and Issei Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 51–60, 2019. (Cited on page 3.)
- [22] Nurislam Tursynbek, Ilya Vilkovskiy, Maria Sindeeva, and Ivan Oseledets. Adversarial turing patterns from cellular automata. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2683–2691, 2021. (Cited on page 3.)
- [23] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018. (Cited on page 3.)
- [24] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. *arXiv preprint arXiv:1811.11304*, 2018. (Cited on pages 3 and 4.)
- [25] Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. Defending against universal perturbations with shared adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4928–4937, 2019. (Cited on page 3.)
- [26] Laurent Meunier, Meyer Scetbon, Rafael B Pinot, Jamal Atif, and Yann Chevaleyre. Mixed nash equilibria in the adversarial examples game. In *International Conference on Machine Learning*, pages 7677–7687. PMLR, 2021. (Cited on page 3.)
- [27] Zifei Zhang, Kai Qiao, Jian Chen, and Ningning Liang. Based on max-min framework transferable adversarial attacks. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pages 1075–1082. IEEE, 2021. (Cited on page 3.)
- [28] Ambar Pal and René Vidal. A game theoretic analysis of additive adversarial attacks and defenses. *Advances in Neural Information Processing Systems*, 33:1345–1355, 2020. (Cited on page 3.)
- [29] Julien Perolat, Mateusz Malinowski, Bilal Piot, and Olivier Pietquin. Playing the game of universal adversarial perturbations. *arXiv preprint arXiv:1809.07802*, 2018. (Cited on pages 3 and 4.)

- [30] Amit Deshpande, Sandesh Kamath, and KV Subrahmanyam. Universal adversarial attack using very few test examples. 2019. (Cited on page 3.)
- [31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Robustness of classifiers to universal perturbations: A geometric perspective. *arXiv preprint arXiv:1705.09554*, 2017. (Cited on page 3.)
- [32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. (Cited on page 4.)
- [33] Ohad Shamir. A stochastic pca and svd algorithm with an exponential convergence rate. In *International Conference on Machine Learning*, pages 144–152. PMLR, 2015. (Cited on page 7.)
- [34] Ohad Shamir. Convergence of stochastic gradient descent for pca. In *International Conference on Machine Learning*, pages 257–265. PMLR, 2016. (Cited on page 7.)
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. (Cited on page 8.)
- [36] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. (Cited on page 8.)
- [37] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998. (Cited on page 8.)
- [38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. (Cited on page 8.)
- [39] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. (Cited on page 8.)
- [40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. (Cited on page 8.)
- [41] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. (Cited on page 8.)
- [42] Yanghao Zhang, Wenjie Ruan, Fu Wang, and Xiaowei Huang. Generalizing universal adversarial attacks beyond additive perturbations. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1412–1417. IEEE, 2020. (Cited on page 8.)
- [43] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on page 8.)
- [44] Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958. (Cited on page 16.)

Appendix A Proofs

A.1 Proof of Theorem 1

According to Theorem 1's assumption, for every perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$, the following optimization problems are equivalent:

$$\begin{aligned} & \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i) \\ \stackrel{(a)}{\equiv} & \min_{f \in \mathcal{F}, \mathbf{b} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \mathbf{b} + \boldsymbol{\delta}), y_i) \\ \stackrel{(b)}{\equiv} & \min_{f \in \mathcal{F}, \mathbf{b}' \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \mathbf{b}'), y_i) \end{aligned}$$

In the above, (a) is a consequence of the theorem's assumption that for every function $f \in \mathcal{F}$ and bias vector $\mathbf{b} \in \mathbb{R}^d$, $f_{\mathbf{b}} \in \mathcal{F}$ is still a function in \mathcal{F} . Also, (b) follows from the change of variable $\mathbf{b}' = \mathbf{b} + \boldsymbol{\delta}$ in the optimization problem. Since, the equivalent optimization problem has no dependence on perturbation $\boldsymbol{\delta}$, the optimal value of the optimization problem is independent from the choice of $\boldsymbol{\delta}$. Therefore, the proof of the theorem's first part is complete.

We give a proof by contradiction for the theorem's second part. To do this, we suppose that for a pure strategy $\boldsymbol{\delta}^* : \|\boldsymbol{\delta}^*\| \leq \epsilon$ and a general mixed strategy over \mathcal{F} characterized by probability distribution P^* , a Nash equilibrium in the universal adversarial example game is attained. The Nash equilibrium with pure strategy for the universal adversary implies that:

$$\begin{aligned} \max_{\boldsymbol{\delta}: \|\boldsymbol{\delta}\| \leq \epsilon} \mathbb{E}_{f \sim P^*} \left[\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i) \right] & \leq \mathbb{E}_{f \sim P^*} \left[\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}^*), y_i) \right] \\ & \leq \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}^*), y_i). \end{aligned}$$

However, as shown earlier, due to the theorem's assumption on function class \mathcal{F} , for every vector $\boldsymbol{\delta} \in \mathbb{R}^d$ we have

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}^*), y_i) = \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i). \quad (14)$$

Therefore, every $\tilde{\boldsymbol{\delta}} \in \mathbb{R}^d$ satisfies:

$$\begin{aligned} \max_{\boldsymbol{\delta}: \|\boldsymbol{\delta}\| \leq \epsilon} \mathbb{E}_{f \sim P^*} \left[\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i) \right] & \leq \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \tilde{\boldsymbol{\delta}}), y_i) \\ & \leq \mathbb{E}_{f \sim P^*} \left[\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \tilde{\boldsymbol{\delta}}), y_i) \right]. \end{aligned}$$

The above inequalities imply that the function $g^*(\boldsymbol{\delta}) := \mathbb{E}_{f \sim P^*} \left[\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i + \boldsymbol{\delta}), y_i) \right]$ takes the same minimum value for every ϵ -norm-bounded $\boldsymbol{\delta}$. As a result, the existence of a Nash equilibrium with a pure adversary strategy implies that at the optimal classifier strategy every norm-bounded universal perturbation including the zero perturbation leads to a Nash equilibrium, and the minimum averaged loss does not change by adding any non-zero perturbations. This contradiction of attaining a Nash equilibrium with a trivial zero-universal-perturbation completes the theorem's proof.

