A Training and implementation details

We train the baseline ensembles for 200 epochs using SGD with momentum 0.9 and weight decay 0.0001. The initial learning rate is 0.1, and we decay it by $10\times$ at the 100-th and the 150-th epochs. Any models pre-trained on the clean dataset can serve as the starting point for the training of DVERGE. In our implementation, DVERGE starts from the trained baseline ensembles. We follow the aforementioned learning rate schedule, but using a carefully-tuned one is likely to bring extra performance gain. We reproduce ADP [12] and GAL [13] according to either the released code or the paper with recommended hyperparameters and setups. Specifically, they both use Adam optimizer [37] with an initial learning rate of 0.001. Also note that GAL requires the ReLU function to be replaced with leaky ReLU to avoid gradient vanishing. The other configurations stay the same as those of baseline ensembles.

Ensembles with adversarial training follow the baseline's training setup. We use 10-step PGD with $\epsilon = 8/255$ and a step size $\alpha = 2/255$ [3]. More specifically, adversarial examples w.r.t. the whole ensemble are generated in each step of the training process and are used to update model parameters. When combining DVERGE with adversarial training, however, adversarial examples are generated on each sub-model instead of the whole ensemble. We empirically find these choices help each case achieve its best robustness.

We use 0.5 as the input transformation probability for M-DI^2- FGSM [30] and 0.2 as the γ for SGM [31] when generating these two attacks as recommended by their respective papers.

All models are implemented and trained with PyTorch [38] on a single NVIDIA TITAN XP GPU. Evaluation is performed based on AdverTorch [39]. As shown in Table 1, the training of DVERGE is marginally faster than that of adversarial training (AdvT). As they both need extra back propagations to either distill non-robust features or find adversarial examples, DVERGE uses only intermediate features for distillation while adversarial training requires the information back propagated from the final output. As for previous methods, though ADP requires the least time budget, it does not improve the robustness much as shown in Figure 4. And the significantly improved robustness would be worth the extra training cost of DVERGE over GAL. In addition, according to Figure 2, DVERGE could reduce the transferability within the ensemble at the very early stage of the training, so later training epochs have the potential to be simplified to a fine-tuning process without diversity loss, which will require much less training time. Further mitigating the computational overhead of DVERGE would be one of our future goals.

Table 1: Training time comparison on a single TITAN XP GPU. All times are evaluated for training a ResNet-20 ensemble with 3 sub-models for 200 epochs.

Method	Baseline	ADP [12]	GAL [13]	AdvT [3]	**DVERGE**
Training time (h)	1.0	2.0	7.5	11.5	**10.5**

B Analysis on the training ϵ of DVERGE

Table 2: The effect of ϵ on optimizing the feature distillation objective in Equation (1) and the resulting diversity loss measured with Equation (5). f_i and f_j are two ResNet-20 models trained in a standard way on CIFAR-10. $x'_{f_i^l}$ is short for $x'_{f_i^l}(x, x_s)$. The step size used for feature distillation is chosen as $\epsilon/$#steps.

ϵ	#steps	feature distillation objective $\mathop{\mathbb{E}}_{(x,y),(x_s,y_s),l} \left\| f_i^l(x'_{f_i^l}) - f_i^l(x) \right\|_2$	diversity loss $\mathop{\mathbb{E}}_{(x,y),(x_s,y_s),l} \left[\mathcal{L}_{f_j}(x'_{f_i^l}, y_s) \right]$
0.03	10	1.056	1.726
0.05	10	0.793	3.302
0.07	10	0.703	4.738

One important hyperparameter of DVERGE is the ϵ used for feature distillation. This section provides some initial exploration on the effect of using different ϵ. We start by looking at how ϵ affects the

optimization of the feature distillation in Equation (1) and the resulted diversity loss in Equation (5). The results evaluated with 1,000 CIFAR-10 testing images on two pre-trained ResNet-20 models are shown in Table 2. We find that a larger ϵ enables more accurate feature distillation as a smaller distance between the internal representation of the distilled image $x'_{f_i^l}$ and the target image x is achieved. As a result, the distilled image from model f_i can lead to a higher diversity loss on another model f_j, which intuitively encourages the training routine of DVERGE to enforce a greater diversity and therefore a lower transferability between the two models. We empirically confirm this intuition in Figure 6, where we vary the training ϵ and measure the transferability between sub-models. For instance, when ensembles have three sub-models, increasing ϵ from 0.03 to 0.07 decreases the transferability from 8%-10% to 3%-6%. Interestingly, however, for ensembles with five or eight sub-models, although we do observe a drop in the transferability between most of the sub-model pairs by using a larger ϵ, some pairs of the sub-models remain highly vulnerable to one another. In particular, observe that when training an ensemble of 5 sub-models with an ϵ of 0.07, 79% of adversarial examples from the second sub-model can fool the fourth one and 48% of examples transfer in the reverse direction. We leave a thorough and rigorous analysis of this phenomenon to future work.

