Stochastic Universal Adversarial Perturbations with Fixed Optimization Constraint and Ensured High-probability Transferability

Yulin Jin[1,5], Xiaoyu Zhang[*3,4], Haoyu Tong[1], Jian Lou[3], Kai Wu[3], Haibo Hu[†1,2], Xiaofeng Chen[3]

[1]Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University.
[2]Research Centre for Privacy and Security Technologies in Future Smart Systems, The Hong Kong Polytechnic University.
[3] State Key Laboratory of ISN, Xidian University, Xi'an, Shaanxi, China.
[4] Key Laboratory of Data and Intelligent System Security Ministry of Education, China.
[5] The State Key Laboratory of Blockchain and Data Security, Zhejiang University.
yulin.jin@connect.polyu.hk, xiaoyuzhang@xidian.edu.cn, haoyu.tong@connect.polyu.hk, louj5@mail.sysu.edu.cn, kwu@xidian.edu.cn, haibo.hu@polyu.edu.hk, xfchen@xidian.edu.cn

Abstract

Adversarial perturbations (APs) have become a great concern in image classification tasks. The most challenging branch, universal adversarial perturbations (UAPs), are exploited to fool most of the unseen samples. Such one-to-all perturbations have the merit of transferability, which has strong practical significance. In this paper, we firstly define the transferability gap and the algorithm stability of the UAP algorithm, and prove the relationship between them. In analyzing the UAP algorithm stability, we prove that the convergence domain of existing UAP algorithms with dynamic constraints is excessively small, which degrades the capacity of UAPs. Thus, we further propose a new expected constraint and prove that UAPs in the expected constraint suit any sample in a high probability. Besides, we propose a Stochastic Universal Adversarial Perturbation (SUAP) that involves additive noise and the expected constraint. Finally, by treating the proposed algorithm as a stochastic differential equation, we prove an upper bound of the UAP algorithm stability of SUAP, which decreases exponentially at the beginning and then increases with a sublinear rate to at most a fixed constant. Experimental results show that SUAP is aligned with our analysis. .

Code — https://github.com/clustering-effect/SUAP

Introduction

Machine learning systems are vulnerable to various attacks (Xiao et al. 2022, 2025b,a; Bai et al. 2025; Wang et al. 2024). In adversarial attacks, numerous studies exploit imperceptible perturbations, commonly known as Adversarial Perturbations (APs), that are delicately crafted by adversarial algorithms to deceive Deep Neural Networks (DNNs) into generating incorrect or intentionally targeted predictions (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016). The most compelling branch among these attacks is the one-to-all Universal Adversarial Perturbations (UAPs) (Moosavi-Dezfooli et al. 2017; Shafahi et al. 2020; Mopuri, Garg, and Babu 2017). The adversary collects a dataset and exploits it in the UAP algorithm to generate UAPs by maximizing average loss across all samples in this dataset (referred to as known samples hereafter), yielding the attack capability to successively deceive more samples outside this dataset (referred to as unknown samples) during inference. On the contrary, many conventional AP algorithms are limited to only affecting a single known sample that has been utilized in their AP algorithms. Therefore, UAPs attract increasing research interest due to their notable characteristic of greater transferability, which enables the attack to be devised in more practical inference scenarios without the prior knowledge of the victim samples beforehand.

So far, existing UAP algorithms can be roughly divided according to their attack purposes (nontargeted or targeted), and threat model (white-box or black-box). The prototype of the nontargeted UAP algorithm (Moosavi-Dezfooli et al. 2017) is devised as the creed *"craft the perturbation at the intersection of the outer of each sample's decision boundary"*. A series of research further formulate the targeted and nontargeted UAP algorithm as a constrained learning problem (Zhang et al. 2020, 2021; Weng et al. 2023; Shafahi et al. 2020). Additionally, tracking the interpretability research of DNNs, another category of UAP algorithms exploits the intermediate output of DNNs to achieve black-box UAPs which are transferable across networks (Mopuri, Garg, and Babu 2017; Mopuri, Ganeshan, and Babu 2018; Khrulkov and Oseledets 2018). While the existence of UAPs has been thoroughly studied (Dezfooli et al. 2017; Jetley, Lord, and Torr 2018), none of these studies presents a thorough theoretical analysis of the transferability of UAP algorithms, despite transferability being the most prominent characteristic of UAPs. **Due to the page limitation, we leave the detailed related work in Appendix C.**

In this paper, we investigate the transferability of UAPs from a theoretical perspective. Analyzing the transferability involves elucidating the capability of UAPs on unknown samples. In stochastic learning, the *generalization gap* (Hardt, Recht, and Singer 2016), which is regarded as a mirror reflection of the transferability of UAPs, identifies the capacity of a learnable parameter on unknown inputs.

[*]*Co-Corresponding author
[†]*Co-Corresponding author

Motivated by the generalization gap, we first well define the *Transferability Gap* to describe the transferability of targeted and nontargeted UAPs. We further prove that the UAP algorithm in the black-box setting could show a poorer transferability gap due to an additional positive term, which is upper bounded by the quality of the model approximated by the adversary, indicating the overfitting of UAP algorithms to the substitute model in the black-box setting.

We demystify the transferability gap by referring to the *algorithm stability* (Hardt, Recht, and Singer 2016; Bousquet and Elisseeff 2002; Bousquet, Klochkov, and Zhivotovskiy 2020), whose expectation is an upper bound of the generalization gap. We consequently define the UAP algorithm stability whose expectation is also proved to be an upper bound of the white-box transferability gap. Existing research usually upper bound the algorithm stability by the Lipschitz assumption, however, the realization of UAP algorithm (Zhang et al. 2025; Moosavi-Dezfooli et al. 2017; Anil, Vinod, and Narayan 2024; Shafahi et al. 2020; Pan, Li, and Yao 2024) incorporates an extra clipping operation that modifies the perturbation in each step, which probably transfers loss to be a non-Lipshitz function. Fortunately, research in convergence of constraint optimization demonstrates that the clipping would attach a reflected vector to the input (Lamperski 2021; Bubeck, Eldan, and Lehec 2015, 2018). Inspired by that, we further prove an upper bound of the UAP algorithm stability which is related to the expected size of the reflected vector, relieving the stress of infinite Lipschitz constant caused by the clipping.

For calculating the expected size of the reflected vector, we revisit the clipping in the realization of existing UAP algorithms (Zhang et al. 2025; Moosavi-Dezfooli et al. 2017; Anil, Vinod, and Narayan 2024; Shafahi et al. 2020; Pan, Li, and Yao 2024), which is determined by the dynamic constraint for adapting UAPs to all samples. We prove that precedent UAP algorithms with dynamic constraint naturally converge in the intersection of all possible constraints, which can be excessively strict or even an empty set. As a result, dynamic constraint could sacrifice the capacity of UAPs as they converge. We alter the dynamic constraint to a fixed ℓ_∞-norm expected constraint, which is sufficiently broad and remarkably overlaps with every single constraint. We prove that UAPs in the expected constraint suit the constraint of any sample with a high probability. In this case, we guarantee an upper bound of the expected size of the reflected vector for UAP algorithms with the expected constraint under a large probability.

