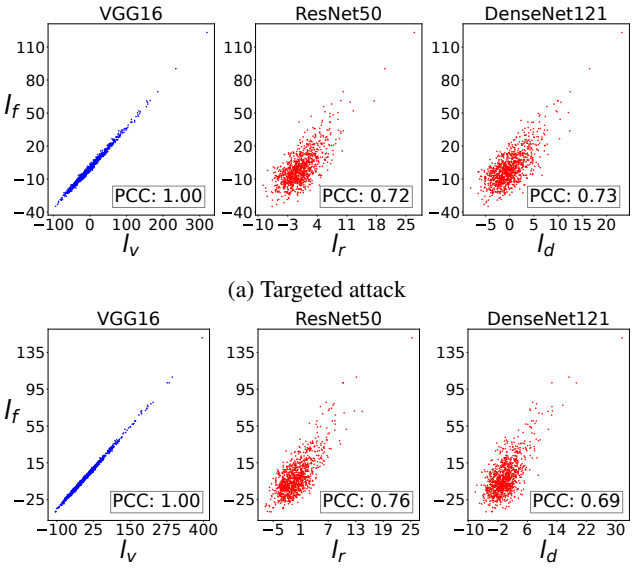
Exploring Non-target Knowledge for Improving Ensemble Universal Adversarial Attacks

Juanjuan Weng[1], Zhiming Luo[1,3*], Zhun Zhong[2], Dazhen Lin[1], Shaozi Li[1]

[1]Department of Artificial Intelligence, Xiamen University, China
[2]Department of Information Engineering and Computer Science, University of Trento, Italy
[3]Fujian Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University, China
wengjuan@stu.xmu.edu.cn, zhunzhong007@gmail.com, {zhiming.luo, dzlin, szlig}@xmu.edu.cn

Abstract

The ensemble attack with average weights can be leveraged for increasing the transferability of universal adversarial perturbation (UAP) by training with multiple Convolutional Neural Networks (CNNs). However, after analyzing the Pearson Correlation Coefficients (PCCs) between the ensemble logits and individual logits of the crafted UAP trained by the ensemble attack, we find that one CNN plays a dominant role during the optimization. Consequently, this average weighted strategy will weaken the contributions of other CNNs and thus limit the transferability for other black-box CNNs. To deal with this bias issue, the primary attempt is to leverage the Kullback–Leibler (KL) divergence loss to encourage the joint contribution from different CNNs, which is still insufficient. After decoupling the KL loss into a target-class part and a non-target-class part, the main issue lies in that the non-target knowledge will be significantly suppressed due to the increasing logit of the target class. In this study, we simply adopt a KL loss that only considers the non-target classes for addressing the dominant bias issue. Besides, to further boost the transferability, we incorporate the min-max learning framework to self-adjust the ensemble weights for each CNN. Experiments results validate that considering the non-target KL loss can achieve superior transferability than the original KL loss by a large margin, and the min-max training can provide a mutual benefit in adversarial ensemble attacks. The source code is available at: https://github.com/WJJLL/ND-MM.



(a) Targeted attack

(b) Non-targeted attack

Figure 1: The PCC analysis between fused logits and single-model logit vector for (a) the targeted UAP and (b) the non-targeted UAP. l_v, l_r and l_d is the logit computed by the VGG16, ResNet50, and DenseNet121, respectively. l_f is the fused logits. (More details can be found in Sec.)

Introduction

The rapid development of Convolutional Neural Networks (CNNs) has wittiness great success in various fields, *e.g.*, image classification (He et al. 2016), object detection (Ren et al. 2015), image segmentation (Long, Shelhamer, and Darrell 2015). However, recent studies (Goodfellow, Shlens, and Szegedy 2015a,b) indicated that the CNNs are highly vulnerable to adversarial samples, *i.e.*, adding a small quasi-imperceptible perturbation can make the CNNs output wrong predictions for an input. Besides, the adversarial samples show strong transferability in attacking other unknown black-box CNNs, and several different methods (Kurakin, Goodfellow, and Bengio 2018; Dong et al. 2018, 2019; Zhao, Liu, and Larson 2021) have been proposed to improve the transferability of adversarial samples.

Instead of crafting the perturbations for each input individually, Moosavi-Dezfooli et al. (2017) found the existence of input-agnostic universal adversarial perturbation (UAP), that a single perturbation δ can fool a CNN model on a majority of inputs. After that, a wide range of methods (Mopuri, Garg, and Babu 2017; Zhang et al. 2020b; Benz et al. 2020) have been proposed for crafting more destructive and purposeful UAPs. Notably, Zhang et al. (2020b) further analyzed the properties of UAPs based on the Pearson Correlation Coefficient (PCC) between the logits of adversarial samples and the UAPs and found that the UAPs contain the dominant features for the adversarial samples. Besides, Zhang et al. (2020b) crafted the targeted UAPs that

*Corresponding author

can change the decision of an adversarial sample to a predefined target class (targeted attack).

In the other aspect, using multiple CNNs to generate the adversarial samples can effectively increase the transferability of attacking unknown black-box models (Liu et al. 2016; Dong et al. 2018). For crafting UAP, the ensemble strategy also can improve the performance. However, we also observe a severe bias issue of the crafted UAP with the ensemble attacks in the non-targeted and targeted attacks (as shown in Fig.1). Specifically, when computing the PCCs between the fused logits and individual logits by using the crafted UAP δ as the input to the training CNNs, one PCC will almost equal 1, which is significantly higher than the others. Based on the analysis in (Zhang et al. 2020b), this phenomenon indicates that one model will have the dominant role in the ensemble learning, and the contributions of other training models are been weakened. Thus, the widely used ensemble training with the average weights for each training model will limited the transferability of UAPs.

