Data-free Universal Adversarial Perturbation with Pseudo-semantic Prior
- *Supplementary Materials* -

Chanhui Lee Yeonghwan Song Jeany Son
AI Graduate School, GIST
{as584868, yeonghwan.song}@gm.gist.ac.kr jeany@gist.ac.kr

A. Additional Experiments on CNN Models

Impact of Each Proposed Component. In the main manuscript, we evaluate the impact of each component on attack performance by generating UAPs using ResNet152. In this section, we extend our ablation study to other models, with the results summarized in Figure 8. AlexNet is abbreviated as AN, ResNet152 as RN152, GoogleNet as GN, ResNet50 as RN50, DenseNet121 as DN121, MobileNet-v3-Large as MN-v3, and Inception-v3 as Inc-v3. We observe consistent trends across models, where the incorporation of pseudo-semantic priors (PSP), sample reweighting, and input transformation enhances the attack performance of the generated UAPs. However, the effect of PSP is less pronounced in AlexNet, while input transformations have a reduced impact on white-box attack performance. Additionally, in experiments with VGG19 and Inception-v3, several models demonstrate reduced performance when sample reweighting is applied alone. Despite these minor degradations, our full model, which combines all components, achieves a significantly higher black-box fooling rate on average, demonstrating its robustness even when individual components show limited effectiveness.

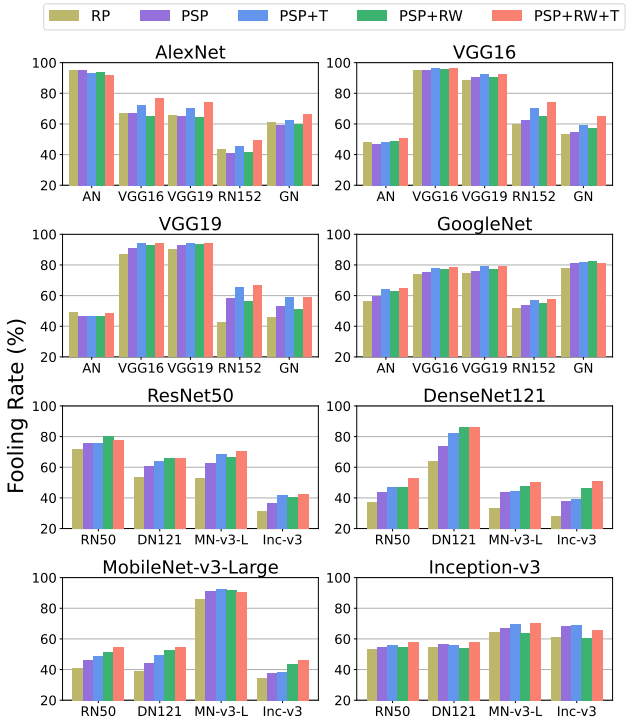

Figure 8. Ablation study on each proposed component in PSP-UAP on various CNN models. RP and PSP refer to training a UAP using random noises and semantic samples drawn from pseudo-semantic prior. RW and T denote the use of sample reweighting, and input transformation, respectively. All experiments, including, RP are conducted with the number of samples, N, set to 10.

Additional Experiments for Transferability. We conduct additional experiments to explore the black-box attack transferability across various models further. We generate UAPs on ResNet50, DenseNet121, MobileNet-v3-Large, and Inception-v3, and attack AlexNet, VGG16, VGG19, ResNet152, and GoogleNet. The results in Table 5 demonstrate that our method persistently surpasses TRM-UAP in attack performance, even when the target model changes, highlighting its superiority.

Impact of Epsilon. We evaluate the impact of ϵ which is a constraint parameter that restricts the pixel intensity of the generated UAPs used in Eq. (8) of the main manuscript. Note that, for experiments in the main manuscript, we set $\epsilon = 10$, following the conventional setting of data-free UAP methods. To further analyze its effect, we compare the FR of our method with TRM-UAP using various ϵ values of 8, 10, and 16. The results, shown in Table 6, show that our method consistently outperforms TRM-UAP in terms of FR across different values of ϵ. These experiments demonstrate that the pseudo-semantic prior retains sufficient value as the data prior, even under varying levels of constraints.