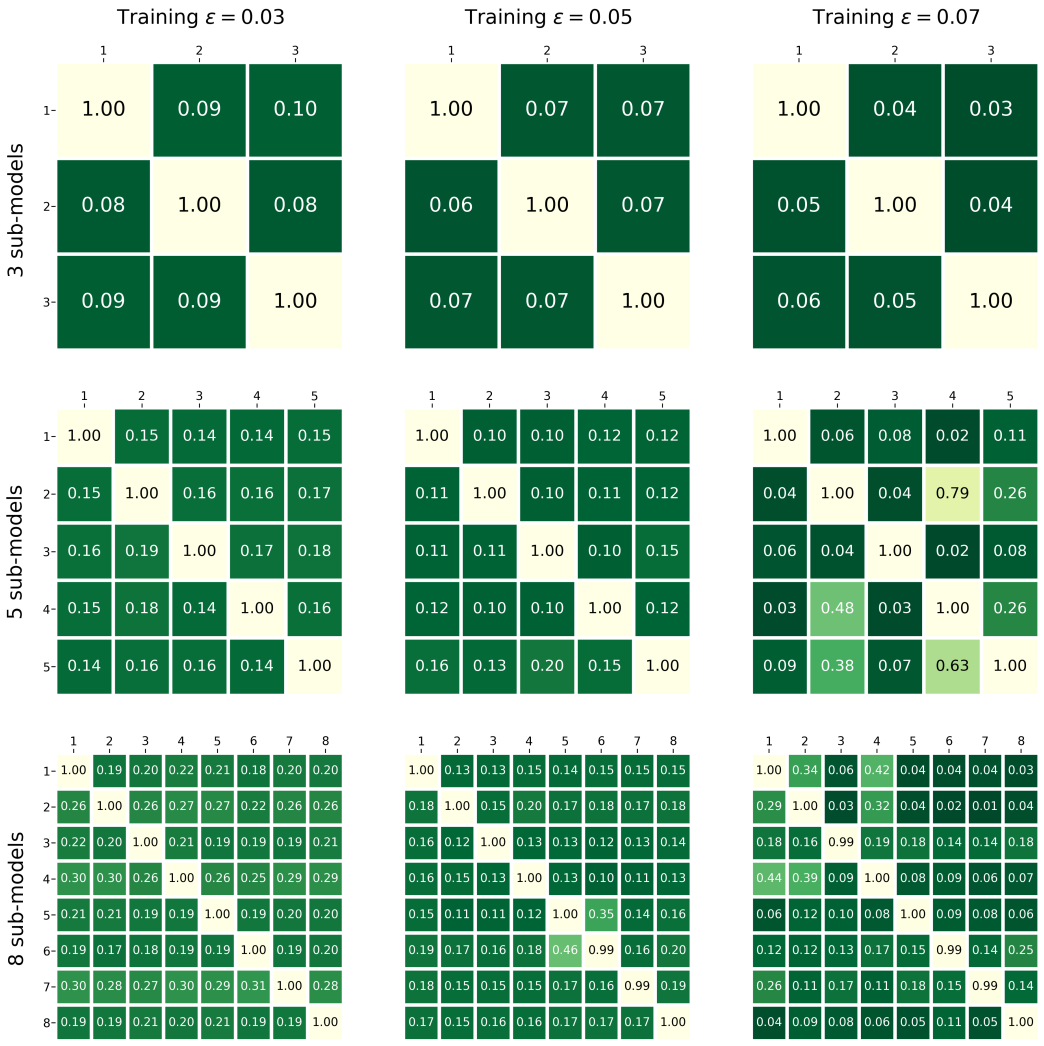

Figure 6: Transferability within DVERGE ensembles trained with different ϵ. The evaluation setting follows that of Figure 3, where the attack perturbation strength is 0.03.

Finally, we look at the clean accuracy and robustness achieved by DVERGE ensembles trained with different ϵ. In Table 3, we observe that training with a larger ϵ leads to a higher black-box

transfer robustness but a lower clean accuracy. The trend between white-box robustness and ϵ is not monotonic though, which we suspect is related to the observation that the attack transferability worsens between some pairs of sub-models under a larger training ϵ, as shown in Figure 6. As the robustness against white-box attacks is not the main focus of DVERGE, we will explore the relationship, both qualitatively and quantitatively, between the transferability among sub-models and the achieved white-box robustness of the whole ensemble in the future.

Table 3: Robustness of DVERGE ensembles trained with different ϵ. In each table block, we report (clean accuracy) / (black-box transfer accuracy under perturbation strength 0.03) / (white-box accuracy under perturbation strength 0.01).