Besides, inspired by the field of Langevin sampling, we devise a noisy UAP algorithm that involves a scaled Gaussian noise and the expected constraint, dubbed as **S**tochastic **U**niversal **A**dversarial **P**erturbation (SUAP). We further prove that SUAP enjoys an upper bound of the algorithm stability by utilizing the theory of stochastic differential equation. And we consequently upper bound the white-box transferability gap of the proposed SUAP. **Empirical results in Appendix A.3 shows that the transferability gap of SUAP is aligned with our bound.**

In summary, we make the following contributions:

- We define the transferability gap and the algorithm stability of UAP algorithms. We prove that the white-box transferability gap is upper bounded by the expected UAP algorithm stability (Main result 1, Proposition 1).
- We prove that the dynamic constraint in UAP algorithms leads to a tiny convergence domain, which limits the effectiveness of UAP. (Main result 2, Theorem 1) We further propose an expected constraint that is much broader than the dynamic one, and prove that UAPs in the expected constraint are compatible to any sample in a high probability. (Main result 3, Proposition 2)
- We propose a noisy UAP algorithm, SUAP, which involves additive noise and the expected constraint. We prove that the upper bound of stability of SUAP decreases exponentially at the beginning and then increases to at most a fixed constant. (Main result 4, Corollary 6) **Empirical results in Appendix A show the satisfactory white-box transferability gap of SUAP.**

Transferability Gap of UAP Algorithms

Notion and terminology

In the basic setting of supervised learning, there is an instance space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y}$ denote the input and output space. Any example $z \in \mathcal{Z}$ can be sampled by a population p_z. A dataset $Z = \{z_i = (x_i, y_i) | 1 \leq i \leq n\} \in (\mathcal{X} \times \mathcal{Y})^n$ can be constructed by repeatedly sample z from p_z, the dataset Z then consequently obeys the joint population $p_{z_1,\dots,z_n}$. The learnable parameter of a neural network is $\theta \in \Theta \subset \mathbb{R}^m$, where Θ is the hypothesis space, and a loss function $\mathcal{L} : \Theta \times \mathcal{Z} \mapsto \mathbb{R}^+$ characterizes how poor of θ performs.

In the field of UAPs, the adversary is assumed to hold a generation set $Z_A \in (\mathcal{X} \times \mathcal{Y})^n$ collected by himself to generate a UAP denoted by $r \in \mathcal{X}$ via the attack algorithm $U : \Theta \times (\mathcal{X} \times \mathcal{Y})^n \mapsto \mathcal{X}$ under a series of individual constraints $\mathcal{K}_z$, i.e., $\forall z = (x, y) \sim p_z$, $\mathcal{K}_z = \{r | \|r\|_\infty \leq \varepsilon, x + r \in [0, 1]^d\}$, $r \in \mathcal{K}_z, \varepsilon > 0$. The conditions $\|r\|_\infty \leq \varepsilon$ and $x + r \in [0, 1]^d$ respectively indicate the imperceptible and the image pixel value constraint. $\Pi_z : \mathbb{R}^d \mapsto \mathbb{R}^d$ is the function that clips any input in $\mathbb{R}^d$ into $\mathcal{K}_z$. U is considered as a randomized function due to the random initialization and sampling. During the generation of UAPs, the constraint $\mathcal{K}_z$ is dynamic and specific to the fetched sample. The generation of UAP $r = U(\theta, Z)$ can be formulated as the following optimization problem,

$$r = \underset{r \in [-\varepsilon, \varepsilon]^d}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\theta, x_i + \Pi_{z_i} r, y_i). \tag{1}$$

In the existing literature, it is a common assumption that θ directly originates from the target model in the white-box setting; otherwise, the adversary could train the substitute parameter $\hat{\theta}$ on Z_A to approximate the θ of the target model in the white-box setting.

Transferability gap based on generalization gap

Compared with conventional APs, the most significant characteristic of UAPs is the extraordinary transferability on unseen samples. To analyze the extent of the transferability

a UAP algorithm U can achieve and to understand which factors could influence it, we start with driving the formal definition of transferability. The crux lies in measuring how UAPs behave differently on the generation set Z_A compared to other unseen samples in mathematics. In stochastic learning, the vastly studied generalization gap provides us opportunities to measure such a difference, which identifies the performance gap of a learnable parameter θ on the training set Z and unknown test set. The generalization gap is formulated as the expected difference between the empirical loss $R_Z[\theta] = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\theta, x_i, y_i)$ and the population loss $R[\theta] = \mathbb{E}_z[\mathcal{L}(\theta, x, y)]$ computed over the training set and the whole population including unseen samples. Motivated by the generalization gap, by reckoning UAPs as a sort of learnable parameter, the corresponding empirical loss of UAP algorithm on the generation set Z_A is $R_{Z_A}[\hat{\theta}, r] = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\hat{\theta}, x_i + \Pi_{z_i}(r), y_i)$, and the population loss is $R[\theta, r] = \mathbb{E}_z[\mathcal{L}(\theta, x+\Pi_z(r), y)]$, $r = U(\hat{\theta}, Z_A)$. The clipping function Π_z is used to promote the generated UAP r to fit the individual constraint of any sample. We consequently give the definition of the transferability gap in the same way, as follows.

Definition 1. Given U, p_z, $p_{z_1,...,z_n}$, Z_A with n samples z_i, $1 \le i \le n$, the parameter $\hat{\theta}$ held by the adversary and that of the targeted model θ, and $r = U(\hat{\theta}, Z_A)$, the transferability gap τ is the absolute value of the expected difference

$$\tau \overset{\text{def}}{=} |\mathbb{E}_{U,\hat{\theta},\theta}\mathbb{E}_{z_1,...z_n}[R_{Z_A}[\hat{\theta}, r] - R[\theta, r]]|. \tag{2}$$

Remark 1. Differ from the formulation of the generalization gap, we introduce the additional expectation $\mathbb{E}_{\hat{\theta},\theta}$ here. The physical meaning is that, the transferability gap τ describes "the *expected* performance gap of the *UAP perturbation* $\boldsymbol{r}$ on *seen samples with substitute model* $\hat{\boldsymbol{\theta}}$ *and unseen samples with target model* $\boldsymbol{\theta}$*"*.

Assumption 1. Assuming the architecture of the target model is accessible to the adversary.

Assumption 1 is reasonable, as many popular DNNs have been made open-source and can be easily accessed. In this case, the transferability gap defined τ in Definition 1 covers both white-box and black-box settings depending on whether $\hat{\theta}$ is equivalent to θ. In both settings, a UAP algorithm (whether targeted or nontargeted) with low transferability is expected to perform similarly on the generation set and the other unaccessible samples. On the contrary, a larger transferability indicates poorer expected capability on unseen samples, although it performs well on the generation set Z_A. We denote τ_W and τ_B as the transferability gap in the white-box and black-box settings, and further prove that the black-box transferability gap τ_B is probably poorer than τ_W due to an additional positive item, representing the possible overfitting of UAP algorithm in the black-box setting.

