Meta Gradient Adversarial Attack
Supplementary Material

Zheng Yuan[1,2], Jie Zhang[1,2], Yunpei Jia[1,2], Chuanqi Tan[3], Tao Xue[3], Shiguang Shan[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3] Tencent
{zheng.yuan, yunpei.jia}@vipl.ict.ac.cn; {zhangjie,sgshan}@ict.ac.cn;
{jamestan,emmaxue}@tencent.com

1. Models

We list all the models used in our experiments here again for more friendly reading. And more details of these models are provided.

Our architecture utilizes a total of 10 white-box models to generate adversarial examples. In each iteration, multiple models are randomly selected to compose a meta-task. Under the scenario of black-box attack, we evaluate 6 and 7 models on ImageNet and CIFAR10, respectively. All the models used in white-box and black-box settings are shown in Tab. 1. For ImageNet, Ens3_Inceptionv3, Ens4_Inceptionv3, Ens_InceptionResNetv2 and Adv_Inceptionv3 are defense models adversarially trained with ensemble adversarial training [19][1]. R&P [22][2], NIPS-r3 [3] and CERTIFY [1] are also defense models. Other models are normally trained on ImageNet and publicly available[4]. For CIFAR10, Adv_ResNet-18, Adv_DenseNet-121 and Adv_GoogLeNet are adversarially trained with FGSM, and Adv_ResNet-18_ll is adversarially trained with LeastLikely [9]. k-WTA [21], GBZ [10] and ADP [13] are also defense models. The rest models are normally trained on CIFAR10.

Table 1: Models on CIFAR10 and ImageNet. The first 10 models are treated as white-box models and the rest models are black-box models in our experimental settings.

	No.	ImageNet	CIFAR10
white-box models	1	Inceptionv3 [18]	ResNet-18 [6]
	2	Inceptionv4 [16]	ResNetv2-18 [7]
	3	InceptionResNetv2 [16]	GoogLeNet [17]
	4	ResNetv2-152 [7]	ResNeXt-29 [24]
	5	Ens3_Inceptionv3 [19]	SENet-18 [8]
	6	Ens4_Inceptionv3 [19]	RegNetX-200mf [14]
	7	Ens_InceptionResNetv2 [19]	DLA [25]
	8	ResNetv2-101 [7]	Shake-ResNet-26_2x64d [4]
	9	MobileNetv2_1.0 [15]	Adv_ResNet-18
	10	PNasNet [12]	Adv_DenseNet-121
black-box models	11	Adv_Inceptionv3 [19]	PyramidNet-164 [5]
	12	NasNet_mobile [26]	CbamResNeXt [20]
	13	MobileNetv2_1.4 [15]	Adv_GoogLeNet
	14	R&P [22]	Adv_ResNet-18_ll
	15	NIPS-r3	k-WTA [21]
	16	CERTIFY [1]	GBZ [10]
	17		ADP [13]

2. Impact of Hyperparameters

In this section, we analyze the influence of the hyperparameters: the number of models n selected for the ensemble-based attacks during the meta-train step and the number of tasks T sampled during the entire generation.

The number of sampled tasks T. Our architecture simulates the process of white-box and black-box attacks iteratively by sampling different model combinations as a meta-task, and the number of sampled tasks T may influence the attack success rate of the generated adversarial examples against the defenses. We compare the attack effects of the generated adversarial examples under white-box and black-box settings when T ranges from 10 to 120 in Tab. 2. It can be seen that the more sampled tasks are taken, the higher attack success rate can be achieved, especially for black-box settings. The reason behind it lies in that more scenarios of white-box and black-box attacks are simulated by sampling more tasks and the gap of white-box and black-box attacks are gradually narrowed, leading to better transferability of adversarial examples against black-box models. However, on the other hand, increasing the number of sampled tasks also increases the time generating adversarial examples. Considering a trade-off of both efficiency and effectiveness, the value of T is recommended to be 40.

The number of ensembled models n in meta-train

[1]https://drive.google.com/drive/folders/10cFNVEhLpCatwECA6SPB-2g0q5zZyfaw
[2]https://github.com/cihangxie/NIPS2017_adv_challenge_defense
[3]https://github.com/anlthms/nips-2017/tree/master/mmd
[4]https://github.com/tensorflow/models/tree/master/research/slim

Table 2: The attack success rates of the adversarial examples generated under **different sampled tasks** T against the white-box and black-box models on ImageNet. The number of iterations K in meta-train step is 2. The number of ensembled models n in meta-train step is 5. The index of models in the table is the same as Tab. 1.

