Supplementary Material for
Meta Style Adversarial Training for Cross-Domain Few-Shot Learning

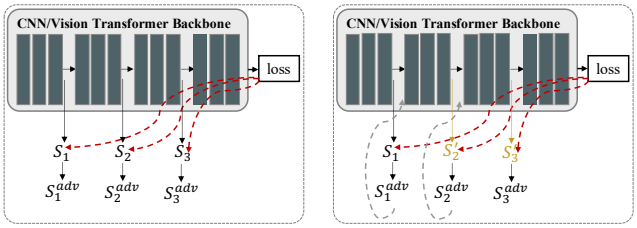
We first provide more implementation details in Sec. A; then we show more experimental results including plugging StyleAdv into different FSL/CD-FSL methods, building StyleAdv upon the PGD attacker, optimizing the model using different losses, and more ablation studies in Sec. B; Finally, in Sec. C, we provide more visualization results.

A. More Implementation Details

A.1. Progressive Attacking Method

To better help understand our proposed progressive attacking strategy, we compare it with the vanilla individual attacking approach. The illustrations are provided in Figure 4. For simplification, we use S_1, S_2, and S_3 to represent the styles extracted from blocks E_1, E_2, and E_3, respectively. Correspondingly, S_1^{adv}, S_2^{adv}, and S_3^{adv} represent the adversarial styles.

We would like to highlight two points: 1. The vanilla individual attacking method takes each block separately, which may lead to inconsistencies between features in different blocks. 2. By contrast, our progressive attacking method accumulates the adversarial signals, generating smooth adversarial features. Overall, we take the dependencies between blocks into account and produce a more coherent set of adversarial features via the progressive attacking way.



(a) Vanilla Individual Attacking Method (b) Ours Progressive Attacking Method

Figure 4. Illustrations of the vanilla/progressive attacking methods.

A.2. Loss Functions

Given the clean and perturbed episode features $F_{\mathcal{T}}$ and $F_{\mathcal{T}}^{adv}$, recall that StyleAdv contains four sub losses: the global classification loss $\mathcal{L}_{cls}$, the original FSL loss $\mathcal{L}_{fsl}$, the adversarial FSL loss $\mathcal{L}_{fsl}^{adv}$, and the consistency loss $\mathcal{L}_{cons}$.

Global Classification Loss: The $\mathcal{L}_{cls}$ is the cross entropy (CE) loss between the predictions of the global classification scores $f_{cls}(F_{\mathcal{T}})$ and the global class labels Y.

Original/Adversarial FSL Loss: Instead of using the global labels Y, meta-learning adopts local FSL class labels Y_{fsl} for query images by adjusting the global labels to the set of $[0, 1, 2, ..., N-1]$, where N denotes the N classes contained in the episode. Since we perturb the episode at style level while maintain the semantic content unchanged, the synthesized adversarial data still belong to the same FSL label Y_{fsl}. The FSL losses thus are calculated as $\mathcal{L}_{fsl} = CE(P_{fsl}, Y_{fsl})$, $\mathcal{L}_{fsl}^{adv} = CE(P_{fsl}^{adv}, Y_{fsl})$, where $P_{fsl} = f_{fsl}(F_{\mathcal{T}})$, $P_{fsl}^{adv} = f_{fsl}(F_{\mathcal{T}}^{adv})$.

Consistency Loss: The $\mathcal{L}_{cons}$ is introduced to constrain the consistency between the prediction P_{fsl} and P_{fsl}^{adv}. Specifically, it is calculated by the KL divergence loss which is defined below:

$$\mathcal{L}_{cons} = \frac{1}{BN}\sum_{i=1}^{B}\sum_{j=1}^{N} P_{fsl_{(i,j)}}(\log P_{fsl_{(i,j)}} - \log P_{fsl_{(i,j)}}^{adv}), \tag{11}$$

where $P_{fsl}, P_{fsl}^{adv} \in \mathcal{R}^{NM\times N}, B = NM$.

A.3. Competitors

In this paper, besides the existing CD-FSL methods, totally six competitors including "Attack Image", "Attack Feature" (as in Table 2) and "StyleGaus", "MixStyle", "AdvStyle", and "DSU" (as in Table 3) are adapted. Thus, we give an introduction to the implementation details of these proposed competitors.