Proposition 1 (Proved in Appendix B.1)**.** *Given U, p_z, $p_{z_1,...,z_n}$, Z_A with n samples z_i, $1 \le i \le n$, $\hat{\theta}$ and θ, $r = U(\hat{\theta}, Z_A)$, and 1-Lipschitz constant $L_{\Theta,1}$ of $\mathcal{L}$ in the hypothesis space Θ, the following inequality holds,*

$$0 \le \tau_B - \tau_W \le L_{\Theta,1} W_1(\hat{\theta}, \theta), \tag{3}$$

Remark 2. The inequality in Proposition 1 requires the existence of 1-Lipschitz constant $L_{\Theta,1}$ for the loss function $\mathcal{L}$. We assert that the requirement is reasonable and easy to satisfy since prevalent UAP algorithms (Zhang et al. 2020, 2021; Weng et al. 2023) tend to select loss functions with bounded gradients.

Assumption 2. $\mathcal{L}$ is 1-Lipschitz both in hypothesis space Θ and input space $\mathcal{X}$, where $|\mathcal{L}(\theta_1, x_1, y_1) - \mathcal{L}(\theta_2, x_2, y_2)| \le L_{\Theta,1}\|\theta_1 - \theta_2\| + L_{\mathcal{X},1}\|x_1 - x_2\|$, $\theta_1, \theta_2 \in \Theta$, $x_1, x_2 \in \mathcal{X}$, $L_{\Theta,1}, L_{\mathcal{X},1} \ge 0$.

The technique to bound the item $W_1(\hat{\theta}, \theta)$ depends on the specific algorithms and assumptions used to train $\hat{\theta}$ and θ. For example, some previous research has investigated the distribution of model parameter trained by the Stochastic Gradient Langevin (SGD) and Stochastic Gradient Langevin Descent (SGLD) algorithms (Azizian et al. 2024; Cheng et al. 2020; Brosse, Durmus, and Moulines 2018; Chen, Du, and Tong 2020). The 1-Wasserstein distance $W_1(\hat{\theta}, \theta)$ then can be bounded in this case. One would say the black-box transferability gap τ_B can be explicitly bounded by substituting the expectation $\mathbb{E}_{\hat{\theta},\theta}$ in Def 1 into the supremum $\sup_{\hat{\theta},\theta}$, however, the bound will be notoriously loose since the parameter space Θ is considerably vast. Bounding $W_1(\hat{\theta}, \theta)$ with determined parameter distributions of $\hat{\theta}$ and θ delivers more strict bounds on black-box transferability gap τ_B. Since the black-box transferability τ_B can be bounded by $\tau_W + L_{\Theta,1}W_1(\hat{\theta}, \theta)$, the remaining task is to bound the white-box transferability gap τ_W where $\hat{\theta} = \theta$.

Bound the White-box Transferability Gap via UAP Algorithm Stability

Construction of UAP algorithm stability

Calculating the white-box transferability gap τ_W is rather a tough task since the population loss $R[\theta, r]$ of the UAP r is hard to approximate. Noteworthy that, by giving a set $\overline{Z}_A = \{\overline{z}_1, ..., \overline{z}_n\}$, where $\overline{z}_i = (\overline{x}_i, \overline{y}_i)$ and z_j are *i.i.d.*, $1 \le i, j \le n$, $Z_A^{(i)} = \{z_1, ..., \overline{z}_i, ..., z_n\}$, $r^{(i)} = U(\theta, Z_A^{(i)})$, the population risk $R[\theta, r]$ can be reconstructed as the following equation shows,

$$R[\theta, r] = \mathbb{E}_{z_1,...,z_n}\mathbb{E}_{\overline{z}_1,...,\overline{z}_n}[\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\theta, \overline{x}_i + \Pi_{\overline{x}_i} r^{(i)}, \overline{y}_i)]. \tag{4}$$

Remark 3. The expectation over $z_1, ..., z_n, \overline{z}_1, ..., \overline{z}_n$ is used implicitly in the generation of the perturbation $r^{(i)}$.

By utilizing Eq.(4), numerous research builds the theory of the algorithm stability which describes the change of a learning algorithm in loss on substituting one training sample. The generalization gap of a learning algorithm is upper bounded by the stability algorithm. Similarly, we define the UAP algorithm stability $\mathcal{S}$ and show a similar relationship between $\mathcal{S}$ and the white-box transferability gap τ_W.

Definition 2. Given U, θ, Z_A and $\overline{Z}_A$ both contain n samples $z_i = (x_i, y_i)$ and $\overline{z}_i = (\overline{x}_i, \overline{y}_i)$, where $\overline{z}_i$ and z_j are *i.i.d.*, $1 \leq i, j \leq n$, $Z_A^{(i)} = \{z_1, ..., \overline{z}_i, ..., z_n\}$, $r = U(\theta, Z_A)$, $r^{(i)} = U(\theta, Z_A^{(i)})$, and 1-Lipschitz constant $L_{\mathcal{X},1}$ of $\mathcal{L}$ in the input space $\mathcal{X}$, the UAP algorithm stability $\mathcal{S}$ is

$$\mathcal{S} \stackrel{\text{def}}{=} \mathbb{E}_{U,\theta}[\frac{1}{n}\sum_{i=1}^{n} L_{\mathcal{X},1}\|\Pi_{\overline{z}_i} r - \Pi_{\overline{z}_i} r^{(i)}\|]. \tag{5}$$

Remark 4. The existing research of algorithm stability is formulated by the difference of the loss. However, they usually further upper bound the algorithm stability by Lipschitz assumption, which is aligned with the form of the stability $\mathcal{S}$ in Def 2. Noteworthy that, we don't need to use $\sup_\theta$ in the definition of $\mathcal{S}$ to trivially upper bound the transferability gap τ_W, as shown in Proposition 2, the stability defined in Def 2 is strong enough to upper bound τ_W.

Proposition 2 (Main result 1, proved in Appendix B.2)**.** *Under the setting in Definition 2 and p_z, $p_{z_1,...,z_n}$, we have*

$$\tau_W \leq \mathbb{E}_{z_1,...,z_n}\mathbb{E}_{\overline{z}_1,...,\overline{z}_n}[\mathcal{S}]. \tag{6}$$

Proposition 2 points out that the white-box transferability gap τ_W can be bounded by the expectation of UAP algorithm stability $\mathcal{S}$. Delivering the UAP algorithm stability $\mathcal{S}$ is based on tackling the distance between two clipped UAPs r and $r^{(i)}$ generated on datasets Z_A and $Z_A^{(i)}$ respectively.

