Transferability Bound Theory: Exploring Relationship between Adversarial Transferability and Flatness

Mingyuan Fan[1], Xiaodan Li[1,3], Cen Chen[1,2]*, Wenmeng Zhou[3], Yaliang Li[4]

[1]School of Data Science & Engineering, East China Normal University, China
[2] The State Key Laboratory of Blockchain and Data Security, Zhejiang University, China
[3]Alibaba Group, Hangzhou, China
[4]Alibaba Group, Bellevue, WA, USA
`fmy2660966@gmail.com`, `cenchen@dase.ecnu.edu.cn`,
`{fiona.lxd, wenmeng.zwm, yaliang.li}@alibaba-inc.com`

Abstract

A prevailing belief in attack and defense community is that the higher flatness of adversarial examples enables their better cross-model transferability, leading to a growing interest in employing sharpness-aware minimization and its variants. However, the theoretical relationship between the transferability of adversarial examples and their flatness has not been well established, making the belief questionable. To bridge this gap, we embark on a theoretical investigation and, for the first time, derive a theoretical bound for the transferability of adversarial examples with few practical assumptions. Our analysis challenges this belief by demonstrating that the increased flatness of adversarial examples does not necessarily guarantee improved transferability. Moreover, building upon the theoretical analysis, we propose TPA, a Theoretically Provable Attack that optimizes a surrogate of the derived bound to craft adversarial examples. Extensive experiments across widely used benchmark datasets and various real-world applications show that TPA can craft more transferable adversarial examples compared to state-of-the-art baselines. We hope that these results can recalibrate preconceived impressions within the community and facilitate the development of stronger adversarial attack and defense mechanisms. The source codes are available in `https://github.com/fmy266/TPA`.

1 Introduction

The transferability of adversarial examples [5, 9] suggests a phenomenon where adversarial examples designed to fool one local proxy model can also be generalized to mislead other unknown target models, even with different model parameters and architectures. This intriguing property has significant implications for various applications, ranging from the robustness assessment of large-scale models [10, 27] to data privacy protection [17, 23, 32]. However, it is observed that the generated adversarial examples often tend to be overly specialized to the proxy model, showing limited transferability [27, 30, 36]. Driven by this, researchers have dedicated substantial efforts to developing transferability-enhancing techniques [24, 30, 35].

Most transferability-enhancing techniques have drawn inspiration from an analogy between the transferability of adversarial examples and the generalization capability of models [25, 36, 44]. For example, borrowing the idea of Mixup [46], Admix [37] attempts to blend adversarial examples with other samples to generate more transferable adversarial examples. Recently, Foret et al. [14] observed a strong correlation between the flatness of the model's loss landscape and its generalization capability,

---

*Corresponding author.

thus advocating for the optimization of the worst-case loss within a small perturbation radius to regulate the loss landscape. The empirical performance of such sharpness-aware minimization is quite impressive, sparking a surge of interest among researchers and catalyzing the development of its variants. Given the above considerable progress, the exploitation of sharpness-aware minimization and its variants to enhance the transferability of adversarial examples has become one of the hottest research topics [12, 15, 42, 45].

Although empirical results show that transferability-enhancing techniques based on flatness achieve state-of-the-art performance [15, 30, 42], the theoretical relationship between adversarial transferability and flatness has not been well established. As such, it remains questionable whether adversarial examples converging to flat extreme points, referred to as *flat adversarial examples*, necessarily possess better transferability. In fact, the relationship between flatness and generalization capability is still under debate and not well-solved. Recent works [1, 3, 39] have brought to light a thought-provoking finding, i.e., flat extreme points do not always lead to better generalization ability, making the relationship between adversarial transferability and flatness even more mysterious. Given the aforementioned gap in our current understanding, we first theoretically investigate the following critical question:

> **Q1:** *Are flat adversarial examples necessarily associated with improved transferability?*

Our theoretical examination yields a negative answer to Q1. Specifically, we derive a bound of adversarial transferability and demonstrate that, despite being a contributing term, flatness alone is insufficient to determine the value of the bound. Then, our attention naturally shifts to explore Q2:

> **Q2:** *Can our theoretical analysis guide to develop a more principled transfer-based attack?*

One straightforward idea for addressing the Q2 is to directly optimize the derived bound. However, the optimization of this bound requires higher-order gradient information, which is prohibitively expensive in high-dimensional spaces. To circumvent this issue, we introduce Theoretically Provable Attack (TPA), which employs a theoretically-grounded surrogate of the original bound as an alternative. The optimization of the surrogate only requires first-order gradient information, rendering TPA more computationally efficient and practical.

It should be stressed that the surrogate in TPA is theoretically supported, ensuring that its attack effectiveness enjoys theoretical guarantees. Moreover, the empirical evaluation of TPA in the benchmarks and three kinds of real-world applications demonstrates the superior performance of TPA over the state-of-the-art attack methods. Our contributions are three-fold:

- To our best knowledge, we are the first to derive a theoretical bound on the transferability of adversarial examples, with few practical assumptions. Contrary to the prevalent belief among the community, we demonstrate that the flatness of adversarial examples does not necessarily guarantee enhanced transferability. Our theory also advances the understanding of the intrinsic characteristics governing transferability and lays a solid theoretical foundation for future research in this area.
- Building upon our theoretical analysis, we introduce TPA which optimizes a surrogate of the original bound to avoid direct computation of high-order gradients. We demonstrate the effectiveness of the surrogate as the surrogate of the derived bound.
- We conduct extensive experiments in the benchmark dataset ImageNet, showing the impressive performance of TPA. *To our best knowledge, TPA is the first transfer-based attack to achieve an average 90% attack success rate against transformer architectures.* We also test the effectiveness of TPA in three kinds of real-world applications that are under-explored in the existing works, including Google Vision Systems, advanced search engines, as well as multimodal large model applications (e.g., GPT-4 and Claude3).

2 Related Work

2.1 Flatness-based Optimization Methods

Intuitively, the loss landscape of a model on training data mirrors that on test data, suggesting that a slight translation of the loss landscape over training data can approximate the loss landscape over

test data. Given this, a flat extreme point is believed to enjoy a smaller performance gap between training and testing data than sharp extreme points, i.e., better generalization ability. Capitalizing on this intuition, SAM [6, 7, 14] incorporates the following flatness term or its variants into its loss function to improve the model's generalization capability:

$$\max_{||\rho||\leq\epsilon} \mathcal{L}(F_{\theta+\rho}(x), y) - \mathcal{L}(F_\theta(x), y), \tag{1}$$

where $(x, y), \mathcal{L}, F_\theta$ and ϵ are training data pairs, loss function, model parameterized by θ, and perturbation budget, respectively. Equation 1 evaluates the loss landscape's flatness at point θ by quantifying the increase in loss when moving from θ to a neighboring worst-behaved parameter value. Subsequent studies explored various flatness metrics, such as Fisher determinant [20] and gradient norms [47, 48], for regularizing loss landscapes. Nevertheless, recent works [1, 3, 39, 41] have cast a critical light. Dinh et al. [3] demonstrated the existence of networks that have a sharp loss landscape yet manage to work well on test data. Andriushchenko et al. [1] empirically revealed that there is not a strong correlation between flatness and generalization ability, particularly in large neural networks.

2.2 Transferability-enhancing Methods

Transferability-enhancing methods can be roughly divided into input-regularization-based methods, optimization-based methods, and model-based methods. In each iteration, input-regularization-based methods [8, 36, 37, 44] ensemble multiple transformed inputs to craft adversarial examples, and the distinction between these methods is reflected in the adopted transformation techniques. DI [44] suggests to resize and pad inputs, while TI [5] translates the inputs and SI [25] scales the inputs. By observing that existing transformations are all applied on a single image, Admix [37] attempts to admix inputs with an image from other categories. Unlike the spatial domain transformations mentioned above, SSA [27] perturbs inputs in frequency domains to produce more diverse transformed inputs. BSR [35] segments inputs into non-overlapping blocks, shuffling and rotating the blocks randomly.

Optimization-based methods leverage more advanced optimizers [4, 25] and model-based methods enhance the transferability via the lens of model per se [22, 50]. For instance, MI [4] adopts Momentum optimizer. Besides, SGM [40] refines the back-propagation procedure to amplify the gradients of early layers, due to that the features learned by early layers are more shared over different models. StyLess [24] employs stylized networks to prevent adversarial examples from using non-robust style features. More recent advancements delved into the exploration of flatness-based optimization methods to improve the transferability of adversarial examples. Qin et al. [30] introduced RAP, which tailors Equation 1 to craft flatness-enhanced adversarial examples. Some studies [15, 42] chose to punish the gradients of the generated adversarial examples.

3 Theoretical Analysis

In this section, we embark on a theoretical exploration of adversarial transferability, commencing with the canonical setup of transfer-based adversarial attacks. The primary objective of transfer-based attacks is to make adversarial examples crafted using the local proxy model F as effective as possible for the target model F'. Formally, for a given natural samples x with ground-truth labels y, the vanilla transfer-based attack solves the following optimization task to craft adversarial noises δ^* for x as:

$$\delta^* = \arg\min_\delta - \mathcal{L}(F(x + \delta), y), \quad s.t., ||\delta||_\infty \leq \epsilon, \tag{2}$$

where $\mathcal{L}$ is cross-entropy loss and ϵ is perturbation budget. Despite the high attack effectiveness of $x + \delta^*$ on the proxy model, their effectiveness significantly diminishes when transferred to the target model. To better study this problem, we decouple transferability into two factors: *the local effectiveness term*, which measures the loss of generated adversarial examples on the proxy model, along with *the transfer-related loss term*, which quantifies the change in the loss of adversarial examples when transferring from the proxy model to the target model. The local effectiveness term and the transfer-related loss term can be evaluated by $\mathcal{L}(F(x + \delta), y)$ and $D(x + \delta, y) = \mathcal{L}(F'(x + \delta), y) - \mathcal{L}(F(x + \delta), y)$, respectively. Intuitively, transferable adversarial examples are characterized by high attack performance on the proxy model together with minimal increase in the transfer-related loss term upon moving to the target model. Before presenting our theoretical results, we first make several practical assumptions.

