Generate Universal Adversarial Perturbations for Few-Shot Learning

Yiman Hu **Yixiong Zou*** **Ruixuan Li*** **Yuhua Li**
School of Computer Science and Technology, Huazhong University of Science and Technology
{imane, yixiongz, rxli, idcliyuhua}@hust.edu.cn

Abstract

Deep networks are known to be vulnerable to adversarial examples which are deliberately designed to mislead the trained model by introducing imperceptible perturbations to input samples. Compared to traditional perturbations crafted specifically for each data point, Universal Adversarial Perturbations (UAPs) are input-agnostic and shown to be more practical in the real world. However, UAPs are typically generated in a close-set scenario that shares the same classification task during the training and testing phases. This paper demonstrates the ineffectiveness of traditional UAPs in open-set scenarios like Few-Shot Learning (FSL). Through analysis, we identify two primary challenges that hinder the attacking process: the task shift and the semantic shift. To enhance the transferability of UAPs in FSL, we propose a unifying attacking framework addressing these two shifts. The task shift is addressed by aligning proxy tasks to the downstream tasks, while the semantic shift is handled by leveraging the generalizability of pre-trained encoders.The proposed Few-Shot Attacking FrameWork, denoted as FSAFW, can effectively generate UAPs across various FSL training paradigms and different downstream tasks. Our approach not only sets a new standard for state-of-the-art works but also significantly enhances attack performance, exceeding the baseline method by over 16%.

1 Introduction

Deep neural networks[16, 12] have made significant advancements in a variety of computer vision tasks. Nowadays, there is a growing trend of pre-training a model that achieves strong generalization capabilities and subsequently fine-tuning it for different unseen downstream tasks. A promising method to handle this open set problem is Few-Shot Learning (FSL), which learns a model that can rapidly adapt to unseen tasks with only a limited number of samples.

Meanwhile, deep networks have shown to be vulnerable to adversarial attacks [41, 14, 5], which are deliberately designed to deceive a trained model by introducing imperceptible perturbations to the input samples. Given the widespread adoption of the pre-training and fine-tuning paradigm, it is crucial to acknowledge and address the security concerns associated with such approaches.

For a more applicable way to attack various downstream tasks, we focus on Universal Adversarial Perturbations (UAPs) [26]. This kind of perturbation can be applied to all images once generated, eliminating the need for crafting image-dependent perturbations for each task.

However, current approaches for generating UAPs have primarily concentrated on close-set scenarios [26, 15, 30], where the classification tasks for both training and finetuning are essentially identical. Despite achieving a high Attack Success Rate (ASR) in their respective settings, our research reveals

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3261

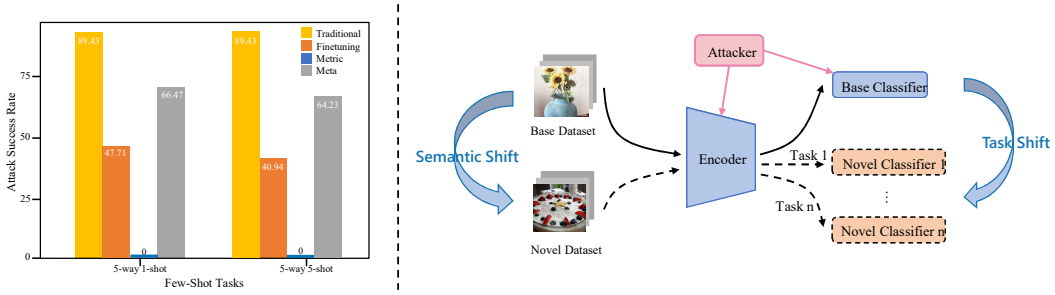


Figure 1: (Left) A decline in the Attack Success Rate (ASR) can be observed when conventional UAPs are applied to the FSL scenario. The yellow bar represents performance in traditional scenarios, while the other three indicate performance in different FSL training paradigms. For the metric-based paradigm, traditional methods even fail to generate UAPs. The detailed setting can be found in Section 4.1 (Right). Solid arrows represent the pre-training phase, while dotted arrows depict the testing phase. The attacker can only manipulate the pre-trained model, which suffers from the semantic shift of different datasets and the task shift from base to novel tasks.

that in FSL scenarios these techniques are ineffective, and in some cases, incapable of crafting UAPs, as shown in the left panel of Fig. 1.

To better understand how to effectively attack the FSL tasks, we conducted a comprehensive evaluation and analysis of the challenges involved, namely task shift and semantic shift, as depicted in the right panel of Fig. 1. To further fill up these two shifts and achieve a more generalizable UAP, we establish a baseline attacking framework as an initial point and progressively improve upon it. To address the task shift, we emphasize the significance of learning task bias for UAP, and handle this shift by constructing proxy tasks. To tackle the semantic shift, we demonstrate the effectiveness brought by the encoder's generalizability and eliminate the need for fine-tuning in the process of UAP generation.

By systematically addressing the shifts, we successfully generate the single perturbation only based on the parameters of the pre-trained encoder, without any prior knowledge of the pre-training dataset or downstream tasks. The ASRs of our universal perturbation surpass others by 16.49% for 5-way 1-shot tasks and 17.27% for 5-way 5-shot tasks. Additionally, our attacking framework(FSAFW) unifies the generation of UAPs in various FSL training paradigms, including finetuning-based [10, 47, 7], meta-based [13, 35, 39], and metric-based [43, 36, 50, 20] approaches, yielding state-of-the-art results in all cases. We summarize our main contributions as follows:

• We propose a new standard for the study of UAP in FSL scenarios, which highlights the limitations of traditional UAPs in the context of FSL.

• We provide a thorough analysis of the associated challenges, the task shift and semantic shift, which diminish the effectiveness of UAPs on downstream FSL tasks.

• We construct an attacking framework and fill up the two shifts step by step to enhance the attack performance, which is effective across various FSL paradigms.

• Our proposed method significantly advances state-of-the-art methods in FSL attacking performance, with an increase of over 16% in ASR in our standard.