Reconstruct UAP algorithms via reflected vector

To analyze the term $\frac{1}{n}\sum_{i=1}^{n}\|\Pi_{z_i} r - \Pi_{z_i} r^{(i)}\|$, we first investigate the effect of the clipping function Π_z. We focus on the relationship between a generated UAP r and each constraint $\mathcal{K}_z$, $z \sim p_z$, by revisiting the existing UAP algorithms U. The realization of existing UAP algorithms (Zhang et al. 2025; Moosavi-Dezfooli et al. 2017; Shafahi et al. 2020; Pan, Li, and Yao 2024)can be recursively represented by the following equation,

$$r_{k+1} = \Pi_{z_k}(r_k + \mathcal{G}(r_k, z_k)), z_k \in Z_A, \tag{7}$$

where $\mathcal{G}$ outputs an updated vector to modify the UAP r_k in the k-th iteration, *i.e.*, gradients on the cross-entropy or other self-defined loss functions, and the clipping Π_{z_k} varies with the dynamic constraint $\mathcal{K}_{z_k}$. We further use the technique (Lamperski 2021) where the clipping is a function which projects the r_k to itself subtracted by a reflected vector $v_k = r_k - \Pi_{z_k} r_k$, $v_k \in \mathcal{N}_{z_k}$ and $\mathcal{N}_{z_k,r_k} = \{v \in \mathbb{R}^d | \langle v, r' - \Pi_{z_k} r_k\rangle \leq 0, \forall r' \in \mathcal{K}_{z_k}\}$ is the normal cone of the individual constraint $\mathcal{K}_{z_k}$ at $\Pi_{z_k} r_k$. Noteworthy that $\mathcal{K}_z = \{r | \|r\|_\infty \leq \epsilon\} \cap ([0,1]^n - x)$ is a convex set, since $\{r | \|r\|_\infty \leq \epsilon\}$ and $([0,1]^n - x)$ are both convex sets. Therefore, the normal cone $\mathcal{N}_{z_k,r_k}$ exists. Eq.(7) will be reconstructed by Eq.(8) below,

$$r_{k+1} = r_k + \mathcal{G}(r_k, z_k) - v_k, v_k \in \mathcal{N}_{z_k,r_k}, z_k \in Z_A. \tag{8}$$

The UAP algorithm stability S can also be bounded by the inequality Eq.(9).

Corollary 1. *Given the settings in Definition 2, we have*

$$\begin{aligned}\mathcal{S} \leq \mathbb{E}_{U,\theta}[\frac{1}{n}\sum_{i=1}^{n} L_{\mathcal{X},1}(\|r - r^{(i)}\| + \|v_i - v_i^{(i)}\|)],\\ v_i \in \mathcal{N}_{\overline{z}_i,r_i}, v_i^{(i)} \in \mathcal{N}_{\overline{z}_i,r_i^{(i)}}.\end{aligned} \tag{9}$$

Then, the term $\frac{1}{n}\sum_{i=1}^{n}\|\Pi_z r - \Pi_z r^{(i)}\|$ that we are pursuing is now transferred to $\frac{1}{n}\sum_{i=1}^{n}(\|r - r^{(i)}\| + \|v_i - v_i^{(i)}\|)$.

Reflected vector on existing UAP algorithms with dynamic constraint strategy

Considering the ideal situation that r and $r^{(i)}$ are converged and included in all $\mathcal{K}_z$, $z \sim p_z$, the item $\|v_i - v_i^{(i)}\|$ would be negligible. The realization of existing UAP algorithms are devoted to achieving those ideal UAPs. (Zhang et al. 2025; Moosavi-Dezfooli et al. 2017; Shafahi et al. 2020; Pan, Li, and Yao 2024) As shown in Eq.(7) and (8), r_k is restricted to suit all $\mathcal{K}_z$ for fitting all possible $z \in Z_A$. In other words, the existing UAP algorithms clip r_k in a dynamic constraint strategy. An intuition is that such a dynamic constraint strategy tends to push r_k into the intersection of each $\mathcal{K}_z$, $z \in Z_A$ when the algorithm converges. We strictly prove this intuition as shown in Theorem 1 via the Cauchy Convergence Criteria.

Theorem 1 (Main result 2, proved in Appendix B.3)**.** *Given θ, Z_A, the consequence r_t whose evolution complies with Eq.(7). Then if r_t converges, $\lim_{t\mapsto\infty} r_t \in \bigcap_z \mathcal{K}_z$, $z \in Z_A$.*

Theorem 1 indicates if the adversary holds a Z_A with a sufficiently large n, r and $r^{(i)}$ would both in $\bigcap_z \mathcal{K}_z$ and $\|v_i - v_i^{(i)}\|$ is negligible, we formally prove it in Corollaries 2 and 3.

Corollary 2 (Appendix B.4)**.** $\lim_{n\mapsto\infty} \bigcap_{z\in Z_A} \mathcal{K}_z = \bigcap_{z\in\mathcal{Z}} \mathcal{K}_z$.

Corollary 3 (Appendix B.5)**.** $\forall i, \lim_{n\mapsto\infty} \|v_i - v_i^{(i)}\| = 0$.

The constraint $\bigcap_z \mathcal{K}_z$ can be too strict. In fact, in some image classification tasks, i.e., dual-value datasets, $\bigcap_z \mathcal{K}_z$ can easily be a trivial set $\{0\}$. Specifically, images usually include some pure black or white patterns more or less, which leads $\bigcap_z \mathcal{K}_z$ to $\{0\}$. Some examples of the too strict $\bigcap_z \mathcal{K}_z$ on various commonly used datasets are shown in the Appendix A.1. That is, the existing UAP algorithms U could hold good stability provided a sufficiently large generation dataset Z_A, by sacrificing the effectiveness of UAPs. In this case, good UAP algorithm stability $\mathcal{S}$ makes no sense even if the algorithm converges since the UAP algorithm fails to generate an effective UAP although the item $\|v_i - v_i^{(i)}\|$ is small.

Reflected vector on the UAP algorithm with a fixed expected constraint

We consider if there exists a constraint strategy for searching UAPs so that UAPs in the constraint would be clipped

slightly when facing unseen samples. Intuitively, the expectation of $\mathcal{K}_z$ is expected to suits most of samples, denoted as $\mathbb{E}\mathcal{K}$, may be broad enough which satisfies the requirement since each individual constraint $\mathcal{K}_z$ can not deviate from it too much. Denote the maximum and minimum of each component of $\mathcal{K}_z$ as $\mathcal{K}_z^{j,max}$ and $\mathcal{K}_z^{j,min}$, define $\mathbb{E}\mathcal{K}^j = [\mathbb{E}_z[\mathcal{K}_z^{j,min}], \mathbb{E}_z[\mathcal{K}_z^{j,max}]]$, $1 \leq j \leq d$ as the j-th component of the expectation of $\mathcal{K}_z$. Considering $\mathcal{K}_z$ is a random variable, by utilizing McDiarmid's Inequality, the probability that the size of the j-th component v^j of the clipped vector v is larger than a positive constant ϱ decreasing exponentially, which means that most of $\mathcal{K}_z$ overlaps with $\mathbb{E}\mathcal{K}$ in a large extent. Lemma 1 presents an element-wise proof that $\mathcal{K}_z$ can not deviate from $\mathbb{E}\mathcal{K}$ excessively.