To address this bias issue, the primary attempt is leveraging the Kullback–Leibler (KL) divergence loss to enforce that the logit distribution of each CNN is similar to the fused logits, thus encouraging their joint contributions to increase the final transferability. After using the KL loss, the bias can be effectively alleviated in the first few training iterations. But, the bias phenomenon still exists when training with more iterations. To take a closer look of this issue, we decouple the KL loss into a target class-related part and a non-target class-related part, following the reformulation of knowledge distillation in (Zhao et al. 2022). The non-target class-related part is conditioned on the sum probability of all non-targeted classes. Since the loss functions for crafting adversarial samples aim to increase the probability of the target class in the targeted attack (or predicted class of the adversarial sample in the non-targeted attack) to achieve the fooling goal, consequently, the probability of the target class will be close to 1, and other non-target classes will be close to 0. Thus, the knowledge of the non-target classes will be neglected in the original KL loss function, which is insufficient to reduce the bias issue. Additionally, the non-target knowledge has not been considered in previous studies for crafting adversarial samples with higher transferability.

Inspired by the above analysis, we raise the question: *"Can we increase the transferability of UAP by only considering the knowledge in non-target classes to narrow the gap between the logits distribution of each CNN and the fused logits?"* Therefore, in this study, we leverage the non-target KL loss function only based on the logits of non-target classes while omitting the logit of the target class during the whole training process of the ensemble adversarial attack. Besides, to further boost the transferability, we incorporate the min-max learning framework (Wang et al. 2021) into our loss to adjust the ensemble weights for each training CNN, instead of using the average weights.

To sum up, the main contributions of this study for improving transferability of UAPs are as follows:

- The original KL loss is insufficient to deal with the dominant bias issue in the convention ensemble attacks. After decoupling the KL loss into a target part and a non-target part, we find the non-target knowledge will be omitted and propose using a non-target KL to solve the issue.
- Experimental results demonstrate the effectiveness of the non-target KL for improving the transferability of UAPs than the original KL loss. Further incorporating the min-max learning framework can provide extra benefits for the performance.

Related Work

In this section, we will review the recent advances related to our study in following two aspects: **Universal Attacks** and **Ensemble Attacks**.

Universal Attacks

Following the finding of UAP in Moosavi-Dezfooli et al. (2017), a wide range of different methods have been proposed for generating more disruptive and purposeful UAPs. In this study, we mainly categorized them into **feature-based** and **decision-based**.

Feature-based methods crafted the UAPs that can incorrectly activate the neurons in the hidden layers of the CNNs. For example, Mopuri, Garg, and Babu (2017) proposed the training data-free Fast Feature Fool (FFF) method to optimize the UAP δ that can maximize the activation of neurons when feeding it to the surrogate CNN. Follow up, Mopuri, Ganeshan, and Babu (2019) further boosted the fooling ability of FFF by leveraging the statistics or the original images from the training dataset as prior information. Li et al. (2019) exploited the model uncertainty to generate a more disruptive UAP, in which the Monte Carlo sampling is employed to activate more neurons and a textural bias is adapted as a statistical uncertainty. Khrulkov and Oseledets (2018) used the (p, q)-singular vectors of the Jacobian metrics of the hidden layers' features for computing the UAP.

Decision-based methods focus on disturbing the decision boundary of the adversarial samples. For example, Li et al. (2019) learned the UAPs for disturbing the image retrieval systems by corrupting the pair-wise and list-wise relationship among the retrieval ranking list. Zhang et al. (2020b) first found that the UAPs contain dominant features of the adversarial samples and then proposed margin-based loss functions to craft the UAPs by enlarging the logit margins between the targeted class and other non-targeted classes in the targeted attack. Besides, Zhang et al. (2021) observed the dominant label phenomenon in the UAPs and leveraged a simple self-supervision cosine similarity loss function to minimize the logits' similarity between a clean sample and its adversarial counterpart. Benz et al. (2020) and Zhang et al. (2020a) achieved the targeted attacks that can only affect the selected classes and have less impact on other non-selected classes. These targeted attacks were trained by combining the margin-based loss function in (Zhang et al. 2020b) and the cross-entropy loss function. The cross-entropy loss mainly forces the samples from the non-selected classes to maintain their original labels.

Ensemble Attacks

The ensemble attack (Liu et al. 2016; Dong et al. 2018) is an effective strategy to increase the adversarial transferability by training with multiple CNN models. Considering K training CNN models, the general objective function can be formulated as

$$\mathcal{L}_{ens} = \mathcal{L}\left(\sum_{k=1}^{K} w_k \boldsymbol{p}_k(x+\delta)\right), \quad s.t. \sum_{k=1}^{K} w_k = 1, \quad (1)$$

where $\mathcal{L}(\cdot)$ computes loss function for ensemble outputs in targeted attack or non-targeted attack, w_k is the ensemble weight for k-th model, and $\boldsymbol{p}_k(x+\delta)$ is the outputs (*e.g.*, logits, probabilities) of adversarial sample $x+\delta$ from k-th model. In (Liu et al. 2016), the probabilities were used for achieving the ensemble attack. Dong et al. (2018) leveraged the logits for ensemble attack, which empirically obtained better performance than ensemble in probabilities and ensemble over loss functions $\sum_{k=1}^{K} w_k \mathcal{L}\left(\boldsymbol{p}_k(x+\delta)\right)$. However, (Liu et al. 2016; Dong et al. 2018) simply used the average weight for each model in experiments, *i.e.,* $w_k = 1/K$, thus the performance was limited. Wang et al. (2021) further explored the min-max framework for producing the ensemble attacks which can self-adjust the weights w_i for each model, optimizing the following loss function,

$$\mathcal{L}_{ens} = \min_{\delta} \max_{\boldsymbol{w}} \sum_{k=1}^{K} w_k \mathcal{L}\left(\boldsymbol{p}_k(x+\delta)\right), \ \ s.t. \sum_{k=1}^{K} w_k = 1. \quad (2)$$