A.2 Proof of Proposition 1

To show this proposition, note that under the smoothness assumption in the paper, we have:

$$\begin{aligned}
& \ell(f(\mathbf{x}_i), y_i) + \tau_i \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) - \frac{\rho}{2} \tau_i^2 \|\delta\|^2 \\
& \leq \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) \\
& \leq \ell(f(\mathbf{x}_i), y_i) + \tau_i \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) + \frac{\rho}{2} \tau_i^2 \|\delta\|^2.
\end{aligned} \tag{15}$$

As a result, since $\|\delta\|^2 \leq B$, we obtain the followings:

$$\begin{aligned}
& \max_{\tau_i \in \mathbb{R}} \left\{ \ell(f(\mathbf{x}_i), y_i) + \tau_i \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) - \frac{\lambda + B\rho}{2} \tau_i^2 \right\} \\
& \leq \max_{\tau_i \in \mathbb{R}} \left\{ \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) - \frac{\lambda}{2} \tau_i^2 \right\} \\
& \leq \max_{\tau_i \in \mathbb{R}} \left\{ \ell(f(\mathbf{x}_i), y_i) + \tau_i \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) - \frac{\lambda - B\rho}{2} \tau_i^2 \right\}.
\end{aligned} \tag{16}$$

Note that both the upper-bound and lower-bound in the above inequalities represent quadratic optimization problems, where under the assumption that $\lambda > B\rho$, the optimal solutions to the lower-bound and upper-bound optimization problems will be the followings implied by the first-order necessary condition:

$$\tau_i^l = \frac{1}{\lambda + B\rho} \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i), \quad \tau_i^u = \frac{1}{\lambda - B\rho} \delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i). \tag{17}$$

Plugging the optimal solutions into the bounds will lead to the following inequalities:

$$\begin{aligned}
& \ell(f(\mathbf{x}_i), y_i) + \frac{1}{2(\lambda + B\rho)} \left(\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) \right)^2 \\
& \leq \max_{\tau_i \in \mathbb{R}} \left\{ \ell(f(\mathbf{x}_i + \tau_i \delta), y_i) - \frac{\lambda}{2} \tau_i^2 \right\} \\
& \leq \ell(f(\mathbf{x}_i), y_i) + \frac{1}{2(\lambda - B\rho)} \left(\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) \right)^2.
\end{aligned} \tag{18}$$

Therefore, the proof is complete.

A.3 Proof of Theorem 2

Consider the target max-min optimization problem:

$$\max_{\|\delta\| \leq 1} \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\ell(f(\mathbf{x}_i), y_i) + \eta \left(\delta^\top \nabla_{\mathbf{x}} \ell(f(\mathbf{x}_i), y_i) \right)^2 \right] = L_S(f) + \eta \delta^\top G_S(f) G_S(f)^\top \delta.$$

In the above max-min optimization problem, we can use the trace operator to rewrite the objective function as

$$\begin{aligned}
& L_S(f) + \eta \delta^\top G_S(f) G_S(f)^\top \delta \\
& = L_S(f) + \eta \text{Tr}(\delta^\top G_S(f) G_S(f)^\top \delta) \\
& = L_S(f) + \eta \text{Tr}(G_S(f) G_S(f)^\top \delta \delta^\top)
\end{aligned}$$

Here $\text{Tr}(\cdot)$ denotes the trace operator, and the above equality holds since the trace operator is linear and satisfies $\text{Tr}(AB) = \text{Tr}(BA)$ as long as the matrix multiplications AB, BA are well-defined. Based on the

above discussion, we apply a change of variables and define the matrix variable $\Delta = \boldsymbol{\delta}\boldsymbol{\delta}^\top$ which leads to the following equivalent max-min optimization problem:

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \min_{f \in \mathcal{F}} L_S(f) + \eta \text{Tr}(G_S(f)G_S(f)^\top \Delta).$$

Here, \mathcal{S}_+^d denotes the $d \times d$ -positive-semidefinite (PSD) cone, $\|\cdot\|_*$ stands for the nuclear norm, i.e. the sum of a matrix's singular values, and $\text{rank}(\cdot)$ is the rank of a matrix. Note that the constraints on matrix variable Δ requires $\Delta = \boldsymbol{\delta}\boldsymbol{\delta}^\top$ for some vector $\|\boldsymbol{\delta}\| \leq 1$. Also, since $G_S(f)G_S(f)^\top$ and Δ are both PSD matrices, every solution to the above bilevel optimization problem will take the maximum allowable norm value, i.e. $\|\Delta^*\|_* = 1$ or equivalently for a PSD matrix we have $\text{Tr}(\Delta^*) = 1$. Therefore, assuming the loss function only takes non-negative values, the above max-min problem has the same solution as the following max-min problem

$$\begin{aligned} & \max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \min_{f \in \mathcal{F}} L_S(f) \text{Tr}(\Delta) + \eta \text{Tr}(G_S(f)G_S(f)^\top \Delta) \\ &= \max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \min_{f \in \mathcal{F}} \text{Tr}\left((L_S(f)I_d + \eta G_S(f)G_S(f)^\top)\Delta\right). \end{aligned} \quad (19)$$

We can define the equivalent problem using the matrix set $\mathcal{M}_{\mathcal{F}} := \{L_S(f)I_d + \eta G_S(f)G_S(f)^\top : f \in \mathcal{F}\}$:

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \min_{M \in \mathcal{M}_{\mathcal{F}}} \text{Tr}(M^\top \Delta). \quad (20)$$

According to the theorem's assumption $\mathcal{M}_{\mathcal{F}}$ is a convex and compact subset of PSD matrices. Also, note that the objective function $\text{Tr}(M^\top \Delta)$ is bi-linear in PSD matrix variables Δ and M . We also note that the following superset of the maximization problem's feasible set $\{\Delta \in \mathcal{S}_+^d : \|\Delta\|_* \leq 1\}$ is by definition convex and compact. As a result, Sion's minimax theorem [44] implies that the following min-max and max-min problems share a common saddle-point solution (Δ^*, M^*) :

$$\max_{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1} \min_{M \in \mathcal{M}_{\mathcal{F}}} \text{Tr}(M^\top \Delta) = \min_{M \in \mathcal{M}_{\mathcal{F}}} \max_{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1} \text{Tr}(M^\top \Delta). \quad (21)$$

However, note that since the L_2 -operator norm ($\|\cdot\|_2$) and nuclear norms are dual to each other:

$$\max_{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1} \text{Tr}(M^\top \Delta) = \|M\|_2.$$