Assumption 3.1. *The underlying distribution of x is continuous and bounded: there exists a constant B_1 such that $\forall x$, it holds that $p(x) \leq B_1$.*

Assumption 3.2. *The gradient norm of $p(x)$ is bounded: there exists a constant B_2 such that $\forall x$, it holds that $||\nabla p(x)|| \leq B_2$.*

Assumption 3.3. *The probability of adversarial examples occurring in nature is less than that of natural samples: $p(x + \delta) \leq p(x)$.*

Assumption 3.4. *Proxy-model-generated adversarial examples have a greater loss on the proxy model than on the target model: $\mathcal{L}(F'(x + \delta), y) \leq \mathcal{L}(F(x + \delta), y)$.*

Assumption 3.5. *The proxy model is based on ResNet-like architecture, where each layer applies a linear transformation followed by a ReLU activation.*

Theorem 3.1. *(See Appendix A for Proof.) Let Assumption $3.1 \sim 3.5$ stand. For small $||\delta||_2^2$, it holds that:*

$$\mathbb{E}_{p(x)}\{||D(x+\delta,y)||_2^2\} \leq \underbrace{\mathbb{E}_{p(x)}\{||D(x,y)||_2^2 + C\,||\delta||_2^2\,||\nabla D(x,y)||_2^2\}}_{\text{The inherent model difference component}}$$

$$+ \underbrace{(1+C)\mathbb{E}_{p(x)}\{||\delta||_2^2\,||\nabla \log F(x+\delta)||_2^2\}}_{\text{The first-order gradient component}} + \underbrace{2\,\mathbb{E}_{p(x)}\{||\delta||_2^2 \sum_i |\nabla^2 \log F(x+\delta)[i,i]|\}}_{\text{The second-order gradient component}}, \quad (3)$$

where $C = \frac{B_1}{B_1 + B_2 ||\delta||}^2$ and $[i,i]$ denotes the element at the intersection of the i-th row and the j-th column in the given matrix. Let the sum of the three terms on the right side of Equation 3 be denoted as K. Then, we have:

$$||\mathcal{L}(F'(x+\delta),y)||_2^2 \geq |(||\mathcal{L}(F(x+\delta),y)||_2^2 - K)|. \quad (4)$$

We use $p(x)$ to denote the underlying distribution of x and posit Assumption $3.1 \sim 3.5$. Assumption 3.1 arises naturally since natural samples do not occur with infinite probability. Building upon Assumption 3.1, a straightforward deduction is that the gradients of $p(x)$ are bounded; otherwise, there exists at least one sample that appears with infinite probability. Assumptions 3.3 and Assumption 3.4 are naturally valid. Regarding Assumption 3.3, adversarial samples are specifically crafted rather than commonly occurring in nature. Moreover, discussions about transfer-based attacks only make sense under the conditions set by Assumption 3.4. Assumption 3.5 is satisfiable because the attacker can freely choose which proxy model to use. Theorem 3.1 establishes bounds for $||D(x+\delta,y)||_2^2$ and $||\mathcal{L}(F'(x+\delta),y)||_2^2$. We mainly discuss the bound of $||D(x+\delta,y)||_2^2$, which is composed of three key components, namely the inherent model difference, the first-order gradient, and the second-order gradient components. For the sake of simplicity in our discussion, $||\delta||_2^2$ is considered as a constant.

The inherent model difference component. This component is intuitively interpreted as a measure of the similarity between the proxy and target models by evaluating their output differences regarding natural samples x. A higher value of the component implies that adversarial examples produced by the proxy model are less likely to transfer to the target model.

In practice, the value of this component tends to be modest due to the following reasons. Firstly, the target model is expected to perform well on x, as misclassification of the target model to x would render crafting adversarial examples for x trivial. Secondly, as the attacker is capable of fine-tuning the proxy model on x, the loss of x in the proxy model can also be small. Furthermore, if the discrepancy between the two models is small, their gradient differences also are minor as well [11]. Thus, both $||D(x,y)||_2^2$ and $||\nabla D(x,y)||_2^2$ are of small values.

Moreover, this component illuminates the selection criteria for an appropriate proxy model. While previous studies have highlighted that proxy and target models exhibiting structural and training data similarities tend to facilitate the transferability of adversarial examples across these models, our theoretical insights suggest a more nuanced criterion. Specifically, a proxy model needs only yield predictions for x that are closely aligned with those of the target model, thereby relaxing the requirements of architectural and data similarities.

---

[2]In fact, there is $C \leq 1$, and using C of 1 could be useful in certain situations, such as calculating the specific value of upper bound for $||D(x+\delta,y)||_2^2$. Here, for the sake of the tightness of our bound, we have not done so.

The first-order gradient component and the second-order gradient component. Our analysis here challenges the prevailing belief [12, 15, 30, 42, 45] that flat adversarial examples necessarily induce improved transferability. In Theorem 3.1, the factors that are relevant to the direction of δ include the first-order and second-order gradient components. The first-order gradient component serves as a direct measurement of the flatness associated with adversarial examples. Nonetheless, the presence of the second-order gradient component unveils that transferability is not exclusively contingent on flatness. In other words, an adversarial example that appears flat but has a higher second-order gradient component might not be optimal for the target model. To elucidate this point, consider the function $f(x) = \sin(x^2)$ as an example. Figure 1 presents the norms of first-order gradient y_1 and the second-order
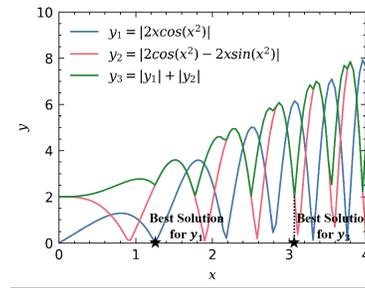

Figure 1: The visualization of the first-order gradient, the second-order gradient of $y = \sin x^2$. The black stars symbolize the location where the minimum values of y_1 and y_3 are achieved.

gradient y_2 of $f(x)$, as well as their sum y_3. Upon examination, it is observed that optimizing the norms of y_1 alone does not yield optimal solutions for optimizing y_3. This highlights the inadequacy of minimizing only the first-order gradient component in ensuring the minimization of the bound.

4 The Proposed Approach TPA

4.1 Overview

In Section 3, the adversarial transferability is decoupled into the local effectiveness term and the transferability-related loss term and we derive a bound for the latter. It is natural to jointly optimize both the local effectiveness term and the bound for generating more transferable adversarial examples. The first term maintains the effectiveness of generated adversarial examples against the proxy model while the bound controls the performance degradation upon transfer to the target model, thus achieving better transferability. However, the bound involves the second-order gradient component that is quite expensive to optimize. To this end, as shown in Section 4.2, TPA instead develops a computationally feasible surrogate for the bound that retains optimization integrity, substantiated through theoretical validation. Section 4.3 introduces an approximate solution to efficiently tackle the optimization target.

4.2 Optimization Formulation

The optimization target of TPA is formalized as follows:

$$\delta^* = \arg\min_{\delta} -\mathcal{L}(F(x+\delta), y) + \lambda \mathbb{E}_{\Delta \sim U(-b,b)}\{||\nabla \mathcal{L}(F(x+\delta+\Delta), y)||_2\},$$
$$s.t., ||\delta||_\infty \leq \epsilon, b \geq 0, \tag{5}$$

where $U(-b, b)$ is uniform distribution between $-b$ and b, λ is penalty magnitude, and $x + \delta + \Delta$ denotes a sample extracted uniformly from the region around $x + \delta$. In Equation 5, we employ $-\mathcal{L}(F(x+\delta), y)$ as the local effectiveness term, while $||\nabla \mathcal{L}(F(x+\delta+\Delta), y)||_2$ serves as a surrogate for our bound. Notice that our proposed surrogate is designed to regulate both the first- and second-order gradient component, excluding the inherent model difference component. This is because the inherent model difference component remains a constant term given specific perturbation norm constraint and the proxy-target model pair.

We next demonstrate the effectiveness of the surrogate. The surrogate minimizes the gradients of samples around $x + \delta$, intuitively leading to a minimization of the gradient norms of $x + \delta$ themselves as well. Our main investigation is determining if the surrogate can effectively minimize the second-order gradient component. The second-order gradient quantifies the rate at which the first-order gradient changes. Through penalizing the gradient norms of samples around $x + \delta$, the surrogate naturally encourages the gradient norms of samples around $x + \delta$ towards zero, thereby implicitly

moderating the second-order gradient. Theoretically, it is demonstrated as follows:

$$\sum_i |\nabla^2 \log F(x+\delta)[i,i]| = \sum_i |\lim_{\mu \to 0} \frac{\nabla \log F(x+\delta+\mu)[i] - \nabla \log F(x+\delta)[i]}{u}|$$

$$= || \lim_{\mu \to 0} \{ (\nabla L(F(x+\delta+\mu),y) - \nabla L(F(x+\delta),y)) \cdot \frac{1}{u} \} ||_1. \tag{6}$$

According to Equation 6, if the gradient norms of the samples around $x+\delta$ approach zero, the second-order gradient components also tend towards zero. Thus, the surrogate is capable of effectively regulating both the first- and second-order gradients.

We see that the optimization target of TPA bears a high visual similarity to those in some studies [15, 42] which penalties the gradient norm of $x+\delta$. Despite this superficial similarity, there exists a fundamental difference in their effects. As elaborated in Section 3, penalizing only the gradient norm of $x+\delta$ is unable to regulate the second-order gradient component.