On the other hand, Yuan et al. (2021) proposed Meta Gradient Adversarial Attack (MGAA) based on FGSM-related techniques (Goodfellow, Shlens, and Szegedy 2015a; Kurakin, Goodfellow, and Bengio 2018; Dong et al. 2019; Xie et al. 2019) to simulate white-box and black-box scenarios for boosting the transferability. Xiong et al. (2022) proposed the stochastic variance reduced ensemble (SVRE) attack to reduce the gradient variance of the ensemble models and take full advantage of the ensemble attack. In contrast with previous methods, we will leverage the non-target knowledge which have not been considered in previous studies for improving the transferability in ensemble attacks.

Method

In this section, we first describe the baseline model used for crafting the UAPs. Then, we introduce the details of using non-target knowledge for improving the transferablity of adversarial examples. Finally, we present the min-max training procedure for adjusting ensemble weights.

Baseline

Given K surrogate CNNs $\{M_1, M_2, ..., M_K\}$, we can compute their corresponding logits $\{\boldsymbol{l}_1(x+\delta), \boldsymbol{l}_2(x+\delta), .., \boldsymbol{l}_K(x+\delta)\}$ for an input adversarial sample $x+\delta$. Their fused logits is denoted as $\boldsymbol{f}(x+\delta) = \sum_k w_k \boldsymbol{l}_k(x+\delta)$. To optimize the δ in the targeted and the non-targeted attacks, we use the following loss functions for training built upon the ensemble loss Eq. 1.

Targeted Attack: The margin-based loss function proposed in (Zhang et al. 2020b) is used for targeted attack,

$$\mathcal{L}_t = \max\left(\max_{i \neq t} \boldsymbol{f}_i(x+\delta) - \boldsymbol{f}_t(x+\delta), -\kappa\right), \quad (3)$$

where $\boldsymbol{f}_i(x+\delta)$ denotes the i-th logit related to class i, t is the targeted class, κ is hyper-parameter which is set to 10.

Non-Targeted Attack: For the non-targeted attack, we adopt the widely used negative Cross-Entropy function for crafting the UAPs, denoted as:

$$\mathcal{L}_{nt} = \mathbb{1}_c^T \cdot \log\left(\text{softmax}(\boldsymbol{f}(x+\delta)\right) \quad (4)$$

where $c = \arg\max \boldsymbol{f}(x)$ is the estimated label of the clean sample x, and $\mathbb{1}_c$ is the corresponding one-hot vector.

Pearson Correlation Coefficient Analysis: Although the ensemble training strategy could improve the overall transferability, we also find that the generated UAPs still bias to one of the training CNNs. Following the PCC analysis in (Zhang et al. 2020b), Fig. 1 shows the $\text{P}\left(\boldsymbol{f}(\delta), \boldsymbol{l}_i(\delta)\right)$ between the fused logits $\boldsymbol{f}(\delta)$ and the logits $\boldsymbol{l}_i(\delta)$ of each individual model M_i based on the δ learned from ensemble models (*i.e.*, VGG16, ResNet50, DenseNet121). The PCC with VGG16 is 1.00 in both targeted and non-targeted attacks, which is significantly higher than the PCCs with ResNet50 and DenseNet121. **This phenomenon reveals the VGG16 plays a dominate role during the optimization of ensemble attack, thereby weakening the contributions of others.** Consequently, simply averaging the outputs of different CNNs to obtain the ensemble adversarial attack will limit the transferability of attacking other black-box CNNs.

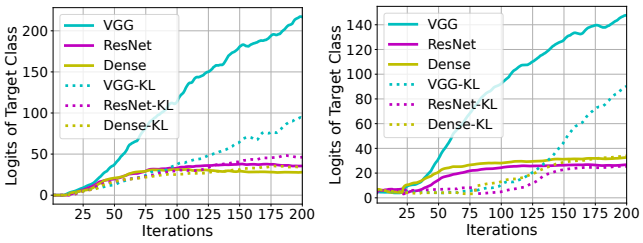
Non-Target Kullback-Leibler

To deal with the bias issue as shown in Fig. 1, we can leverage the Kullback–Leibler (KL) divergence loss function to encourage each $\boldsymbol{l_k}(x+\delta)$ having a similar distribution with the fused logits $\boldsymbol{f}(x+\delta)$ during the optimization process. For brevity, we omit the $(x+\delta)$ and can have the KL loss function $\mathcal{L}_{\text{KL}}$ for increasing the similarity as,

$$\begin{aligned}
\mathcal{L}_{\text{KL}} &= \sum_{k=1}^{K} \text{KL}\big(\text{softmax}(\boldsymbol{f}) \,||\, \text{softmax}(\boldsymbol{l_k})\big) \\
&= \sum_{k=1}^{K} \sum_{i=1}^{C} p_f^i \log(\frac{p_f^i}{p_k^i}) \\
&= \sum_{k=1}^{K} \Big[p_f^t \log(\frac{p_f^t}{p_k^t}) + \sum_{i \neq t}^{C} p_f^i \log(\frac{p_f^i}{p_k^i})\Big]
\end{aligned} \quad (5)$$

where p_f^i and p_k^i are the probabilities of class i after the softmax over the fused logits f and the logits of the k-th model, respectively. t is the target class in the targeted attack or the predicted class of adversarial example in the non-targeted attack[1], and C is the total number of classes.