2 Related Works

Adversarial attacks can be categorized into two types: image-dependent attacks and universal attacks. Image-dependent attacks have been widely studied [14, 23, 5, 22, 11]. The concept of Universal Adversarial Perturbations (UAPs) was initially introduced by [26] using an iterative Deepfool attack [27] applied to individual image samples. [28] developed a data-independent approach to generate UAPs. Furthermore, [49] validated the efficacy of using random source images. Several studies have explored the use of generative models to produce more generalized and natural-appearing UAPs. [15] were the first to utilize the generative network. [30] introduced GAP, a framework applicable to both classification and semantic segmentation models. [52] trained a generator by leveraging a

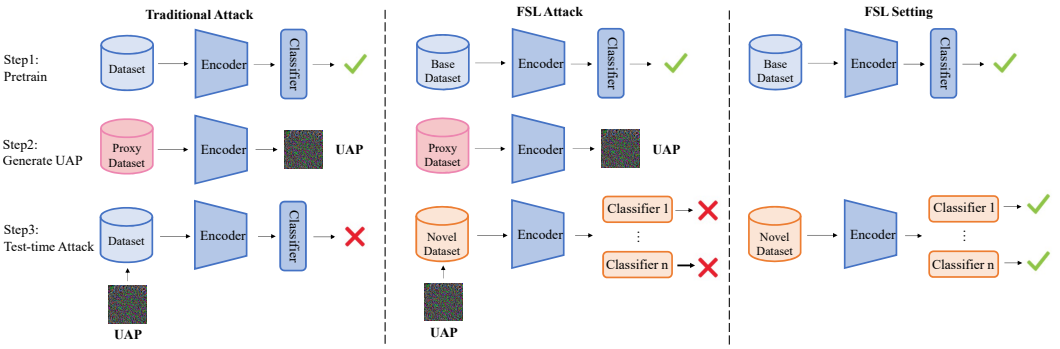


Figure 2: Comparison of different scenarios. In contrast to traditional settings, the FSL framework introduces two distinct challenges: the task shift from pre-training to testing, and the semantic shift from the base to the novel dataset. Attackers must overcome the two shifts to mount successful attacks in the FSL scenario.

contrastive loss function. However, none of these studies address the specific challenges associated with attacking few-shot learning tasks.

Few-shot learning is a machine learning paradigm that aims to recognize novel classes from a few labeled samples [25]. Existing FSL methods can be broadly grouped into three categories according to [21]: finetuning-based, meta-based, and metric-based methods. Finetuning-based strategies [47, 7, 10, 42, 32, 24] follow a transfer learning process that includes pre-training on base classes and fine-tuning on novel classes. Meta-based approaches [13, 33, 38, 35, 39, 3, 31] adopts a meta-learning paradigm to learn the cross-task knowledge through the optimization between the meta-learner and base-learner. In this way, the model adopts a quick adaptation to the novel dataset. Metric-based techniques [2, 40, 43, 36, 50, 20] focus on learning transferable representations and making predictions based on the distance between feature representations. This strategy eliminates the need for test-time fine-tuning. These paradigms have significantly advanced the progress of few-shot learning, yet a unified method for generating UAPs across all paradigms remains undeveloped.

3 UAP Setting for Few-Shot Learning

In this section, we outline the threat model of attacking the few-shot tasks and present key definitions and notations to facilitate a clearer understanding of the concepts involved.

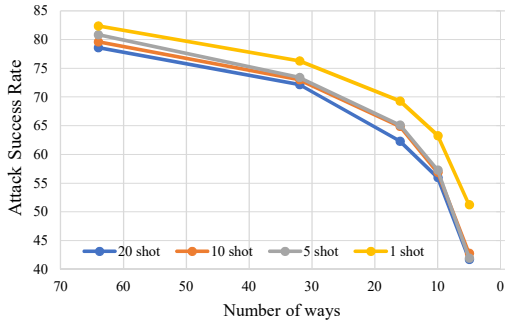
3.1 Threat Model

We consider an attacker who aims to craft a universal adversarial perturbation to attack a pre-trained victim model and further impair the performance of downstream few-shot tasks. The attacker only has access to the pre-trained model (*e.g.*, by downloading from public repositories), but can not obtain the datasets used for pre-training and have no knowledge of the following few-shot tasks. Once the perturbation is generated, the attacker attaches it to each query sample. The crafted perturbation is imperceptible, and is expected to greatly mislead the few-shot classification.

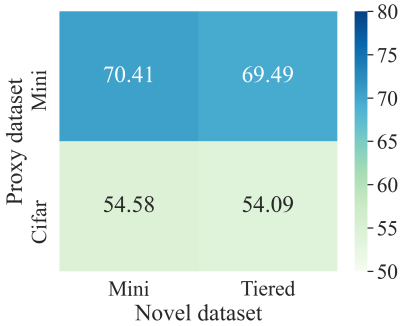
3.2 Definition and Notations

In the domain of few-shot learning, abundant annotated images of the base dataset $\mathcal{D}_b$ can be used for pre-training. Subsequently, the model would be fine-tuned using limited samples from the novel dataset $\mathcal{D}_n$, where the categories in $\mathcal{D}_n$ do not overlap with those in $\mathcal{D}_b$. In the pre-training stage, the victim model $f(\cdot)$ that composed of an encoder $f_e(\cdot)$ and a base classifier $f_{bc}(\cdot)$ is trained on $\mathcal{D}_b$. In the evaluation stage, different forms of tasks are sampled from $\mathcal{D}_n$. Each task contains a support set $\mathcal{S}$ used for finetuning and a query set $\mathcal{Q}$ used for evaluation. The support set includes n different classes with k samples per class, referred to as n-way k-shot. Similarly, the query set contains n classes with q samples per class. For each task, a novel class classifier $f_{nc}(\cdot)$ is newly trained on $\mathcal{S}$. For any image x in $\mathcal{Q}$, $f_{nc}(f_e(x), \mathcal{S})$ yields the classifier's output. The goal of FSL is to learn a

https://doi.org/10.52202/079017-3261



(a) An illustration of the ASRs under different downstream tasks.



(b) An illustration of the ASRs under different datasets.

Figure 3: The ASRs under different task shifts and semantic shifts. In (b), Cifar, Mini, Tiered are different datasets detailed in Section 6.1.

generalizable $f_e(\cdot)$ that exhibits good performance on $\mathcal{Q}$ across various tasks after learning from the limited examples in $\mathcal{S}$.

Once get the pre-trained encoder $f_e(\cdot)$, an attacker can craft the adversarial images as follows: $x^{adv} = x + \delta$, where δ represents the perturbation. To ensure that the adversarial images are visually indistinguishable from the original ones, δ is usually restricted by a chosen distance metric, such as the ℓ_∞ norm. This constraint can be expressed as $\|x^{adv} - x\|_\infty = \|\delta\|_\infty \leq \epsilon$. An effective approach to generate δ is to train a generator $g_\theta(\cdot)$ on the proxy dataset $\mathcal{D}_p$. The generator takes a random vector z as input and it aims to produce perturbations that satisfy the given constraints while maximizing adversarial impact.

4 A Closer Look at UAP Generation in Few-Shot Learning

In this session, we conduct a thorough analysis of the challenges associated with generating Universal Adversarial Perturbations (UAPs) in the context of Few-Shot Learning (FSL). We first compare the attack performance in the traditional scenario with that in the FSL scenario. Then we point out the lower performance in the FSL scenario is due to the presence of two shifts.

4.1 Poor Performance of Traditional UAPs in the FSL Scenario

To better understand the performance of existing attacking methods on FSL tasks, we directly apply the traditional method in the FSL scenario. We compare the Attack Success Rate (ASR) in the FSL scenario with that observed in the original scenario. The ASR refers to the success rate of the UAP in fooling the classifier. A higher ASR means a stronger attack capability.