Lemma 1 (Proved in Appendix B.6)**.** *Given $\varepsilon > 0$, $\varrho > 0$, marginal probability distribution p_{z^j} of j-th component of z, and a UAP $r \in \mathbb{E}\mathcal{K}$, we have the probability inequality* $\mathbb{P}(\|r^j - \Pi_{\mathcal{K}_z} r^j\| \geq \varrho) = \mathbb{P}(\|v^j\| \geq \varrho) \leq e^{\frac{-2\varrho^2}{\epsilon^2}}$.

By bounding each component of the clipped vector v^j, we further extend the result to v in Theorem 2.

Theorem 2 (Main result 3, proved in Appendix B.7)**.** *Given given $\varepsilon > 0$, $\varrho > 0$ and a UAP $r \in \mathbb{E}\mathcal{K}$, we have the probability inequality* $\mathbb{P}(\|r - \Pi_{\mathcal{K}_z} r\|) = \mathbb{P}(\|v\| \geq \varrho) \leq e^{\frac{-2\varrho^2}{\epsilon^2}}$.

Therefore, the UAP r in the expected constraint $\mathbb{E}\mathcal{K}$ suits any $z \sim p_z$ well. We consequently alter the old-fashioned dynamic constraint strategy as the fixed expected constraint $\mathbb{E}\mathcal{K}$ and analyze the UAP algorithm stability S of UAP algorithms with such a unique constraint. We improve the upper bound of UAP algorithm stability $\mathcal{S}$ in Corollary 1 as shown in Corollary 4.

Corollary 4. *Given the settings in Definition 2, $\varrho > 0$, the following inequality holds with probability* $(1 - e^{\frac{-2\varrho^2}{\varepsilon^2}})^2$*,*

$$\mathcal{S} \leq \mathbb{E}_{U,\theta}[\frac{1}{n}\sum_{i=1}^{n} L_{\mathcal{X},1}(\|r - r^{(i)}\|)] + 2\varrho. \quad (10)$$

Assumption 3. Assume the generation set Z_A contains a large amount of samples so that $\mathbb{E}\mathcal{K} \approx \hat{\mathbb{E}}\mathcal{K}$.

Remark 5. The process for approximating $\mathbb{E}\mathcal{K}$ is shown in Algorithm 1 in Appendix A.2. Corollary 4 can be realized when $\mathbb{E}\mathcal{K}$ can be approximated properly. In this case, the large amount of samples in Z_A is required. Hence, the assumption 3 is necessary. Empirical results in Appendix A.1) show demonstrates the effectiveness of Algorithm 1.

Therefore, under Assumption 3, further analysis focuses on the distance between two UAPs r and $r^{(i)}$ generated on datasets that only have one different sample. Noteworthy that the reflected vector still makes a contribution in the following analysis since it involves in the generation of UAPs with approximated constraint $\hat{\mathbb{E}}\mathcal{K}$. The recursion of UAP algorithms with the approximated constraint $\mathbb{E}\mathcal{K}$ can be modified from Eq.(8), as shown in Eq.(11)

$$r_{k+1} = r_k + \mathcal{G}(r_k, z_k) - v_k, v_k \in \mathcal{N}_{\hat{\mathbb{E}}\mathcal{K}, r_k}, z_k \in Z_A, \quad (11)$$

where $\mathcal{N}_{\hat{\mathbb{E}}\mathcal{K}, r_k} = \{v \in \mathbb{R}^d | \langle v, r' - \Pi_{\hat{\mathbb{E}}\mathcal{K}} r_k \rangle \leq 0, \forall r' \in \hat{\mathbb{E}}\mathcal{K}\}$ is the normal cone of the approximated expected constraint $\hat{\mathbb{E}}\mathcal{K}$ at $\Pi_{\hat{\mathbb{E}}\mathcal{K}} r_k$.

UAP with Expected Constraint and Ensured High-probability Bounds

The remaining item $\mathbb{E}_{U,\theta}[\frac{1}{n}\sum_{i=1}^{n} L_{\mathcal{X},1}\|r - r^{(i)}\|]$, which is the upper bound of UAP algorithm stability $\mathcal{S}$ in Corollary 4, depends on the realization of the updatable vector $\mathcal{G}$ in Eq.(11). Motivated by the field of Langevin stochastic learning (Bubeck, Eldan, and Lehec 2018; Lamperski 2021; Welling and Teh 2011; Raginsky, Rakhlin, and Telgarsky 2017), which significantly decreases the generalization gap and unifies the learning results by adding scaled noise, here we propose a noisy update vector $\mathcal{G}$ as shown in Eq.(12) which may have a small upper bound of the term $\mathbb{E}_{U,\theta}[\frac{1}{n}\sum_{i=1}^{n} \|r - r^{(i)}\|]$,

$$\mathcal{G}(z_k, r_k) = \eta \nabla_{r_k} \mathcal{L}(\theta, x_k + r_k, y_k) + \sqrt{2|\eta|/\beta} W_k, \quad (12)$$

where η is the learning rate, W_k is a standard Gaussian Noise, and β is a temperature constant. The realization of $\mathcal{G}$ in Eq.(12) resembles the widely used gradient ascent or descent algorithm by adding a noise item (Welling and Teh 2011; Raginsky, Rakhlin, and Telgarsky 2017). The intuition is that, UAPs r and $r^{(i)}$ share the same operation of adding the identical noise during each step of their evolution, so they are probably similar. With additive noise, the recursion of UAP can be deemed as a discrete Stochastic Differential Equation (SDE), where we can bound the distance between UAPs r and $r^{(i)}$ also in terms of SDE. Noteworthy that the realization of $\mathcal{G}$ represents the category of nontargeted or targeted UAP algorithms relying on whether η is positive or negative. The recursion in Eq.(11) under the proposed realization of $\mathcal{G}$ is shown in Eq.(13),

$$r_{k+1} = r_k + \eta \nabla_{r_k} \mathcal{L}(\theta, x_k + r_k, y_k) + \sqrt{2|\eta|/\beta} W_k - v_k, \quad v_k \in \mathcal{N}_{\hat{\mathbb{E}}\mathcal{K}, r_k}, z_k \in Z_A. \quad (13)$$

We name the proposed UAP algorithm in Eq.(13) as **S**tochastic **U**niversal **A**dversarial **P**erturbation (SUAP). The algorithmic pseudo-code of nontargeted SUAP is presented in Algorithm 2 in Appendix A.2. In the rest of the section, we prove that SUAP has a concrete upper bound of UAP algorithm stability $\mathcal{S}$ which decreases with the iteration k. Specifically, $\mathcal{S}$ of SUAP can be upper bounded by the following four steps: **1)** Construct the continuous-time SUAP as an SDE, and extend the discrete SUAP to the continuous-time filtration; **2)** Bound the expected distance between discrete SUAP and continuous-time SUAP; **3)** Construct a monotonically increased function h on the item $\frac{1}{n}\sum_{i=1}^{n} \|r - r^{(i)}\|$ in continuous-time SUAP via $Itô$'s Lemma. Thus obtain the conditions of the function h to decrease exponentially with time; **4)** Prove the existence of the function which compels the UAP algorithm stability $\mathcal{S}$ of continuous-time SUAP decreases with time. By combing the results of steps 2) and 4), we bound the UAP algorithm stability of SUAP, and also the transferability gap.