Therefore, based on the theorem's first assumption that for every minimizer $f^* \in \mathcal{F}$ for the min-max problem, the matrix $G_S(f^*)$ has a unique top singular value or equivalently the PSD matrix $M_{f^*} = L_S(f^*)I_d + \eta G_S(f^*)G_S(f^*)^\top$ has a unique top eigenvalue, then the corresponding maximization solution $\Delta_{f^*}^*$ will be rank-1, as the matrix's nuclear norm needs to be concentrated on the top right-singular vector of $G_S(f^*)$. As a result, there exists a shared solution (f^*, Δ^*) for the min-max and max-min problems in (21) where Δ^* is a rank-1 matrix. Since this solution satisfies the maximization constraints of the original max-min problem in (20) with the maximization feasible set being a subset of the feasible set in (20), (f^*, Δ^*) will also be a solution to (20). Similarly, (f^*, Δ^*) is also a solution to the min-max version of (20), since the max-min inequality implies that

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \min_{M \in \mathcal{M}_{\mathcal{F}}} \text{Tr}(M^\top \Delta) \leq \min_{M \in \mathcal{M}_{\mathcal{F}}} \max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq 1}} \text{Tr}(M^\top \Delta)$$

and in the above min-max problem, matrix M_{f^*} achieves the lower-bound given by the max-min formulation. Thus, (f^*, Δ^*) is a saddle-point for the original max-min optimization problem, and therefore results in a

pure Nash equilibrium to the approximate universal adversarial direction game. Hence, the proof of the theorem’s first part is finished.

Next, under the theorem’s second assumption that for every minimizer $f^* \in \mathcal{F}$ of the min-max problem, the corresponding matrix $G_S(f^*)$ has a top singular value with multiplicity at most r or equivalently the PSD matrix $M_{f^*} = L_S(f^*)I_d + \eta G_S(f^*)G_S(f^*)^\top$ has a maximum eigenvalue with multiplicity at most r , then we have a saddle point solution (Δ^*, M_{f^*}) for (21) where Δ^* is of rank r . Therefore, we assume that the orthonormal unit-norm vectors in $\{\delta_1^*, \dots, \delta_r^*\}$ are the top eigenvectors of Δ^* . Note that the solution (Δ^*, M_{f^*}) will solve the following problem since the maximization problem’s feasible set in the following problem is a subset of the the one in (21).

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq r}} \min_{M \in \mathcal{M}_{\mathcal{F}}} \text{Tr}(M^\top \Delta).$$

In addition, (Δ^*, M_{f^*}) will solve the min-max problem corresponding to the above task, because it achieves the lower-bound coming from the following max-min inequality

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq r}} \min_{M \in \mathcal{M}_{\mathcal{F}}} \text{Tr}(M^\top \Delta) \leq \min_{M \in \mathcal{M}_{\mathcal{F}}} \max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq r}} \text{Tr}(M^\top \Delta).$$

As a result, the rank- r Δ^* combined with a mixed strategy for the classifier player choosing $f \in \mathcal{F}$ results in a mixed Nash equilibrium for the following max-min game:

$$\max_{\substack{\Delta \in \mathcal{S}_+^d: \|\Delta\|_* \leq 1 \\ \text{rank}(\Delta) \leq r}} \min_{P_f \in \mathcal{P}_{\mathcal{F}}} \mathbb{E}_{f \sim P_f} \left[L_S(f) \text{Tr}(\Delta) + \eta \text{Tr}(G_S(f)G_S(f)^\top \Delta) \right].$$

This Nash equilibrium implies the existence of mixed strategies for the universal adversarial direction and classifier players where the universal adversary always generates the perturbation from the rank- r subspace of Δ^* ’s range spanned by orthonormal vectors in $\{\delta_1^*, \dots, \delta_r^*\}$. Therefore, the theorem’s proof is complete.

Appendix B Additional Numerical Experiments

B.1 Additional Numerical Results on UAD-PCA, UAD-grad, and UAP-grad Attacks

Here, we present the complete numerical results for the visualizations of adversarial perturbations generated by UAD-PCA, UAD-grad and UAP-grad in Figure 4. The proposed UAD-PCA and UAD-grad both generated rather semantically meaningful noise patterns, while UAP-grad synthesizes seemingly less meaningful perturbations without a noteworthy pattern. We further report the fooling rates and achieved adversarial test accuracies of the three universal attack algorithms, on CIFAR-10, CIFAR-100 and TinyImageNet datasets. Note that the adversarial test accuracies are presented in Table ??; fooling rates are given in Tables 3, 5, 4 and ??. Furthermore, in histogram Figures 2, 3, we visualize the distribution of optimal τ_i ’s for UAD attacks on various datasets and architectures (including EfficientNetV2-S). We can see that the UAD framework is more expressive than UAP, allowing for both positive and negative τ_i perturbation coefficients with varying strengths between $[-1, 1]$.

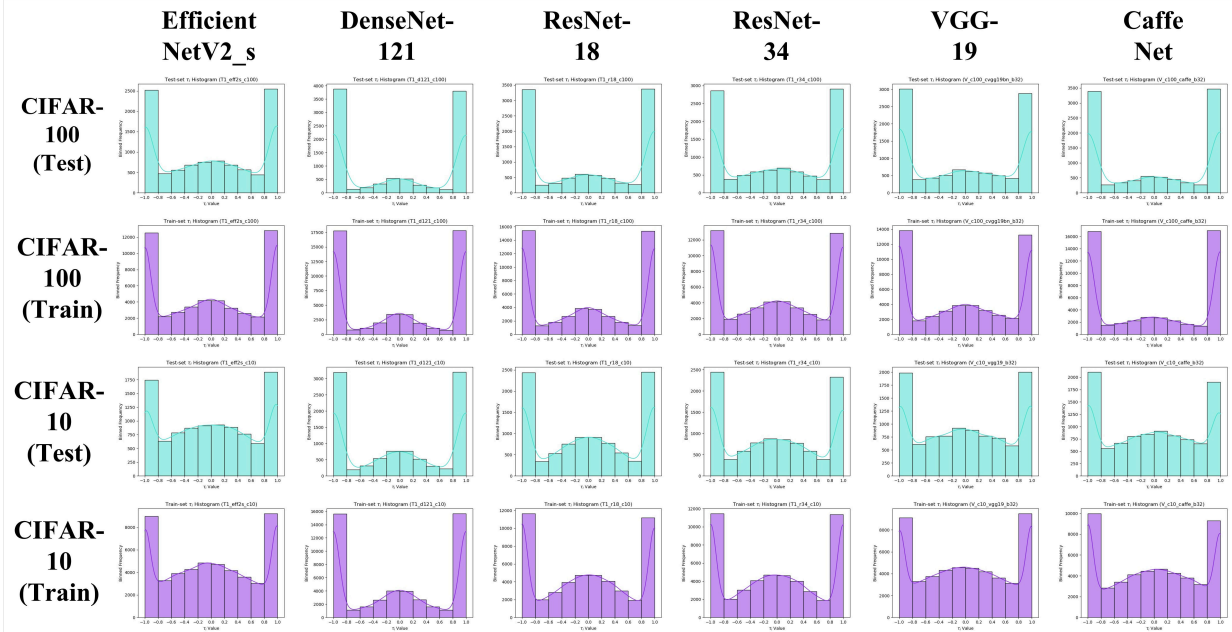


Figure 2: Distribution of $\tau_i \in [-1, 1]$ of UAD attacks on CIFAR-100 and CIFAR-10.