T	white-box models										black-box models					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	98.7	98.6	97.9	97.8	98.4	98.0	96.4	97.0	97.7	98.2	94.5	96.9	96.7	95.6	95.8	66.8
20	99.1	99.2	98.5	98.4	99.1	98.6	97.7	98.3	98.5	98.6	96.1	97.9	97.9	96.8	97.0	68.7
30	99.3	99.1	98.7	98.3	99.1	99.1	98.0	98.4	99.0	99.0	97.1	98.2	98.2	97.1	97.0	69.0
40	99.5	99.5	98.9	98.5	99.2	99.0	98.2	98.6	99.0	98.9	97.4	98.2	98.3	97.7	97.6	70.2
60	99.5	99.5	99.2	98.8	99.2	99.0	98.3	**98.9**	98.9	99.3	97.8	98.5	98.5	98.0	**98.3**	71.1
80	99.5	99.5	**99.3**	**98.9**	99.3	99.1	98.2	98.8	99.1	99.3	97.9	98.4	**98.7**	97.9	97.9	71.3
120	**99.7**	**99.8**	**99.3**	98.8	**99.4**	**99.3**	**98.5**	**98.9**	**99.2**	**99.4**	**98.2**	**98.7**	98.6	**98.1**	**98.3**	**71.6**

step. In the meta-train step, we use an ensemble of multiple models to calculate the gradients and update the adversarial examples. We compare the attack success rates against the white-box and black-box models with an ensemble of different numbers of models n during the update in Tab. 3. It can be seen that, when the number of ensembled models increases, the success rates against the white-box and black-box attacks become higher and higher. But when n is greater than 5, the increase in attack success rates is not obvious, which shows that our architecture is not sensitive to the hyperparameter n to a certain extent. Considering that the more ensembled models in each iteration, the higher the computational complexity is needed. Therefore, it is a suitable choice to set the number of ensembled models to be 5.

3. More Results of the Untargeted Attack

Except the experiments of the untargeted attack with T being 40 illustrated in the manuscript, we also conduct the experiments with T being 10, which is the common setting in MIM [2], DIM [23] and TIM [3]. As shown in Tab. 4, our proposed MGAA also outperforms the state-of-the-art methods under the setting of T being 10. Moreover, when comparing the results of our method under the setting of T being 10 with the baseline methods under the setting of T being 40, we can see that the time cost is nearly equal, but our method still achieves higher attack success rates under both white-box and black-box settings.

4. Attack under Various Perturbation Budgets

We conduct experiments of the attack under various perturbation budgets. The curve of attack success rate vs. perturbation budgets is shown in Fig. 1. The curve with dotted lines are the results of TI-DIM, and the curve with solid lines are the results of our method. We can clearly see that our MGAA consistently achieves higher attack success rates in both white-box and black-box attacks under various perturbation budgets.

5. Ablation Study

We conduct an ablation study to demonstrate the effectiveness of each part in our proposed MGAA architecture. The version of MGAA without meta-test is actually the existing methods like TI-DIM [3], *i.e.*, using an ensemble of multiple models to update the adversarial examples in each step of the iteration. The version of MGAA without meta-train is to use only one randomly selected model to update the adversarial perturbation in each step. From Tab. 5, we can see that the meta-train step plays a more important role than the meta-test step. And the full version of our MGAA architecture achieves higher attack success rates than both meta-train only and meta-test only versions.

6. The Cosine Similarity of the Gradients

We calculate the cosine similarity between the generated adversarial perturbations on ten **white-box models** and the gradient directions of three **black-box models**, *i.e.*, Adv_Inceptionv3, NasNet, and MobileNetv2_1.4. The range of cosine similarity is -1 to 1, and the bigger value means the more similar direction. As shown in Tab. 6, the generated adversarial perturbations by our MGAA are closer to the gradient directions of both three black-box models consistently, which verifies the theoretical analysis in Sec. 3.3 in the paper that MGAA can narrow the gaps of gradient directions between the white-box and black-box models.

7. Minimum Adversarial Noises

We conduct the experiment to see the minimum adversarial noise needed to fool each image. From Tab. 7 we can see that the minimum adversarial noise needed in our MGAA is less than TI-DIM method under various metric evaluations.