#sub-models \ training ϵ	0.03	0.05	0.07
3	92.9% / 4.2% / 22.7%	92.7% / 26.6% / 32.3%	91.4% / 53.2% / 40.0%
5	92.3% / 30.5% / 43.1%	91.5% / 57.2% / 48.9%	90.2% / 66.5% / 42.3%
8	91.3% / 42.8% / 51.9%	91.1% / 63.6% / 57.9%	89.2% / 71.3% / 52.4%

C Additional results

C.1 Decision region visualization

We visualize the decision regions learned by DVERGE ensembles around more testing images from the CIFAR-10 dataset in Figure 7.

C.2 Transferability within the ensemble under different testing ϵ

In addition to Figure 3, we provide more results of the transferability between sub-models under different attack ϵ in Figure 8. In all cases, DVERGE achieves the lowest level of transferability among all ensemble methods.

Table 4: Accuracy v.s. ϵ against black-box transfer attacks generated from hold-out baseline ensembles. The number in the first column after the slash is the number of sub-models within the ensemble. The results are averaged over three independent runs.

ϵ	clean	0.01	0.02	0.03	0.04	0.05	0.06	0.07
baseline/3	93.9%	9.6%	0.1%	0%	0%	0%	0%	0%
baseline/5	93.9%	9.4%	0%	0%	0%	0%	0%	0%
baseline/8	94.4%	9.7%	0%	0%	0%	0%	0%	0%
ADP/3 [12]	92.8%	20.9%	0.6%	0%	0%	0%	0%	0%
ADP/5 [12]	93.2%	21.8%	0.6%	0%	0%	0%	0%	0%
ADP/8 [12]	93.4%	21.2%	0.4%	0%	0%	0%	0%	0%
GAL/3 [13]	88.3%	76.6%	59.4%	39.8%	23.2%	11.5%	5.8%	2.5%
GAL/5 [13]	91.1%	78.3%	56.9%	33.3%	15.7%	6.3%	2.9%	0.7%
GAL/8 [13]	92.4%	75.1%	46.5%	21.7%	7.5%	2.0%	0.4%	0.0%
DVERGE/3	91.3%	83.2%	69.7%	51.0%	30.7%	15.8%	6.0%	1.5%
DVERGE/5	91.9%	83.8%	71.8%	55.0%	37.2%	21.4%	10.5%	3.2%
DVERGE/8	91.1%	85.2%	76.0%	65.0%	50.2%	34.9%	22.5%	11.2%
AdvT/3 [3]	77.2%	76.3%	74.8%	73.2%	70.9%	68.7%	66.0%	62.7%
AdvT/5 [3]	78.6%	77.8%	76.1%	73.8%	71.3%	69.0%	66.0%	63.3%
AdvT/8 [3]	79.4%	78.2%	76.9%	74.9%	72.3%	69.8%	66.8%	63.9%
DVERGE+AdvT/3	81.4%	79.7%	77.6%	75.3%	72.5%	69.1%	66.4%	62.6%
DVERGE+AdvT/5	83.4%	81.6%	79.3%	76.6%	74.1%	70.3%	66.3%	61.7%
DVERGE+AdvT/8	85.0%	82.8%	80.2%	76.8%	73.0%	68.7%	64.0%	57.9%