[1]For simplicity, we will denote the predicted class with maximum logit of the adversarial sample in the non-targeted attack as the target class t in following sections.



(a) Targeted attack (b) Non-targeted attack

Figure 2: The logits trend over iterations of the UAPs in the Targeted attack and the Non-targeted attack.

In Figure 2, we plot the logit trend of the UAPs over iterations of the target class t, and can observe that the KL loss function is effective in alleviating the bias issue during the first 100 to 125 iterations. However, the target logit of VGG16 will rapidly increase when training with more iterations, and thus it will neglect the effects of KL loss since the probability of target class t is close to 1 and other non-target classes will close to 0 after the softmax.

To address this problem, we first delve into the KL loss and decouple it into the target class related part and the non-target classes related part following the reformulation of knowledge distillation in (Zhao et al. 2022). Let's define the re-normalized probabilities after using softmax among non-target classes as $\hat{\boldsymbol{p}}^{nt} = \{\hat{p}^1, \cdots, \hat{p}^{t-1}, \hat{p}^{t+1} \cdots, \hat{p}^C\}$ (*i.e.*, $\hat{p}^i = \frac{e^{z_i}}{\sum_{j\neq t}^{C} e^{z_j}}$ and z_i is the logit of class i). According to the probability $p^i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$ and $p^{nt} = 1 - p^t = \frac{\sum_{j\neq t}^{C} e^{z_j}}{\sum_{j=1}^{C} e^{z_j}}$, we can get $p^i = p^{nt} \cdot \hat{p}^i$. Then, the KL loss function Eq. 5 can be reformulated as:

$$
\begin{aligned}
\mathcal{L}_{\mathrm{KL}} &= \sum_{k=1}^{K} \Big[p_f^t \log(\frac{p_f^t}{p_k^t}) + \sum_{i\neq t}^{C} p_f^i \log(\frac{p_f^i}{p_k^i}) \Big] \\
&= \sum_{k=1}^{K} \Big[p_f^t \log(\frac{p_f^t}{p_k^t}) + \sum_{i\neq t}^{C} p_f^{nt} \cdot \hat{p}_f^i \log(\frac{p_f^{nt} \cdot \hat{p}_f^i}{p_k^{nt} \cdot \hat{p}_k^i}) \Big] \\
&= \sum_{k=1}^{K} \Big[\underbrace{p_f^t \log(\frac{p_f^t}{p_k^t}) + p_f^{nt} \log(\frac{p_f^{nt}}{p_k^{nt}})}_{\text{Target KL}} + \underbrace{p_f^{nt} \sum_{i\neq t}^{C} \hat{p}_f^i \log(\frac{\hat{p}_f^i}{\hat{p}_k^i})}_{\text{Non-target KL}} \Big].
\end{aligned}
\tag{6}
$$

After the reformulation, we can find the first part optimize the KL divergence only related to the target class, and the second part related to the non-target classes. Since the fused logit of the target class f_t rapidly increases, then $p_f^t \approx 1$ and $p_f^{nt} \approx 0$. Hence, the original KL loss function will omit the knowledge from the non-target classes. On the other aspects, the first part in Eq. 6 has the similar effect as Eq. 3 and Eq. 4, which enlarges the logit of the target class. Therefore, the original KL loss extensively utilizes target-class related knowledge to enforce the similarity between the logits of each model and the fused logits, **while largely overlooking the knowledge among non-target classes. Thus it is still inadequate to deal with the bias issue and limits the transferability of adversarial examples.**

The above analysis inspires us to only leverage the knowledge contained in non-target classes to narrow the gap between logits distribution of each model and the fused logits distribution for improving the transferability. The corresponding non-target KL loss $\mathcal{L}_{NKL}$ for addressing the bias issue can be depicted as:

$$
\begin{aligned}
\mathcal{L}_{NKL} &= \sum_{k=1}^{K} \mathrm{KL}\Big(\mathrm{softmax}(\hat{\boldsymbol{f}}) \,||\, \mathrm{softmax}(\hat{\boldsymbol{l}_k}) \Big) \\
&= \sum_{k=1}^{K} \mathrm{KL}\left(\hat{\mathbf{p}}^f \| \hat{\mathbf{p}}^k \right)
\end{aligned}
\tag{7}
$$

where $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{l}_k}$ are logits without the target class t.

Finally, we combine the ensemble loss functions $\mathcal{L}_t$ (Eq. 3) or $\mathcal{L}_{nt}$ (Eq. 4) and the non-target KL loss function Eq. 7 to optimize the universal perturbation δ. The whole loss function is

$$
\mathcal{L}_{all}^t = \mathcal{L}_t + \lambda \mathcal{L}_{NKL}, \quad \mathcal{L}_{all}^{nt} = \mathcal{L}_{nt} + \lambda \mathcal{L}_{NKL}, \tag{8}
$$

where λ is a hyper-parameter controlling the influence of non-target knowledge.