Robustness against Defenses. In Table 7, we validate the robustness of our method against JPEG compression [1]

Model	Attack	AN	VGG16	VGG19	RN152	GN	Average
RN50	TRM	46.46±0.80	73.82±0.85	72.43±0.91	52.64±1.18	58.59±1.57	60.79
	PSP	**51.80**±0.80	**82.02**±0.60	**82.09**±0.65	**60.90**±1.14	**62.22**±1.22	**67.81**
DN121	TRM	45.79±1.64	49.95±1.54	49.60±0.98	31.36±0.84	47.87±2.36	44.91
	PSP	**59.04**±0.76	**67.79**±0.85	**69.86**±0.99	**43.72**±0.26	**72.82**±2.04	**62.65**
MN-v3	TRM	45.47±0.49	49.13±0.71	48.69±0.64	28.67±0.58	36.15±0.92	41.62
	PSP	**66.50**±1.30	**77.52**±0.51	**75.96**±0.50	**49.56**±0.72	**69.78**±0.35	**67.86**
Inc-v3	TRM	**58.72**±0.56	71.77±0.25	70.82±0.12	45.84±0.47	62.87±0.41	62.01
	PSP	54.84±0.55	**78.38**±0.64	**75.52**±0.55	**52.82**±0.54	**65.24**±0.78	**65.36**

Table 5. Black-box attack transferability across models is analyzed. UAPs crafted on ResNet50, DenseNet121, MobileNet-v3-Large, and Inception-v3 are evaluated on AlexNet, VGG16, VGG19, ResNet152, and GoogleNet.

Model	δ_∞ constraint	Attack	RN50	DN121	MN-v3-L	Inc-v3	Average
RN50	$\epsilon = 8$	TRM-UAP	55.39*	39.80	39.02	22.87	39.27
		PSP-UAP	**66.41***	**50.90**	**54.06**	**28.98**	**50.09**
	$\epsilon = 10$	TRM-UAP	73.26*	54.42	61.25	37.36	56.57
		PSP-UAP	**77.60***	**66.11**	**70.50**	**42.32**	**64.13**
	$\epsilon = 16$	TRM-UAP	94.61*	80.74	75.21	58.16	77.18
		PSP-UAP	**94.88***	**90.53**	**90.35**	**74.21**	**87.49**
DN121	$\epsilon = 8$	TRM-UAP	29.82	59.12*	30.43	24.70	36.01
		PSP-UAP	**37.56**	**67.51***	**44.38**	**32.34**	**45.45**
	$\epsilon = 10$	TRM-UAP	35.24	70.10*	34.17	32.11	42.91
		PSP-UAP	**53.03**	**85.81***	**50.22**	**50.73**	**59.95**
	$\epsilon = 16$	TRM-UAP	64.64	88.80*	60.90	51.88	66.55
		PSP-UAP	**77.89**	**96.84***	**77.10**	**73.87**	**81.42**
MN-v3-L	$\epsilon = 8$	TRM-UAP	37.41	36.35	79.71*	30.79	46.06
		PSP-UAP	**43.47**	**44.41**	**79.94***	**35.39**	**50.80**
	$\epsilon = 10$	TRM-UAP	39.47	40.37	73.07*	30.11	45.76
		PSP-UAP	**54.38**	**54.62**	**90.39***	**46.29**	**61.42**
	$\epsilon = 16$	TRM-UAP	63.21	63.95	96.70*	47.49	67.83
		PSP-UAP	**81.40**	**83.45**	**99.03***	**76.83**	**85.18**
Inc-v3	$\epsilon = 8$	TRM-UAP	43.02	44.55	54.33	48.85*	47.68
		PSP-UAP	**46.53**	**45.43**	**57.12**	**52.58***	**50.41**
	$\epsilon = 10$	TRM-UAP	53.53	54.93	67.16	64.22*	59.96
		PSP-UAP	**57.60**	**57.50**	**70.20**	**65.38***	**62.67**
	$\epsilon = 16$	TRM-UAP	78.90	79.06	88.40	91.81*	84.54
		PSP-UAP	**83.58**	**82.21**	**89.24**	**93.56***	**87.14**

Table 6. FR (%) results for the UAPs constrained by $\epsilon = 8, 10$ and 16, crafted on ResNet50, DenseNet121, MobileNet-v3-Large, and Inception-v3. * denotes the white-box model.