Continuous-time and extended discrete SUAP

Denote the perturbation in continuous-time SUAP as r_t^C, Eq.(14) shows the recursion equation of it.

$$\begin{aligned} \mathrm{d}r_t^C &= \eta\nabla_{r_t^C}\mathcal{L}(\theta, x_t + r_t^C, y_t)\mathrm{d}t + \sqrt{\frac{2|\eta|}{\beta}}\mathrm{d}B_t - v_t^C\mathrm{d}t, \\ & v_t^C \in \mathcal{N}_{\hat{\mathbb{E}}\mathcal{K}, r_t^C}, z_t \in Z_A, \end{aligned} \tag{14}$$

where $r_0^C = r_0$, and continuous-time SUAP samples the same data to discrete SUAP at time $t = k$. Eq.(15) shows the motivation for us to study the continuous-time SUAP,

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{U,\theta}[\|r_t - r_t^{(i)}\|] \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{U,\theta}[\|r_t - r_t^C\| + \|r_t^C - r_t^{(i),C}\| + \|r_t^{(i),C} - r_t^{(i)}\|], \end{aligned} \tag{15}$$

where B_t is a Brownian motion, $r_t^{(i),C}$ is the continuous-time SUAP generated on $Z_A^{(i)}$ which differs from Z_A at the i-th sample. Apparently, the expectation of $\|r_t - r_t^C\|$ and $\|r_t^{(i)} - r_t^{(i),C}\|$ over the randomness of SUAP and θ have the same upper bound, which restricts the expected distance from the continuous SUAP r_t^C to the original discrete one r_t. Therefore, by bounding the expected distance between the discrete and continuous SUAP, the remaining issue is analyzing how $\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|$ varies along with the time. Because the item is time-continuous, the theory in a differential manner can be applied to study $\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|$. However, the discrete SUAP only exists as $t = k$, $k \in \mathbb{N}$. To bound the item $\mathbb{E}_{U,\theta}[\|r_t - r_t^C\|]$, we extend the discrete SUAP to the continuous-time filtration $\{r_t^E | t \geq 0, t \in \mathbb{R}^+\}$, where r_t^E follows the recursion in Eq.(16),

$$\begin{aligned} r_t^E = r_k^E &+ \eta\int_k^t \nabla_{r_k}\mathcal{L}(\theta, x_k + r_k, y_k)\mathrm{d}s \\ &+ \sqrt{\frac{2|\eta|}{\beta}}W_t - \int_k^t v_k\mathrm{d}s, \\ &\text{where } v_k \in \mathcal{N}_{\hat{\mathbb{E}}\mathcal{K}, r_k}, z_k \in Z_A, t \in [k, k+1). \end{aligned} \tag{16}$$

Since $r_k^E = r_k$ at the time $t = k$, $\mathbb{E}_{U,\theta}[\|r_t^E - r_t^C\|]$ can be derived by continuous-time analysis and equals to the pursued item $\mathbb{E}_{U,\theta}[\|r_t - r_t^C\|]$ at any $t = k$.

Remark 6. Different from $r_{\lfloor t \rfloor}$ which keeps being invariant for $t \in (k, k+1)$ and has discontinuities at each $t = k$, r_t^E act as a linear connection between each r_k and r_{k+1} pair.

Bound the expected distance between r_t^E and r_t^C

We concern the following inequality (17) to obtain the upper bound of $\mathbb{E}_{U,\theta}[\|r_t^E - r_t^C\|]$,

$$\|r_t^E - r_t^C\| \leq \|r_t^E - r_{\lfloor t \rfloor}^C\| + \|r_{\lfloor t \rfloor}^C - r_t^C\|. \tag{17}$$

Under Assumptions 2, we improve the upper bound of the terms $\mathbb{E}_{U,\theta}[\|r_t^E - r_{\lfloor t \rfloor}^C\|]$ and $\mathbb{E}_{U,\theta}[\|r_{\lfloor t \rfloor}^C - r_t^C\|]$ in the previous work (Lamperski 2021). We bound the concerned expected distance between r_t^E and r_t^C in Theorems 3 and 4 below.

Theorem 3 (Proved in Appendix B.8)**.** *Given r_t^C in Eq.(14), we have,*

$$\mathbb{E}_{U,\theta}\|r_{\lfloor t \rfloor}^C - r_t^C\| \leq 2\sqrt{\sqrt{d}|\eta|\varepsilon(L_{\mathcal{X},1} + \frac{\sqrt{d}}{2\beta\varepsilon})}.$$

Theorem 4 (Proved in Appendix B.9)**.** *Given r_t^E, r_t^C in Eqs.(14) and (16), we have that, for $t \in [k, k+1)$, $k \geq 1$,*

$$\begin{aligned} &\mathbb{E}_{U,\theta}[\|r_t^E - r_{\lfloor t \rfloor}^C\|] \\ &\leq \min\{\sqrt{(2k+1)4\sqrt{d}|\eta|\varepsilon(L_{\mathcal{X},1} + \frac{\sqrt{d}}{2\beta\varepsilon})}, 2\varepsilon\sqrt{d}\}. \end{aligned}$$

Next, we are going to handle $\mathbb{E}_{U,\theta}[\|r_t^C - r_t^{(i),C}\|]$ by constructing a monotonically increased real function h on it, and analyze how the SDE of h evolves.

Construct the SDE of monotonically increased function h on $\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|$

The main idea is, given a monotonically real function h on $\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|$ in the approximated expected constraint $\hat{\mathbb{E}}\mathcal{K}$, we can obtain a new SDE for h. If $\mathbb{E}_{U,\theta}[\mathrm{d}(\lambda_t h)]$ is always negative, $\lambda_t > 0$, the UAP algorithm stability $\mathcal{S}$ subsequently decreases with t since the upper bound of $\mathcal{S}$ depends on $\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|$. To apply $Itô$'s Lemma, we assume there exists such a twice differentiable and monotonically increased real function $h(\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|)$, and we consider another function $\tilde{h}(\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|) = e^{|\eta|at}h(\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|)$, $a > 0$. In this case, $\lambda_t = e^{|\eta|at}$. Therefore, if $\mathbb{E}_{U,\theta}[\frac{\mathrm{d}(\tilde{h})}{\mathrm{d}t}] \leq 0$, we say h decreases exponentially, where $\mathbb{E}_{U,\theta}[h(\frac{1}{n}\sum_{i=1}^{n}\|r_t^C - r_t^{(i),C}\|)] \leq e^{-|\eta|at}\mathbb{E}_{U,\theta}[h(\frac{1}{n}\sum_{i=1}^{n}\|r_0^C - r_0^{(i),C}\|)]$.