Min-Max Training

Wang et al. (2021) have demonstrated the effectiveness of self-adjusting the $\boldsymbol{w}$ in learning ensemble attacks, we also leverage the min-max training framework for further boosting the transferability. In contrast with the Eq. 2 that ensemble over multiple losses in (Wang et al. 2021), we optimize the bi-level min-max function over the fused logits,

$$
\begin{aligned}
\mathcal{L}_{ens} = \min_{\delta} \max_{\boldsymbol{w}} \sum_{k=1}^{K} \mathcal{L}_{all}\big(w_k \boldsymbol{l}_k(x+\delta)\big) - \gamma ||\boldsymbol{w} - \frac{1}{K}||_2, \\
s.t. \sum_{k=1}^{K} w_k = 1,
\end{aligned}
\tag{9}
$$

where γ is a regularization parameter and set to the number of models K in this study.

The above Eq. 9 is a bi-level optimization problem (Liu et al. 2021), and can be solved by optimizing the inner maximization and the outer minimization iteratively. For the inner maximization, the $\boldsymbol{w}$ is optimized by the following gradient ascent algorithm:

$$
\boldsymbol{w} = \boldsymbol{w} + \alpha \nabla_{\boldsymbol{w}} \Big[\mathcal{L}_{all}\big(\sum_{k=1}^{K} w_k \boldsymbol{l_k}(\boldsymbol{x}+\delta) \big) - K ||\boldsymbol{w} - \frac{1}{K}||_2 \Big], \tag{10}
$$

where α is the learning rate for inner maximization. In this sense, the trainable $\boldsymbol{w}$ can encode the difficulty level of attacking each model. For the outer minimization, we employ the ADAM optimizer and mini-batch training, which is as same as in (Zhang et al. 2020b), denoted as below:

$$
\delta = \mathrm{Clip}\Big[\delta - \beta \nabla_{\delta} \mathcal{L}_{all}\big(\sum_{k=1}^{K} w_k \boldsymbol{l_k}(\boldsymbol{x}+\delta) \big) \Big], \tag{11}
$$

where β is the learning rate of outer optimization, and the Clip operation ensures the learned δ within the valid perturbation ϵ-ball.

Attacks	Models	Ens-Logits ntFR	Ens-Loss ntFR	Min-Max ntFR	Ours-Loss ntFR	Ours-Logits ntFR
White-box	VGG16	**95.77**	**95.94**	86.76	90.88	91.17
	ResNet50	74.67	68.62	87.60	89.05	89.41
	DenseNet121	68.38	61.72	84.76	86.49	86.25
	Avg.	79.61	75.42	86.38	**88.81**	**88.94**
Black-box	VGG11	77.25	76.36	72.54	77.37	77.98
	VGG13	89.62	89.51	82.10	85.82	86.21
	ResNet18	59.98	56.99	69.36	74.17	74.04
	ResNet101	53.87	48.54	70.38	74.02	73.91
	DenseNet161	57.80	53.06	70.48	75.15	74.65
	VGG19-BN	89.07	88.92	84.82	88.68	89.00
	WideResNet50-2	64.09	60.10	75.39	78.49	78.66
	GoogleNet	50.33	47.13	54.78	61.54	61.83
	Avg.	67.75	65.07	72.48	**76.91**	**77.04**

(a) Non-targeted Attack

Attacks	Models	Ens-Logits		Ens-Loss		Min-Max		Ours-Loss		Ours-Logits	
		tFR	ntFR	tFR	ntFR	tFR	ntFR	tFR	ntFR	tFR	ntFR
White-box	VGG16	69.66	91.57	68.97	89.42	61.32	86.06	58.80	82.37	58.50	82.25
	ResNet50	32.89	61.25	52.52	72.00	57.69	75.03	61.10	77.41	63.22	78.29
	DenseNet121	29.77	58.68	47.43	69.62	55.12	74.38	59.59	76.06	60.55	76.45
	Avg.	44.11	70.50	56.31	77.01	58.04	78.49	**59.83**	**78.61**	**60.76**	**79.00**
Black-box	VGG11	13.37	67.90	16.11	67.60	15.54	65.64	17.40	64.28	18.25	64.96
	VGG13	26.81	80.68	27.80	79.08	25.51	76.00	31.05	73.80	31.56	73.99
	ResNet18	7.84	51.53	15.95	56.68	17.99	58.17	19.61	60.79	21.35	61.51
	ResNet101	6.74	43.91	16.09	51.19	17.68	53.32	22.24	56.63	23.01	57.35
	DenseNet161	15.70	47.92	27.90	55.11	32.46	58.76	37.00	60.86	38.72	61.60
	VGG19-BN	30.77	77.79	32.27	77.48	31.51	74.90	38.52	74.39	39.61	74.28
	WideResNet50-2	10.99	49.02	24.46	55.96	28.64	58.34	32.27	60.91	33.03	61.14
	GoogleNet	1.87	38.27	4.97	41.37	5.63	42.50	7.12	44.79	8.62	45.78
	Avg.	14.26	57.13	20.69	60.56	21.87	60.95	**25.65**	**62.06**	**26.77**	**62.58**

(b) Targeted Attack

Table 1: Comparison with previous methods in both non-targeted attack and targeted attack. (The targeted fooling ratio (tFR) and non-targeted fooling ratio (ntFR) in Targeted Attack are the mean value over 8 different targeted classes.)