Table 3: The attack success rates of the adversarial examples generated under **different number of ensembled models** n in meta-train step against the white-box and black-box models on ImageNet. The number of sampled tasks T is 40. The number of iterations K in meta-train step is 5.

n	white-box models										black-box models					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	**99.9**	99.6	99.7	99.1	99.5	99.6	99.0	99.0	99.4	99.4	98.6	98.8	98.8	98.6	98.6	70.3
4	99.5	99.9	**99.9**	99.4	99.7	99.5	99.1	99.4	99.3	99.6	98.7	99.1	98.9	99.0	98.8	70.6
5	**99.9**	**100**	99.7	99.5	99.8	99.7	98.9	99.5	**99.5**	99.7	98.6	**99.3**	99.1	98.7	98.6	71.3
6	99.8	99.9	99.6	99.5	**99.9**	**99.8**	99.1	**99.6**	99.4	99.7	98.7	99.1	99.2	98.7	98.8	71.6
7	**99.9**	99.9	99.7	99.5	99.7	**99.8**	99.1	99.4	99.4	99.6	**98.9**	**99.3**	99.2	98.7	98.8	**72.2**
8	**99.9**	**100**	99.8	**99.7**	**99.9**	**99.8**	99.1	**99.6**	99.4	99.7	98.8	**99.3**	**99.3**	99.0	**99.0**	71.9
9	**99.9**	**100**	99.8	**99.7**	**99.9**	**99.8**	**99.3**	99.5	99.4	**99.8**	98.6	**99.3**	99.2	**99.1**	**99.0**	72.0

Table 4: The attack success rates of the adversarial examples from our proposed Meta Gradient Adversarial Attack and some state-of-the-art methods against the white-box and black-box models on **ImageNet** under **untargeted attack** setting. The number of ensembled models n in meta-train step is 5. The number of iterations K in meta-train step is 8.

(a) The number of sampled tasks T and the number of iteration in baseline methods are all 10.

Method	white-box models										black-box models						Time
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	(*s*/img)
SI-NI [11]	99.6	96.5	96.1	94.4	98.8	99.1	92.4	93.9	99.0	97.2	43.0	88.3	90.5	49.3	53.1	38.1	22.76
MIM [2]	99.4	99.5	99.2	98.2	99.5	99.8	99.1	98.6	98.7	97.7	44.1	91.5	93.9	66.0	70.3	34.9	6.48
MGAA w/ MIM	**100**	**100**	**100**	**99.8**	**100**	**100**	**100**	**99.5**	**99.8**	**99.3**	**46.5**	**95.3**	**97.0**	**68.2**	**73.9**	**37.4**	22.54
DIM [23]	99.5	99.7	99.4	98.6	99.5	99.6	98.5	98.5	98.8	98.8	78.5	98.1	98.3	95.4	87.7	44.7	8.23
MGAA w/ DIM	**100**	**100**	**100**	**99.6**	**99.9**	**99.8**	**98.9**	**99.6**	**99.7**	**99.6**	**79.9**	**99.3**	**99.4**	**96.5**	**97.3**	**48.7**	22.96
TI-DIM [3]	98.5	98.5	97.3	97.2	98.0	97.7	95.8	97.1	97.1	97.7	93.3	96.3	95.7	95.1	94.9	67.8	7.09
MGAA w/ TI-DIM	**99.8**	**99.9**	**99.7**	**99.4**	**99.7**	**99.5**	**98.9**	**99.3**	**99.3**	**99.3**	**98.4**	**99.0**	**99.0**	**98.6**	**98.7**	**70.5**	21.33

(b) The number of sampled tasks T and the number of iteration in baseline methods are all 40.

Method	white-box models										black-box models						Time
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	(*s*/img)
SI-NI [11]	99.7	97.5	97.4	96.3	98.8	98.6	90.8	95.6	99.7	98.2	48.2	90.8	92.9	50.6	58.5	38.9	68.29
MIM [2]	99.6	99.7	99.4	98.7	99.8	99.8	99.5	99.0	99.1	98.2	44.4	92.6	94.1	65.4	72.2	34.4	17.51
MGAA w/ MIM	**100**	**100**	**100**	**99.9**	**100**	**100**	**100**	**99.9**	**99.9**	**99.9**	**52.0**	**96.0**	**96.9**	**67.1**	**74.9**	**37.0**	71.24
DIM [23]	99.4	99.8	99.5	98.6	99.4	99.5	98.5	98.6	98.9	98.8	79.4	98.0	98.3	95.0	95.3	44.8	22.22
MGAA w/ DIM	**100**	**100**	**100**	**99.9**	**100**	**100**	**99.9**	**99.9**	**100**	**99.9**	**88.0**	**99.9**	**99.8**	**98.9**	**98.9**	**49.3**	69.26
TI-DIM [3]	98.9	99.1	98.2	98.3	98.9	98.6	97.3	98.0	98.1	98.3	96.3	97.5	97.5	96.7	96.8	67.8	19.13
MGAA w/ TI-DIM	**100**	**100**	**99.9**	**99.8**	**99.9**	**99.8**	**99.4**	**99.8**	**99.6**	**100**	**99.1**	**99.4**	**99.5**	**99.4**	**99.0**	**71.6**	67.28

Table 5: The ablation study of different parts in our proposed MGAA architecture.