4.3 Approximate Solution

The highly non-linear and non-convex nature of $F(\cdot)$ render the analytic solution of Equation 5 to be hardly derived [21]. As a result, standard practice for solving Equation 5 is employing gradient-based optimization methods [2]. However, as shown in Equation 7, the gradients of Equation 5 involve Hessian matrix evaluated at multiple points $x+\delta+\Delta_i$ for $\Delta_i, i = 1, \cdots, N$, each of which is troublesome to evaluate in high-dimension spaces [13, 29].

$$- \nabla \mathcal{L}(F(x+\delta),y) + \frac{\lambda}{N} \sum_{i=1}^{N} \nabla ||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2$$

$$= -\nabla \mathcal{L}(F(x+\delta),y) + \frac{\lambda}{N} \sum_{i=1}^{N} H_{x+\delta+\Delta_i} \frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}. \tag{7}$$

To get rid of directly evaluating Hessian matrix in Equation 7, we propose an approximate estimation for $H_{x+\delta+\Delta_i} \frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}$. To achieve this, we utilize Taylor expansion on $\mathcal{L}(F(x+\delta+\Delta_i+\phi),y)$, assuming ϕ is small enough for making expansion feasible:

$$\mathcal{L}(F(x+\delta+\Delta_i+\phi),y) = \mathcal{L}(F(x+\delta+\Delta_i),y) + \nabla \mathcal{L}(F(x+\delta+\Delta_i),y)\phi, \tag{8}$$

Furthermore, by differentiating both sides of Equation 8, we obtain:

$$\nabla \mathcal{L}(F(x+\delta+\Delta_i+\phi),y) = \nabla \mathcal{L}(F(x+\delta+\Delta_i),y) + H_{x+\delta+\Delta_i}\phi, \tag{9}$$

Notice that, the desired item $H_{x+\delta+\Delta_i} \frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}$ arises if setting ϕ along the direction $\frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}$. By implementing the idea, there is:

$$H_{x+\delta+\Delta_i} \frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2} =$$

$$\frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i+k\frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}),y)}{k} - \frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{k}. \tag{10}$$

Wherein, $\phi = k\frac{\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)}{||\nabla \mathcal{L}(F(x+\delta+\Delta_i),y)||_2}$ and k is a small constant to cater ϕ being small. By substituting Equation 10 into Equation 7, we obtain the approximately estimated gradients of Equation 5. Compared to directly evaluating Hessian matrix that requires quadratic storage and cubic computation time [13, 29], our approximate solution only involves first-order gradients (linear computation time) so as to make TPA more efficient computationally. Moreover, linear expansion used in Equation 9 results in an approximation error of $O(k)$ [33].

5 Simulation Experiment

5.1 Setup

Dataset. We randomly select 10000 images from ImageNet.

Table 1: The attack success rates (%) of different attacks on normal models. Best results are in bold.

| Proxy Model	Method	ResNet50	DenseNet121	EfficientNet	InceptionV3	MobileNetV2	SqueezeNet	ShuffleNetV2	ConvNet	RegNet	MNASNet	WideResNet50	VGG19	ViT	Swin
ResNet50	MI	**100.00**	87.91	76.43	68.57	66.51	86.66	78.11	60.84	72.50	81.33	90.77	81.99	43.06	55.89
	NI	**100.00**	88.14	76.79	68.57	66.10	86.43	78.06	60.64	72.55	82.35	91.17	82.98	41.68	54.87
	DI	99.97	80.77	68.75	69.74	66.05	77.58	72.83	49.39	65.82	79.54	82.91	76.51	40.33	46.71
	TI	**100.00**	73.47	56.28	55.99	55.79	68.85	62.04	42.83	59.08	66.33	78.60	68.78	34.85	40.51
	VT	**100.00**	89.62	81.63	73.47	76.86	90.21	83.83	76.28	82.83	92.86	95.51	86.12	54.77	57.87
	SSA	**100.00**	95.51	91.56	79.62	77.40	93.60	87.60	79.90	85.99	93.60	96.53	95.13	54.21	59.64
	RAP	**100.00**	95.01	94.78	93.81	93.88	93.87	94.64	90.80	94.54	95.25	93.97	94.92	62.62	60.77
	BSR	**100.00**	97.02	95.16	93.58	94.19	94.52	94.88	88.46	94.64	96.70	93.62	93.80	82.32	77.05
	Ours	99.85	**99.69**	**99.52**	**98.70**	**99.52**	**99.72**	**99.82**	**94.34**	**97.37**	**99.82**	**99.44**	**98.83**	**93.52**	**90.98**
ResNet152	MI	83.44	85.92	77.70	70.97	66.33	85.54	77.65	62.17	74.59	81.30	90.36	79.52	45.43	56.22
	NI	84.26	87.32	77.70	70.38	66.38	86.10	78.06	62.88	74.67	81.48	90.89	78.70	43.93	54.31
	DI	86.76	79.06	70.56	71.38	66.89	75.87	73.06	52.70	68.75	78.67	83.01	73.80	43.62	47.45
	TI	84.21	73.42	58.57	57.60	56.51	68.14	62.42	45.08	61.35	66.48	78.57	66.48	37.91	41.30
	VT	89.54	89.57	84.67	83.62	77.98	92.07	87.76	72.42	89.29	92.5	91.38	90.87	56.38	55.87
	SSA	94.62	95.23	93.06	88.55	78.55	96.05	93.37	76.79	95.33	93.24	94.74	93.11	57.63	59.46
	RAP	98.86	94.44	93.74	93.48	93.61	92.37	93.07	90.01	92.89	94.82	93.22	94.22	61.48	58.98
	BSR	98.05	94.62	93.55	94.08	93.75	92.42	93.30	89.75	92.52	94.04	93.64	94.82	82.22	77.86
	Ours	**99.85**	**99.67**	**99.39**	**98.95**	**99.18**	**99.01**	**99.64**	**94.11**	**97.96**	**99.77**	**99.57**	**98.57**	**90.19**	**86.27**
DenseNet121	MI	89.01	99.97	81.35	73.95	70.33	87.88	80.18	65.00	73.65	84.41	86.63	85.64	47.17	61.10
	NI	90.28	99.98	82.55	74.13	70.10	87.65	81.15	65.51	74.26	86.10	87.98	86.28	47.60	61.45
	DI	82.45	99.90	73.24	72.88	70.43	78.90	76.58	51.33	66.22	81.22	78.47	79.52	43.52	49.92
	TI	76.33	99.95	61.38	60.15	59.01	70.41	66.33	46.15	58.78	70.84	72.76	73.60	38.88	44.21
	VT	92.12	99.95	90.33	86.58	81.48	92.02	90.18	75.19	85.78	88.75	81.20	81.56	49.18	52.93
	SSA	94.82	99.97	91.86	83.67	83.67	96.66	92.76	82.18	88.04	88.11	90.48	90.13	50.92	59.64
	RAP	97.40	99.96	95.48	95.06	94.67	97.91	94.94	90.77	92.51	96.10	95.04	94.43	63.13	61.65
	BSR	98.24	99.99	94.68	94.24	94.56	96.27	94.96	90.52	93.01	95.63	89.20	95.42	81.52	79.25
	Ours	**99.69**	99.87	**99.34**	**98.32**	**99.52**	**99.57**	**99.82**	**94.41**	**97.37**	**99.77**	**99.08**	**99.03**	**93.70**	**91.98**
DenseNet201	MI	90.41	95.89	83.16	74.90	70.56	87.53	80.69	69.21	76.33	85.08	87.60	83.34	51.28	64.31
	NI	91.20	96.56	83.19	75.48	70.20	87.32	80.87	70.18	77.24	86.25	88.55	84.67	50.48	64.54
	DI	84.57	90.28	76.02	75.71	70.74	77.68	77.76	56.71	70.99	82.53	81.38	78.57	47.78	53.67
	TI	78.39	88.11	63.85	62.37	59.67	70.00	66.61	49.21	63.29	71.30	74.11	70.74	41.53	46.38
	VT	93.44	95.91	83.06	87.40	80.71	89.92	93.04	79.92	84.89	79.16	82.12	79.16	42.37	55.66
	SSA	96.20	97.27	92.42	84.11	82.88	94.59	90.31	77.93	90.33	87.42	89.82	87.70	42.50	62.12
	RAP	96.01	98.99	94.42	95.15	93.27	97.56	93.58	89.05	91.04	95.64	93.74	92.52	62.66	61.31
	BSR	96.85	97.94	94.24	95.08	93.26	96.26	93.54	92.24	92.30	95.16	94.56	92.46	81.02	79.51
	Ours	**99.72**	**99.90**	**99.67**	**98.83**	**99.23**	**99.31**	**99.85**	**96.96**	**98.34**	**99.77**	**99.62**	**99.16**	**93.54**	**92.85**
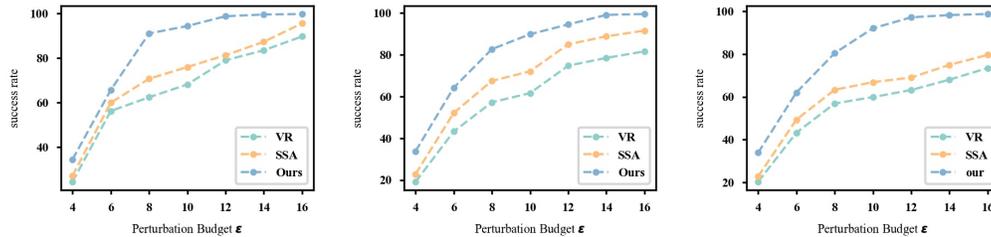
Figure 2: The attack effectiveness of TPA with varying perturbation budgets. The adversarial examples are crafted on ResNet50.

Models. We consider 14 models including ResNet50, DenseNet121, MobileNetV2, EfficientNet, VGG19, InceptionV3, WideResNet50, MNASNet, RegNet, ShuffleNetV2, SqueezeNet, ConvNet, ViT, and Swin. with the first twelve being convolutional networks and the final two based on transformer architectures. Moreover, we also validate the performance of TPA on secured models including adversarial training with L_2 [31] and L_∞ [34] constraint as well as robust training with Styled ImageNet (SIN) and the mixture of Styled and natural ImageNet (SIN-IN) [16].