Attack	JPEG compression					Ensemble methods	
	AN	VGG16	VGG19	RN152	GN	Inc-v3$_{ens3}$	IncRes-v2$_{ens}$
TRM	53.57	58.38	53.86	39.94*	45.44	17.0	11.5
Ours	**56.74**	**73.62**	**69.41**	**58.61***	**62.00**	**19.8**	**13.1**

Table 7. Robustness evaluation of our method and TRM against defense methods: JPEG compression and ensemble adversarially trained models.

and ensemble adversarially trained models, ens3-adv-Inc-v3 and ens-adv-Inc-Res-v2 [6]. Our method consistently shows higher robustness than TRM-UAP, with UAP crafted on ResNet152.

Diverse Surrogate Models We craft UAPs on various models in the main manuscript. To further demonstrate the effectiveness of our method on recent architectures, we additionally evaluate it using ConvNext-B [3] and DeiT3 [5] as surrogate models. As shown in Table 8, our method achieves strong performance on ConvNext-B but underperforms on DeiT3, which is consistent with the behavior observed in TRM. We attribute this to the fact that both our baseline and TRM were originally designed for CNN-based models, which may result in limited effectiveness on ViT-based architectures.

Attack	Model	CN-B	DeiT3	Others	Model	CN-B	DeiT3	Others
TRM	CN-B	40.71*	10.10	34.67	DeiT3	14.18	6.73*	36.01
Ours		**86.57***	**13.94**	**59.49**		**19.98**	**9.54***	**43.05**

Table 8. FR (%) comparison for ConvNext-B (CN) and DeiT3. *Others* denotes the average FR (%) on AlexNet, VGG16, VGG19, ResNet152, and GoogleNet.