Experiments

In this section, we conduct experiments on the ImageNet dataset (Deng et al. 2009) to evaluate the effectiveness of the proposed method on the non-targeted and targeted attacks.

Experimental Setup

Datasets. For training and testing the UAPs, we randomly select $50k$ images from the ImageNet training set for training, and evaluate the attacking performance on ImageNet validation set ($50k$ images).

CNN models. For crafting the UAPs, we leverage 3 white-box models with different network architectures for training, *i.e.*, DenseNet121 (Huang et al. 2017), VGG16 (Simonyan and Zisserman 2015) and ResNet50 (He et al. 2016). For evaluating the transferability of attacking unknown black-box CNNs, we choose 8 models, including: VGG11, VGG13,VGG19-BN (Simon, Rodner, and Denzler 2016), ResNet18, ResNet101, WideResNet50-2 (Zagoruyko and Komodakis 2016), DenseNet161 and GoogleNet (Szegedy et al. 2015). Additionally, we also evaluate the performance of attacking 4 defense mechanisms, including, the neural representation purifier (NRP) based on input-level purifier (Naseer et al. 2020), the augmentation-based Augmix (Hendrycks et al. 2019), SIN (Geirhos et al. 2018) with stylized ImageNet dataset, and the adversarial training models (Salman et al. 2020).

Hyper-parameters. In the training phase, the hyper-parameters are set as follows: the number of classifier models $K = 3$, the batch size $N = 20$, the number of training epochs $T = 5$, the learning rate of inner maximization $\alpha = 0.003$. For other parameters, we follow the settings in (Zhang et al. 2020b), *i.e.*, the perturbation magnitude $\epsilon = 10$, and the initial learning rate of the Adam optimizer (outer minimization) $\beta = 0.005$.

Evaluation metrics. For the non-targeted attack, we adopt the fooling ratio (ntFR) to evaluate the performance, which calculates the ratio of samples whose prediction changes when the UAPs are added to the original images. For the targeted attack, we adopt both the fooling ratio and the targeted fooling ratio (tFR) to validate the performance. The tFR computes the percentage of adversarial samples (except the samples of the target class) that are successfully attacked to a pre-defined target class. In this study, the targeted attack is evaluated over 8 randomly selected target classes, follow-

ing (Zhang et al. 2020b).

Comparison with Previous Methods

In this section, we compare our proposed method with previous methods (Dong et al. 2018; Wang et al. 2021), and report the results in Table 1. The Ens-Logits and Ens-Loss are the ensemble on logits and ensemble on loss functions used in (Dong et al. 2018). The Min-Max is the self-adjustable ensemble training strategy used in (Wang et al. 2021). All the comparison methods are trained with the same loss function as described in Section .

From Table 1, we have the following findings. **1)** Our proposed methods can obtain better attacking performance than (Dong et al. 2018; Wang et al. 2021) in both non-targeted and targeted scenarios. Besides, there is no significant difference between the ensemble on logits and the ensemble on loss under our proposed ensemble framework. **2)** By comparing the Ens-Logits and Ens-Loss, we can find that the Ens-Logits can achieve better performance in non-targeted attack, while Ens-Loss is better for the targeted attack. Additionally, the bias issue to the VGG16 exists obviously in the Ens-Logits for the targeted attack. **3)** Self-adjusting the weights $\boldsymbol{w}$ of different models in the (Wang et al. 2021) can largely alleviate the bias issue and increase the transferability for attacking the unknown black-box models in non-targeted attack, while slightly increasing the performance in the targeted attack when compared with the Ens-Loss.

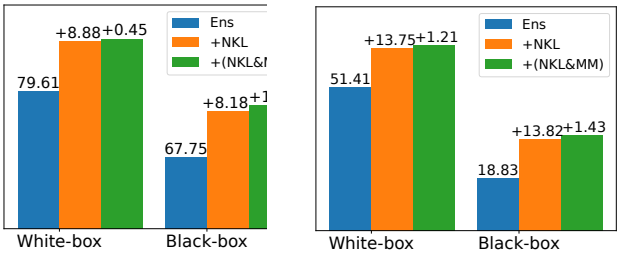
Ablation Studies

In this part, we conduct a series of ablation experiments in the non-targeted and targeted attacks as follows: **1)** We compare the performance of the non-target KL (NKL) with the original KL divergence loss function. **2)** We analyze the effectiveness of each component in our proposed method, *i.e.*, NKL and min-max training (MM). **3)** We study the impact of the hyper-parameters λ for controlling the influence of non-target knowledge in our proposed method. Here, we follow the settings in (Zhang et al. 2020b) and choose the class 'sea lion' as the target class for targeted attacks. All the experiments are reported with an average value of 5 times.

Non-target KL *v.s.* Original KL: From Table 2, we can find that incorporating the original KL loss can increase the performance of the ensemble attack over the baseline. Significantly, the average tFR is increased from 51.41% to 63.12%, 18.83% to 27.66% in white-box and black-box targeted attacks, respectively. However, we also notice that the crafted UAPs are still largely biased to the VGG16.