We adopt the *Generative Adversarial Perturbation (GAP)*[30] method, due to its straightforward applicability to FSL tasks and its high attack performance within its original context. The GAP method generates UAP through a ResNet Generator[17], denoted as g_{GAP}. The perturbation δ is produced as follows:

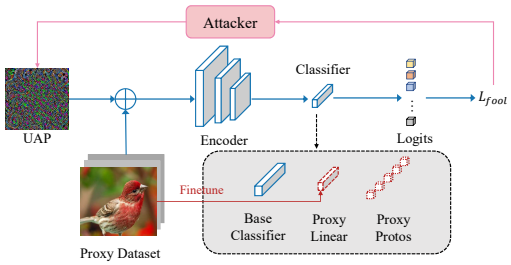
$$\delta = g_{GAP}(z), \tag{1}$$

where z represents a random vector. The generator is optimized with respect to the following loss term:

$$L_{GAP} = -\log(\mathcal{H}(f(x+\delta), y)). \tag{2}$$

Here, (x, y) is the data point sampled from proxy dataset, $f(\cdot)$ denotes the model pre-trained on the base dataset, and $\mathcal{H}(f(x^{adv}), y)$ represents the cross-entropy loss.

For the traditional GAP setting, we compute the mean ASR on VGG16, VGG19, and ResNet152 reported in the original paper to gauge the average performance in the traditional scenario. In the FSL scenario, where the attacker does not have access to the base or novel datasets, we utilize CIFAR-FS [4] as the proxy dataset to generate UAPs. We attack three pre-trained FSL models representative of different paradigms: finetuning-based, metric-based, and meta-based approaches.



(a) An illustration of the attacking framework.

Methods	TS	SS	ASR 1-shot	ASR 5-shot
Base classifier	✗	✗	65.07±0.36	61.73±0.30
Proxy linear	✓	✗	70.87±0.36	68.87±0.27
Base linear	✓	✓	76.75±0.32	74.67±0.21
Proxy protos	✓	✓	**80.04**±0.30	**77.69**±0.21

(b) An illustration of different ASRs.

Figure 4: (a) The Base Classifier is the pre-trained classifier. Proxy Linear and Proxy Protos are newly designed to address the task shift and the semantic shift, as detailed in Section 5. (b) The detailed Attack Success Rate(ASR) of different methods on 5-way 1-shot and 5-way 5-shot tasks. TS, SS represents whether the Task Shift and the Semantic Shift are handled respectively.

As illustrated in the left panel of figure 1, GAP achieves a pretty high mean ASR of 89.43% in traditional tasks. However, when switched to the FSL setting, there is a marked decrease in performance, with a reduction of at least 25% observed in both 5-way 1-shot and 5-way 5-shot tasks. Moreover, the GAP method is unable to generate UAPs for metric-based models, as its generation process depends on the presence of a fixed classifier, which is lacking for metric-based methods. This observation leads us to pose the question: **What makes the attack success rate of UAPs decrease so much in FSL?** In the next subsection, we will discuss the factors.

4.2 Two Shifts in FSL Affect the Attack Transferability

We compare the differences in the UAP generation process between the traditional scenario and the FSL scenario, as depicted in Figure 2. In traditional scenarios, UAP generation relies on a fixed classifier that remains unchanged throughout both the training and testing phases. In contrast, FSL scenarios present a variety of classification tasks during testing—encompassing different categories and shapes, such as 5-way 1-shot and 5-way 5-shot, thereby posing challenges to UAP generalization. Therefore, our first hypothesis is that **the task shift in FSL hinders the transferability of the UAP**.

To validate our hypothesis, we conducted experiments to evaluate the performance of the UAP across different downstream tasks. The experiments vary the number of ways from 5 to 64 and the number of shots from 1 to 20. To maintain the semantic consistency between the pre-training and testing datasets, we sampled an additional 100 images for each category in the training split of *mini*-ImageNet to serve as the downstream dataset. The results are presented in Figure 3 (a), demonstrating that as the downstream task becomes less similar to the pre-training task(64-way full shot), the attacking performance of the UAP diminishes.

From Figure 2, it can also be observed that the categories in training and testing datasets do not overlap in the FSL scenario. This semantic shift may contribute to a reduction in the attacking performance. Therefore, we derive a second hypothesis that **the semantic shift in FSL hinders the transferability of the UAP**. To verify the impact of the semantic shift, we keep the downstream task constant and vary the proxy dataset and downstream dataset. We choose a model pre-trained on the *mini*-ImageNet as the victim model. As illustrated in Figure 3 (b), the attack performance declines with increasing distances between both the downstream dataset and the pre-trained dataset(mini), as well as between the proxy dataset and the pre-trained dataset(mini).

To summarize, we empirically demonstrate that the two shifts that existed in the FSL scenario degrade the attack performance of traditionally generated UAPs.

5 Method

In this section, we propose a baseline framework aimed at generating effective Universal Adversarial Perturbations (UAPs) within a Few-Shot Learning (FSL) scenario. Starting with an initial attacking framework, we progressively refine our approach to address the two shifts mentioned before. Finally, we have developed a universally applicable UAP generation strategy, capable of attacking models

https://doi.org/10.52202/079017-3261

trained under all kinds of FSL paradigms and showing strong generalizability to different downstream tasks.

5.1 A Baseline Attacking Framework

The attacking framework is illustrated in Figure 4 (a). The victim model, built on a ResNet12[16] backbone, is pre-trained using the training split of the *mini*-ImageNet[43] dataset, following the training paradigm established in [10]. For the attack, we train a generator to transform random vectors into a UAP, which is also used in [52]. Since the attacker lacks access to both the training and testing data from the *mini*-ImageNet dataset, we opt for the training split of the CIFAR-FS dataset[4] as a proxy. To optimize our generator g_θ, we employ a negative cross-entropy loss as the attacking objective, defined by the following equation:

$$L_{fool} = -\mathcal{H}(f_{bc}(f_e(x + g_\theta(z))), y). \tag{3}$$

As the original approach, we utilize a 64-way classifier $f_{bc}(\cdot)$ from the baseline model to infer, without any fine-tuning. The results are presented in the first row of Figure 4 (b), denoted as **base classifier**. The results are far from satisfactory compared to the performance in traditional tasks, highlighting the need for further refinement in the FSL scenario.