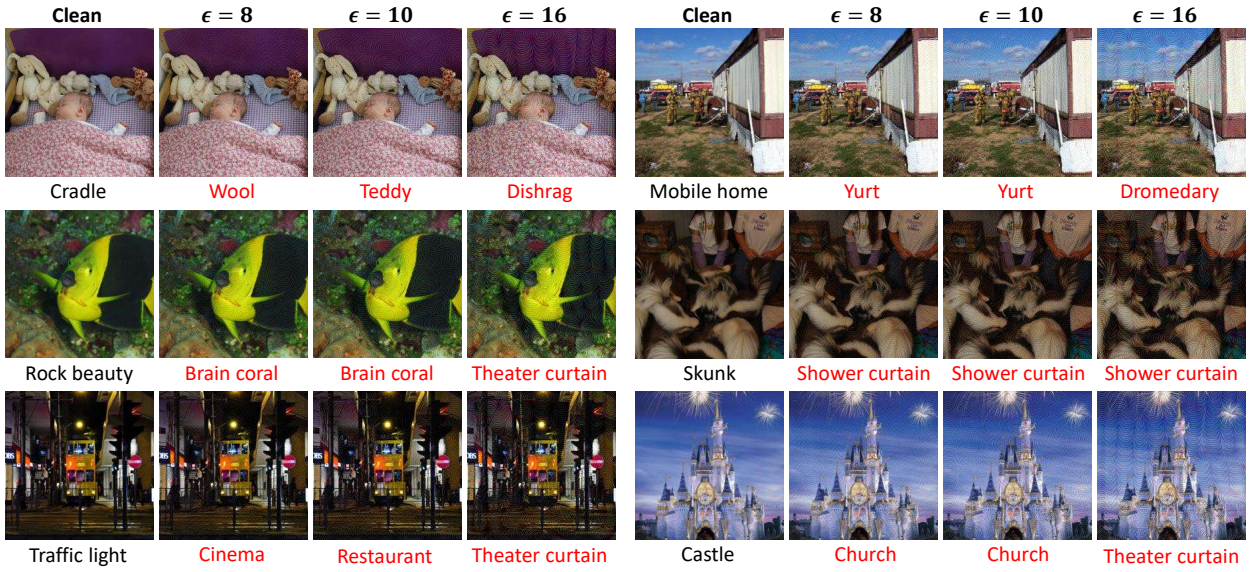

Figure 9. Qualitative results of our method. The leftmost column represents the original images, while the remaining three columns correspond to adversarial images generated with $\epsilon = 8$, 10, and 16 (from left to right). The predicted labels are displayed below each image. The UAPs are crafted on ResNet152.

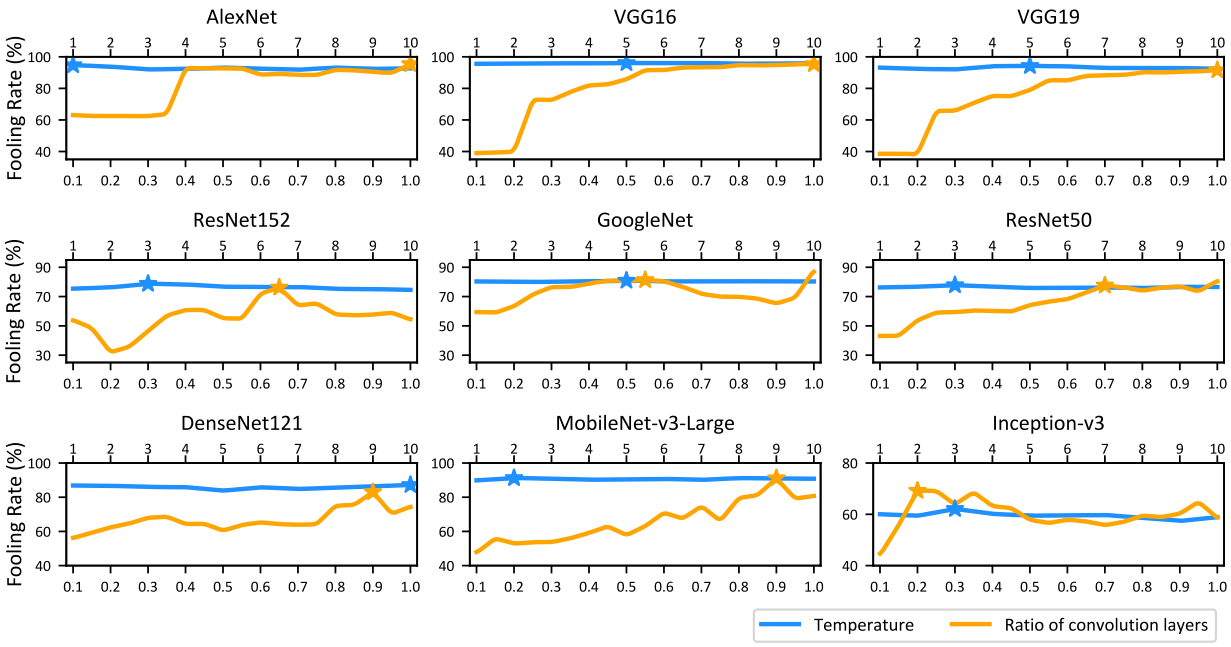

Figure 10. Parameter study on the ratio of convolutional layers and the temperature parameter for sample reweighting.

Qualititive Results We illustrate adversarial examples attacked by our generated UAPs using ResNet152 in Figure 9 with different $\epsilon \in \{8, 10, 16\}$. As expected, smaller ϵ values result in minimal degradation to the original image, whereas larger ϵ values highlight more artifacts introduced by the UAP. Similarly, as shown in Table 6, smaller ϵ values lead to lower performance compared to larger ϵ values. We also visualize the final UAPs crafted for each model and the intermediate UAPs used during the training phase to generate the pseudo-semantic prior in Figure 12. As discussed in the main manuscript, visually diverse patterns can be observed across different iterations, even on the same surro-