Baselines. Nine state-of-the-art attacks are used as competitors for TPA: BIM [21], DI [44], MI [4], NI [25], TI [5], VT [36], SSA [27], RAP [30], BSR [35], and Self-Universality [38]. Self-Universality is a targeted attack and thus we only report its performance in targeted setting.

Evaluation metric. Attack success rate (ASR, ↑) is used as the evaluation metric that is defined as the misclassified rate of adversarial examples by target models.

Hyperparameter configurations. For baselines, we set hyperparameters used in their original papers. For TPA, we set $\lambda = 5, b = 16, k = 0.05, N = 10$. Moreover, for all methods, we set iteration of 20^3, ϵ of 16, and step size of 1.6.

5.2 Attack Results

Attack results on normal models. Here we examine the attack effectiveness of our method on both convolutional neural networks and transformer-like neural networks and Table 1 reports the attack results. The following observation can be made. Simply speaking, as shown in Table 1, our

---

[3]Since our approximation involves Hessian matrix that is known to accelerate convergence, we increase the number of iterations to ensure the convergence of all attacks. See Appendix B.4 for attack results with iteration of 10. Overall, the increased iterations did not significantly affect the attack performance (less than 1%).

Table 2: The attack success rates (%) of different methods on secured models. Three different robust training methods are considered: adversarial training with L_2 perturbation ($L2 - \{0.03 \sim 5\}$) [31] and L_∞ perturbation (AdvIncV3 and EnsAdvIncResV2) [34], robust training with Styled ImageNet (SIN) and the mixture of Styled and natural ImageNet (SIN-IN) [16]. The best results are in bold.

Proxy Model	Method	AdvIncV3	EnsAdvIncResV2	SIN	SIN-IN	L2-0.03	L2-0.05	L2-0.1	L2-0.5	L2-1	L2-3	L2-5
ResNet50	BIM	41.81	33.52	50.51	51.56	63.67	64.06	62.81	61.10	63.98	68.27	74.13
	MI	58.78	43.24	75.46	96.05	82.83	83.47	80.31	74.74	75.00	75.08	77.98
	NI	58.11	43.11	75.13	96.79	83.57	83.70	79.97	74.52	75.15	75.33	78.11
	DI	54.34	45.03	75.84	92.04	78.27	78.19	75.61	69.57	70.94	71.53	75.74
	TI	48.52	39.72	66.35	88.27	75.51	75.05	72.76	67.04	69.21	70.87	75.61
	VT	59.72	50.89	69.03	94.06	77.27	77.22	74.21	68.24	69.80	70.92	75.48
	SSA	60.13	50.48	70.31	97.63	80.61	80.46	76.84	69.06	70.28	71.25	75.69
	RAP	60.17	51.02	84.34	97.94	81.72	81.43	80.22	75.49	75.06	72.48	76.05
	BSR	64.02	56.35	88.47	98.21	85.34	85.57	83.09	78.31	78.31	76.28	80.61
	Ours	**74.57**	**72.81**	**99.52**	**99.90**	**96.20**	**96.45**	**96.48**	**93.65**	**91.76**	**88.19**	**87.24**
ResNet152	BIM	41.84	33.85	48.95	42.14	63.11	63.47	62.76	60.92	64.11	68.19	74.06
	MI	58.78	45.15	71.40	82.58	82.55	82.96	80.13	74.87	75.23	75.36	78.21
	NI	58.98	44.74	71.79	83.83	82.70	83.04	80.08	74.80	75.13	75.28	78.14
	DI	54.97	47.58	72.50	80.00	77.27	76.86	75.38	69.46	70.26	71.56	75.59
	TI	49.34	41.48	63.11	72.76	74.21	74.44	72.22	66.61	69.06	71.35	75.48
	VT	51.22	43.42	65.10	78.62	76.58	76.07	73.44	68.11	69.72	71.02	75.41
	SSA	51.30	41.94	67.68	84.62	79.34	79.31	76.61	69.29	70.20	71.20	75.54
	RAP	52.38	43.36	76.19	85.47	81.12	81.21	79.54	75.42	74.95	72.16	75.39
	BSR	53.28	45.56	80.95	88.37	84.95	84.84	82.78	77.57	77.60	75.61	80.42
	Ours	**71.66**	**79.46**	**99.21**	**99.80**	**95.64**	**95.43**	**95.15**	**91.76**	**90.05**	**85.36**	**84.72**
DenseNet121	BIM	41.99	34.08	48.80	40.26	63.09	63.11	62.55	60.92	64.01	68.27	74.16
	MI	59.31	45.03	71.25	79.44	81.71	83.27	80.08	74.64	76.02	75.77	78.65
	NI	59.03	45.08	72.07	80.31	81.53	83.14	79.80	74.87	75.69	75.79	78.57
	DI	55.00	47.19	73.21	77.19	76.99	77.07	75.94	69.31	70.69	71.68	75.69
	TI	49.13	40.71	62.24	67.58	73.62	73.57	71.96	67.60	68.85	71.10	75.38
	VT	60.26	51.73	64.26	75.28	76.28	76.38	74.21	68.55	69.97	71.05	75.54
	SSA	61.84	51.86	69.08	84.06	79.64	79.85	76.94	69.67	70.51	71.45	75.77
	RAP	62.17	53.02	75.62	84.92	81.62	81.10	79.65	75.45	74.78	72.00	75.64
	BSR	64.02	56.35	81.14	88.25	85.14	84.69	82.24	78.22	77.39	76.18	79.78
	Ours	**72.19**	**79.64**	**99.16**	**99.64**	**95.31**	**95.69**	**95.56**	**92.65**	**90.82**	**86.63**	**85.71**
DenseNet201	BIM	42.07	34.18	48.93	40.99	62.88	63.62	62.78	60.94	64.11	68.42	74.13
	MI	59.92	45.20	71.33	81.53	82.98	83.19	80.03	75.54	76.20	75.71	78.60
	NI	59.90	44.41	72.27	82.42	82.55	83.06	80.18	75.28	75.79	75.66	78.21
	DI	56.53	48.70	74.69	80.03	77.12	77.45	76.73	70.05	71.43	71.71	75.82
	TI	50.08	41.20	62.68	69.34	74.21	74.29	72.60	67.68	69.67	71.02	75.64
	VT	61.76	52.86	65.23	76.51	76.66	76.86	74.59	68.67	70.26	71.07	75.64
	SSA	62.30	53.76	69.59	85.05	79.64	79.39	76.76	70.20	70.92	71.38	75.99
	RAP	63.17	54.02	75.70	85.27	80.76	81.06	80.16	74.74	74.16	71.70	75.33
	BSR	65.02	56.35	81.41	88.03	85.20	85.10	83.06	77.45	77.56	76.17	80.10
	Ours	**70.99**	**78.21**	**98.98**	**99.67**	**95.41**	**95.94**	**94.92**	**91.56**	**89.44**	**85.51**	**84.57**

Table 3: The targeted attack success rates of different methods. The proxy model is ResNet50.

Method	DenseNet121	EfficientNet	InceptionV3	ConvNet	WideResNet50	VGG19	ViT	Swin
MI	4.20	0.60	0.00	0.20	5.20	0.80	0.00	0.20
DI	1.60	0.20	0.20	0.00	2.00	0.70	0.00	0.10
VT	4.70	1.10	0.50	1.10	5.40	1.30	0.40	0.40
SSA	5.40	2.30	1.50	0.70	6.60	3.60	0.80	1.20
RAP	18.60	13.90	8.20	6.70	12.20	7.10	7.30	5.80
BSR	19.40	14.00	8.60	7.90	14.50	9.30	7.70	6.50
Self-Universality	22.00	18.00	9.60	8.70	15.70	8.50	9.70	6.70
Ours	**31.90**	**25.20**	**14.30**	**12.20**	**22.60**	**13.20**	**13.40**	**9.80**

method defeats all the baselines by a significant margin and is the winning attack. For instance, when employing ResNet50 as the proxy model, our attack witnesses an average gain of roughly 9% on attack success rate, even when compared to the state-of-the-art transfer-based attack SSA.

Attack results on secured models. We evaluate the effectiveness of TPA on models defended by robust training. We stick to employing undefended models as proxy models and this is a more challenging setting, given that proxy and target models are more divergent. The attack results are reported in Table 2 and we draw the following conclusions. Similar to the attack results on normal models, TPA still can craft more transferable adversarial examples against robust models. In particular, as can be seen in Table 2, TPA enhances ASRs over SSA by a large margin.

Targeted attack results. We also investigate the attack effectiveness of TPA for targeted attacks, which is a more challenging attack setting compared to untargeted attack setting. Following the evaluation setting for targeted attacks [49], Table 3 reports the targeted attack results of different attack methods. As can be seen in Table 3, the attack effectiveness of TPA on the targeted setting is so striking, even when compared to Self-Universality, which is specifically designed for targeted attacks. compared with baselines.

Attack results with varying perturbation budgets. Small perturbation budgets ϵ raise the difficulty in conducting transfer-based attacks and we further evaluate the attack effectiveness of different

Table 4: The scoring for the effectiveness of adversarial examples against real-world applications. We randomly extract 100 samples from ImageNet and generate adversarial examples for them using TPA and ResNet50. We enlist a volunteer to assess the consistence between the image contents with the predictions made by applications. A lower rating reflects a higher effectiveness of the attack.

Score ($\downarrow$)	Classification	Object Detection	Google Search	Bing Search	Yandex Search	Baidu Search	GPT-4	Claude3
5	1	3	0	0	0	0	2	0
4	7	21	10	6	5	4	15	12
3	13	7	18	11	13	4	27	26
2	9	4	16	21	17	10	30	30
1	70	65	56	62	65	82	26	32

methods with varying perturbation budgets. Figure 2 illustrates the ASRs of VT, SSA, and TPA. In short, TPA still dominates all settings in terms of ASRs.

6 Evaluation in Real World Applications

This section evaluates the performance of TPA against real applications, which is more challenging but also leads to a more reliable evaluation due to the following three reasons:

- **Application architecture and complexity.** Real-world applications are likely to employ sophisticated models that are difficult to replicate in a controlled research environment. Besides, such applications probably incorporate practical defenses.