5.2 Fill Up the Task Shift through Task Alignment

As demonstrated in Section 4.2, a misalignment between upstream and downstream tasks can adversely affect the performance of UAPs. Given that the attacker cannot foresee the specific downstream tasks, we modify the tasks used in UAP generation to better align with those downstream tasks. Importantly, we only aim to approximate the shape of the downstream tasks rather than achieving exactness, which is sufficient to let the UAP acquire the necessary task bias. This is supported by the ablation study detailed in Section 6.3. We sample *proxy tasks* with the shape of 5-way 1-shot from the proxy dataset $\mathcal{D}_p$. For each task, we construct a linear classifier $f_{pc}(\cdot)$ based on its support set $\mathcal{S}_p$. We refer to this method as **proxy linear**. The objective function for training the classifier is given by:

$$L_{pc} = \mathcal{H}(f_{pc}(f_e(x)), y), \tag{4}$$

where (x, y) represents the data sampled from $\mathcal{S}_p$. Once the proxy linear classifier has been trained, the attacker generates perturbations guided by the perturbed query set $\mathcal{Q}_p^{adv}$ from the proxy task, as depicted in Figure 4 (a). The fooling loss is quite like equation 3 except that the classifier is newly constructed and the data (x, y) is sampled from $\mathcal{Q}_p$, rather than the entire proxy dataset. The fooling loss is calculated as follows to optimize the generator g_θ :

$$L_{fool} = -\mathcal{H}(f_{pc}(f_e(x + g_\theta(z))), y), \tag{5}$$

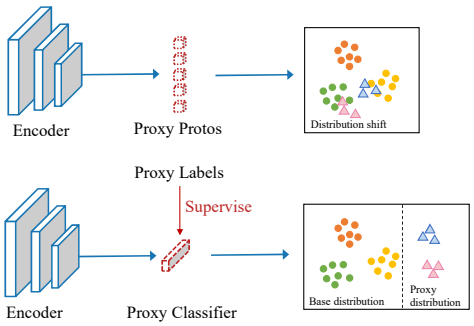
where (x, y) is sampled from $\mathcal{Q}_p$.

By integrating task alignment into the training of UAPs, we empower them to generalize more effectively to downstream tasks, even in the absence of direct access to those tasks. From the results presented in the second row of Figure 4 (b), it is evident that by addressing the task shift, there is a marked improvement in the ASR, with an increase of 5% for 1-shot tasks and 7% for 5-shot tasks.

5.3 Fill Up the Semantic Shift by Leveraging the Encoder's Generalizability

Analysis of the semantic shift. In Section 4.2, we discussed the impact of the semantic shift on UAPs, a phenomenon widely present in FSL. As attackers lack access to upstream or downstream datasets, our strategy emphasizes utilizing the generalization capabilities of the pre-trained encoder. A well-trained encoder possesses the capability to transfer useful information to the novel dataset, even trained on the base dataset. Consequently, we hypothesize that *if the proxy dataset closely aligns with the base dataset, the generated UAP can be more transferable to the novel dataset.*

We substitute the proxy dataset with the base dataset to create an ideal scenario for testing our assumption. While attackers cannot access the base dataset in real-world situations, utilizing it in our experiments allows us to better understand and validate our hypotheses. To be specific, we sample different tasks from the base dataset and train a linear classifier for each task, following the process outlined in equation 4. The results are documented in the third row of Figure 4 (b), denoted as **base**



(a) A qualitative illustration.

	Log	LS	ASR	
			1-shot	5-shot
eq 7	✗	✗	80.04±0.30	77.69±0.21
eq 8	✓	✗	80.72±0.30	78.35±0.21
eq 9	✓	✓	**81.56**±0.29	**79.00**±0.18

(b) An illustration of different ASRs.

Figure 5: (a) A qualitative illustration showing the effect of the supervisory signal. (b) The attack performance comparison of different fooling losses.

linear. Compared to the Proxy liner, a further improvement can be observed in the table. The results suggest that a closer simulation of the base dataset distribution enhances the transferability of the UAP to the novel dataset.

Alleviation of the semantic shift. While utilizing the base dataset proves effective, the attacker can only utilize the proxy dataset. As discussed in [46], when the proxy dataset is fed into the encoder, its output distribution tends to shift towards the base dataset, which is exactly what we need.

However, the previously mentioned proxy linear method introduces a supervisory signal that corrects the distribution, as shown in Figure 5 (a). The output distribution, initially altered by the encoder, is adjusted by the constructed proxy classifier.

This causes the UAP to lose its connection with the base dataset during training, thereby diminishing its ability to generalize to downstream tasks. Consequently, we abandon the proxy linear classifier and adopt a metric-based evaluation to mitigate the influence of labels from the proxy dataset. For each task sampled from the proxy dataset, we extract the output features from the support set and compute the mean of each class (i.e., class prototypes). The prototype for class k in support set $\mathcal{S}_p$ is calculated using the equation:

$$c_k = \frac{1}{N_k} \sum_{y=k} f_e(x). \tag{6}$$

(x, y) is sampled from $\mathcal{S}_p$ and N_k represents the number of samples for class k in the support set. We then employ a distance metric $D(\cdot, \cdot)$ to quantify the distance between the query feature and each class prototype. The classification probability for a query sample x in class k is then determined by:

$$p(y = k|x) = \frac{e^{D(f_e(x), c_k)}}{\sum_{k'} e^{D(f_e(x), c_{k'})}}, \tag{7}$$

where k' represents a class in the task. In accordance with prior work [36], we utilize cosine similarity as the distance metric.

The **proxy protos** strategy facilitates a closer approximation to the novel dataset, regardless of the specific proxy dataset used. The efficacy of this approach is documented in the final row of Figure 4 (b), which presents the most favorable results among all methods tested.

A further improvement. As suggested by [30], an alternative formulation for the fooling loss is as follows:

$$L_{fool} = -\log \mathcal{H}(f_{proto}(f_e(x + g(z))), y), \tag{8}$$

where f_{proto} denotes the predicted classification computed using the prototype-based approach previously described. Furthermore, to avoid reliance on the proxy query label and enhance transferability, we can supervise the perturbation generator with smoothed labels, denoted as $\hat{y}$. The fooling loss is redefined as:

$$L_{fool} = -\log \mathcal{H}(f_{proto}(f_e(x + g(z))), \hat{y}). \tag{9}$$

https://doi.org/10.52202/079017-3261

Table 1: 5-way 1-shot ASR results of our framework in different FSL victim models.

Victim	Proxy	Backbone	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	Cifar	RN12	81.56±0.29	58.94±0.43	77.84±0.28	70.34±0.29	69.03±0.36	73.31±0.32
		RN18	71.55±0.35	74.67±0.28	72.11±0.29	70.00±0.31	60.02±0.34	67.66±0.30
	Mini	RN12	81.63±0.28	68.55±0.46	78.08±0.28	77.22±0.28	68.54±0.36	74.91±0.33
		RN18	72.32±0.36	80.01±0.26	76.87±0.30	77.42±0.30	76.09±0.31	69.36±0.32
	Tiered	RN12	79.68±0.30	67.51±0.43	77.81±0.28	77.09±0.27	72.87±0.34	75.23±0.33
		RN18	72.11±0.36	79.45±0.27	76.36±0.30	77.11±0.30	74.80±0.32	73.31±0.31
Tiered	Cifar	RN12	76.03±0.32	62.56±0.29	68.26±0.28	76.49±0.27	76.73±0.32	78.47±0.34
		RN18	69.27±0.32	73.82±0.27	69.27±0.29	70.31±0.30	67.60±0.31	63.66±0.39
	Mini	RN12	78.91±0.30	75.67±0.29	75.02±0.31	77.68±0.28	76.71±0.32	78.80±0.34
		RN18	74.74±0.33	80.03±0.24	75.68±0.32	78.08±0.29	71.65±0.33	70.58±0.43
	Tiered	RN12	78.84±0.30	75.99±0.28	75.48±0.31	77.64±0.28	76.83±0.32	78.94±0.35
		RN18	73.51±0.32	80.44±0.25	75.33±0.33	78.25±0.28	71.45±0.34	69.15±0.43

Table 2: 5-way 5-shot ASR results of our framework in different FSL victim models.