Attack Image & Attack Feature: Generally, the "Attack Image" and "Attack Feature" share the same forward pipeline as our StyleAdv. To summarize, given a clean episode, all these three methods first perturb the original data via adversarial attack and then optimize the whole network under the supervision of both clean and adversarial perturbed episodes. The loss defined in Eq. 10 which contains four sub losses is utilized to optimize the network. Besides, the hyper-parameters are also kept consistent. While different from attacking styles as in StyleAdv, "Attack Image" attacks the episode at the image pixel level. That is, given clean episode $(\mathcal{T}, Y)$, the attacked $\mathcal{T}^{adv}$ is defined as,

$$\mathcal{T}^{adv} = \mathcal{T} + k_{RT}\cdot\mathcal{N}(0, I) + \epsilon\cdot sign(\nabla_{\mathcal{T}} J(\theta_E, \theta_{cls}, \mathcal{T}, Y)). \tag{12}$$

While "Attack Feature" generates the adversarial feature $F_{\mathcal{T}^{adv}}$ from the clean episode feature $F_{\mathcal{T}}$ as,

$$F_{\mathcal{T}}^{adv} = F_{\mathcal{T}} + k_{RT}\cdot\mathcal{N}(0, I) + \epsilon\cdot sign(\nabla_{F_{\mathcal{T}}} J(\theta_E, \theta_{cls}, F_{\mathcal{T}}, Y)). \tag{13}$$

To ensure a more fair comparison, the features of different blocks are attacked in the same progressive strategy as StyleAdv. Concretely, using F_1 denotes the feature extracted by the first block E_1 i.e. $F_1 = E_1(\mathcal{T})$. The F_1^{adv} can be easily obtained as in Eq. 13. However, for the subsequent block E_2, rather than obtaining F_2 as $E_2(F_1)$, we have $F_2^{'} = E_2(F_1^{adv})$. Attacking $F_2^{'}$ results in the F_2^{adv}. Similarly, we obtain the F_3^{adv}, thus get the final feature $F_{\mathcal{T}}^{adv}$ as the result of applying max pooling into the F_3^{adv}.

StyleGaus: The only difference between StyleGuas and StyleAdv lies in that rather than synthesizing new styles by adversarial attack as in Eq. 7 and Eq. 8, StyleGaus adds random Gaussian noises into the style (μ, σ) as below:

$$\mu^{adv} = \mu + k \cdot \mathcal{N}(0, I), \sigma^{adv} = \sigma + k \cdot \mathcal{N}(0, I), \quad (14)$$

where k is set as $\frac{16}{255}$. Note that all the other implement details e.g. network modules, pipeline, losses, and progressive augment manner are the same as StyleAdv.

MixStyle: The results of adapting MixStyle [66] for CD-FSL are introduced from wave-SAN [10]. Typically, the MixStyle competitor is constructed by randomly sampling two episodes from the source training set and using the mixed style of these two episodes as the new style.

AdvStyle: We implement the AdvStyle that attacks the style on images according to the pseudo codes provided in its paper. However, AdvStyle is initially proposed for segmentation, while we tackle the CD-FSL problem. Once we set the task as N-way K-shot, we could not take data of different sizes as input. Thus, rather than concating the original episode and the style-attacked episode as the input, we perform FSL tasks for these two episodes in parallel and use the sum of two FSL losses to optimize the network. For fair comparisons, the attacking ratio is set as $[0.008, 0.08, 0.8]$.

DSU: The DSU is adapted into CD-FSL by replacing our style attacking method as their method – modeling a Gaussian style distribution for each current batch of training data, and then randomly sample a new style from the Gaussian style. The core codes for modeling the style as uncertain Gaussian are provided by DSU.

A.4. Details for Finetuning

For each novel testing episode, as stated in Sec. 4, we generate pseudo training episodes and use them for finetuning the meta-trained model. Empirically, the finetuning stage is sensitive to the learning rates and tuning iterations. Thus, we provide the specific finetuning details as in Table 4. Overall, compared to the ViT-small with large pretrained parameters as initialization, the ResNet-10 (RN10) trained purely on the single source dataset requires a bigger learning rate; compared to the 5-shot models, finetuning 1-shot models needs fewer training iterations.