Model	Attack	AN	VGG16	VGG19	RN152	GN	Avg.
AN	TRM	**93.53***	60.10	57.08	27.31	32.70	54.14
	PSP-I	91.59*	74.95	72.70	47.66	65.54	70.49
	PSP-D	91.77*	**76.56**	**74.07**	**49.20**	**66.00**	**71.52**
VGG16	TRM	47.53	94.30*	89.68	61.43	53.95	69.38
	PSP-I	48.90	96.10*	91.86	70.75	58.45	73.21
	PSP-D	**50.40**	**96.26***	**92.60**	**74.10**	**64.89**	**75.65**
VGG19	TRM	46.01	89.82	91.35*	47.19	46.48	64.17
	PSP-I	46.57	94.07	93.88*	66.08	57.33	71.59
	PSP-D	**48.93**	**94.55**	**94.56***	**67.13**	**58.83**	**72.80**
RN152	TRM	53.56	77.20	73.30	67.46*	57.54	65.81
	PSP-I	57.17	87.40	86.34	84.85*	71.86	77.24
	PSP-D	**58.82**	**88.59**	**87.35**	**85.65***	**76.00**	**79.29**
GN	TRM	60.10	**79.66**	**79.98**	**58.85**	**85.32***	**72.78**
	PSP-I	**66.06**	78.88	79.61	56.95	81.04*	72.51
	PSP-D	65.22	78.43	79.26	57.63	81.43*	72.39

Table 9. Ablation study on the sample reweighting temperature parameter, τ. PSP-I and PSP-D refer to fixing the τ to 4 and adapting it for each model.

Model	Attack	RN50	DN121	MN-v3	Inc-v3	Avg.
RN50	TRM	73.26*	54.42	61.25	37.36	56.57
	PSP-I	76.41*	64.89	69.32	42.03	63.16
	PSP-D	**77.60***	**66.11**	**70.50**	**42.32**	**64.13**
DN121	TRM	35.24	70.10*	34.17	32.11	42.91
	PSP-I	**53.30**	84.95*	49.79	49.59	59.40
	PSP-D	53.03	**85.81***	**50.22**	**50.73**	**59.95**
MN-v3	TRM	39.47	40.37	73.07*	30.11	45.76
	PSP-I	**54.88**	53.56	89.85*	45.92	61.05
	PSP-D	54.38	**54.62**	**90.39***	**46.29**	**61.42**
Inc-v3	TRM	53.53	54.93	67.16	64.22*	59.96
	PSP-I	57.56	57.15	69.94	64.83*	62.37
	PSP-D	**57.60**	**57.50**	**70.20**	**65.38***	**62.67**

Table 10. Ablation study on the sample reweighting temperature parameter, τ, for additional CNN models. PSP-I and PSP-D refer to fixing the τ to 4 and adapting it for each model.

gate model. This demonstrates that our method effectively crafts UAPs even in the absence of prior knowledge by generating diverse semantic samples.

B. Ablation Study on Hyperparameters

In this section, we demonstrate an ablation study on the hyperparameters used in our PSP-UAP framework, including the ratios of convolutional layers to calculate the loss, temperature parameters in the sample reweighting, and the ranges for rotation, scaling, and shuffling in the input transformation. To determine the optimal set of parameters, we follow the setting used in previous works [2, 4].

Ratio of Convolutional Layers We follow the same process outlined in TRM-UAP [2] to determine l' in Eq. (8) by searching for the optimal ratio of convolutional layers. For this, we use only our pseudo-semantic priors, excluding sample reweighting and input transformation. Figure 10 shows the results, with yellow lines indicating outcomes and the yellow star marking the convolutional layer ratios used in our experiments. Based on this, the ratios are set to 100%, 100%, 100%, 65%, 55%, 70%, 90%, 90%, 20% for AlexNet, VGG16, VGG19, ResNet152, GoogleNet, ResNet50, DenseNet121, MobileNet-v3-Large, Inception-v3, respectively. Note that, for a fair comparison with TRM-UAP in Table 3 of our main manuscript and Table 5 in this supplementary material, we made every effort to conduct comprehensive experiments to determine the optimal positive truncation rate (PTR) and negative truncation rate (NTR) for TRM-UAP.