Victim	Proxy	Backbone	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	Cifar	RN12	79.00±0.18	63.41±0.27	78.31±0.21	70.42±0.22	67.96±0.28	74.88±0.22
		RN18	71.82±0.21	73.87±0.24	71.37±0.25	69.74±0.23	59.94±0.31	69.75±0.22
	Mini	RN12	79.05±0.19	75.01±0.22	78.40±0.21	79.14±0.18	70.09±0.27	76.90±0.20
		RN18	73.89±0.21	79.31±0.20	77.46±0.21	79.09±0.17	76.34±0.19	68.70±0.24
	Tiered	RN12	76.78±0.21	74.03±0.23	78.43±0.20	79.02±0.18	74.26±0.22	76.91±0.21
		RN18	73.14±0.23	78.99±0.20	77.42±0.21	78.92±0.17	74.75±0.21	75.93±0.21
Tiered	Cifar	RN12	75.09±0.21	59.40±0.30	68.96±0.23	76.94±0.19	78.33±0.17	75.91±0.21
		RN18	70.32±0.24	71.30±0.25	70.73±0.23	71.55±0.22	70.87±0.22	66.09±0.28
	Mini	RN12	78.43±0.17	78.07±0.17	76.05±0.22	78.72±0.16	78.34±0.17	76.42±0.22
		RN18	78.01±0.16	79.41±0.18	77.90±0.21	79.25±0.15	76.65±0.17	74.85±0.26
	Tiered	RN12	78.48±0.16	78.09±0.18	76.61±0.22	78.79±0.16	78.37±0.17	77.09±0.23
		RN18	76.60±0.18	79.83±0.18	77.88±0.21	79.14±0.16	75.43±0.19	73.06±0.27

The adoption of this revised fooling loss yields further performance enhancements, as evidenced in Figure 5 (b). Consequently, we have adopted equation 9 as the final formulation for our fooling loss.

5.4 Conclusion and Discussion

Through a thorough comparison of the traditional scenario and the FSL scenario, we point out two critical shifts that reduce the attack performance on FSL tasks. To address the task shift, we sample proxy tasks to mirror the shape of the downstream tasks. To handle the semantic shift, we employ class prototypes to train UAPs, circumventing the introduction of the proxy labels. Moreover, the application of smoothed labels yields additional enhancements in attack performance.

Our final method can be summarized in the following steps: *(1)* Sample 5-way 1-shot tasks from the proxy dataset. *(2)* Calculate prototypes for each class in the proxy tasks. *(3)* Compute the fooling loss based on the distance between the query sample and class prototypes. *(4)* Optimize the perturbation generator to produce a highly transferable UAP in FSL. By following these steps, our method systematically generates universal adversarial perturbations that are robust and generalizable across different tasks.

6 Experiments

6.1 Implementation Details

Datasets. We utilize three widely used datasets in Few-Shot Learning (FSL) as the proxy dataset: CIFAR-FS[4], *mini*-ImageNet[43], and Tiered-ImageNet[34]. Both CIFAR-FS and *mini*-ImageNet

Table 3: Comparison of different attack methods on ASR for 5-way 1-shot tasks.

Victim	Method	Mark	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	UAN	SPW-18	52.27±0.33	47.68±0.42	43.64±0.26	-	-	-
	GAP	CVPR-18	47.71±0.31	49.40±0.35	66.47±0.32	-	-	-
	AdvEncoder	ICCV-23	76.68±0.31	57.37±0.38	68.51±0.31	59.10±0.29	66.63±0.34	72.85±0.31
	FSAFW	Ours	**81.56±0.29**	**58.94±0.43**	**77.84±0.28**	**70.34±0.29**	**69.03±0.36**	**73.31±0.32**
Tiered	UAN	SPW-18	40.34±0.31	33.00±0.27	51.91±0.28	-	-	-
	GAP	CVPR-18	49.72±0.33	58.23±0.28	61.19±0.30	-	-	-
	AdvEncoder	ICCV-23	75.99±0.32	62.16±0.29	53.82±0.30	71.01±0.31	60.23±0.33	68.86±0.32
	FSAFW	Ours	**76.03±0.32**	**62.56±0.29**	**68.26±0.28**	**76.49±0.27**	**76.73±0.32**	**78.47±0.34**

Table 4: Comparison of different attack methods on ASR for 5-way 5-shot tasks.

Victim	Method	Mark	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	UAN	SPW-18	46.09±0.30	45.42±0.30	42.23±0.26	-	-	-
	GAP	CVPR-18	40.94±0.28	45.71±0.29	64.23±0.29	-	-	-
	AdvEncoder	ICCV-23	74.19±0.23	55.12±0.28	67.34±0.26	59.47±0.25	67.76±0.26	74.72±0.21
	FSAFW	Ours	**79.00±0.18**	**63.41±0.27**	**78.31±0.21**	**70.42±0.22**	**67.96±0.28**	**74.88±0.22**
Tiered	UAN	SPW-18	32.97±0.29	23.45±0.23	51.75±0.27	-	-	-
	GAP	CVPR-18	44.90±0.30	52.40±0.28	59.52±0.28	-	-	-
	AdvEncoder	ICCV-23	75.03±0.23	58.23±0.28	53.25±0.28	70.93±0.25	60.40±0.30	69.55±0.22
	FSAFW	Ours	**75.09±0.21**	**59.40±0.30**	**68.96±0.23**	**76.94±0.19**	**78.33±0.17**	**75.91±0.21**

datasets comprise images from 100 categories, divided into 64 classes for training, 16 for validation, and 20 for testing. Tiered-ImageNet, a more extensive subset of ImageNet [9], consists of 608 classes (779,165 images) organized into 34 high-level categories, which are further split into 351 training classes, 97 validation classes, and 160 testing classes. To prevent the attacker from gaining knowledge of the downstream images, only the training portions of these datasets are utilized for the generation of UAPs.

Training Details. All the victim models are downloaded from the Libfewshot[21], a comprehensive Library for FSL. In line with the library's classification of FSL training paradigms, we selected Baseline and Baseline++ [10] as representatives of the finetuning-based paradigm, ANIL [31] and R2D2 [3] for the meta-based paradigm, and ProtoNet [36] along with DN4 [20] for the metric-based paradigm. For each paradigm, we adopted ResNet12 and ResNet18[16] as the backbone of the victim models, following the previous works[8, 48]. All these victim models are pre-trained on the *mini*-ImageNet and Tiered-ImageNet.