Backbone	LargeP	Task	Optimizer	Iter	LR
RN10	-	5-way 5-shot	Adam	50	{0, 0.001}
RN10	-	5-way 1-shot	Adam	10	{0, 0.005}
ViT-small	DINO/IN1K	5-way 5-shot	SGD	50	{0, 5e-5}
ViT-small	DINO/IN1K	5-way 1-shot	SGD	20	{0, 5e-5}

Table 4. The finetuning details for our ResNet10 (RN10) and ViT-small based models. The "LargeP" denotes the large-scale pretrained model. The "Iter" and the "LR" represent the tuning iterations and the learning rate, respectively.

B. More Experimental Results

B.1. Working in A Plug-and-Play Manner.

We highlight that our StyleAdv is complementary to other CD-FSL methods and can be used in a plug-and-play manner. To validate that, we show the results of plugin our StyleAdv into several different base models. The results are reported in Table. 5.

From the results, we draw the conclusion that our StyleAdv is model-agnostic and improves other FSL/CD-FSL methods effectively. Concretely, taking four different FSL/CD-FSL methods as base models, our StyleAdv promotes performance in most cases. Taking 5-way 1-shot as an example, we on average improve the RelationNet [44], the GNN [12], the FWT [48], and the PMF [19] by 4.81%, 4.51%, 3.75%, and 2.29%, respectively. Similar improvements can be observed in 5-shot results.

B.2. Working with Different Attack Algorithms?

As stated in Sec. 3.3, our style adversarial attack method is built upon the FGSM algorithm, thus we may wonder whether StyleAdv can still work with different attack algorithms. To that end, we further propose a variant style attack method (Style-PGD) by adapting the PGD algorithm. Formally,

$$\mu_0^{adv} = \mu + k_{RT} \cdot \mathcal{N}(0, I), \sigma_0^{adv} = \sigma + k_{RT} \cdot \mathcal{N}(0, I), \quad (15)$$

$$\mu_t^{adv} = \mu_{t-1}^{adv} + \epsilon \cdot \text{sign}(\nabla_\mu J(\theta_E, \theta_{f_{cls}}, \mathcal{A}(F_\mathcal{T}, \mu, \sigma), Y)), \quad (16)$$

$$\sigma_t^{adv} = \sigma_{t-1}^{adv} + \epsilon \cdot \text{sign}(\nabla_\sigma J(\theta_E, \theta_{f_{cls}}, \mathcal{A}(F_\mathcal{T}, \mu, \sigma), Y)). \quad (17)$$

The comparison results of the base GNN model, Style-PGD, and Style-FGSM are given in Table 6. Results show that both Style-PGD and Style-FGSM have a performance improvement against the base GNN. This basically shows that our StyleAdv is not sensitive to different attack algorithms. Besides, we also observe that Style-PGD is worse than Style-FGSM. This shows that the one-step attack is enough and more suitable to generate desired adversarial noises. Multi-step attacking may cause the generated styles too difficult to train the model. Besides, this significantly increases the burden of training. Thus, in this paper, we stick to the one-step Style-FGSM as our attack method.

B.3. Effectiveness of Each Loss Item.

To show the effectiveness of each item, we conduct ablation studies on different losses. Concretely, we compare our StyleAdv which is optimized by four sub losses with that of "w/o $\mathcal{L}_{cls}$", "w/o $\mathcal{L}_{cons}$", "w/o $\mathcal{L}_{fsl}, \mathcal{L}_{cons}$", and "w/o $\mathcal{L}_{fsl}^{adv}, \mathcal{L}_{cons}$". The 5-way 1-shot results are given in Table 7.

We first notice that all these variants perform worse than our method. This generally shows that each loss helps. More