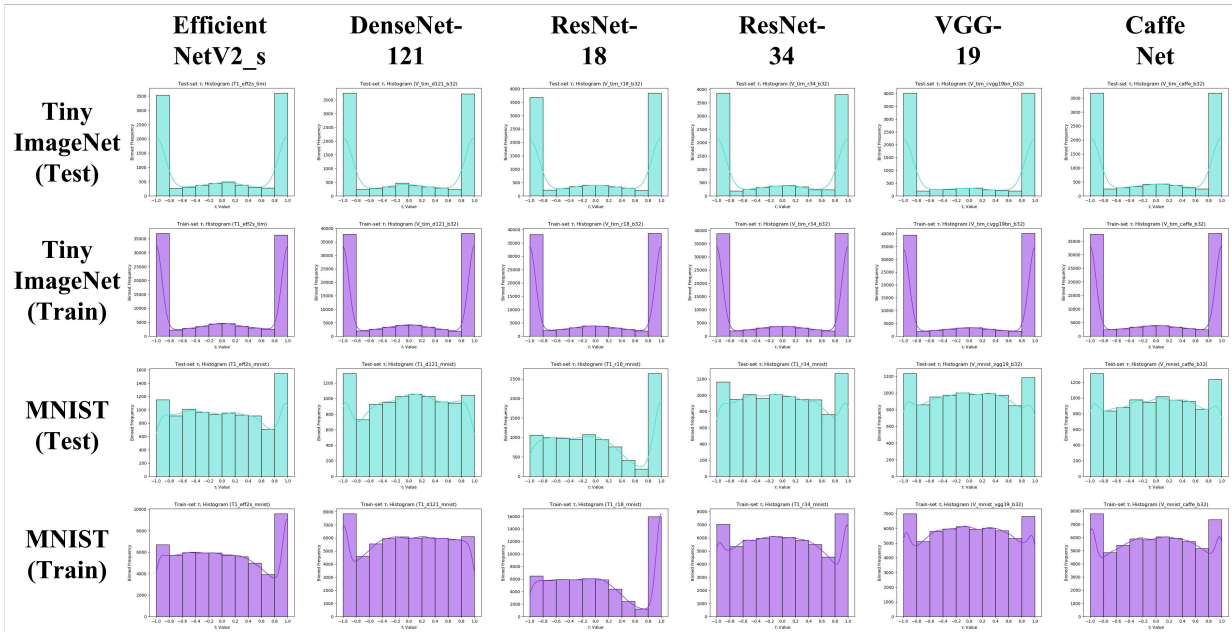


Figure 3: Distribution of $\tau_i \in [-1, 1]$ of UAD attacks on TinyImageNet and MNIST.

B.2 Additional Numerical Results on Generalizability and Transferability of UADs vs. UAPs

To measure the transferability and strength of the perturbations, we quantified the explained transferred fooling rate of UADs and UAPs across different DNN architectures. $\tau_i \delta_a$ is the transferred (designed for a

different model) universal attack’s perturbation for sample \mathbf{x}_i , while $\tau_i \boldsymbol{\delta}_m$ is the original (designed for the same model) attack’s perturbation. Consider the following evaluation measures:

$$\begin{aligned}
 TP &= \sum_i (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) \neq f(\mathbf{x}_i)) \ \& \ (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) = f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_m)) \\
 TN &= \sum_i (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) \neq f(\mathbf{x}_i)) \ \& \ (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) \neq f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_m)) \\
 FP &= \sum_i (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) = f(\mathbf{x}_i)) \ \& \ (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) \neq f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_m)) \\
 FN &= \sum_i (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) = f(\mathbf{x}_i)) \ \& \ (f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_a) \neq f(\mathbf{x}_i + \tau_i \boldsymbol{\delta}_m))
 \end{aligned}$$

We defined and evaluated the following *Transferred Fooling Rate (TFR)* score in our experiments:

$$\text{TFR} = \frac{TP + TN}{TP + FN + TN + FP} \tag{22}$$

Intuitively, TP (true positive / explained and transferred ability to fool) means both the transferred and original adversarial perturbations fool the source model (i.e., the model prediction on the transfer-perturbed sample is not equal to that on unperturbed sample but is equal to that on the original-perturbed sample); TN (true negative / unexplained and untransferred ability to fool) means the unexplained ability to fool the source model; FP (false positive / explained and transferred inability to fool) means neither the original nor transferred adversarial perturbations are able to fool the network; FN (false negative / unexplained and untransferred inability to fool) means the transferred attack neither fools the model nor does it have a prediction that corresponds to the original perturbed result. Since the measurement already accounts for the transferability between models, a higher TFR indicates both better transferability and fooling ability. We measured TFR scores across different DNN architectures and datasets; the results suggest that our proposed UAD attack exhibits higher transferability than gradient-based UAPs.

Finally, singular value decomposition (SVD) was performed on the matrix of the loss function’s gradients with respect to the input image samples on the TinyImageNet dataset, in order to compare the universality of perturbation. These results are visualized in Figures 5, 6, 7, 8. We observed that the loss’s gradient matrix $G_S(f)$ used for generating the UAD perturbations has always a unique top singular value across train and test sets.