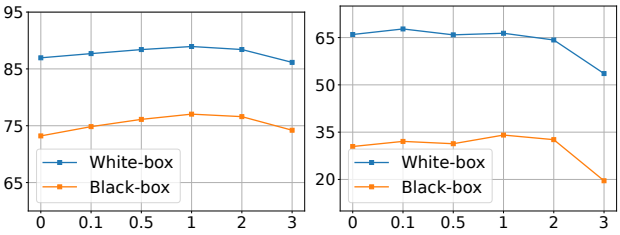
On the other aspect, we can obtain a more consistent FR and ntFR for the three white-box CNNs when incorporating the NKL into the ensemble attacks. Besides, in the black-box transfer scenarios, the average ntFR in the non-targeted attacks and the average tFR in the targeted attacks are around 5% higher than the original KL. These results suggest that the non-target knowledge plays an essential role for dealing the bias and improving the transferability of the UAPs.

Effectiveness of each component: The NKL in Section and the MM in Section are two main components for improving the transferability in our proposed method. In the part, we evaluate the effectiveness by gradually adding them into the based models. From Fig. 3, we can find the NKL can significantly boost the performance in both white-box and black-box attacks. For example, for the non-targeted attacks, the average fooling rates are increased by 8.88% and 8.18% in white-box attacks and black-box attacks, respectively. The average targeted fooling rates are raised by 13.75% and 13.82% for white-box targeted attacks and black-box targeted attacks, respectively. When further using the min-max training, the ntFR and tFR will slightly increase. These experimental results validate the mutual benefits of these two components in our method.



(a) Non-targeted attack (ntFR) (b) Targeted attack (tFR)

Figure 3: Ablation study on the performance of different components for non-targeted and targeted attacks. (MM: Min-Max training)



(a) Non-targeted attack (ntFR) (b) Targeted attack (tFR)

Figure 4: Parameter sensitivity analysis of using different λ in Eq 8 for (a) Non-targeted attack and (b) Targeted attack.

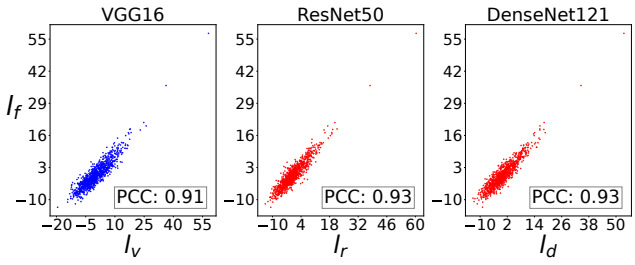
Impact on hyper-parameter: In this section, we evaluate the impact of the NKL under different λ in Eq 9 with the min-max training strategy. As shown in Fig. 4(a), the ntFR for both white-box and black-box CNNs is increasing along with the increment of λ, and the optimal value is $\lambda = 1$ in the non-targeted attack. And further increasing λ will decrease the performance. In the targeted attack (Fig. 4(b)), the optimal value for white-box CNNs is $\lambda = 0.1$. For black-box CNNs, the highest tFR is achieved at $\lambda = 1$. Moreover, we can observe a significant performance drop when $\lambda = 3$.

Visualization of the PCC

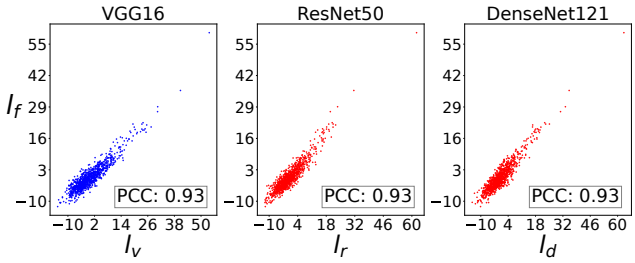
In Figure 5, we visualize the $\mathrm{P}(\boldsymbol{f}(\delta), \boldsymbol{l}_i(\delta))$ of the UAP δ learned by our proposed method. As can be seen, the PCC values (0.93 and 0.94) are very similar to each other in both the targeted attack and the non-targeted attack. This visualization can further verify that our proposed method can deal

Attack	Model	Non-targeted Attaks			Targeted Attaks					
		Ens	+KL	+NKL	Ens		+KL		+NKL	
		ntFR	ntFR	ntFR	tFR	ntFR	tFR	ntFR	tFR	ntFR
White-box	VGG16	95.77	95.27	90.89	81.61	92.61	73.05	88.34	65.99	84.16
	ResNet50	74.67	79.77	89.02	33.74	61.45	57.24	74.04	65.44	78.88
	DenseNet121	68.38	75.75	85.55	38.89	61.56	59.06	73.43	64.04	78.27
	Avg.	79.61	83.60	88.49	51.41	71.87	63.12	78.60	65.16	80.44
Black-box	VGG11	77.25	77.59	76.22	14.46	68.48	13.90	64.53	16.10	65.08
	VGG13	89.62	89.23	85.54	34.45	82.04	33.65	77.85	34.22	75.36
	ResNet18	59.98	64.04	73.19	11.84	53.90	21.90	59.13	31.09	65.16
	ResNet101	53.87	59.64	72.57	8.21	43.28	27.01	53.46	32.46	58.54
	DenseNet161	57.80	64.17	73.21	23.55	48.35	42.82	59.03	49.47	64.40
	VGG19-BN	89.07	89.54	88.05	44.08	79.19	46.76	77.23	49.79	77.02
	WideResNet50-2	64.09	68.37	77.61	12.76	49.80	30.17	57.03	40.72	62.92
	GoogleNet	50.33	53.72	61.04	1.29	38.84	5.04	41.20	7.33	45.72
	Avg.	67.75	70.79	75.93	18.83	57.98	27.66	61.18	32.65	64.28

Table 2: The comparison between the non-target KL (NKL) and the original KL.