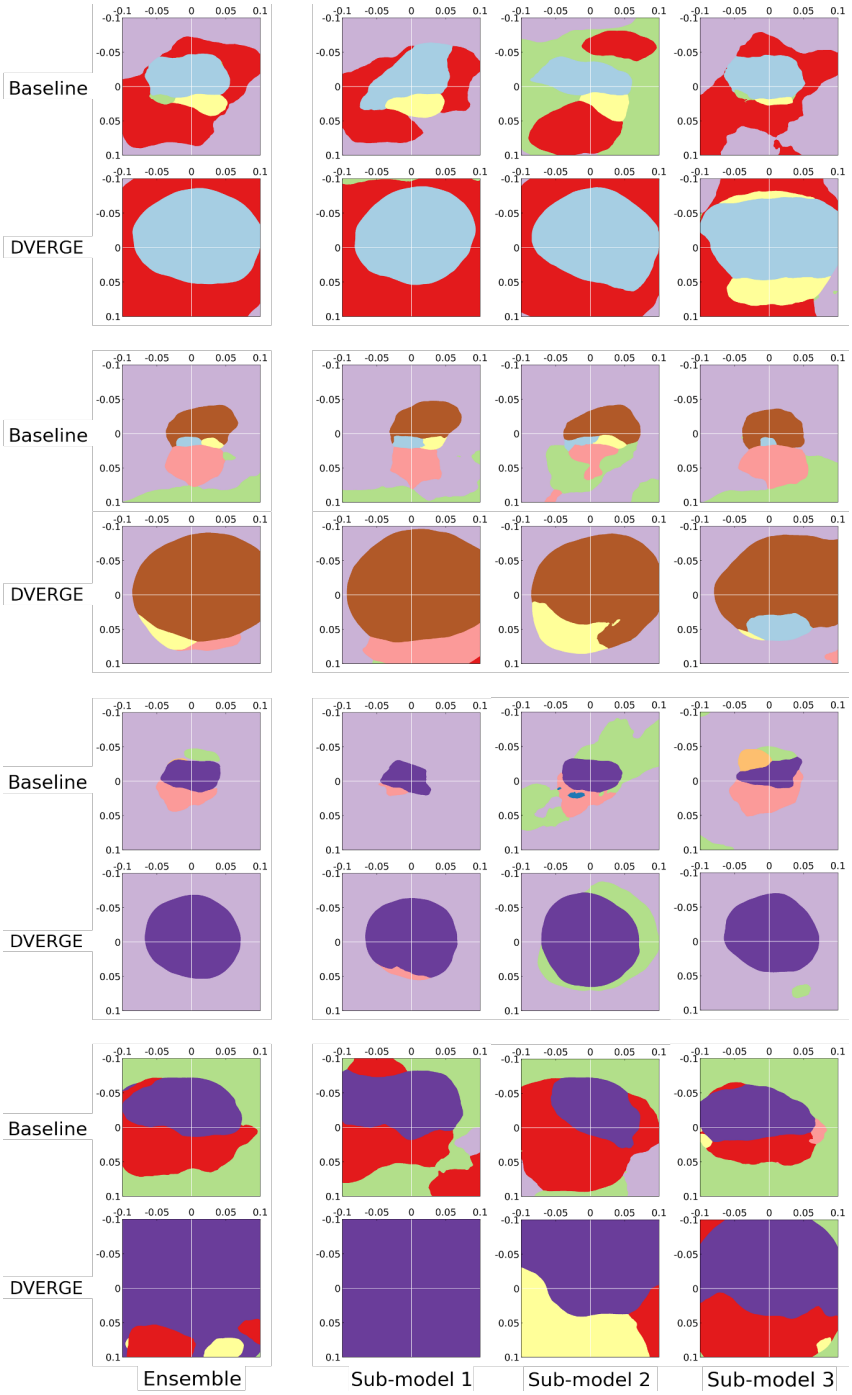

Figure 7: Additional decision region plots of ensembles with 3 ResNet-20 sub-models trained on CIFAR-10. Each pair of two rows is generated with one testing image. The first row is for the baseline ensemble, and the second row is for the DVERGE ensemble. The axes are chosen in the same way as in Figure 1.

C.3 Numerical results for robustness

We report numerical results that correspond to Figure 4 and Figure 5 in Table 4 (black-box transfer accuracy) and Table 5 (white-box accuracy), respectively. Note that ADP presents higher white-box

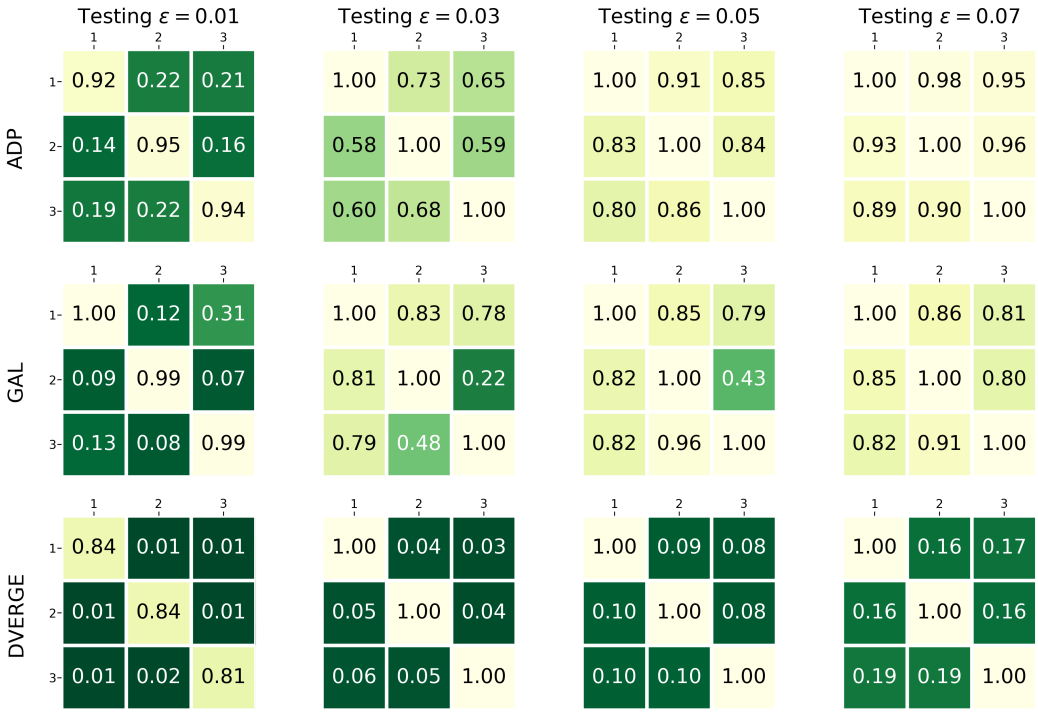

Figure 8: Transferability results under different testing ϵ.