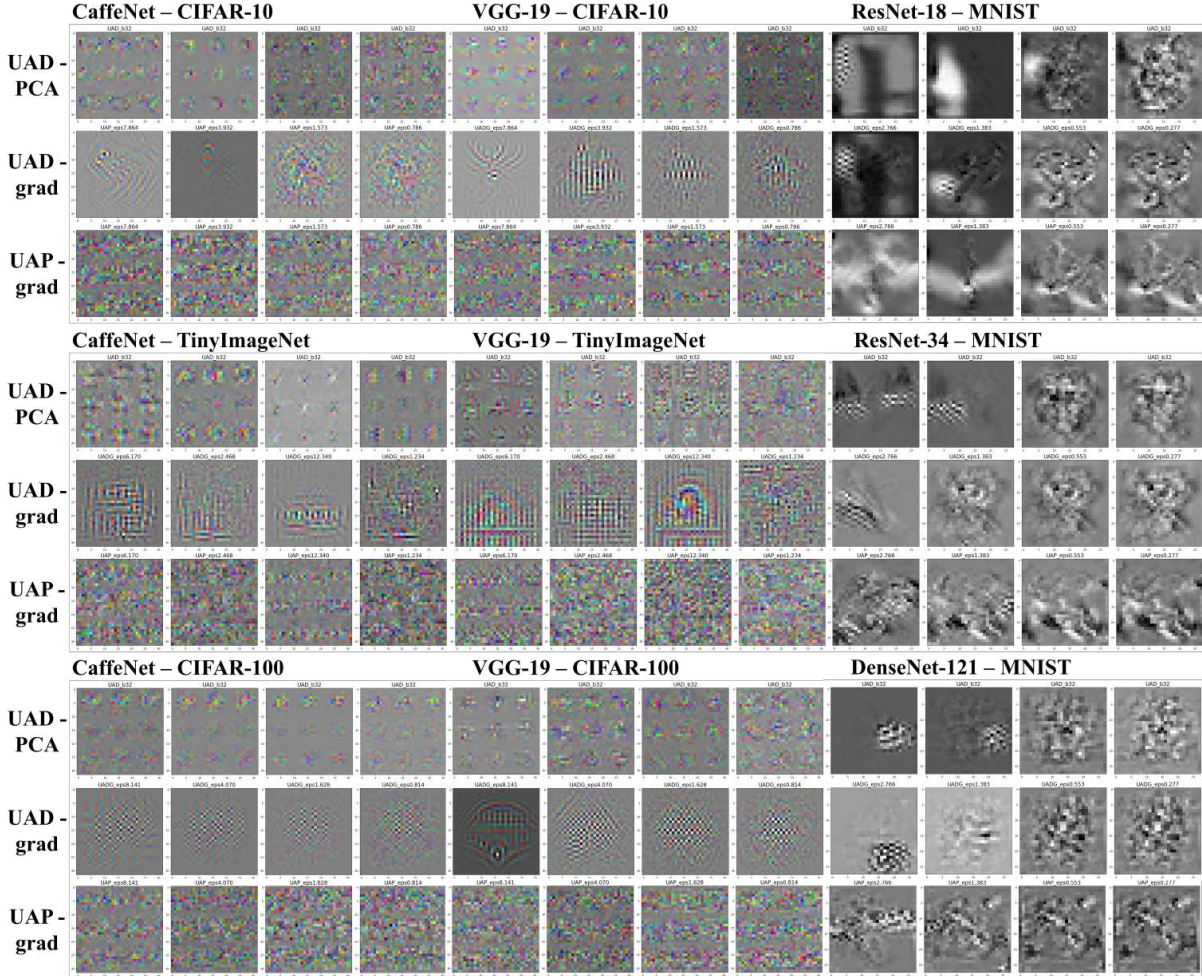


Figure 4: Visualizations of UAD-PCA, UAD-grad, UAP-grad adversarial noise at $\epsilon = 0.1, 0.05, 0.02, 0.01 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$

UAD-PCA Fooling Rates							
Model		ResNet-18	ResNet-34	DenseNet-121	CaffeNet	VGG-19	EfficientNet
Dataset							
CIFAR-10 (Train)	$\epsilon = 0.1$	87.930	85.948	87.230	73.874	70.704	71.670
	$\epsilon = 0.05$	49.860	57.962	77.420	25.788	25.878	28.062
	$\epsilon = 0.02$	6.574	7.286	28.374	10.380	8.544	9.818
	$\epsilon = 0.01$	2.204	3.304	9.338	9.658	5.268	9.170
CIFAR-10 (Test)	$\epsilon = 0.1$	88.350	85.380	86.940	74.010	71.180	69.410
	$\epsilon = 0.05$	51.110	57.500	77.700	24.050	24.720	23.240
	$\epsilon = 0.02$	6.480	8.560	30.340	4.130	8.070	2.910
	$\epsilon = 0.01$	2.080	2.380	10.520	1.350	1.960	1.180
CIFAR-100 (Train)	$\epsilon = 0.1$	91.604	89.034	97.326	95.926	86.738	88.414
	$\epsilon = 0.05$	70.064	61.936	90.630	77.440	68.436	57.102
	$\epsilon = 0.02$	33.416	24.396	45.984	40.958	29.776	28.056
	$\epsilon = 0.01$	13.368	9.822	15.164	30.364	14.846	21.326
CIFAR-100 (Test)	$\epsilon = 0.1$	91.910	89.330	97.540	95.300	86.980	84.840
	$\epsilon = 0.05$	72.130	65.050	91.740	73.990	71.090	49.000
	$\epsilon = 0.02$	40.010	31.130	52.460	30.550	34.970	14.800
	$\epsilon = 0.01$	18.650	13.040	23.730	11.070	12.620	2.980
TinyImageNet (Train)	$\epsilon = 0.1$	98.101	98.358	98.096	98.368	99.023	98.390
	$\epsilon = 0.05$	83.353	82.876	79.080	80.425	93.549	88.526
	$\epsilon = 0.02$	62.006	64.609	59.091	70.242	71.471	69.781
	$\epsilon = 0.01$	56.095	53.904	48.460	66.078	67.022	62.069
TinyImageNet (Test)	$\epsilon = 0.1$	98.160	98.520	98.260	98.100	99.400	97.380
	$\epsilon = 0.05$	77.360	77.660	74.160	60.410	91.950	76.700
	$\epsilon = 0.02$	40.120	48.740	41.280	31.160	42.780	33.790
	$\epsilon = 0.01$	25.920	20.110	19.910	13.680	19.640	9.580

Table 3: UAD-PCA fooling rates with $\epsilon = 0.1, 0.05, 0.02, 0.01 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$.