Setting	white-box models										black-box models					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MGAA w/o meta-train	95.7	93.7	92.4	96.0	94.3	93.1	89.3	96.3	97.9	96.2	81.5	89.4	89.3	81.8	82.6	52.5
MGAA w/o meta-test	98.9	99.1	98.2	98.3	98.9	98.6	97.3	98.0	98.1	98.3	96.3	97.5	97.5	96.7	96.8	67.8
MGAA	**100**	**100**	**99.9**	**99.8**	**99.9**	**99.8**	**99.4**	**99.8**	**99.6**	**100**	**99.1**	**99.4**	**99.5**	**99.4**	**99.0**	**71.4**

Note: MGAA w/o meta-test is actually the same as existing method like TI-DIM [3].

Table 6: The cosine similarity between the generated adversarial perturbations on ten **white-box models** and the gradient directions directly computed on three **black-box models**.

Method	Adv_Inceptionv3	NasNet	MobileNetv2_1.4
TI-DIM	-0.104	-0.070	-0.084
MGAA w/ TI-DIM	0.113	0.071	0.083

Table 7: The mean of minimum adversarial noises needed to fool images on ImageNet.

Method	L_∞	L_1	L_2
TI-DIM	7.4576	5.3112	9.8975
MGAA w/ TI-DIM	4.2960	3.1203	5.8767

References

[1] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. pages 1310–1320. PMLR, 2019. 1

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE Conf. Comput. Vis. Pattern*

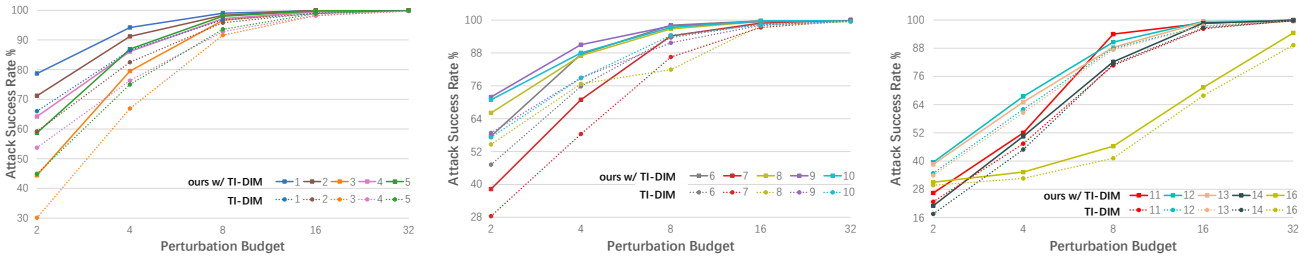

Figure 1: The attack success rates vs. perturbation budget curve on ImageNet. The dotted lines are the results of TI-DIM, and the solid lines are the results of our method. The number in the legend means the index of models in Tab. 1.

Recog., pages 9185–9193, 2018. 2, 3

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4312–4321, 2019. 2, 3, 4

[4] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 1

[5] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5927–5935, 2017. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Eur. Conf. Comput. Vis.*, pages 630–645, 2016. 1

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7132–7141, 2018. 1

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1

[10] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? pages 3804–3814. PMLR, 2019. 1

[11] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *Int. Conf. Learn. Represent.*, 2020. 3

[12] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Eur. Conf. Comput. Vis.*, pages 19–34, 2018. 1

[13] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. pages 4970–4979. PMLR, 2019. 1

[14] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10428–10436, 2020. 1

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4510–4520, 2018. 1

[16] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 1

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–9, 2015. 1

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2818–2826, 2016. 1

[19] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 1

[20] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 1

[21] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. *arXiv preprint arXiv:1905.10510*, 2019. 1

[22] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 1

[23] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2730–2739, 2019. 2, 3

[24] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1492–1500, 2017. 1

[25] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2403–2412, 2018. 1

[26] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8697–8710, 2018. 1