1-shot	Method	ChestX	ISIC	EuroSAT	CropDisease	Cub	Cars	Places	Plantae	Average
RelationNet [44]	-	21.95±0.20	30.53±0.30	49.08±0.40	53.58±0.40	41.27±0.40	30.09±0.30	48.16±0.50	31.23±0.30	38.24
	+ **StyleAdv**	22.39±0.30	32.19±0.46	58.55±0.66	62.37±0.68	45.94±0.59	31.91±0.48	53.06±0.67	38.02±0.54	43.05 (**4.81**↑)
GNN [12]	-	22.00±0.46	32.02±0.66	63.69±1.03	64.48±1.08	45.69±0.68	31.79±0.51	53.10±0.80	35.60±0.56	43.55
	+ **StyleAdv**	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06 (**4.51**↑)
FWT [48]	-	22.04±0.44	31.58±0.67	62.36±1.05	66.36±1.04	47.47±0.75	31.61±0.53	55.77±0.79	35.95±0.58	44.14
	+ **StyleAdv**	22.91±0.37	35.05±0.56	68.03±0.81	73.84±0.78	48.68±0.72	34.88±0.58	59.15±0.84	40.60±0.66	47.89 (**3.75**↑)
PMF* [19]	-	21.73±0.30	30.36±0.36	70.74±0.63	80.79±0.62	78.13±0.66	37.24±0.57	71.11±0.71	53.60±0.66	55.46
	+ **StyleAdv**	22.92±0.32	33.05±0.44	72.15±0.65	81.22±0.61	84.01±0.58	40.48±0.57	72.64±0.67	55.52±0.66	57.75 (**2.29**↑)
5-shot	**Method**	**ChestX**	**ISIC**	**EuroSAT**	**CropDisease**	**Cub**	**Cars**	**Places**	**Plantae**	**Average**
RelationNet [44]	-	24.07±0.20	38.60±0.30	65.56±0.40	72.86±0.40	56.77±0.40	40.46±0.40	64.25±0.40	42.71±0.30	50.66
	+ **StyleAdv**	25.38±0.31	42.99±0.44	72.42±0.56	80.70±0.51	63.94±0.56	43.71±0.57	69.55±0.56	52.05±0.54	56.34 (**5.68**↑)
GNN [12]	-	25.27±0.46	43.94±0.67	83.64±0.77	87.96±0.67	62.25±0.65	44.28±0.63	70.84±0.65	52.53±0.59	58.84
	+ **StyleAdv**	26.07±0.37	45.77±0.51	86.58±0.54	93.65±0.39	68.72±0.67	50.13±0.68	77.73±0.62	61.52±0.68	63.77 (**4.93**↑)
FWT [48]	-	25.18±0.45	43.17±0.70	83.01±0.79	87.11±0.67	66.98±0.68	44.90±0.64	73.94±0.67	53.85±0.62	59.77
	+ **StyleAdv**	25.53±0.36	47.36±0.53	85.74±0.55	92.32±0.45	70.25±0.68	49.97±0.66	78.78±0.60	60.23±0.65	63.77 (**4.00**↑)
PMF [19]	-	27.27	50.12	85.98	92.96	-	-	-	-	-
	+ **StyleAdv**	26.97±0.33	47.73±0.44	88.57±0.34	94.85±0.31	95.82±0.27	61.73±0.62	88.33±0.40	75.55±0.54	72.44

Table 5. **Results of our StyleAdv working in a plug-and-play way.** Methods trained on mini-Imagenet and evaluated in eight various novel target datasets, respectively. "-" represents the base model, "+StyleAdv" means that our StyleAdv is applied to the base model. Results marked in blue perform best (best viewed in color).

	Attack Algorithm	ChestX	ISIC	EuroSAT	CropDisease	Cub	Cars	Places	Plantae	Average
1-shot	GNN [12]	22.00±0.46	32.02±0.66	63.69±1.03	64.48±1.08	45.69±0.68	31.79±0.51	53.10±0.80	35.60±0.56	43.55
	StyleAdv (Style-PGD)	22.74±0.35	32.79±0.53	68.08±0.82	73.02±0.81	47.86±0.70	34.27±0.56	57.13±0.83	39.90±0.63	46.97
	StyleAdv (Style-FGSM)	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06
5-shot	GNN [12]	25.27±0.46	43.94±0.67	83.64±0.77	87.96±0.67	62.25±0.65	44.28±0.63	70.84±0.65	52.53±0.59	58.84
	StyleAdv (Style-PGD)	25.98±0.38	44.49±0.50	84.39±0.57	92.30±0.43	68.50±0.67	48.82±0.64	77.76±0.62	59.62±0.66	62.73
	StyleAdv (Style-FGSM)	26.07±0.37	45.77±0.51	86.58±0.54	93.65±0.39	68.72±0.67	50.13±0.68	77.73±0.62	61.52±0.68	63.77

Table 6. **Results of StyleAdv working with different style attack algorithms.** Models are built upon the ResNet-10 and GNN.