UAD-grad Fooling Rates							
Model		ResNet-18	ResNet-34	DenseNet-121	CaffeNet	VGG-19	EfficientNet
Dataset							
CIFAR-10 (Train)	$\epsilon = 0.1$	79.820	69.328	85.806	67.570	44.644	67.994
	$\epsilon = 0.05$	15.372	18.410	70.008	13.824	14.610	16.926
	$\epsilon = 0.02$	3.476	4.042	19.782	9.606	5.868	4.638
	$\epsilon = 0.01$	2.126	3.106	4.338	9.430	5.206	1.356
CIFAR-10 (Test)	$\epsilon = 0.1$	80.790	69.760	85.820	64.880	46.830	67.940
	$\epsilon = 0.05$	17.990	20.050	71.240	10.530	14.700	17.950
	$\epsilon = 0.02$	4.900	4.470	21.560	2.490	3.390	5.220
	$\epsilon = 0.01$	1.870	2.150	4.440	1.220	1.430	1.630
CIFAR-100 (Train)	$\epsilon = 0.1$	84.848	78.794	96.980	90.322	83.720	82.636
	$\epsilon = 0.05$	65.176	42.456	85.496	64.512	44.574	47.410
	$\epsilon = 0.02$	22.348	14.436	34.382	31.910	21.004	16.438
	$\epsilon = 0.01$	8.102	7.708	8.442	28.640	13.864	2.664
CIFAR-100 (Test)	$\epsilon = 0.1$	86.080	78.640	97.070	88.380	84.970	83.040
	$\epsilon = 0.05$	67.960	47.440	86.980	59.810	49.160	48.640
	$\epsilon = 0.02$	29.640	20.750	42.230	16.190	24.160	19.740
	$\epsilon = 0.01$	10.110	9.330	16.070	4.360	9.470	3.560
TinyImageNet (Train)	$\epsilon = 0.1$	95.510	95.261	94.634	96.780	97.775	98.279
	$\epsilon = 0.05$	73.192	75.848	72.553	80.195	90.642	85.269
	$\epsilon = 0.02$	57.211	57.009	54.092	66.979	71.367	43.215
	$\epsilon = 0.01$	53.432	53.343	47.967	65.229	66.845	16.424
TinyImageNet (Test)	$\epsilon = 0.1$	95.020	94.700	93.870	95.030	97.980	98.530
	$\epsilon = 0.05$	62.350	65.990	64.280	60.300	87.700	84.070
	$\epsilon = 0.02$	30.910	31.000	33.870	21.480	42.840	42.570
	$\epsilon = 0.01$	18.200	17.920	19.150	8.360	17.120	16.880

Table 4: UAD-grad fooling rates with $\epsilon = 0.1, 0.05, 0.02, 0.01 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$.

UAP Fooling Rates							
Model		ResNet-18	ResNet-34	DenseNet-121	CaffeNet	VGG-19	EfficientNet
Dataset							
CIFAR-10 (Train)	$\epsilon = 0.1$	56.288	66.018	80.908	30.546	29.944	56.814
	$\epsilon = 0.05$	10.394	10.916	66.486	11.812	11.230	5.188
	$\epsilon = 0.02$	3.018	3.960	7.104	9.660	5.366	1.446
	$\epsilon = 0.01$	2.180	3.102	3.692	9.500	5.072	0.662
CIFAR-10 (Test)	$\epsilon = 0.1$	59.500	67.180	81.240	31.160	31.910	56.640
	$\epsilon = 0.05$	12.960	12.300	68.040	7.760	11.030	5.620
	$\epsilon = 0.02$	4.020	4.200	8.980	2.330	2.330	1.740
	$\epsilon = 0.01$	1.520	2.170	3.960	0.940	1.150	0.830
CIFAR-100 (Train)	$\epsilon = 0.1$	81.466	66.408	80.606	82.538	62.346	68.112
	$\epsilon = 0.05$	49.262	31.618	67.475	59.060	37.532	29.208
	$\epsilon = 0.02$	11.462	10.974	22.131	29.412	15.318	4.476
	$\epsilon = 0.01$	7.134	7.208	19.724	28.376	13.364	1.938
CIFAR-100 (Test)	$\epsilon = 0.1$	82.700	67.850	81.305	80.710	64.770	69.960
	$\epsilon = 0.05$	54.460	38.440	68.878	54.580	42.370	34.020
	$\epsilon = 0.02$	17.200	16.640	26.137	9.270	14.780	5.880
	$\epsilon = 0.01$	8.060	8.280	11.253	3.770	6.400	2.570
TinyImageNet (Train)	$\epsilon = 0.1$	94.807	95.835	95.706	93.622	97.230	97.853
	$\epsilon = 0.05$	73.221	75.126	70.545	77.820	87.035	76.720
	$\epsilon = 0.02$	55.989	55.817	51.278	66.454	69.487	31.901
	$\epsilon = 0.01$	52.773	52.894	46.956	65.076	66.724	9.077
TinyImageNet (Test)	$\epsilon = 0.1$	93.870	95.250	95.130	89.770	96.930	98.270
	$\epsilon = 0.05$	61.300	65.480	61.530	56.220	82.040	75.450
	$\epsilon = 0.02$	28.040	27.750	29.340	19.060	36.210	30.800
	$\epsilon = 0.01$	16.860	17.270	16.880	7.730	16.100	8.160

Table 5: Gradient-based UAP fooling rates with $\epsilon = 0.1, 0.05, 0.02, 0.01 \cdot \mathbb{E}[\|\mathbf{X}\|_2]$.

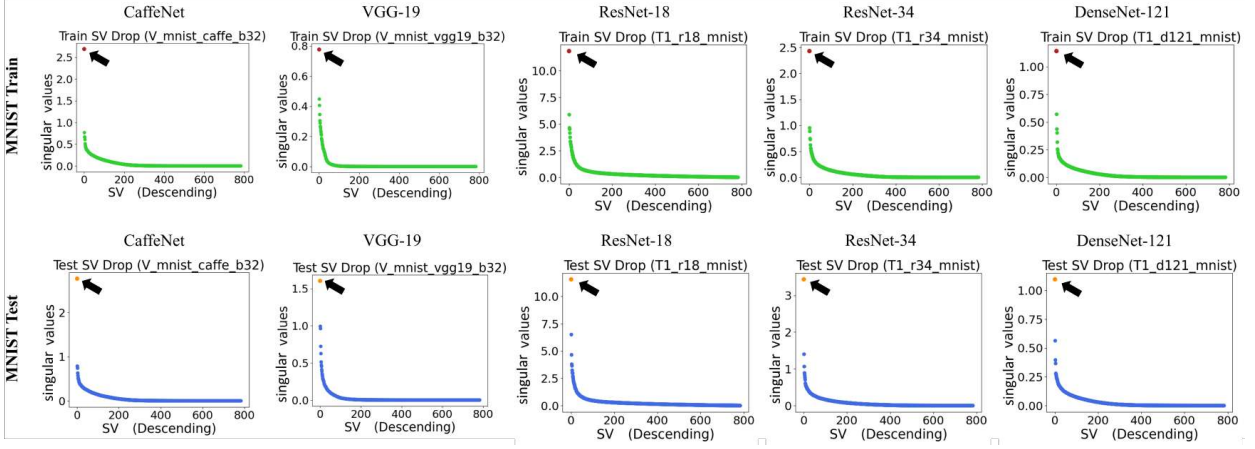


Figure 5: Singular values of $G_S(f)$ on MNIST; top singular value is denoted with an arrow.