Noteworthy that r_t^C and each $r_t^{(i),C}$ add different noise at any t and fetch samples circularly in Z_A and $Z_A^{(i)}$ by the same order from 1 to n, for further analysis, we rewrite the incursion of $r_t^{(i),C}$ in the following Eq.(18),

$$\begin{aligned} \mathrm{d}r_t^{(i),C} =& \eta\nabla_{r_t^{(i),C}}\mathcal{L}(\theta, x_t^{(i)} + r_t^{(i),C}, y_t^{(i)}) \\ &+ \sqrt{\frac{2|\eta|}{\beta}}(1 - 2u_t^{(i)}u_t^{(i),\top})\mathrm{d}B_t - v_t^{(i),C}\mathrm{d}t, \end{aligned} \tag{18}$$

denote $\rho_t^{(i)} = r_t^C - r_t^{(i),C}$, where $u_t^{(i)} = \frac{\rho_t^{(i)}}{\|\rho_t^{(i)}\|}$ is a unit vector, $(x_t^{(i)}, y_t^{(i)})$ is the sample fetched by continuous-time SUAP at time t from dataset $Z_A^{(i)}$. Because $(1 - 2u_t^{(i)}u_t^{(i),\top})$

is an orthogonal transformation, $(1-2u_t^{(i)}u_t^{(i),\top})\mathrm{d}B_t$ is a differential of Brownian motion which differs from $\mathrm{d}B_t$.

According to $Itô$'s Lemma, denote $\boldsymbol{\rho_t} = \frac{1}{n}\sum_{i=1}^{n}\|\rho_t^{(i)}\|$, $\mathrm{d}e^{|\eta|at}h$ is given by Eq.(19),

$$\begin{aligned}\mathrm{d}(\tilde{h}) = a|\eta|e^{|\eta|at}h(\boldsymbol{\rho_t})\mathrm{d}t + e^{|\eta|at}h_t^{'}(\boldsymbol{\rho_t})\mathrm{d}\boldsymbol{\rho_t} \\ + \frac{1}{2}e^{|\eta|at}h_t^{''}(\boldsymbol{\rho_t})(\mathrm{d}\boldsymbol{\rho_t})^2.\end{aligned} \tag{19}$$

Assumption 4. Assume the loss function is l-smooth in the input space $\mathcal{X}$ where $\|\nabla_{x_1}\mathcal{L}(\theta,x_1,y_1) - \nabla_{x_2}\mathcal{L}(\theta,x_2,y_2)\| \le l\|x_1-x_2\|, l>0$.

Remark 7. Assumption 4 is commonly used in the field of stochastic learning. Under this assumption, we obtain Proposition 3 and Corollary 5 by improving the results in the previous work (Lamperski 2021).

Proposition 3 (Proved in Appendix B.10)**.** *Given $d, n, \varepsilon > 0$, η, under the Assumption 4,we have,*

$$\mathrm{d}\boldsymbol{\rho_t} \le \sqrt{\frac{8|\eta|}{\beta}}\frac{1}{n}\sum_{i=1}^{n}u_t^{(i),\top}\mathrm{d}B_t + (\frac{2l\varepsilon|\eta|\sqrt{d}}{n} + l|\eta|\boldsymbol{\rho_t})\mathrm{d}t. \tag{20}$$

Corollary 5 (Proved in Appendix B.11)**.** $(\mathrm{d}\boldsymbol{\rho_t})^2 \le \frac{8|\eta|}{\beta}\mathrm{d}t$.

Plugging the results in Proposition 3 and Corollary 5 into Eq.(19), and assume h satisfies $h(0)=0$, $h^{'}(0)=1$, and $-\frac{1}{2\varepsilon\sqrt{d}} < h^{''}(p) < 0$, we obtain $h^{'}(\boldsymbol{\rho_t}) < h^{'}(0)$, thus,

$$\begin{aligned}\mathbb{E}_{U,\theta}[\mathrm{d}(\tilde{h})] \le \frac{4|\eta|}{\beta}e^{|\eta|at}\mathbb{E}_{U,\theta}[(\frac{a\beta}{4}h(\boldsymbol{\rho_t}) + \\ \frac{l\beta\varepsilon\sqrt{d}}{2n}h^{'}(\boldsymbol{\rho_t}) + h^{''}(\boldsymbol{\rho_t}) + \frac{\beta l\boldsymbol{\rho_t}}{4})\mathrm{d}t].\end{aligned} \tag{21}$$

Therefore, if Eq.(22) holds for all $p \in [0, 2\varepsilon\sqrt{d}]$,

$$\frac{a\beta}{4}h(p) + \frac{l\beta\varepsilon\sqrt{d}}{2n}h^{'}(p) + h^{''}(p) + \frac{l\beta}{4}p = 0, \tag{22}$$

$\mathbb{E}_{U,\theta}[\frac{\mathrm{d}(\tilde{h})}{\mathrm{d}t}] \le 0$ and h decreases exponentially with time t. In the next step, we prove the existence of function h.

The function h which satisfies Eq.(22) is a quadratic nonhomogeneous differential equation with constant coefficients. Recall that h satisfies that $h(0)=0$, $h^{'}(0)=1$, and $-\frac{1}{2\varepsilon\sqrt{d}} < h^{''}(p) < 0$. As shown in Proposition 4, we prove that there exists a function h by choosing a, β, and n.

Proposition 4 (Proved in Appendix B.12)**.** *Given $c_1 = \frac{l\beta\varepsilon\sqrt{d}}{4n}$, $c_2 = \sqrt{\frac{a\beta}{4}}$, $c_3 = -\frac{l\beta}{4}$, and a, n, β meet the following two conditions, $a = \frac{\beta dl^2\varepsilon^2}{4n^2}$, $n \in (\sqrt{\frac{\beta ld\varepsilon^2}{4}}, \min\{\frac{\beta dl\varepsilon^{\frac{3}{2}}}{4}, \frac{\sqrt{4\beta ld\varepsilon^2+e^2}+e}{4}\})$ Then, there exists a function $h(p)$ satisfying Eq.(22), we show it's formulation via solving a second-order constant-coefficient nonhomogeneous differential equation, we have,*

$$h(p) = (\frac{c_2^2+c_3}{c_2^2}p + \frac{2c_3}{c_2^3})e^{-c_2p} + \frac{c_3}{c_2^2}p - \frac{2c_1c_3}{c_2^4}. \tag{23}$$

Proposition 4 discusses the existence of the function h. In fact, by choosing a and β properly, the requirements are easy to meet. We unite the results in Proposition 2, 4, Corollary 4, Theorem 3, and 4 to bound the UAP algorithm stability $\mathcal{S}$ and white-box transferability τ_W as shown in Corollary 6.