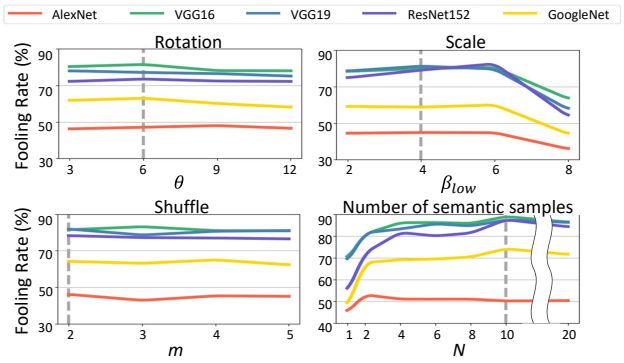

Figure 11. Hyperparameter analysis on ImageNet train set for input transformation and the number of semantic samples. The hyperparameters used in our experiments are marked with gray dashed line.

Temperature Parameters After determining the optimal convolution layer ratio, we use it as a basis to find the temperature parameter τ, used in Eq. (6) for the temperature-scaled softmax output, by incrementally increasing it from 1 to 10 in steps of 1. The results are shown in Figure 10, with blue lines representing the outcomes and the blue stars indicating the temperature values used in our experiments. Our observations indicate that variations in the temperature parameter τ have minimal impact on the results. In Table 9 and Table 10, we report the performances of our PSP-UAP with a fixed temperature ($\tau = 4$, referred to as PSP-I) alongside PSP-UAP with optimal temperature values (PSP-D) and TRM-UAP for comparison. Even with a fixed temperature, the performance difference is minimal, and our method consistently outperforms TRM-UAP by a significant margin. This highlights the robustness of our approach, achieving strong results over TRM-UAP even without tuning the temperature parameter.

Hyperparameter Search on ImageNet Train Set We conduct experiments on a randomly selected subset of 1,000 images from the ImageNet train set to determine hyperparameters for input transformation and the number of seman-