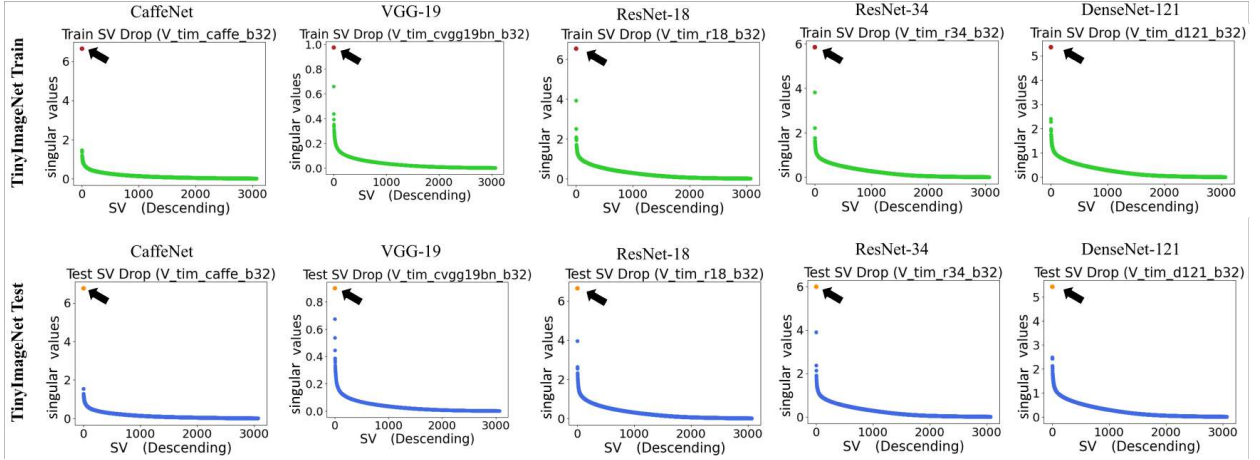


Figure 6: Singular values of $G_S(f)$ on TinyImageNet train and test sets; the top singular value is denoted with an arrow.

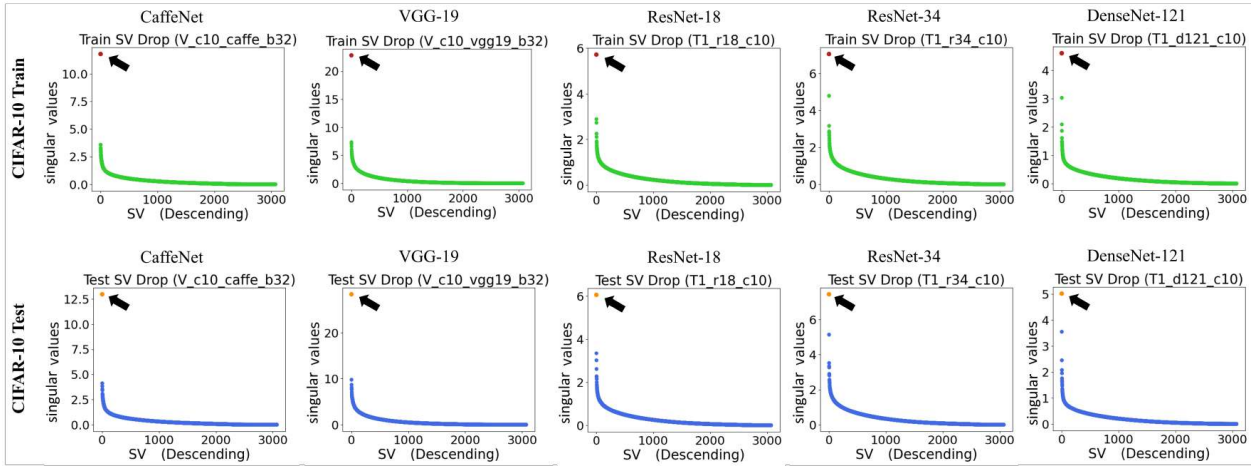


Figure 7: Singular values of $G_S(f)$ on CIFAR-10 train and test sets; the top singular value is denoted with an arrow.

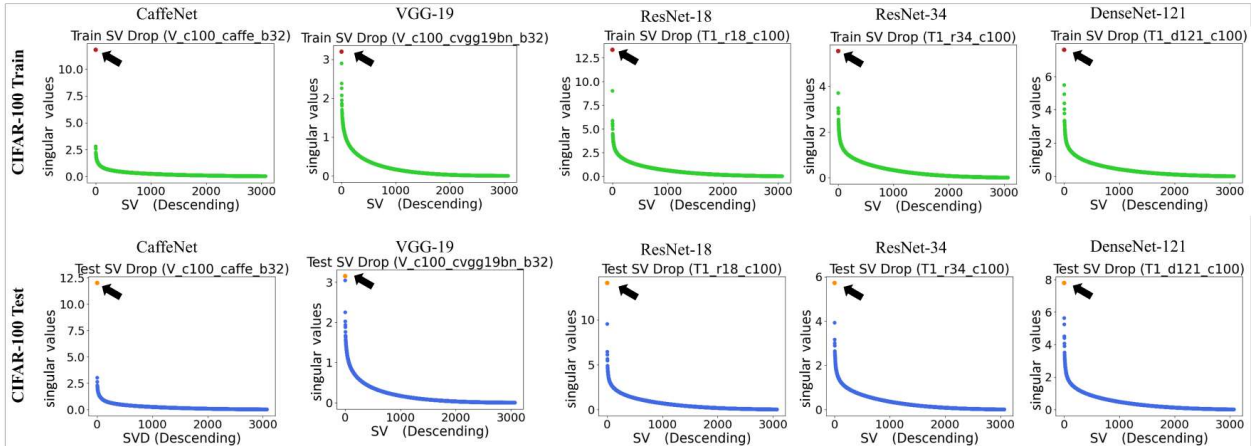


Figure 8: Singular values of $G_S(f)$ on CIFAR-100 train and test sets; the top singular value is denoted with an arrow.