- **Training setting.** Publicly available models mostly share similar training recipes (ImageNet). However, the recipes of real industry environments are far more complicated.

- **Structure of the output.** Real-world systems often output multiple hierarchical labels with associated confidences, instead of logits. Along this, the output space of real applications is vastly larger than that of proxy models, i.e., inconsistency in label space.

Google MLaaS platforms. We attack Google Cloud Vision Application including image classification and object detection, which is recognized as one of the most advanced AI services. We craft 100 adversarial examples against the application. We first collect the application's responses to these examples and ask a volunteer to score the consistency between these examples with the corresponding responses [4]. The scoring ranges from 1 (totally wrong) to 5 (precise), with higher scores indicating weaker attack performance. See the appendix C for further details about the scoring process. Table 4 shows the striking effectiveness of TPA against Google Service, while Appendix D provides visualization results. The ASRs for image classification and object detection are around 70% and 80%, respectively, considering scores of 1 or 2 as a successful attack. Besides, we find that ineffective adversarial examples often involve entities of persons, a label not covered in ImageNet. As a result, TPA is not guided to corrupt the features of "Person", leading to the ineffectiveness of these adversarial examples.

Reverse image search engines. Given an image of interest, reverse image search engines enable searching for the most similar ones and creates great convenience and benefits. The task is remarkably different from image classification and object detection and here we test the effectiveness of TPA against search engines. We attack Top-4 search engines suggested by the site, including Google, Bing, Yandex, and Baidu. We reuse adversarial ones crafted for Google Service and score their effectiveness ranging from 1 to 5, inversely related to the similarity of retrieved images to the original image. Table 4 reports attack results and Appendix D shows the images retrieved by four search engines for original and adversarial ones. Four engines present notable vulnerabilities against adversarial examples by TPA, particularly Baidu Picture Search, which returns completely unrelated images.

Multi-modal AI Platforms. Multimodal AI platforms, which process and integrate various types of data, are inherently more difficult to attack due to their complexity and their ability for contextual integration across multiple modalities. We evaluate the effectiveness of our method on the multimodal AI platforms of OpenAI and Amazon, specifically on GPT-4 and Claude3, respectively. Table 4 reports the numerical results where TPA achieves ASRs as high as 56% and 62% for GPT-4 and Claude3 if scores of 1 or 2 are deemed as a successful attack. Specifically, Figure 1 provides an

---

[4]See Appendix E for more information on the recruitment of volunteers and the evaluation procedure.

Figure 3: Example.

GPT-4 Response: The image appears to be a digital painting or an animated, glitch-like effect applied to a photograph of a green cabbage or lettuce being chopped with a knife. The colors are vivid, and the motion suggests that the knife is in the act of slicing through the vegetable.

Claude3 Response: This image shows various plastic bags filled with different types of produce and vegetables. The bags contain items like lettuce or leafy greens, carrots, and what appears to be herbs or greens of some kind. The plastic bags allow the contents to be visible from the outside.

example wherein both GPT-4 and Claude3 fail to accurately classify the green fruit and mistakenly perceive a knife and plastic bag instead.

7 Ending Remark

In this paper, we established a bound for adversarial transferability and demonstrated that flat adversarial examples do not inherently enjoy improved transferability. Drawing upon our theoretical insights, we developed TPA, which optimizes both the local effectiveness term and a surrogate of the transfer-related loss term to generate adversarial examples. We conducted extensive experiments to validate the effectiveness of TPA across standard benchmarks and multiple real-world applications. It is our hope that our theoretical analysis will inspire further research that delves deeper into adversarial transferability, driving forward the development of more sophisticated and effective adversarial attacks and defenses.

Acknowledgments and Disclosure of Funding

References

[1] Maksym Andriushchenko, Francesco Croce, Maximilian Mueller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *International Conference on Machine Learning*, 2023. URL `https://api.semanticscholar.org/CorpusID:256846369`.

[2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[3] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 2017. URL `https://api.semanticscholar.org/CorpusID:7636159`.

[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[5] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4307–4316, 2019.

[6] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Y. F. Tan. Efficient sharpness-aware minimization for improved training of neural networks. *ArXiv*, abs/2110.03141, 2021. URL `https://api.semanticscholar.org/CorpusID:238419436`.

[7] Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Y. F. Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *ArXiv*, abs/2205.14083, 2022. URL `https://api.semanticscholar.org/CorpusID:249152070`.

[8] Mingyuan Fan, Cen Chen, Chengyu Wang, and Jun Huang. Exploiting pre-trained models and low-frequency preference for cost-effective transfer-based attack. *ACM Transactions on Knowledge Discovery from Data*.

[9] Mingyuan Fan, Cen Chen, Chengyu Wang, and Jun Huang. On the trustworthiness landscape of state-of-the-art generative models: A comprehensive survey. *arXiv preprint arXiv:2307.16680*, 2023.

[10] Mingyuan Fan, Cen Chen, Chengyu Wang, Wenmeng Zhou, and Jun Huang. On the robustness of split learning against adversarial attacks. In *ECAI 2023*, pages 668–675. IOS Press, 2023.

[11] Mingyuan Fan, Cen Chen, Chengyu Wang, Wenmeng Zhou, and Jun Huang. On the robustness of split learning against adversarial attacks. In *ECAI*, 2023.

[12] Yan Fang, Zhongyuan Wang, Jikang Cheng, Ruoxi Wang, and Chao Liang. Promoting adversarial transferability with enhanced loss flatness. *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1217–1222, 2023. URL `https://api.semanticscholar.org/CorpusID:261126940`.

[13] Roger Fletcher. Practical methods of optimization. 1988.

[14] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *ArXiv*, abs/2010.01412, 2020. URL `https://api.semanticscholar.org/CorpusID:222134093`.

[15] Zhijin Ge, Fanhua Shang, Hongying Liu, Yuanyuan Liu, and Xiaosen Wang. Boosting adversarial transferability by achieving flat local maxima. *ArXiv*, abs/2306.05225, 2023. URL `https://api.semanticscholar.org/CorpusID:259108262`.

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bygh9j09KX`.

[17] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *ArXiv*, abs/2101.04898, 2021. URL `https://api.semanticscholar.org/CorpusID:231592390`.

[18] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. 2020. URL `https://api.semanticscholar.org/CorpusID:59842968`.

[19] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6085, 2018. URL `https://api.semanticscholar.org/CorpusID:54434655`.

[20] Zhiwei Jia and Hao Su. Information-theoretic local minima characterization and regularization. In *International Conference on Machine Learning*, pages 4773–4783. PMLR, 2020.

[21] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. 2016.

[22] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Loddon Yuille. Learning transferable adversarial examples via ghost networks. *ArXiv*, abs/1812.03413, 2020.

[23] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *International Conference on Machine Learning*, pages 20763–20786. PMLR, 2023.

[24] Kaisheng Liang and Bin Xiao. Styless: Boosting the transferability of adversarial examples. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8163–8172, 2023. URL `https://api.semanticscholar.org/CorpusID:258297932`.

[25] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv: Learning*, 2020.

[26] Zihao Liu, Qi Liu, Tao Liu, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868, 2018. URL `https://api.semanticscholar.org/CorpusID:3863554`.

[27] Yuyang Long, Qi li Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. 2022.

[28] Muzammal Naseer, Salman Hameed Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Murat Porikli. A self-supervised approach for adversarial robustness. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 259–268, 2020. URL `https://api.semanticscholar.org/CorpusID:219559020`.

[29] Yaguan Qian, Yu qun Wang, Bin Wang, Zhaoquan Gu, Yu-Shuang Guo, and Wassim Swaileh. Hessian-free second-order adversarial examples for adversarial learning. *ArXiv*, abs/2207.01396, 2022.

[30] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *Advances in Neural Information Processing Systems*, 2022.

[31] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[32] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. In *Proceedings of the 40th International Conference on Machine Learning*, pages 29894–29918, 2023.

[33] Elias M. Stein and Rami Shakarchi. Real analysis: Measure theory, integration, and hilbert spaces. 2005.

[34] F Tramèr, D Boneh, A Kurakin, I Goodfellow, N Papernot, and P McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, 2018.

[35] Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. Boosting Adversarial Transferability by Block Shuffle and Rotation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2024.

[36] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, 2021.

[37] Xiaosen Wang, Xu He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16138–16147, 2021.

[38] Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yueping Jiang. Enhancing the self-universality for transferable targeted attacks. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12281–12290, 2023. URL `https://api.semanticscholar.org/CorpusID:257663620`.

[39] Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *ArXiv*, abs/2307.11007, 2023. URL `https://api.semanticscholar.org/CorpusID:259991268`.

[40] Dongxian Wu, Yisen Wang, Shutao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *ArXiv*, abs/2002.05990, 2020.

[41] Lei Wu and Weijie J. Su. The implicit regularization of dynamical stability in stochastic gradient descent. *ArXiv*, abs/2305.17490, 2023. URL `https://api.semanticscholar.org/CorpusID:258959231`.

[42] Tao Wu, Tie Luo, and Donald C. Wunsch. Gnp attack: Transferable adversarial examples via gradient norm penalty. *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3110–3114, 2023. URL `https://api.semanticscholar.org/CorpusID:259501820`.

[43] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Loddon Yuille. Mitigating adversarial effects through randomization. *ArXiv*, abs/1711.01991, 2017. URL `https://api.semanticscholar.org/CorpusID:3526769`.

[44] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Loddon Yuille. Improving transferability of adversarial examples with input diversity. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2725–2734, 2019.

[45] Yinhu Xu, Qi Chu, Haojie Yuan, Zixiang Luo, Bin Liu, and Nenghai Yu. Enhancing adversarial transferability from the perspective of input loss landscape. In *International Conference on Image and Graphics*, 2023. URL `https://api.semanticscholar.org/CorpusID:265068861`.

[46] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

[47] Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20247–20257, 2023. URL `https://api.semanticscholar.org/CorpusID:257365008`.