Table 5: Accuracy v.s. ϵ against white-box attacks. The number in the first column after the slash is the number of sub-models within the ensemble. The results are averaged over three independent runs.

ϵ	clean	0.01	0.02	0.03	0.04	0.05	0.06	0.07
baseline/3	93.9%	1.1%	0%	0%	0%	0%	0%	0%
baseline/5	93.9%	1.8%	0%	0%	0%	0%	0%	0%
baseline/8	94.4%	2.8%	0%	0%	0%	0%	0%	0%
ADP/3 [12]	92.8%	9.3%	0.4%	0%	0%	0%	0%	0%
ADP/5 [12]	93.2%	11.2%	0.9%	0.1%	0%	0%	0%	0%
ADP/8 [12]	93.4%	12.7%	4.9%	2.5%	1.3%	0.8%	0.5%	0.2%
GAL/3 [13]	88.3%	9.3%	0.3%	0%	0%	0%	0%	0%
GAL/5 [13]	91.1%	32.2%	7.1%	0.5%	0.1%	0.1%	0%	0%
GAL/8 [13]	92.4%	38.8%	9.4%	0.9%	0.2%	0%	0%	0%
DVERGE/3	91.3%	37.4%	10.2%	2.2%	0.4%	0.2%	0%	0%
DVERGE/5	91.9%	47.7%	19.0%	5.2%	0.7%	0.2%	0.1%	0%
DVERGE/8	91.1%	56.5%	26.3%	10.7%	2.8%	0.5%	0.2%	0.1%
AdvT/3 [3]	77.2%	69.1%	59.2%	48.2%	36.1%	26.1%	17.2%	9.5%
AdvT/5 [3]	78.6%	69.6%	59.5%	48.5%	36.5%	26.2%	16.0%	9.2%
AdvT/8 [3]	79.4%	70.9%	60.8%	48.9%	37.0%	26.6%	17.1%	9.6%
DVERGE+AdvT/3	81.4%	71.4%	59.1%	44.1%	30.4%	19.8%	11.1%	5.5%
DVERGE+AdvT/5	83.4%	73.2%	59.2%	42.2%	28.0%	17.2%	8.3%	3.6%
DVERGE+AdvT/8	85.0%	72.3%	57.8%	40.8%	25.7%	14.8%	6.7%	3.1%

accuracy than black-box transfer accuracy in some cases, e.g., 4.9% > 0.4% for ADP/8 when ϵ is 0.02, which implies ADP might result in obfuscated gradients [40].

In addition, we provide the robustness plots with error bars (indicating standard deviation) computed over three independent runs in Figure 9. DVERGE presents a little higher variation in results, which we suspect is due to the random distillation layer selected in the last training epoch. Refer

to **Appendix C.5** for discussion on the layer effects. However, DVERGE still yields noticeable improvements over other methods across the attack spectrum.