UAD Cosine Similarities for CIFAR-10						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	-0.70	0.54	-0.43	-0.36	-0.16
R34	-0.70	1.00	0.73	0.30	-0.50	-0.41
D121	0.54	0.73	1.00	0.31	-0.20	0.86
Caffe	-0.43	0.30	0.31	1.00	-0.26	0.60
VGG	-0.36	-0.50	-0.20	-0.26	1.00	-0.34
Eff2s	-0.16	-0.41	0.86	0.60	-0.34	1.00

UAP Cosine Similarities for CIFAR-10						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	-0.04	-0.00	-0.02	0.04	-0.06
R34	-0.04	1.00	0.03	0.01	-0.01	0.00
D121	-0.00	0.03	1.00	0.01	-0.00	0.03
Caffe	-0.03	-0.01	-0.01	1.00	-0.03	-0.02
VGG	0.04	-0.01	-0.00	-0.03	1.00	0.06
Eff2s	-0.06	0.00	0.03	-0.02	0.06	1.00

(a) Cosine similarity scores for UAD & UAP on CIFAR-10.

UAD TFR for CIFAR-10						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.884	0.728	0.594	0.417	0.550	0.597
R34	0.772	0.854	0.532	0.406	0.503	0.456
D121	0.788	0.723	0.869	0.411	0.547	0.488
Caffe	0.324	0.320	0.267	0.740	0.447	0.344
VGG	0.295	0.403	0.145	0.241	0.712	0.394
Eff2s	0.321	0.341	0.294	0.231	0.280	0.694

UAP TFR for CIFAR-10						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.595	0.610	0.378	0.298	0.305	0.465
R34	0.483	0.672	0.366	0.282	0.325	0.450
D121	0.481	0.549	0.812	0.333	0.316	0.441
Caffe	0.257	0.281	0.232	0.312	0.273	0.313
VGG	0.199	0.230	0.111	0.183	0.319	0.347
Eff2s	0.105	0.130	0.109	0.094	0.148	0.566

(b) Transferred fooling rates for UAD & UAP on CIFAR-10.

Table 6: UAD and UAP cross-network transferability comparison.

UAD Cosine Similarities for CIFAR-100						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	0.41	-0.53	-0.50	-0.49	-0.23
R34	0.41	1.00	0.38	-0.53	0.41	-0.23
D121	-0.53	0.38	1.00	-0.50	-0.47	-0.13
Caffe	-0.50	-0.53	-0.50	1.00	0.39	-0.37
VGG	-0.49	0.41	-0.47	0.39	1.00	-0.47
Eff2s	-0.23	-0.23	-0.13	-0.37	-0.47	1.00

UAP Cosine Similarities for CIFAR-100						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	0.02	-0.07	0.04	0.03	-0.07
R34	0.02	1.00	-0.01	-0.06	-0.02	0.02
D121	-0.07	-0.01	1.00	0.02	-0.03	-0.02
Caffe	0.04	-0.06	0.02	1.00	0.01	0.01
VGG	0.03	-0.02	-0.03	0.01	1.00	-0.05
Eff2s	-0.07	0.02	-0.02	0.01	-0.05	1.00

(a) Cosine similarity scores for UAD & UAP on CIFAR-100.

UAD TFR for CIFAR-100						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.919	0.770	0.863	0.699	0.708	0.660
R34	0.767	0.893	0.783	0.598	0.732	0.578
D121	0.794	0.763	0.975	0.743	0.773	0.660
Caffe	0.632	0.564	0.688	0.953	0.617	0.547
VGG	0.673	0.642	0.747	0.746	0.870	0.608
Eff2s	0.561	0.520	0.612	0.678	0.693	0.848

UAP TFR for CIFAR-100						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.827	0.724	0.869	0.726	0.688	0.594
R34	0.596	0.679	0.797	0.616	0.620	0.562
D121	0.886	0.713	0.813	0.755	0.653	0.566
Caffe	0.624	0.576	0.701	0.807	0.592	0.505
VGG	0.392	0.636	0.773	0.692	0.648	0.616
Eff2s	0.303	0.365	0.256	0.351	0.389	0.700

(b) Transferred fooling rates for UAD & UAP on CIFAR-100.

Table 7: UAD and UAP cross-network transferability comparison.

UAD Cosine Similarities for TinyImageNet						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	0.59	0.27	0.29	-0.64	-0.52
R34	0.59	1.00	-0.65	-0.39	-0.54	0.58
D121	0.27	-0.65	1.00	0.34	-0.45	0.79
Caffe	0.29	-0.39	0.34	1.00	-0.37	-0.17
VGG	-0.64	-0.54	-0.45	-0.37	1.00	-0.29
Eff2s	-0.52	0.58	0.79	-0.17	-0.29	1.00

UAP Cosine Similarities for TinyImageNet						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	1.00	0.02	-0.03	-0.04	0.02	0.05
R34	0.02	1.00	0.01	0.01	0.04	-0.01
D121	-0.03	0.01	1.00	-0.00	-0.02	0.01
Caffe	-0.04	0.01	-0.00	1.00	-0.00	0.00
VGG	0.02	0.04	-0.02	-0.00	1.00	-0.03
Eff2s	0.05	-0.01	0.01	0.00	-0.03	1.00

(a) Cosine similarity scores for UAD & UAP on TinyImageNet.

UAD TFR for TinyImageNet						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.982	0.956	0.943	0.720	0.887	0.836
R34	0.938	0.986	0.945	0.668	0.878	0.826
D121	0.926	0.914	0.983	0.699	0.836	0.901
Caffe	0.695	0.743	0.760	0.981	0.787	0.883
VGG	0.874	0.886	0.826	0.723	0.994	0.853
Eff2s	0.866	0.915	0.912	0.874	0.877	0.974

UAP TFR for TinyImageNet						
Tar / Src	R18	R34	D121	Caffe	VGG	Eff2s
R18	0.939	0.925	0.932	0.726	0.901	0.853
R34	0.921	0.952	0.903	0.690	0.880	0.832
D121	0.892	0.908	0.951	0.703	0.852	0.768
Caffe	0.808	0.802	0.769	0.898	0.811	0.818
VGG	0.931	0.940	0.909	0.835	0.969	0.801
Eff2s	0.890	0.890	0.831	0.720	0.885	0.983

(b) Transferred fooling rates for UAD & UAP on TinyImageNet.

Table 8: UAD and UAP cross-network transferability comparison.