[48] Yang Zhao, Haoxian Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. *ArXiv*, abs/2202.03599, 2022. URL `https://api.semanticscholar.org/CorpusID:246652190`.

[49] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In *NeurIPS*, 2021.

[50] Yao Zhu, Yuefeng Chen, Xiaodan Li, Kejiang Chen, Yuan He, Xiang Tian, Bo Zheng, Yaowu Chen, and Qingming Huang. Toward understanding and boosting adversarial transferability from a distribution perspective. *IEEE Transactions on Image Processing*, 31:6487–6501, 2022.

A   Proof of Theorem 3.1

We first demonstrate a bound on our transfer loss term $D(x+\delta, y) = \mathcal{L}(F'(x+\delta), y) - \mathcal{L}(F(x+\delta), y)$. By employing Taylor expansion on $D(x+\delta, y) = \mathcal{L}(F'(x+\delta), y) - \mathcal{L}(F(x+\delta), y)$, we derive the following expression:

$$D(x, y) - D(x + \delta, y) = -\nabla L(F'(x+\delta), y)^\top \delta + \nabla L(F(x+\delta), y)^\top \delta. \tag{11}$$

Let us denote the underlying distribution of x as $p(x)$. We employ cross-entropy loss as the loss function. We use $F(x)[y]$ and $F'(x)[y]$ to denote the prediction probabilities of classifying input x as y. We consider minimizing $||\nabla L(F'(x+\delta), y)^\top \delta - \nabla L(F(x+\delta), y)^\top||_2^2$ over $p(x)$. There is:

$$\int p(x)||(\nabla \log F'(x+\delta)[y] - \nabla \log F(x+\delta)[y]^\top \delta||_2^2 dx$$

$$\leq \int p(x)||\delta||_2^2 ||\nabla \log F'(x+\delta)[y] - \nabla \log F(x+\delta)[y]||_2^2 dx \quad (by\ ||ab|| \leq ||a||\ ||b||)$$

$$= \int p(x)||\delta||_2^2 (||\nabla \log F'(x+\delta)[y]||_2^2 + ||\nabla \log F(x+\delta)[y]||_2^2) dx \tag{12}$$

$$- 2\int p(x)||\delta||_2^2 \nabla \log F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] dx.$$

The following proof is divided into two main steps. The first step involves deriving a bound on the second integral. The second step focuses on deriving a bound on $\int p(x)||\nabla \log F'(x+\delta)[y]||_2^2 dx$. Finally, we combine the results from both steps to obtain the bound for the equation above.

Step I: we handle the second integral of the above expression here: $\int p(x)||\delta||_2^2 \nabla \log F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] dx$**.** We have:

$$\int p(x)||\delta||_2^2 \nabla \log F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] dx$$

$$= \int \frac{p(x)}{F'(x+\delta)[y]} ||\delta||_2^2 \nabla F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] dx$$

$$\geq \int p(x)||\delta||_2^2 \nabla F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] dx \quad (use\ F'(x+\delta)[y] \leq 1)$$

$$= \sum_i \int p(x)||\delta||_2^2 \nabla F'(x+\delta)[y][i] \cdot \nabla \log F(x+\delta)[y][i] dx$$

$$= \sum_i p(x)||\delta||_2^2 F'(x+\delta)[y] \cdot \nabla \log F(x+\delta)[y][i]|_{-\infty}^{+\infty} \tag{13}$$

$$- \sum_i \int p(x)||\delta||_2^2 F'(x+\delta)[y]\nabla^2 \log F(x+\delta)[y][i,i] dx$$

$$= -\sum_i \int p(x)||\delta||_2^2 F'(x+\delta)[y]\nabla^2 \log F(x+\delta)[y][i,i] dx \quad (use\ p(\infty) = 0)$$

$$\geq -\sum_i \int p(x)||\delta||_2^2 F(x+\delta)[y]\nabla^2 \log F(x+\delta)[y][i,i] dx.$$

Since adversarial examples generated on the proxy model are often also somewhat effective against the target model, this implies that $\nabla F'(x+\delta)[y]^\top \nabla \log F(x+\delta)[y] \geq 0$. Furthermore, by using $F'(x+\delta)[y] \leq 1$, we can derive the first inequality in the above equation. $[i]$ and $[i,i]$ refer to i-th element in gradient and element in the i-th row and i-th column of Hessian matrix, respectively. From the fourth line to the fifth line, we use integration by parts. Then, we suggest $p(+\infty) = p(-\infty) = 0$, which is natural in the distribution of interest. In the last line, we use $F'(x+\delta)[y] \leq F(x+\delta)[y]$ and $\nabla^2 \log F(x+\delta)[y][i,i] \leq 0$. Since δ is crafted for the proxy model, it naturally follows that $F'(x+\delta)[y] \leq F(x+\delta)[y]$, i.e., Assumption 3.4. Moreover, the second derivative of $log(\cdot)$ is negative, and the second derivative of $F(x+\delta)[y][i,i]$ is 0, which leads to $\nabla^2 \log F(x+\delta)[y][i,i] \leq 0$. The second derivative being zero can be derived from Assumption 3.5, which is easily verifiable.

Step 2: now we derive the bound of $\int p(x)||\nabla \log F'(x+\delta)[y]||_2^2 dx$**.** Notice that here we have:

$$\int p(x+\delta)(||\nabla \log F'(x+\delta)[y]||_2^2 - ||\nabla \log F(x+\delta)[y]||_2^2)dx$$
$$= \int p(x)(||\nabla \log F'(x)[y]||_2^2 - ||\nabla \log F(x)[y]||_2^2)dx. \tag{14}$$

Based on Equation 14, we obtain:

$$\int p(x+\delta)||\nabla \log F'(x+\delta)[y]||_2^2 dx$$
$$= \int p(x+\delta)||\nabla \log F(x+\delta)[y]||_2^2 dx \tag{15}$$
$$+ \int p(x)(||\nabla \log F'(x)[y]||_2^2 - ||\nabla \log F(x)[y]||_2^2)dx.$$

For $\int p(x)(||\nabla \log F'(x)[y]||_2^2 - ||\nabla \log F(x)[y]||_2^2)dx$, by using $||a|| - ||b|| \leq |\,||a|| - ||b||\,| \leq ||a-b||$, we have:

$$\int p(x)(||\nabla \log F'(x)[y]||_2^2 - ||\nabla \log F(x)[y]||_2^2)dx$$
$$\leq \int p(x)||\nabla(\log F'(x)[y] - \log F(x)[y])||_2^2 dx. \tag{16}$$

Thus, there is:

$$\int p(x+\delta)||\nabla \log F'(x+\delta)[y]||_2^2 dx$$
$$\leq \int p(x+\delta)||\nabla \log F(x+\delta)[y]||_2^2 dx \tag{17}$$
$$+ \int p(x)||\nabla(\log F'(x)[y] - \log F(x)[y])||_2^2 dx.$$

Due to $p(x+\delta) \leq p(x)$, there is:

$$\int p(x+\delta)||\nabla \log F'(x+\delta)[y]||_2^2 dx$$
$$\leq \int p(x)||\nabla \log F(x+\delta)[y]||_2^2 dx \tag{18}$$
$$+ \int p(x)||\nabla(\log F'(x)[y] - \log F(x)[y])||_2^2 dx.$$

Furthermore, we let:

$$\begin{aligned} C &= \max \frac{p(x)}{p(x+\delta)} \\ &= \max \frac{p(x)}{p(x) + \nabla p(x)^\top \delta} \quad (\text{notice } \nabla p(x)^\top \delta \geq 0) \\ &= \max \frac{B_1}{B_1 + \nabla p(x)^\top \delta} \\ &= \max \frac{B_1}{B_1 + B_2||\delta||_2} \\ &= \frac{B_1}{B_1 + B_2||\delta||_2}. \end{aligned} \tag{19}$$

Notice that:

$$p(x+\delta) = p(x+\delta)\frac{p(x)}{p(x)} \geq p(x)\min \frac{p(x+\delta)}{p(x)} = \frac{B_1 + B_2||\delta||_2}{B_1}p(x). \tag{20}$$

For Equation 17, we have:

$$\int p(x)||\nabla \log F'(x+\delta)[y]||_2^2 dx$$

$$\leq C \int p(x)||\nabla \log F(x+\delta)[y]||_2^2 dx \tag{21}$$

$$+ C \int p(x)||\nabla(\log F'(x)[y] - \log F(x)[y])||_2^2 dx.$$

Combining results. Combining Equation 12 and 13, we have:

$$\int p(x)||(\nabla \log F'(x+\delta)[y] - \nabla \log F(x+\delta)[y])^\top \delta||_2^2 dx$$

$$\leq \int p(x)||\delta||_2^2(||\nabla \log F'(x+\delta)[y]||_2^2 + ||\nabla \log F(x+\delta)[y]||_2^2)dx$$

$$+ 2\sum_i \int p(x)||\delta||_2^2 F(x+\delta)[y]\nabla^2 \log F(x+\delta)[y][i,i]dx$$

$$\leq (1+C) \int p(x)||\delta||_2^2||\nabla \log F(x+\delta)[y]||_2^2 dx \tag{22}$$

$$+ 2\sum_i \int p(x)||\delta||_2^2|\nabla^2 \log F(x+\delta)[y][i,i]|dx$$

$$+ C \int p(x)||\delta||_2^2||\nabla(\log F'(x)[y] - \log F(x)[y])||_2^2 dx.$$

In the second line of Equation 13, we use $F(x+\delta) \leq 1$. Then, according to $D(x,y) - D(x+\delta,y) = -\nabla L(F'(x+\delta),y)^\top \delta + \nabla L(F(x+\delta),y)^\top \delta$, we have

$$\mathbb{E}_{p(x)}||D(x+\delta,y)||_2^2 \leq \mathbb{E}_{p(x)}||D(x,y)||_2^2 + \mathbb{E}_{p(x)}||\nabla L(F'(x+\delta),y)^\top \delta + \nabla L(F(x+\delta),y)^\top \delta||_2^2. \tag{23}$$

By substituting the result from Equation 22 into the above expression, we can obtain the bound presented in Theorem 3.1. The bound of $\mathcal{L}(F'(x+\delta),y)$ can similarly be obtained using basic norm inequalities, which we will not elaborate on here.