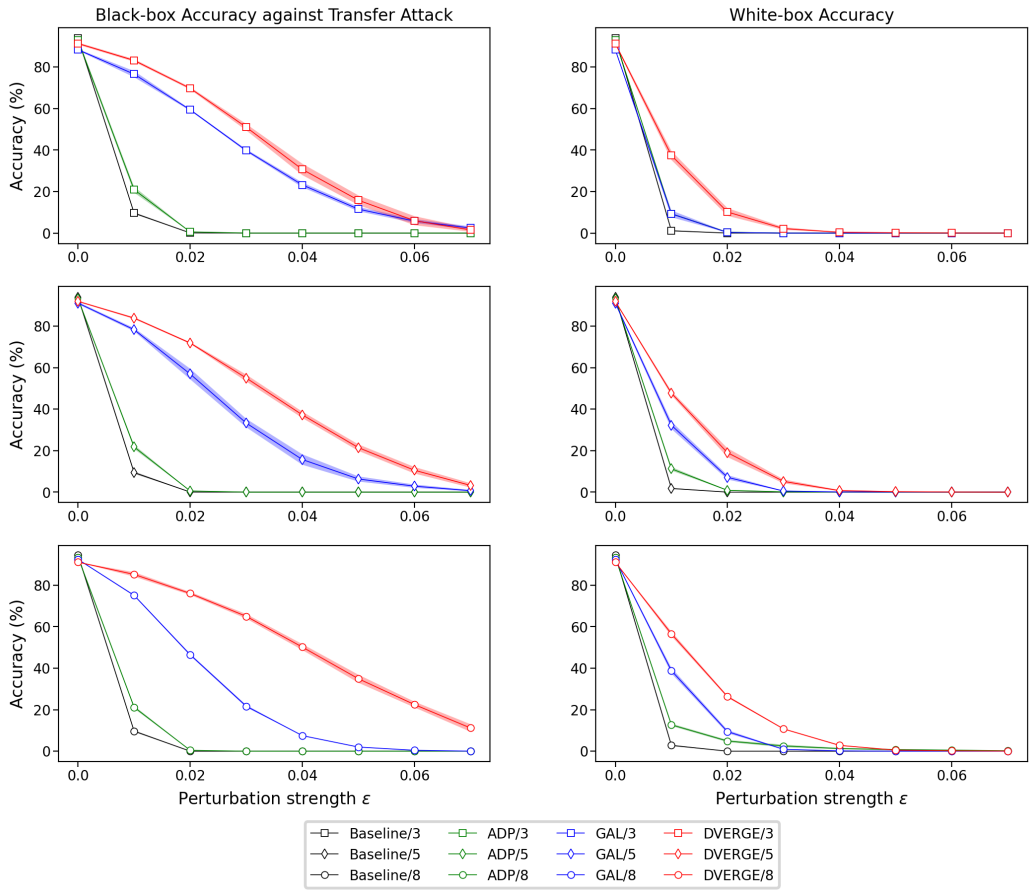

Figure 9: Robustness results with error bars for different ensemble methods. The number after the slash stands for the number of sub-models. From top to bottom, sub-plots on each row report the performance with 3, 5, and 8 sub-models respectively.

A more challenging scenario for adversarial defenses is assuming the attacker is fully aware of the exact defense that the system relies on. In such case, the adversary can train an independent copy of the defended network as the surrogate model. We report black-box transfer accuracy of each method under this setting in Table 6. The same group of attacks as in Section 4.3 are used and ensembles with 3, 5, and 8 sub-models form the collection of surrogate models. According to Table 6, DVERGE still presents the strongest robustness against such a powerful black-box transfer adversary among all ensemble methods.

C.4 Discussion on DVERGE with adversarial training

Formally, the combined training objective of DVERGE and adversarial training is

$$\min_{f_i} \mathbb{E}_{(x,y),(x_s,y_s),l} \left[\underbrace{\lambda \cdot \sum_{j \neq i} \mathcal{L}_{f_i}(x'_{f_j^l}(x, x_s), y_s)}_{\text{DVERGE loss}} + \underbrace{\max_{\delta \in \mathcal{S}} \mathcal{L}_{f_i}(x_s + \delta, y_s)}_{\text{AdvT loss}} \right], \tag{6}$$

where λ is a hyperparameter that balances between the two terms. We set $\lambda = 1.0$ to achieve the results in Figure 5. Results for other ensemble sizes under $\lambda = 1.0$ are shown in Figure 10, where the

Table 6: Accuracy v.s. ϵ against black-box transfer attacks generated from hold-out ensembles that are trained with the exact defense technique used by each ensemble. The number in the first column after the slash is the number of sub-models within the ensemble.

ϵ	clean	0.01	0.02	0.03	0.04	0.05	0.06	0.07
ADP/3 [12]	93.3%	34.7%	6.5%	1.4%	0.2%	0%	0%	0%
ADP/5 [12]	93.1%	34.2%	6.6%	1.6%	0.6%	0%	0%	0%
ADP/8 [12]	93.0%	32.5%	5.6%	1.4%	0.2%	0%	0%	0%
GAL/3 [13]	88.8%	67.8%	36.2%	13.8%	3.7%	0.6%	0.1%	0%
GAL/5 [13]	91.0%	67.5%	31.9%	9.2%	1.8%	0.3%	0%	0%
GAL/8 [13]	92.3%	64.7%	25.8%	5.4%	0.6%	0.1%	0%	0%
DVERGE/3	91.4%	75.4%	50.2%	23.8%	7.6%	2.1%	0.3%	0.2%
DVERGE/5	91.9%	77.2%	53.1%	26.7%	9.5%	2.6%	0.5%	0.2%
DVERGE/8	91.1%	77.7%	57.3%	32.0%	13.9%	3.8%	0.9%	0.2%

standard deviation over three independent runs is also included. The observations stay the same as in Section 4.4. To better reflect the trade-off between clean accuracy and robustness, we also report the results for an ensemble of eight sub-models with $\lambda = 0.5$. In this case, DVERGE loss is weighted less and the training process will favor adversarial training. In turn, the ensemble spends more of its capacity to capture robust features instead of diverse non-robust features. Consequently, compared with $\lambda = 1.0$, we observe a decrease in clean accuracy and an increase in both black-box and white-box robustness when ϵ is large. In addition, the ensemble size is actually another weight factor in Equation (6) as increasing the number of sub-models will naturally lead to larger DVERGE loss such that it outweighs the AdvT loss. As a result, larger (smaller) ensemble sizes for DVERGE+AdvT results in better (worse) clean performance yet worse (better) black-box and white-box robustness under a large ϵ. This assertion can be confirmed by the results in the bottom three rows of Table 4 and Table 5.