Corollary 6 (Main result 4, proved in Appendix B.13)**.** *Given the stop step k_s of SUAP, $d, n, \varepsilon, \varrho > 0$, a, β, h in Proposition 4, η, and Assumption 2, 3, 4, a constant $c_4(n,\beta,l,\varepsilon,d)$, the following inequality holds with the probability $(1-e^{\frac{-2\varrho^2}{\varepsilon^2}})^2$,*

$$\begin{aligned}\mathcal{S} \le & L_{\mathcal{X},1}(2\min\{2\sqrt{\sqrt{d}|\eta|\varepsilon(L_{\mathcal{X},1}+\frac{\sqrt{d}}{2\beta\varepsilon})}(1+\sqrt{2k_s+1}), \\ & 2\varepsilon\sqrt{d}\} + \frac{e^{-a|\eta|k_s}2\varepsilon\sqrt{d}}{c_4(n,\beta,l,\varepsilon,d)}) + 2\varrho.\end{aligned} \tag{24}$$

Recall Proposition 2, we deliver that, the white-box transferability gap τ_W consequently has the same upper bound in Corollary 6. The bound in Corollary 6 decreases exponentially at the beginning of the generation of SUAP, and then increases to a fixed constant with ratio $\mathcal{O}(\sqrt{k})$. **We delegate the empirical results of the white-box transferability gap of SUAP in the white-box setting in Appendix A.3, which are in line with the result in Corollary 6.** (Zhang et al. 2025; Moosavi-Dezfooli et al. 2017; Shafahi et al. 2020; Pan, Li, and Yao 2024), demonstrating the superior white-box transferability gap of SUAP.

Conclusions

We construct the framework for analyzing the transferability and algorithm stability of UAP algorithms. We prove that the white-box transferability gap can be bounded by the expectation of the proposed UAP algorithm stability. Besides, we also prove that the black-box transferability explicitly larger than that in the white-box setting at most an additional item. We prove that the dynamic constraint in UAP algorithms results in a tiny convergence domain that harms the capacity of UAPs. We consequently propose an expected constraint that is much broader, and prove that UAPs in the expected constraint suit any sample with a high probability. We propose a noisy UAP algorithm, dubbed as SUAP, and deliver the upper bound of the UAP algorithm stability of SUAP. We prove that the upper bound of UAP algorithm stability of SUAP decrease exponentially at the beginning and increases with a sublinear rate to at most a fixed constant. Experimental results demonstrate the effectiveness of the proposed expected constraint and SUAP.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No: 92270123), and the Research Grants Council (Grant No: 15226221 and 15224124), Hong Kong SAR, China, also supported by the National Natural Science Foundation of China (Grant No: 62472345, 62206207, 62432012), and also supported by the Open Research Fund of The State Key Laboratory of Blockchain and Data Security, Zhejiang University.

References

Anil, G.; Vinod, V.; and Narayan, A. 2024. Generating universal adversarial perturbations for quantum classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10891–10899.

Azizian, W.; Iutzeler, F.; Malick, J.; and Mertikopoulos, P. 2024. What is the Long-Run Distribution of Stochastic Gradient Descent? A Large Deviations Analysis. In *International Conference on Machine Learning*, 2168–2229. PMLR.

Bai, L.; Ye, Q.; Zhang, X.; Zhang, S.; Liang, Z.; Xu, J.; and Hu, H. 2025. Toward Efficient Inference Attacks: Shadow Model Sharing via Mixture-of-Experts. *arXiv preprint arXiv:2510.13451*.

Bousquet, O.; and Elisseeff, A. 2002. Stability and generalization. *The Journal of Machine Learning Research*, 2: 499–526.

Bousquet, O.; Klochkov, Y.; and Zhivotovskiy, N. 2020. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, 610–626. PMLR.

Brosse, N.; Durmus, A.; and Moulines, E. 2018. The promises and pitfalls of stochastic gradient Langevin dynamics. *Advances in Neural Information Processing Systems*, 31.

Bubeck, S.; Eldan, R.; and Lehec, J. 2015. Finite-time analysis of projected Langevin Monte Carlo. *Advances in Neural Information Processing Systems*, 28.

Bubeck, S.; Eldan, R.; and Lehec, J. 2018. Sampling from a log-concave distribution with projected Langevin Monte Carlo. *Discrete & Computational Geometry*, 59: 757–783.

Chen, X.; Du, S. S.; and Tong, X. T. 2020. On stationary-point hitting time and ergodicity of stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 21: 1–41.

Cheng, X.; Yin, D.; Bartlett, P.; and Jordan, M. 2020. Stochastic gradient and Langevin processes. In *International Conference on Machine Learning*, 1810–1819. PMLR.

Dezfooli, S. M.; Alhussein, F.; Omar, F.; Pascal, F.; and Stefano, S. 2017. Analysis of universal adversarial perturbations. *arXiv preprint arXiv:1705.09554*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hardt, M.; Recht, B.; and Singer, Y. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, 1225–1234. PMLR.

Jetley, S.; Lord, N.; and Torr, P. 2018. With friends like these, who needs adversaries? *Advances in neural information processing systems*, 31.

Khrulkov, V.; and Oseledets, I. 2018. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8562–8570.

Lamperski, A. 2021. Projected stochastic gradient langevin algorithms for constrained sampling and non-convex learning. In *Conference on Learning Theory*, 2891–2937. PMLR.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Mopuri, K. R.; Ganeshan, A.; and Babu, R. V. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10): 2452–2465.

Mopuri, K. R.; Garg, U.; and Babu, R. V. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*.

Pan, C.; Li, Q.; and Yao, X. 2024. Adversarial initialization with universal adversarial perturbation: A new approach to fast adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21501–21509.

Raginsky, M.; Rakhlin, A.; and Telgarsky, M. 2017. Nonconvex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, 1674–1703. PMLR.

Shafahi, A.; Najibi, M.; Xu, Z.; Dickerson, J.; Davis, L. S.; and Goldstein, T. 2020. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5636–5643.

Wang, Y.; Liu, L.; Liang, Z.; Ye, Q.; Hu, H.; et al. 2024. New Paradigm of Adversarial Training: Releasing Accuracy-Robustness Trade-Off via Dummy Class. *arXiv preprint arXiv:2410.12671*.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.

Weng, J.; Luo, Z.; Zhong, Z.; Lin, D.; and Li, S. 2023. Exploring non-target knowledge for improving ensemble universal adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2768–2775.

Xiao, Y.; Ye, Q.; Hu, H.; Zheng, H.; Fang, C.; and Shi, J. 2022. Mexmi: Pool-based active model extraction crossover membership inference. In *Advances in Neural Information Processing Systems*.

Xiao, Y.; Ye, Q.; Hu, L.; Zheng, H.; Hu, H.; Liang, Z.; Li, H.; and Jiao, Y. 2025a. Reminiscence attack on residuals: Exploiting approximate machine unlearning for privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3058–3068.

Xiao, Y.; Ye, Q.; Liang, Z.; Li, H.; Li, R.; Zheng, H.; and Hu, H. 2025b. Class-feature Watermark: A Resilient Black-box Watermark Against Model Extraction Attacks. *arXiv preprint arXiv:2511.07947*.

Zhang, C.; Benz, P.; Imtiaz, T.; and Kweon, I. S. 2020. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14521–14530.

Zhang, C.; Benz, P.; Karjauv, A.; and Kweon, I. S. 2021. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7868–7877.

Zhang, Y.; Xu, Y.; Shi, J.; Zhang, L. Y.; Hu, S.; Li, M.; and Zhang, Y. 2025. Improving Generalization of Universal Adversarial Perturbation via Dynamic Maximin Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 10293–10301.