The applicability for big δ. In our proof, we apply Taylor expansion which demonstrates effectiveness for small changes in the input, i.e., small δ. Our discussion extends to the applicability of our theory when faced with sizable δ values. In fact, the generation of adversarial examples is an iterative process. Taylor expansion remains accurate with small update steps. Moreover, our proof mainly relies on $p(x) \geq p(x+\delta)$. This implies that as long as we acknowledge that a larger δ leads to a decrease in $p(x+\delta)$, our theory continues to hold. It naturally follows that a larger δ typically correlates with diminished classification accuracy.

B Supplementary Experiments

B.1 Attack Results on Other Defenses

We here evaluate the attack performance of TPA against other defenses including R&P [43], NIPS-R3 [5], FD [26], ComDefend [19], RS [18], NRP [28]. These defense mechanisms deviate significantly from adversarial training as they rely on data processing techniques to purify adversarial examples into normal ones. The detailed settings follow Long et al. [27]. Table 5 reports the attack performance of VT, SSA, RAP, and our method against these defenses. TPA still consistently surpasses baselines by a large margin.

B.2 Attention Visualization for Targeted Attacks

Figure 4 shows more visualizations of attention maps. As can be seen, the adversarial examples produced by VT and SSA only can slightly decrease the attention of the target model to entities of

---

[5]https://github.com/anlthms/nips-2017/blob/master/poster/defense.pdf

Table 5: The attack success rates (%) of attacks against various defenses. We use ResNet50 as the proxy model.

Attack	R&P	NIPS-R3	FD	ComDefend	RS	NRP
VT	64.53	70.69	73.4	87.33	52.14	43.97
SSA	72.10	75.32	76.06	91.27	61.86	52.91
RAP	93.15	92.44	92.58	95.59	76.09	76.28
Ours	**95.81**	**97.44**	**94.02**	**97.64**	**85.54**	**83.47**

images, while TPA can significantly distract the attention of the target model from the entities to trivial regions.

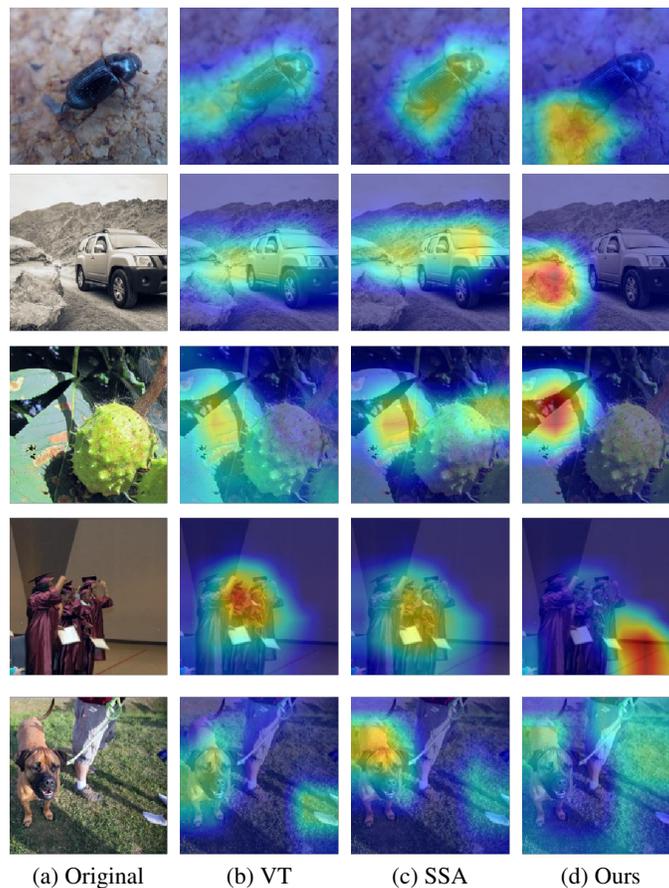|			
(a) Original	(b) VT	(c) SSA	(d) Ours

Figure 4: We conduct targeted attacks and visualize attention maps of the target model to the resultant adversarial images.

B.3 Impact of Hyperparameters

TPA involves four hyperparameters that can significantly impact the attack performance:

- λ is a balance factor between the loss of resultant adversarial examples and the flatness of the region around the adversarial ones. The bigger λ attaches more attention to promoting adversarial examples toward flat regions.
- b suggests how wide the region around adversarial examples is desired to be flat.
- When comes to the implementation of TPA, Hessian matrix is approximately evaluated and a inappropriate k probably induces non-trivial errors.

- We extract N samples around adversarial examples to estimate the flatness of the region around the adversarial ones. Generally, the estimation accuracy raises with increasing N.

The impact of λ. Figure 5(a) illustrates the attack success rates of crafted adversarial examples by TPA with varying λ against target models. We see that as λ increases, the attack effectiveness of TPA presents the tendency to climb initially and decline thereafter. If λ is small like $\lambda = 0.1$, most attention of Equation 5 is put into optimizing L while ignoring the flatness of the region around crafted adversarial examples, *i.e.*, degrading into vanilla transfer-based attacks and causing that the adversarial ones are more likely to be trapped by model-specified sharp regions. As a remedy, properly increasing λ can grab part of the attention of Equation 5 to focus on the flatness of the surrounding region, so that the resultant adversarial examples have more chance to evade the model-specified sharp regions and are more transferable. However, it should be stressed that too bigger λ induces Equation 5 to only focus on the flatness and de-emphasize whether the region poses a threat to the proxy model, leading to a reduction in the attack success rate. Therefore, carefully adjusting λ is necessary.
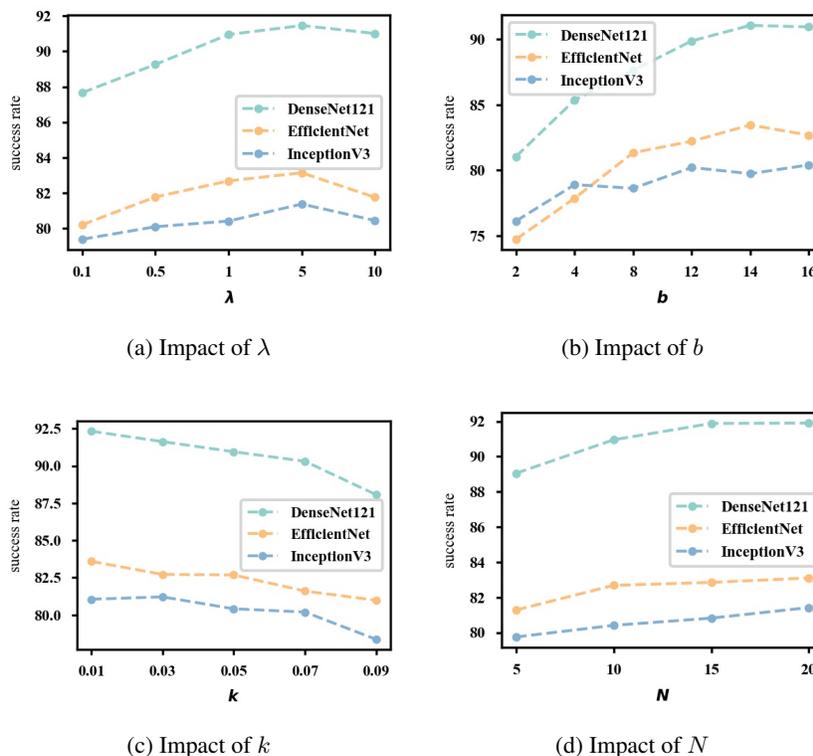

(a) Impact of λ

(b) Impact of b

(c) Impact of k

(d) Impact of N

Figure 5: The attack effectiveness of TPA with varying $\lambda \in \{0.1, 0.5, 1, 5, 10\}, b \in \{1, 2, 4, 8, 12, 16\}, k \in \{0.01, 0.03, 0.05, 0.07, 0.09\}, N \in \{5, 10, 15, 20\}$. The proxy model is ResNet50. We set $\epsilon = 8$.

The impact of b. We examine the attack effectiveness of TPA by changing the hyperparamter b and Figure 5(b) shows the attack results over different proxy-target pairs. Overall, the attack performance of TPA steadily increases with increasing b. The reason for it is that, a small b makes the extracted samples too close to the resultant adversarial examples and causes that the flatness of the region around the resultant adversarial examples cannot be effectively evaluated. Furthermore, increasing b alleviates the issue so as to boost the transferability of produced adversarial examples. Moreover, interestingly, we find that, when employing ResNet50 and EfficientNet, increasing b from 14 to 16 slightly hurts the transferability of produced adversarial examples, and this is probably attributed to the extracted samples being a little far from the crafted adversarial examples.

The impact of k. Figure 5(c) shows the influence of the hyperparameter k to the performance of TPA and we observe that increasing k weakens the attack effectiveness of TPA. In fact, as shown in

Section 4.3, Hessian matrix is approximately replaced by gradient difference, *i.e.*, Equation 10, and the feasibility of the approximation solution depends on a small k to omit the error of approximation induced by Taylor expansion. Hence, it is intuitive why a bigger k incurs degradation on the attack success rates of TPA.

The impact of N. Figure 5(d) shows the attack performance of TPA with different N. Intuitively, by sampling more instances from the region around the resultant adversarial examples, *i.e.*, increasing N, we can make a more accurate estimation of the flatness of the region, which in turn can decrease estimation errors and then strengthen the transferability of generated adversarial examples. As expected, Figure 5(d) validates this point that increasing N promotes the transferability of produced adversarial examples.

B.4   Attacks with Varying Iteration

We here investigate the impact of iterations on attack performance. Table 6 reports the performance of different attacks with varying numbers of iterations. It can be observed that there is negligable change in attack performance when the number of iterations is increased from 10 to 20. The results suggest that the attack methods achieve convergence with 10 iterations.