We adopt the Attack Success Rate (ASR) to evaluate the attack performance of UAPs on 2000 FSL tasks. The generator network, following the approach described in [52], was optimized using the Adam optimizer with an initial learning rate of 0.0002. The perturbation was kept within an ℓ_∞-norm bound of $\epsilon = 10$, considering pixel values in the range of 0 to 255. Additional FSL testing configurations follow the details provided in Libfewshot.

6.2 Results

Effectiveness of our attacking framework. Our investigation into the effectiveness of our attacking framework (FSAFW) across different FSL models on different tasks is demonstrated in 1 and 2. For each FSL paradigm, we targeted four models pre-trained on *mini*-ImageNet and TieredImageNet, utilizing ResNet12 and ResNet18 as backbone architectures. We report the mean Attack Success Rate (ASR, %, top-1) alongside 95% confidence intervals. For the meta-based methods ANIL and R2D2, we chose the victim models trained by 5-way 1-shot episodes denoted as ANIL-1 and R2D2-1 in the table (ANIL-5 and R2D2-5 are present in Section A). Notably, when the victim's dataset is consistent with the proxy dataset, attack performance generally improves. Our proposed attacking framework can achieve comparable ASR even when the proxy dataset is different from the victim one.

 https://doi.org/10.52202/079017-3261

Table 5: Comparison of different contrastive losses on ASR.

Tasks	Ours	InfoNCE	SimCLR	MSE	Cosine	SupCon	feature scatter
5-way 1-shot	**81.56**$\pm$**0.29**	77.67$\pm$0.31	77.26$\pm$0.31	64.19$\pm$0.33	79.32$\pm$0.30	77.48$\pm$0.31	75.85$\pm$0.31
5-way 5-shot	**79.00**$\pm$**0.18**	75.40$\pm$0.23	74.69$\pm$0.23	60.38$\pm$0.29	76.87$\pm$0.22	74.94$\pm$0.23	72.90$\pm$0.23

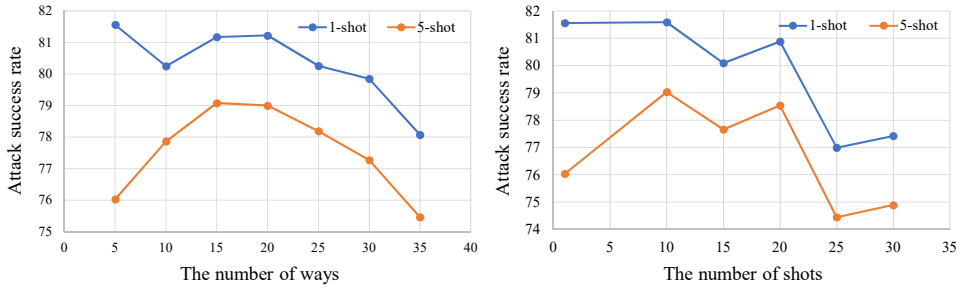


Figure 6: An illustration of the ASR that different forms of proxy tasks bring.

Comparison with other attacking methods. We compare the ASR of our attacking framework with the classic generator-based adversarial methods, such as UAN [15], GAP [30], and AdvEncoder [52], against the aforementioned FSL victim models. For this comparison, we utilized ResNet12 as the backbone and CIFAR-FS as the proxy dataset. ASR results for both 5-way 1-shot and 5-way 5-shot tasks are detailed in 3 and 4, respectively. Our framework consistently outperforms the other methods across victim models trained under various configurations. Note that some attacking methods rely on a fixed classifier in testing to generate UAPs, while models like R2D2, ProtoNet, and DN4 do not have one. These methods are not applicable to such models, and we have indicated this incompatibility with a dash ('-') in the relevant table cells.

6.3 Ablation Study

The attack performance of contrastive losses. To generalize the attack ability to the downstream tasks, a simple way is to adopt the contrastive losses[52]. In this section, we probe the effectiveness of various contrastive losses in crafting generalizable UAPs. We adopt the InfoNCE loss[29], SimCLR loss[6], MSE loss, Cosine similarity loss, SupCon loss[18] and feature scatter loss[51] to train the generator. We use a victim model trained on *mini*-ImageNet and with a backbone of ResNet12. The model is trained and tested under the Baseline[10] paradigm. We compare their ASR performance with that of our method in Table 5. The results suggest that contrastive losses in attacking FSL tasks are inferior to our FSAFW.

The influence of different forms of proxy tasks. We evaluate how different shapes of the proxy tasks affect the attack performance of the generated UAP. For the proxy tasks, we keep the number of shots constant at 1 while varying the number of ways, and vice versa, maintaining 5 ways while changing the number of shots. We utilize 5-way 5-shot and 5-way 1-shot as the shapes for downstream tasks. The attack results are displayed in the left and right panels of Figure 6. It can be observed that when the shapes of the proxy tasks do not deviate too much from the downstream tasks, the attack performance can be maintained. This suggests that when constructing the proxy tasks, a rough estimation of the downstream tasks' shape is enough.

7 Conclusion

In this work, we propose a unifying Few-Shot Attacking FrameWork (FSAFW) to generate transferable UAPs on the FSL scenario. We identify and analyze two major challenges in FSL: the task shift and the semantic shift. We adopt proxy tasks to handle the task shift and proxy prototypes to address the semantic shift. Our framework significantly outperforms existing methods, establishing a new benchmark for launching attacks on FSL tasks.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grants 62206102, 62376103, 62436003 and 62302184; the Science and Technology Support Program of Hubei Province under grant 2022BAA046; Hubei Science and Technology Talent Service Project under grant 2024DJC078; and Ant Group through CCF-Ant Research Fund.

References

[1] Yuanhao Ban and Yinpeng Dong. Pre-trained adversarial perturbations. *Advances in Neural Information Processing Systems*, 35:1196–1209, 2022.

[2] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14493–14502, 2020.

[3] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[4] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[15] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

 https://doi.org/10.52202/079017-3261

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7260–7268, 2019.

[21] Wenbin Li, Ziyi Wang, Xuesong Yang, Chuanqi Dong, Pinzhuo Tian, Tiexin Qin, Jing Huo, Yinghuan Shi, Lei Wang, Yang Gao, et al. Libfewshot: A comprehensive library for few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[22] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[24] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2218–2227, 2020.

[25] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[28] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*, 2017.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[30] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4422–4431, 2018.

[31] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[32] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

[33] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017.

[34] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[35] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[37] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[38] Qianru Sun, Yaoyao Liu, Zhaozheng Chen, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning through hard tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1443–1456, 2020.

[39] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 403–412, 2019.

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[42] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020.

[43] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[45] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[46] Jing Xu, Xu Luo, Xinglin Pan, Yanan Li, Wenjie Pei, and Zenglin Xu. Alleviating the sample selection bias in few-shot learning by removing projection to the centroid. *Advances in Neural Information Processing Systems*, 35:21073–21086, 2022.

[47] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*, 2021.

[48] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International conference on machine learning*, pages 7115–7123. PMLR, 2019.