C.5 Ablation study on layer selection for distillation

As aforementioned, at each epoch of the DVERGE training, we randomly sample a layer from 20 candidate layers in the ResNet-20s to perform feature distillation based on the intuition that this could help avoid overfitting to any specific layer's feature sets. However, it is natural to wonder what the difference is between using each individual layer. While a rigorous investigation on layers' effect will be an important future direction for this work, here we compare the random layer selection with a straightforward alternative which is using a fixed layer for distillation throughout the training.

Table 7: Black-box robustness (ϵ=0.03) / white-box robustness (ϵ=0.01) of each layer choice. The results for random layer selection are directly taken from Table 4 and 5.

ResBlock 1	ResBlock 2	ResBlock 3	Output Layer	Random
50.7% / 39.3%	50.2% / 36.5%	32.8% / 31.0%	37.3% / 33.5%	51.0% / 37.4%

Specifically, we train another four ensembles by distilling only from the output layer of either ResBlock 1, ResBlock 2, ResBlock 3, or the whole network (correspondingly, the 7-th, 13-th, 19-th, and 20-th layer of the ResNet20). Then we report the black-box and white-box robustness results in Table 7 and present the pair-wise transferability results (evaluated under ϵ=0.01) in Figure 11. It can be seen that while training with deep layers (ResBlock 3 and the output layer) could indeed lead to lower transferability between sub-models, training with shallow layers (ResBlock 1 and 2) can introduce a noticeably higher white-box robustness for each sub-model itself, as shown by the lower diagonal numbers in the first and second heatmap of Figure 11. The fact that some degree of white-box robustness can be achieved with shallow layers is surprising and needs further study, but this observation explains why they lead to stronger overall robustness for the ensemble than deep layers in Table 7. Meanwhile, training with random layers can balance the effect from both shallow and deep layer, achieving both individual robustness improvement and low transferability at the same time. These two factors both contribute to the overall black-box robustness of the ensemble. As shown in Table 7, random layer selection provides superior black-box robustness than other choices,