Table 6: The attack effectiveness of different attacks with varying iterations. We use ResNet50 as the proxy model.

Attack	ResNet50	DenseNet121	EfficientNet
VT (10 iter)	100.00	88.76	81.51
VT (20 iter)	100.00	89.62	81.63
SSA (10 iter)	100.00	95.29	90.72
SSA (20 iter)	100.00	95.51	91.56
Ours (10 iter)	99.80	99.08	99.25
Ours (20 iter)	99.85	99.69	99.52

C   Score Implication

We begin by gathering predictions from Google MLaaS platforms for both label detection and object detection over crafted adversarial examples. We assign scores to these predictions along five discrete levels, with scores of 1 through 5 indicating the degree of accuracy: totally wrong, slightly incorrect, strange but not incorrect, relatively reasonable, and precise. A score of 1 suggests that the images do not contain the objects predicted. A score of 2 indicates minor errors in the predictions, such as identifying flowers instead of trees in a tree image. A score of 3 suggests that the main object in the image is not correctly recognized, for example, predicting stones, roads, or tires for a car image. A score of 4 denotes accurately identifying the general type of the main object in the images, while a score of 5 indicates that the system can fully and correctly identify the main object in the image. Next, a volunteer (See Section E for more information on the recruitment of volunteers and the evaluation procedure) is asked to rate the consistency between the predictions and the images based on these scores For search engines, a similar evaluation procedure is followed, where we assess the similarity between the retrieved images and their corresponding original images.

D   A visualization of different attacks against Google Service and Search Engines

Figure 6 shows a visualization of different attacks against Google Service, and we use the format {attack method}-{task} to denote the attack used to craft adversarial examples and the corresponding test task.

For image classification, consistent with our intuition, the original image is predicted into plant-related categories. However, the returned predictions for the adversarial examples produced via VT and SSA are close to the ground-truth label of the original image. Therefore, the adversarial examples cannot be deemed to be threats against Google Cloud Vision. In contrast, the adversarial examples

generated by our method indeed mislead Google Cloud Vision, where the predictions with the highest confidence are bird, water, and beak, and these predictions are fairly irrelevant to the original images.

For object detection, we can obtain similar conclusions. The original image is correctly detected. The adversarial examples produced by VT seem to fool Google Cloud Vision to some extent, while SSA can fully mislead Google Cloud Vision. Also, our adversarial examples trick Google Cloud Vision with higher confidence than VT.

Figure 7 visualizes an example of TPA against four state-of-the-art search engines. We observe that search engines fetch high-quality and similar images for normal samples. However, when we input the generated adversarial examples, the quality of retrieved images noticeably deteriorates, particularly in the case of Baidu.

(a) Original-Classification

(b) VT-Classification

(c) SSA-Classification

(d) Ours-Classification

(e) Original-Object-Detection

(f) VT-Object-Detection
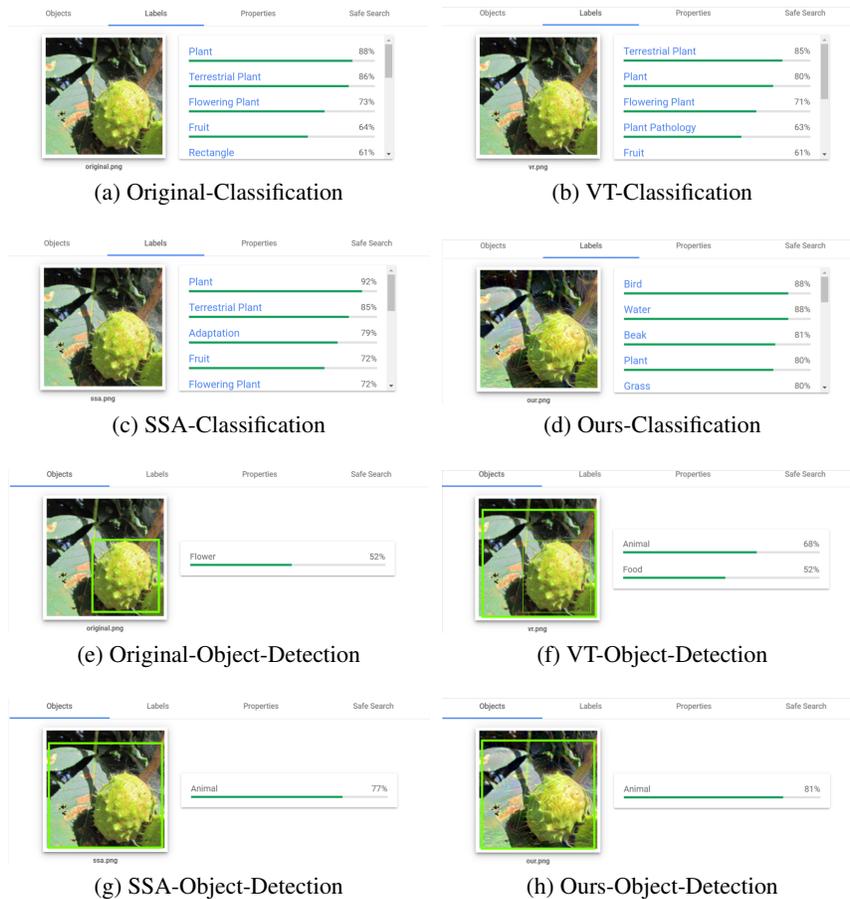
(g) SSA-Object-Detection

(h) Ours-Object-Detection

Figure 6: The attack results on Google Cloud Vision including image classification and object detection tasks. We do not know any knowledge of it and the proxy model is ResNet50.

E   Ethics Statement

This paper designs a novel approach to enhance the transferability of adversarial examples. While this approach is easy-to-implement and seems harmful, it is believed that the benefits of publishing TPA outweigh the potential harms. It is better to expose the blind spots of DNNs as soon as feasible because doing so can alert deployers to be aware of potential threats and greatly encourage AI community to design corresponding defense strategies.

For the human assessment process (Section 6), we generated adversarial examples and collected the predictions of different applications on these adversarial examples. Throughout the entire process, all communication with the volunteer, including recruitment, was conducted anonymously. Similarly,
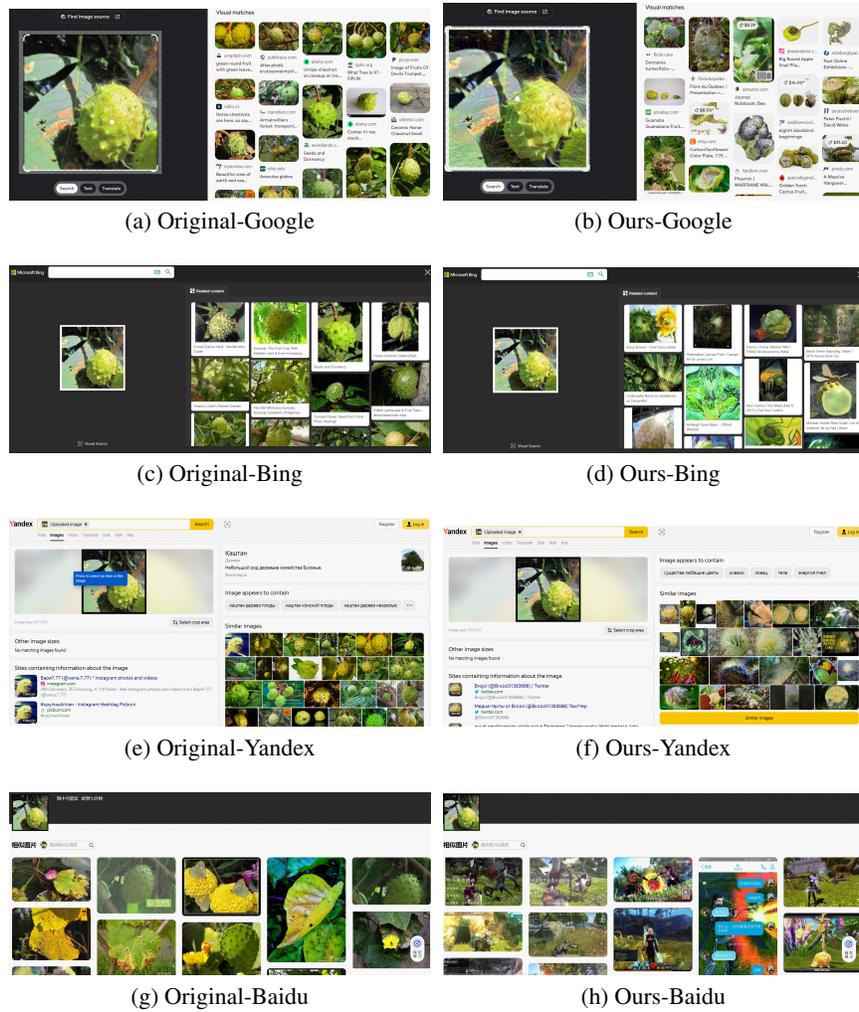
(a) Original-Google

(b) Ours-Google

(c) Original-Bing

(d) Ours-Bing

(e) Original-Yandex

(f) Ours-Yandex

(g) Original-Baidu

(h) Ours-Baidu

Figure 7: An example for attacking four state-of-the-art search engines.

we do not knew the personal information regarding to the volunteer. The volunteer were unaware of our specific objectives, ensuring that there was no interest between the volunteer and us. The volunteer came from a certain university and possessed normal discernment abilities. During the rating process, the volunteer were unaware of whether a given sample was an adversarial example or the attack method used to generate it. Therefore, there was no bias towards any particular type of attack method from the volunteer. Overall, the evaluation process was relatively fair.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims reflect the contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our appendix contains ethics statement.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our appendix contains complete proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide source codes.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide source codes.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We make detailed experiment settings in our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We only run one trial.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: Common GPUs are capable of running our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We follow code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We explain this in Introduction.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not releases any new models or datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We do this.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Our appendix contains this.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: There are no risks in our experiment for study participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.