specifically, comparing "all losses" with "w/o $\mathcal{L}_{cls}$", we observe that a obvious performance improvement is brought by $\mathcal{L}_{cls}$. It is not difficult to understand since the $\mathcal{L}_{cls}$ makes the global classifier optimized thus providing the correct gradients for the Style-FGSM. Also, by comparing the results of ours against that of "w/o $\mathcal{L}_{cons}$", we show that the consistency loss also contributes. It helps alleviate the semantic drift problem caused by perturbing the styles thus promoting the final model. In addition, through the results of removing the $\mathcal{L}_{fsl}^{adv}$ and $\mathcal{L}_{cons}$, the effectiveness of the adversarial styles generated by us is well indicated. The model performance is boosted by introducing such relatively challenging styles. Finally, we find that the original styles also help through the experimental results of "w/o $\mathcal{L}_{fsl}$, $\mathcal{L}_{cons}$".

B.4. More Ablation Studies of StyleAdv.

Our StyleAdv perturbs the initial style using the attacking ratio randomly sampled from ϵ_{list} with a random skip probability p_{skip}. In addition, the operation of random start is applied before attacking. Thus, we perform ablation studies on attacking with/without random start (RT), p_{skip}, and ϵ_{list}. Specifically, for the random skip probability p_{skip}, we set it as 0, 0.2, 0.4 (ours), and 0.6, respectively. For the attacking ratio ϵ_{list}, four regular choices including $[0.2, 0.02, 0.002]$, $[0.4, 0.04, 0.004]$, $[0.8, 0.08, 0.008]$ (ours), and $[1.6, 0.16, 0.016]$ and two relative large options including $\epsilon_{list} = [4]$ and $\epsilon_{list} = [20]$ are conducted. The 5-way 1-shot results are given in Table 8.

1) With/without random start. We first notice that our choice of applying RT performs better than without RT in most cases with an average improvement of 0.42%.

2) Different choices of p_{skip}. For different choices of p_{skip}, we find that except for the Cars and Places, as p_{skip} increases, the accuracy will first rise and then fail, or keep rising in some cases. This generally indicates that an appropriate p_{skip} can trade off the introduced perturbations and the difficulty of the meta task.

3) Different attacking ratios. The phenomenons presented by different ϵ_{list} factually are basically similar to those of p_{skip}. The higher the value of ϵ_{list}, the more diffi-

	Losses	ChestX	ISIC	EuroSAT	CropDisease	Cub	Cars	Places	Plantae	Average
1-shot	w/o $\mathcal{L}_{cls}$	22.36±0.36	34.43±0.57	67.86±0.83	68.46±0.80	48.13±0.73	32.98±0.56	56.44±0.81	38.48±0.63	46.14
	w/o $\mathcal{L}_{cons}$	22.68±0.36	33.10±0.53	70.06±0.84	72.46±0.80	48.34±0.71	33.58±0.55	57.65±0.82	40.06±0.64	47.24
	w/o $\mathcal{L}_{fsl}^{adv}$, $\mathcal{L}_{cons}$	22.05±0.35	32.49±0.53	68.86±0.83	68.93±0.81	47.23±0.72	32.85±0.57	55.88±0.82	37.68±0.62	45.75
	w/o $\mathcal{L}_{fsl}$, $\mathcal{L}_{cons}$	22.34±0.33	34.29±0.56	67.09±0.82	73.23±0.79	46.64±0.69	35.10±0.59	55.61±0.79	40.44±0.66	46.84
	All losses (ours)	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06

Table 7. **Effectiveness of each loss item.** Results conducted under 5-way 1-shot setting. Models are built upon the ResNet-10 and GNN.