[49] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14521–14530, 2020.

[50] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.

 https://doi.org/10.52202/079017-3261

[51] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Ziqi Zhou, Shengshan Hu, Ruizhi Zhao, Qian Wang, Leo Yu Zhang, Junhui Hou, and Hai Jin. Downstream-agnostic adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4345–4355, 2023.

Table 6: 5-way 1-shot ASR results of our framework in different FSL victim models.

Victim	Proxy	Backbone	ANIL-5	R2D2-5	ProtoNet-5	DN4-5
Mini	Cifar	RN12	67.80±0.45	74.77±0.34	61.10±0.39	66.70±0.37
		RN18	59.42±0.45	65.95±0.32	70.11±0.34	63.99±0.32
	Mini	RN12	69.48±0.45	74.76±0.34	68.80±0.41	68.77±0.35
		RN18	60.14±0.45	79.12±0.28	70.45±0.35	65.36±0.32
	Tiered	RN12	69.31±0.43	74.66±0.35	68.34±0.41	70.22±0.35
		RN18	60.06±0.47	78.68±0.28	65.11±0.32	65.32±0.32

Table 7: 5-way 5-shot ASR results of our framework in different FSL victim models.

Victim	Proxy	Backbone	ANIL-5	R2D2-5	ProtoNet-5	DN4-5
Mini	Cifar	RN12	72.75±0.28	79.39±0.17	63.65±0.31	69.05±0.26
		RN18	67.93±0.26	65.90±0.25	76.06±0.18	64.32±0.27
	Mini	RN12	74.96±0.25	79.40±0.17	75.84±0.20	74.30±0.23
		RN18	71.93±0.25	79.48±0.16	75.53±0.18	66.28±0.27
	Tiered	RN12	74.51±0.26	79.38±0.17	75.80±0.19	74.44±0.23
		RN18	69.07±0.26	79.33±0.16	68.52±0.24	66.18±0.27

A More Victim Models

Due to page limitations, we present the Attack Success Rate (ASR) of our method on more victim models trained under different paradigms in this section. Specifically, we selected the victim models trained by 5-way 5-shot episodes under the four paradigms: ANIL [31], R2D2 [3], ProtoNet [36], and DN4 [20]. These models are referred to as ANIL-5, R2D2-5, ProtoNet-5, and DN4-5 in the table. The victim dataset we chose is *mini*-ImageNet, while CIFAR-FS, *mini*-ImageNet, and tieredImageNet were used as proxy datasets. The corresponding results can be found in Table 6 and Table 7. As observed, our method consistently maintains a high Attack Success Rate (ASR) across all victim models, across different Few-Shot Learning (FSL) settings. This demonstrates the effectiveness of our method on models trained with 5-way 5-shot episodes.

B Compared with An Additional UAP Method

We have incorporated the PAP[1] method under the few-shot learning scenario and made a comparative analysis. The results in Table 8 show that our proposed method demonstrates effectiveness not only on the base dataset but also exhibits superior cross-dataset performance without necessitating access to the base dataset.

Table 8: The ASR results compared with the PAP method.

Proxy	Method	1-shot			5-shot		
		Baseline	ANIL-1	ProtoNet-1	Baseline	ANIL-1	ProtoNet-1
Cifar	PAP	44.39±0.30	44.52±0.29	37.70±0.29	36.58±0.26	42.34±0.28	35.69±0.26
	Ours	**81.56±0.29**	**77.84±0.28**	**69.03±0.36**	**79.00±0.18**	**78.31±0.21**	**67.96±0.28**
CUB	PAP	45.59±0.30	44.73±0.29	37.92±0.30	37.86±0.26	42.65±0.27	36.21±0.26
	Ours	**81.17±0.30**	**77.83±0.28**	**71.22±0.36**	**78.29±0.21**	**78.25±0.20**	**72.24±0.28**
Omniglot	PAP	47.91±0.31	45.88±0.28	38.52±0.29	40.69±0.27	43.92±0.27	36.63±0.26
	Ours	**76.32±0.31**	**77.81±0.28**	**67.70±0.35**	**74.46±0.21**	**77.87±0.21**	**69.05±0.26**
GTSRB	PAP	42.62±0.29	44.54±0.29	37.96±0.29	34.67±0.26	42.39±0.27	49.44±0.26
	Ours	**79.99±0.30**	**78.13±0.20**	**69.17±0.38**	**77.55±0.20**	**77.93±0.28**	**69.98±0.30**

https://doi.org/10.52202/079017-3261

Table 9: 5-way 1-shot ASR results of cross domain datasets.

Proxy	Downstream	Method	Baseline	ANIL-1	ProtoNet-1
CUB	Mini	UAN	52.47±0.31	62.85±0.27	-
		GAP	68.47±0.31	67.47±0.28	-
		AdvEncoder	76.57±0.31	72.86±0.27	70.80±0.36
		Ours	**81.17±0.30**	**77.83±0.28**	**71.22±0.36**
Omniglot	Mini	UAN	44.90±0.30	37.58±0.26	-
		GAP	60.95±0.33	63.82±0.35	-
		AdvEncoder	74.16±0.33	47.97±0.34	54.72±0.31
		Ours	**76.32±0.31**	**77.81±0.28**	**67.70±0.35**
GTSRB	Mini	UAN	54.45±0.32	50.17±0.26	-
		GAP	54.09±0.33	68.22±0.31	-
		AdvEncoder	67.24±0.32	62.86±0.30	67.16±0.34
		Ours	**79.99±0.30**	**78.13±0.20**	**69.17±0.38**
Mini	CUB	UAN	76.82±0.36	73.75±0.48	-
		GAP	69.70±0.35	75.08±0.45	-
		AdvEncoder	76.21±0.35	76.82±0.56	71.40±0.40
		Ours	**79.22±0.35**	**78.09±0.58**	**71.60±0.40**
Mini	Omniglot	UAN	36.20±0.40	56.63±0.59	-
		GAP	20.09±0.32	25.40±0.33	-
		AdvEncoder	64.41±0.41	66.27±0.65	71.80±0.38
		Ours	**76.44±0.27**	**69.64±0.64**	**75.27±0.36**
Mini	GTSRB	UAN	74.86±0.34	79.82±0.33	-
		GAP	73.30±0.34	79.04±0.36	-
		AdvEncoder	77.18±0.33	78.27±0.37	76.71±0.37
		Ours	**80.29±0.30**	**80.31±0.36**	**78.41±0.35**

C Cross Domain Experiments

To better validate the generalizability of our attack approach, we conducted experiments on various cross-domain datasets. We used CUB [45], [44], and GTSRB [37] as cross-domain datasets, employing them both as proxy datasets and downstream datasets to implement the attacks. We selected one victim model each from the fine-tuning, meta-based, and metric-based FSL paradigms for our experiments, comparing our approach with previous attack methods. The results are shown in Table 9 and Table 10, demonstrating that our attacking framework exhibits broader generalizability.