(a) Targeted attack



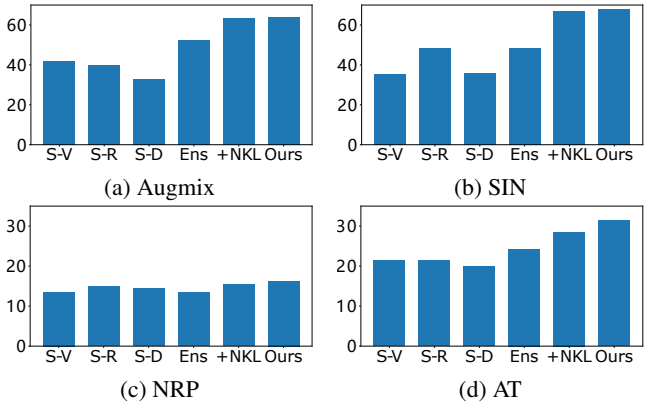
(b) Non-targeted attack

Figure 5: The correlation analysis between fused logit vector and single-model logit vector for targeted (a) and non-targeted UAP (b). The UAP was crafted with our method. l_f is fused logit, l_v is VGG16-model logit.

with the dominant issue towards the VGG16 in the average weighted ensemble attack, as shown in Fig. 1.

Attacking Defense Mechanisms

Finally, we conduct the experiments of attacking several types of defense mechanisms in the non-targeted attack. From Fig. 6, we can have two major conclusions: **1)** Comparing with the UAPs trained on single model, the ensemble attacks can increase the fooling rate against defense methods based on robust training. For the NRP based on input transformation, there is marginal improvement between



(a) Augmix (b) SIN

(c) NRP (d) AT

Figure 6: The fooling rate for non-target attack against different defense mechanisms. The 'S-V', 'S-R' and 'S-D' represent UAPs trained by the single VGG16, ResNet50 and DenseNet121, respectively.

the single-model attacks and multi-models attacks. **2)** The NKL can significantly boost the performance against defense models than the 'Ens'. Besides, the min-max training in ours can further slightly improve the performance. These results further confirm the effectiveness of each component in our method.

Conclusion

In this study, we explored the non-target KL loss that only considers the non-target knowledge to address the dominant bias issue in ensemble attacks. Besides, we leverage the min-max learning framework to adjust the ensemble weights to further boost transferability. Experimental results on both targeted and non-targeted attacks demonstrate that the non-target KL loss outperforms the original KL loss, and the min-max learning can provide mutual benefits for increasing the transferability.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61876159, 62076210, 62276221); the Natural Science Foundation of Fujian Province of China (No. 2022J01002); the Science and Technology Plan Project of Xiamen (No. 3502Z20221025); the Open Project Program of Fujian Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University (No. FKLBDAITI202203).

References

Benz, P.; Zhang, C.; Imtiaz, T.; and Kweon, I. S. 2020. Double targeted universal adversarial perturbations. In *ACCV*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *CVPR*.

Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015a. Explaining and harnessing adversarial examples. In *ICLR*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015b. Explaining and harnessing adversarial examples. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*.

Khrulkov, V.; and Oseledets, I. 2018. Art of singular vectors and universal adversarial perturbations. In *CVPR*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, 99–112. Chapman and Hall/CRC.

Li, J.; Ji, R.; Liu, H.; Hong, X.; Gao, Y.; and Tian, Q. 2019. Universal perturbation attack against image retrieval. In *ICCV*.

Liu, R.; Gao, J.; Zhang, J.; Meng, D.; and Lin, Z. 2021. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE TPAMI*.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *CVPR*.

Mopuri, K. R.; Ganeshan, A.; and Babu, R. V. 2019. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE TPAMI*, 41(10): 2452–2465.

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

Naseer, M.; Khan, S.; Hayat, M.; Khan, F. S.; and Porikli, F. 2020. A self-supervised approach for adversarial robustness. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do adversarially robust imagenet models transfer better? In *NeurIPS*.

Simon, M.; Rodner, E.; and Denzler, J. 2016. Imagenet pre-trained models with batch normalization. *arXiv preprint arXiv:1612.01452*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Wang, J.; Zhang, T.; Liu, S.; Chen, P.-Y.; Xu, J.; Fardad, M.; and Li, B. 2021. Adversarial attack generation empowered by min-max optimization. In *NeurIPS*.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *CVPR*.

Xiong, Y.; Lin, J.; Zhang, M.; Hopcroft, J. E.; and He, K. 2022. Stochastic Variance Reduced Ensemble Adversarial Attack for Boosting the Adversarial Transferability. In *CVPR*.

Yuan, Z.; Zhang, J.; Jia, Y.; Tan, C.; Xue, T.; and Shan, S. 2021. Meta gradient adversarial attack. In *ICCV*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, C.; Benz, P.; Imtiaz, T.; and Kweon, I.-S. 2020a. Cd-uap: Class discriminative universal adversarial perturbation. In *AAAI*.

Zhang, C.; Benz, P.; Imtiaz, T.; and Kweon, I. S. 2020b. Understanding adversarial examples from the mutual influence of images and perturbations. In *CVPR*.

Zhang, C.; Benz, P.; Karjauv, A.; and Kweon, I. S. 2021. Data-Free Universal Adversarial Perturbation and Black-Box Attack. In *ICCV*.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled Knowledge Distillation. In *CVPR*.

Zhao, Z.; Liu, Z.; and Larson, M. 2021. On success and simplicity: A second look at transferable targeted attacks. *NeurIPS*, 34.