1-shot	Choice	ChestX	ISIC	EuroSAT	CropDisease	Cub	Cars	Places	Plantae	Average
RT	✗	22.88±0.35	33.93±0.55	68.27±0.82	72.40±0.80	48.95±0.70	35.36±0.59	58.48±0.81	40.86±0.66	47.64
	✓ **(ours)**	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06
p_{skip}	$p_{skip} = 0$	22.59±0.36	33.06±0.52	67.26±0.81	72.73±0.79	48.11±0.70	35.92±0.59	58.65±0.82	40.43±0.65	47.34
	$p_{skip} = 0.2$	22.97±0.37	33.63±0.54	70.06±0.81	73.85±0.78	48.06±0.71	34.57±0.58	58.43±0.82	39.87±0.65	47.68
	$p_{skip} = 0.4$ **(ours)**	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06
	$p_{skip} = 0.6$	22.54±0.35	34.03±0.55	70.09±0.81	73.35±0.80	48.68±0.72	33.78±0.55	58.28±0.83	40.24±0.64	47.62
ϵ_{list}	$\epsilon_{list} = [20]$	20.83±0.28	23.97±0.34	50.68±0.79	43.12±0.73	29.41±0.50	23.34±0.35	32.79±0.55	25.98±0.41	31.27
	$\epsilon_{list} = [4]$	21.55±0.32	29.06±0.46	62.15±0.78	61.56±0.82	33.41±0.56	28.55±0.44	41.69±0.67	32.77±0.54	38.83
	$\epsilon_{list} = [1.6, 0.16, 0.016]$	22.71±0.36	33.37±0.54	70.98±0.82	73.33±0.79	48.76±0.72	35.34±0.60	58.25±0.81	41.00±0.65	47.97
	$\epsilon_{list} = [0.8, 0.08, 0.008]$ **(ours)**	22.64±0.35	33.96±0.57	70.94±0.82	74.13±0.78	48.49±0.72	34.64±0.57	58.58±0.83	41.13±0.67	48.06
	$\epsilon_{list} = [0.4, 0.04, 0.004]$	22.66±0.36	33.24±0.53	69.10±0.80	72.97±0.79	48.21±0.71	33.67±0.57	57.58±0.80	40.62±0.66	47.26
	$\epsilon_{list} = [0.2, 0.02, 0.002]$	22.47±0.36	32.30±0.52	68.22±0.79	72.06±0.78	47.41±0.71	33.60±0.58	57.57±0.83	40.03±0.64	46.71

Table 8. **Ablation studies on the random start (RT), skip probability p_{skip}, and attacking ratio ϵ_{list}.** 5-way 1-shot meta tasks are conducted. Models are built on ResNet10 and GNN.

cult the meta task is. For the two large choices $\epsilon_{list} = [4]$ and $\epsilon_{list} = [20]$, we find that when the attacking ratio becomes too large, the perturbations added will affect the original semantic label, thus leading to the drastic performance drop. The visualization results of stylized images with large attacking ratios shown in Figure 5 further validate that a suitable attacking ratio is key. In this paper, we set ϵ_{list} as $[0.8, 0.08, 0.008]$ as a trade-off. Note that our model is only trained with the mini-Imagenet without any single target image and we don't tune our model e.g. hyper-parameters for different target sets, thus it is unrealistic for our method to achieve totally consistent performance on eight unseen datasets. Alternatively, the choice with the relatively higher average performance is finally selected.

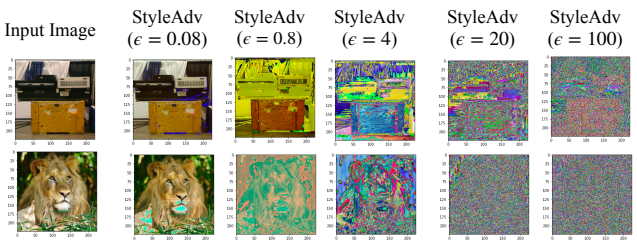


Figure 5. Visualization results of stylized images generated by different attacking ratios.

C. More Visualization Results.

In the main file, as in Figure 3, the stylized images generated by StyleAdv are given. Further, we provide the visualization results of feature maps extracted by both the ResNet-10 (RN10) and the ViT-small backbones. As shown in Figure 6, two examples are illustrated. For each example, both the clean feature maps and the attacked feature maps are shown. The attacking ratio is set as $[0.008]$. Whether for RN10-based features or ViT-small-based features, we visualize 36 channels. Note that the VIT-small feature maps are formed by reshaping the patch tokens as we do in Sec. 3.2.

Results show that: 1) The ViT-small feature maps also correctly reflect the original image information e.g., the shapes. This validates our idea of the patch tokens still remain the spatial information and the whole image feature can be formed by reshaping the patch tokens. This further supports us to apply StyleAdv to the ViT features. 2) Since the visualization is performed on the gray feature maps, the differences between the clean feature maps and the attacked feature maps are somewhat not so significant. However, as highlighted in red bounding boxes, we can still observe minor changes.