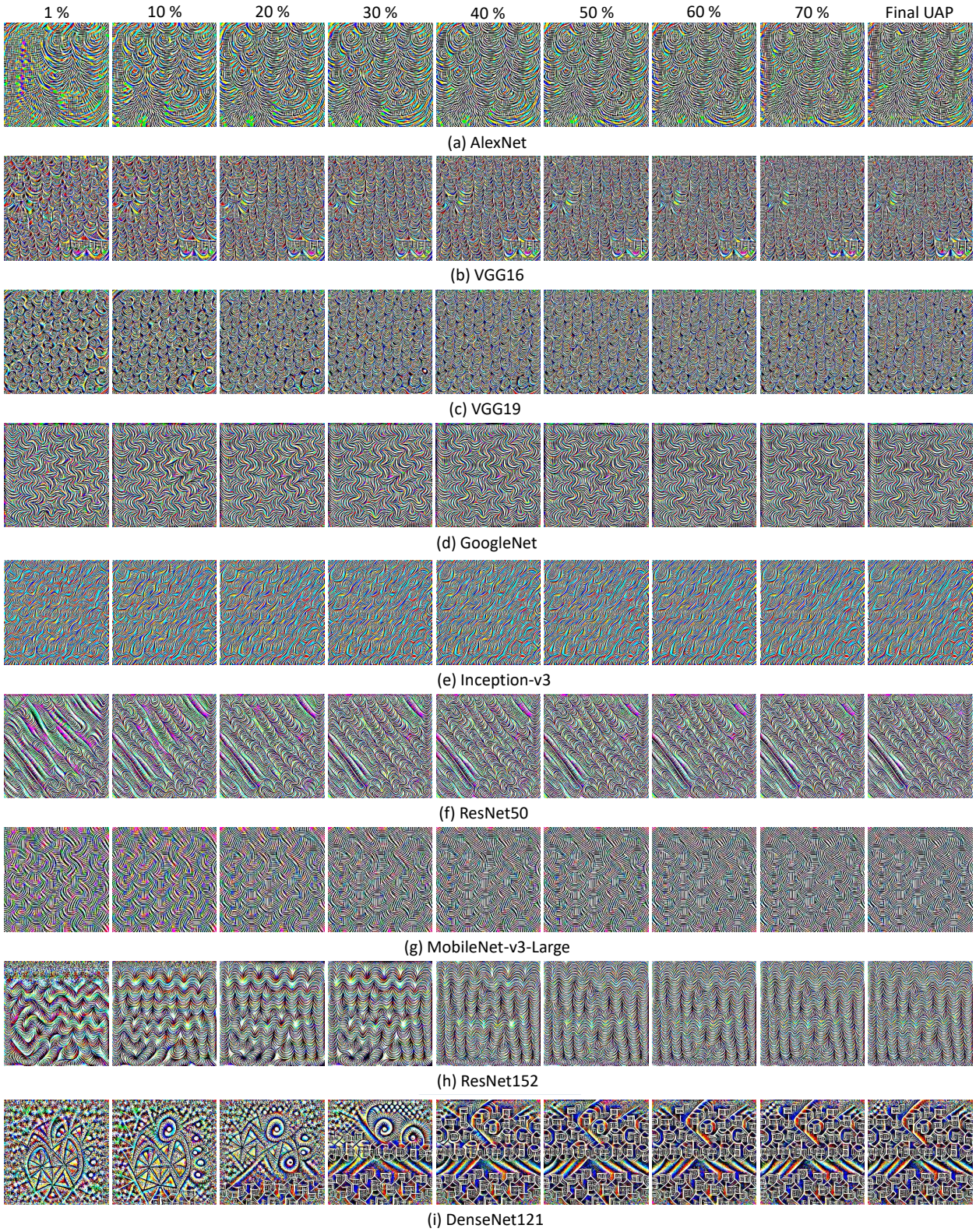

Figure 12. Visualization of the UAPs crafted by various CNN models in the training phase. The percentage above the figure corresponds to the progress of training iterations (*e.g.,* 1000 iterations out of 10000 = 10%)

tic samples. As shown in Figure 11, our method demonstrates consistent performance across various transformation hyperparameters and exhibits a similar tendency to the results in Figure 6 of the main manuscript. In the case of the number of semantic samples, the performance is relatively low when $N = 1$, which is likely due to sample imbalance caused by randomly sampling 1,000 images rather than using a dedicated validation set. Nevertheless, we conduct our main experiments using $N = 10$, selected based on training set results and achieve the highest fooling rate on the validation set compared to other methods.

C. Limitations and Discussions

Applying input transformations in our data-free UAP framework occasionally leads to a decrease in white-box attack performance. Unlike data-dependent approaches that rely on cross-entropy or logits, our method in Eq. (8) utilizes activations from all layers. While this comprehensive use of layer activations provides several advantages, it also increases sensitivity to unintended side effects of input transformations, as shallower features are generally more affected than deeper ones. Consequently, although input transformations boost black-box attack transferability, they may cause a slight decline in white-box performance.

In addition, since our method does not rely on target images or models, the adversarial examples generated may exhibit artifacts from the UAP itself, particularly when the images contain large plain regions, making them less visually clean compared to image-specific attacks. However, this is not a limitation unique to our approach but a common challenge for UAP methods, where a single UAP is used to attack a wide range of images.

Furthermore, while our method achieves strong performance on CNN architectures, it demonstrates limited attack transferability on ViT-based models. This limitation appears to be inherent to both data-free and data-dependent UAP approaches. As a direction for future work, we intend to explore black-box UAP strategies specifically tailored to ViT-based architectures.

References

[1] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

[2] Yiran Liu, Xin Feng, Yunlong Wang, Wu Yang, and Di Ming. Trm-uap: Enhancing the transferability of data-free universal adversarial perturbation via truncated ratio maximization. In *ICCV*, 2023.

[3] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[4] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *TPAMI*, 41(10), 2018.

[5] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*. Springer, 2022.

[6] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.