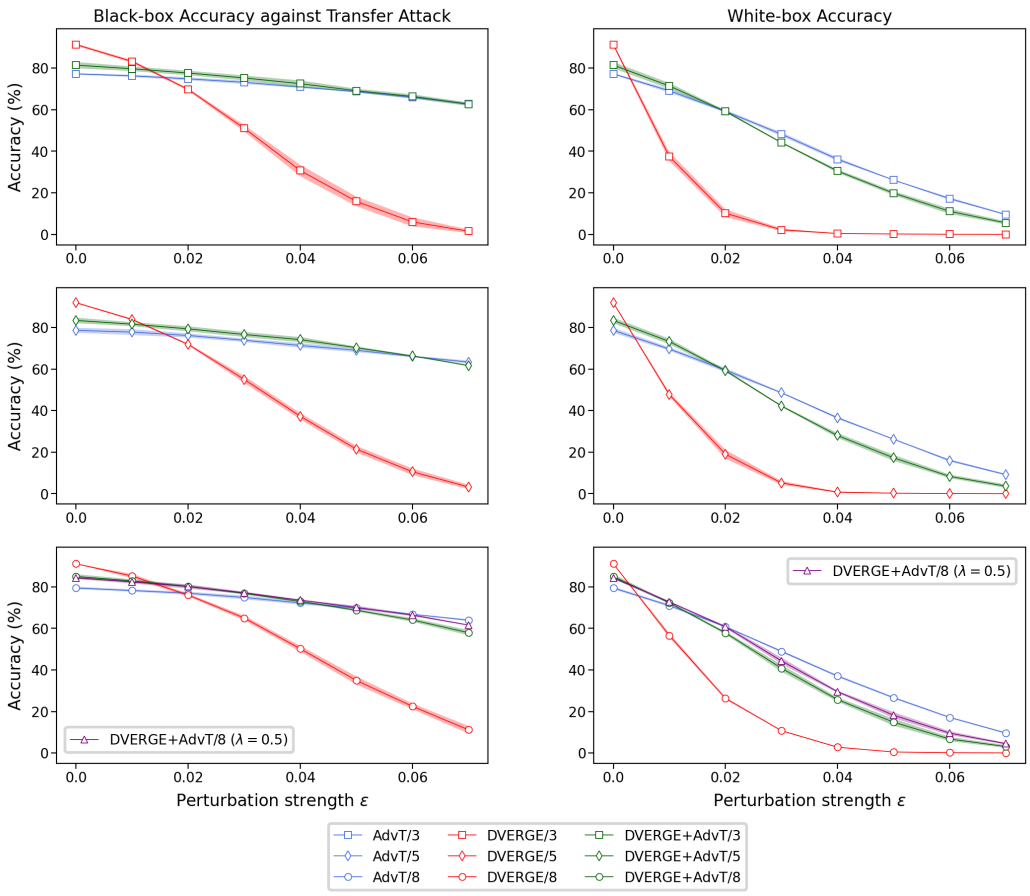

Figure 10: Results for DVERGE combined with adversarial training. From top to bottom, sub-plots on each row report the performance with 3, 5, and 8 sub-models respectively.

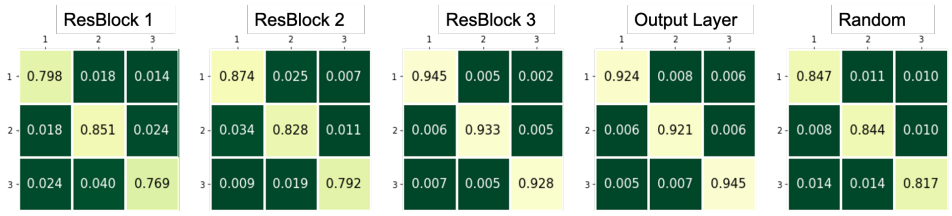

Figure 11: Pair-wise transferability results for different layer selection.

while the achieved white-box robustness is a little lower than the best alternative (ResBlock 1) due to the lack of individual robustness in each sub-model.

C.6 Ablation study on pre-training

As mentioned in Section 3.3, we train DVERGE from a pre-trained ensemble based on the intuition that well-learnt features of the pre-trained models are more informative for distillation and diversification. However, we show in Figure 12 that although slightly worse than using pre-trained models, training from scratch still offers improved robustness over others, implying that pre-training is not strictly necessary for DVERGE.

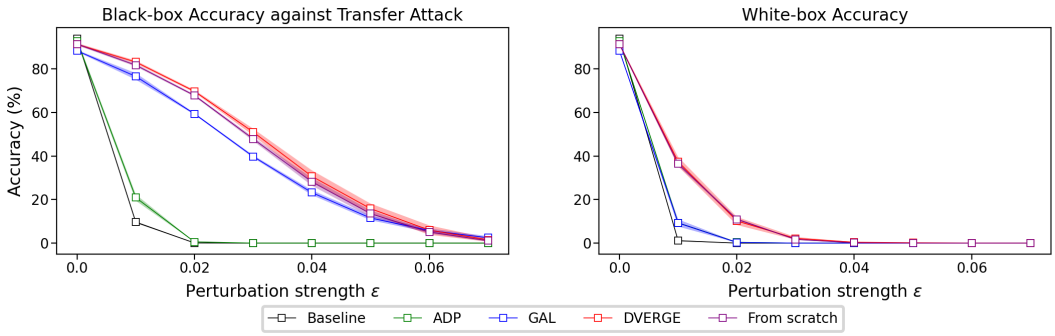

Figure 12: Results of DVERGE with or without pre-training comparing to other methods. All results are evaluated with 3 sub-models. Error bars are evaluated with 3 repeated runs.

D Convergence check

As suggested in [14, 41], we report accuracy vs. the number of attack iterations in Table 8. Note, we use only one random start here for white-box attacks for efficiency. One can observe that using more steps decreases the accuracy by no more than 0.6% for black-box attacks and no more than 1.2% for white-box attacks. Thus, we confirm sufficient steps have been applied and all attacks have converged during the evaluation.

Table 8: Accuracy against attacks with varying number of iterations.

	black-box ($\epsilon = 0.03$)			white-box ($\epsilon = 0.01$)		
	100	500	1000	50	500	1000
DVERGE/3	53.2%	52.6%	53.4%	42.7%	41.6%	41.5%
DVERGE/5	57.2%	56.9%	57.3%	50.4%	49.5%	49.5%
DVERGE/8	63.6%	63.6%	63.2%	58.0%	57.8%	57.8%
DVERGE+AdvT/3	76.2%	76.3%	76.2%	72.9%	72.9%	72.9%
DVERGE+AdvT/5	77.9%	78.0%	77.8%	74.8%	74.8%	74.8%
DVERGE+AdvT/8	77.1%	76.9%	77.4%	72.7%	72.7%	72.7%