D Datasets

The CIFAR-FS [4] dataset is randomly sampled from CIFAR-100 [19] which contains 100 classes with 600 images of 32×32 size per class. The 100 classes are split into 64 classes for training, 16 classes for validation and 20 classes for testing. The average inter-class similarity is sufficiently high to represent a challenge for the current state of the art. Moreover, the limited original resolution of 32×32 makes the task harder and at the same time allows fast prototyping. The samples of this dataset are listed in Figure 7.

The *mini*-ImageNet dataset was proposed by [43] for few-shot learning evaluation. Its complexity is high due to the use of ImageNet images but requires fewer resources and infrastructure than running on the full ImageNet dataset. In total, there are 100 classes with 600 samples of 84×84 color images per class. These 100 classes are divided into 64, 16, and 20 classes respectively for sampling tasks for meta-training, meta-validation, and meta-test. The samples of this dataset are listed in Figure 8

The tieredImageNet dataset [34] is a larger subset of ILSVRC-12 with 608 classes (779,165 images) grouped into 34 higher-level nodes in the ImageNet human-curated hierarchy. This set of nodes is partitioned into 20, 6, and 8 disjoint sets of training, validation, and testing nodes, and the corresponding classes form the respective meta-sets. As argued in Ren et al., this split near the root

Table 10: 5-way 5-shot ASR results of cross domain datasets.

Proxy	Downstream	Method	Baseline	ANIL-1	ProtoNet-1
CUB	Mini	UAN	45.93±0.28	63.53±0.24	-
		GAP	65.91±0.24	67.58±0.25	-
		AdvEncoder	73.67±0.23	72.67±0.22	71.99±0.26
		Ours	**78.29**±0.21	**78.25**±0.20	**72.24**±0.28
Omniglot	Mini	UAN	37.34±0.27	35.05±0.26	-
		GAP	57.22±0.29	61.09±0.31	-
		AdvEncoder	72.07±0.23	44.73±0.31	57.38±0.26
		Ours	**74.46**±0.21	**77.87**±0.21	**69.05**±0.26
GTSRB	Mini	UAN	48.46±0.29	48.73±0.26	-
		GAP	48.68±0.29	65.94±0.28	-
		AdvEncoder	63.99±0.27	60.51±0.27	68.96±0.24
		Ours	**77.55**±0.20	**77.93**±0.28	**69.98**±0.30
Mini	CUB	UAN	77.42±0.20	64.42±0.58	-
		GAP	68.90±0.27	68.89±0.52	-
		AdvEncoder	75.29±0.22	67.31±0.68	72.80±0.27
		Ours	**77.46**±0.21	**69.43**±0.71	**72.86**±0.27
Mini	Omniglot	UAN	26.15±0.39	52.25±0.60	-
		GAP	10.14±0.23	21.69±0.32	-
		AdvEncoder	60.57±0.44	62.99±0.66	68.25±0.34
		Ours	**75.04**±0.25	**67.52**±0.67	**73.15**±0.30
Mini	GTSRB	UAN	74.27±0.23	78.31±0.28	-
		GAP	68.57±0.29	77.77±0.29	-
		AdvEncoder	73.96±0.26	77.50±0.29	78.38±0.23
		Ours	**77.22**±0.24	**78.50**±0.28	**78.77**±0.23

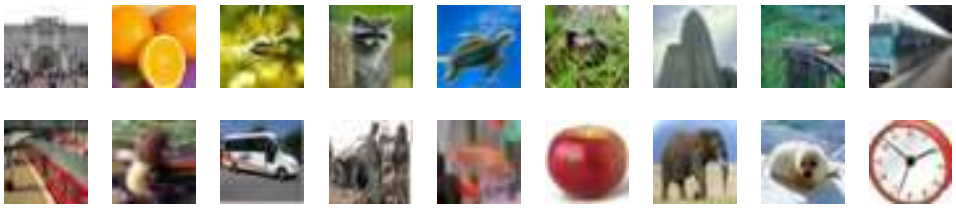


Figure 7: Samples of CIFAR-FS.

of the ImageNet hierarchy results in a more challenging, yet realistic regime with test classes that are less similar to training classes. The samples of this dataset are listed in Figure 9

E Broader Impact

We point out the security problem in a widely used open-set setting, i.e. Few-Shot Learning (FSL). Under this setting, we propose a unified attack framework to generate the UAP, which can fool all kinds of FSL models. The attack method can be used in many vision-based real-world scenarios, as

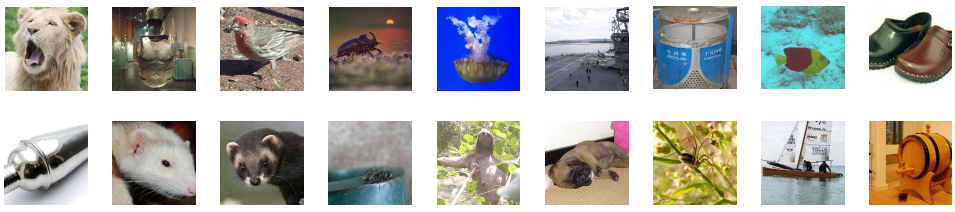


Figure 8: Samples of *mini*-ImageNet.

https://doi.org/10.52202/079017-3261

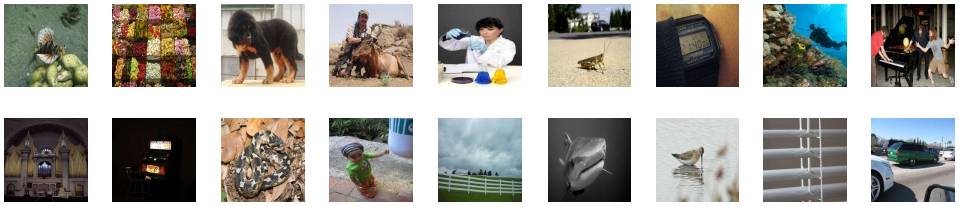


Figure 9: Samples of tieredImageNet.

it doesn't need to acquire the pre-training and downstream data. Also, the attack method can draw attention to the development of a defense method in FSL.

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: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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: We discuss the limitations of the work in the appendix.

   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: [NA]

https://doi.org/10.52202/079017-3261

Justification: We do not have theoretical results or 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 have reported the detailed settings and hyper-parameters. We will release our 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 have reported the detailed settings and hyper-parameters. We will release our 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 have reported the detailed settings and hyper-parameters.

   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: [Yes]

   Justification: We have reported the confidence interval in our ablation studies.

   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).

https://doi.org/10.52202/079017-3261

- 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: [Yes]

   Justification: We have reported the computer resources in the implementation details.

   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 conform, in every respect, with the NeurIPS 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 have discussed the broader impact in the appendix.

    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: Our work does not have the risk for misuse.

    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 have respected the license of assets used in this paper.

    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.

 https://doi.org/10.52202/079017-3261

- 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: We do not have new assets in this paper.

    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: [NA]

    Justification: This paper does not include human subjects information.

    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: This paper does not include human subjects information.

    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.