D. Discussion of Limitations

As indicated in Table 1, wave-SAN outperforms StyleAdv on the Cub dataset. This result suggests that when the visual appearances of the source and target datasets are similar, augmenting the source styles via attacking may result in overly challenging meta-tasks. Although we still improve all the base models, exploring better methods to address this issue could be one of our further work.

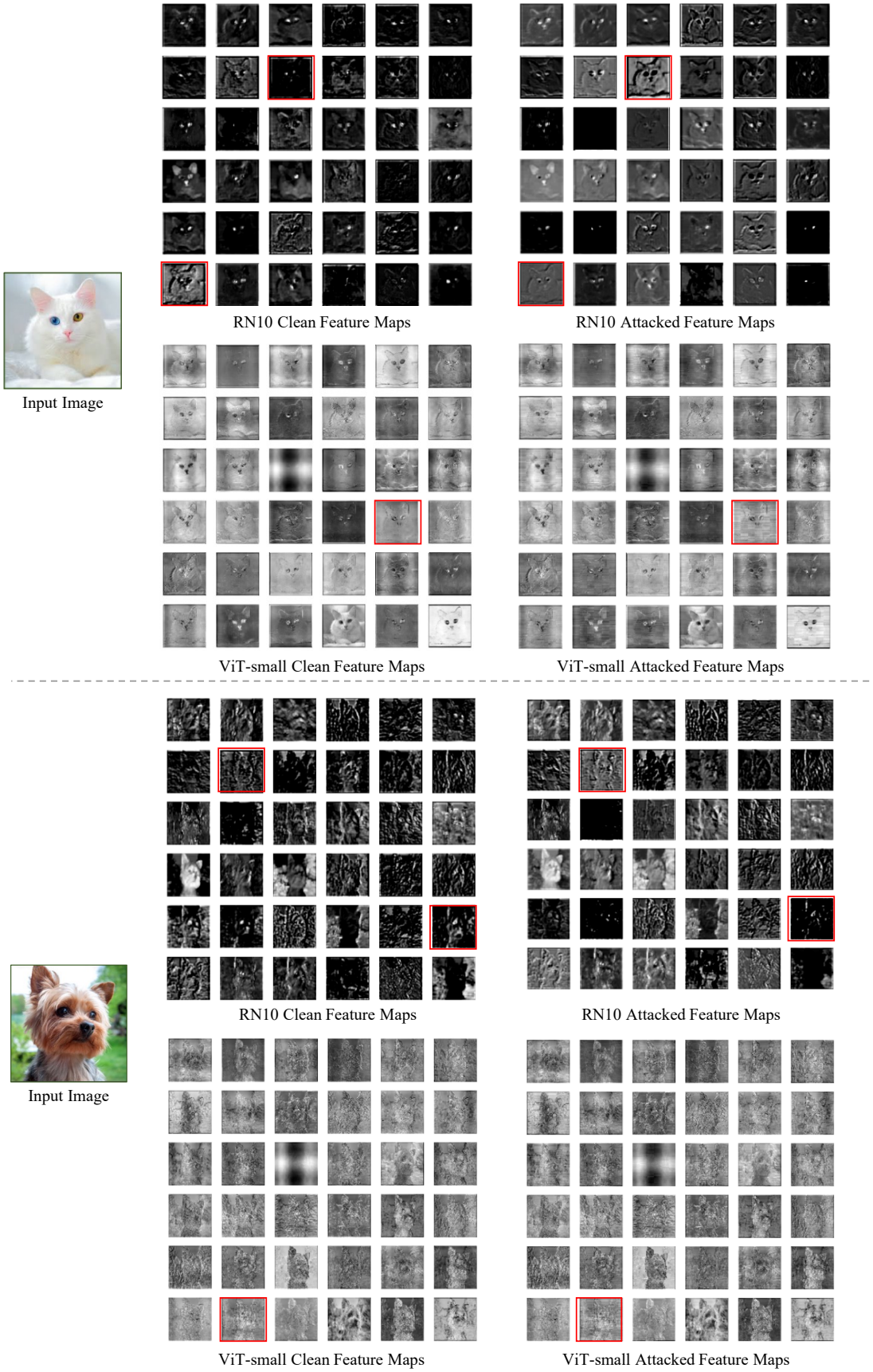


Figure 6. The clean/style-attacked feature maps of ResNet10 (RN10) and ViT-small are visualized. We visualize 36 channels.