Organization. The appendix is organized as follows: In Section A, we describe the hyperparameters and provide the description about evaluation for multiple perturbations. Furthermore, we provide a breakdown of all the attacks across various datasets. In addition, we include the previous reveiws for our paper and the changes made to the manuscript.

A Experimental setup

A.1 Datasets

1. **CIFAR-10.** This dataset [1] contains 60,000 images with 5,000 images for training and 1,000 images for test for each class. Each image is sized 32×32, we use the Wide ResNet 28-10 architecture [2] as a base network for this dataset.
2. **SVHN.** This dataset [3] contains 73257 training and 26032 testing images of digits and numbers in natural scene images containing ten-digit classes. Each image is sized 32×32, we use the Wide ResNet 28-10 architecture similar to the CIFAR-10 dataset as the base network.
3. **Tiny-ImageNet.** This dataset [1] is a subset of ImageNet [4] dataset, consisting of 500, 50, and 50 images for training, validation, and test dataset, respectively. This dataset contains 64×64 size images from 200 classes, we use ResNet50 [5] as a base network for this dataset.

A.2 Training setup

We use the SGD optimizer with momentum 0.9 and weight decay $5 \cdot 10^{-4}$ to train all our models with cyclic learning rate with a maximum learning rate λ that increases linearly from 0 to λ over first $N/2$ epochs and then decreases linearly from $N/2$ to 0 in the remainder epochs, as recommended by [6] for fast convergence of adversarial training. We train all the models for 30 epochs on a single machine with four GeForce RTX 2080Ti using WideResNet 28-10 architecture [2]. We use the maximum learning rate of $\lambda = 0.21$ for all our experiments. We use $\beta = 16$ for all the experiments with our meta noise generator. The generator is formulated as a convolutional network with four $3{\times}3$ convolutional layers with LeakyReLU activations and one residual connection from input to output. All our algorithms are implemented in Pytorch [7]. We use the weight for the KL divergence ($\beta = 6.0$) for TRADES and RST in all our experiments. We replicate all the baselines on SVHN and TinyImageNet since most of the baseline methods have reported their results on MNIST and CIFAR-10. Unfortunately, we found that MSD [8] did not converge for larger datasets even after our extensive hyperparameter-search. We believe that this is due to the the change in formulation of the inner optimization which leads to a difficulty in convergence for larger datasets. Since the authors also report their results on CIFAR-10, we do not use it as a baseline for other datasets.

A.3 Evaluation setup

For ℓ_∞ perturbations, we use PGD [9], Brendel and Bethge attack [10], and AutoAttack [11]. For ℓ_2 perturbations, we use CarliniWagner attack [12], PGD [9], Brendel and Bethge attack [10], and AutoAttack [11]. For ℓ_1 perturbations, we use SLIDE [13], Salt and pepper [14], and EAD attack [15]. For all our experiments and evaluation, we use $\varepsilon = \{0.03, 8, 0.31\}$ and $\alpha = \{0.004, 1.0, 0.1\}$ for ℓ_∞, ℓ_1, and ℓ_2 attacks for CIFAR-10 and SVHN respectively. For Tiny-ImaegeNet we use $\varepsilon = \{0.01, 8, 0.31\}$ and $\alpha = \{0.004, 1.0, 0.1\}$ for ℓ_∞, ℓ_1, and ℓ_2 attacks respectively. We use 10 steps of PGD attack for ℓ_∞, ℓ_2 during training. For ℓ_1 adversarial training, we use 20 steps during training and 100 steps during evaluation. We use the code provided by the authors for evaluation against AutoAttack [11] and Foolbox [14] library for all the other attacks.

Due to the length limit of our paper, we provide a breakdown of all the attacks on CIFAR-10 in Table A.1, SVHN on Wide ResNet 28-10 in Table A.2, Tiny-ImageNet on ResNet50 in Table A.3. Besides, we analyze the noise learned by our meta-learning framework on multiple datasets and the loss landscape on the CIFAR-10 dataset.

[1] https://tiny-imagenet.herokuapp.com/

Table A.1: Summary of adversarial accuracy results for CIFAR-10 on Wide ResNet 28-10 architecture.

	Adv_∞	Adv_1	Adv_2	Trades_∞	RST_∞	Adv_{avg}	Adv_{max}	MSD	MNG-AC
Clean Accuracy	86.8± 0.1	93.3± 0.6	91.7± 0.2	84.7± 0.3	88.9± 0.2	87.1± 0.2	85.4± 0.3	82.3± 0.2	84.9± 0.3
PGD-ℓ_∞	46.9± 0.5	0.40± 0.7	23.6± 0.2	52.0± 0.6	56.9± 0.1	35.2± 0.8	42.2± 1.1	45.4± 0.4	44.5± 1.1
PGD-Foolbox	54.7± 0.4	0.33± 0.6	35.3± 0.4	57.8± 0.5	62.9± 0.3	45.0± 0.4	50.4± 0.4	51.7± 0.8	50.8± 0.8
AutoAttack	44.9± 0.7	0.0± 0.0	20.7± 0.4	48.8± 1.1	53.9± 0.3	33.8± 0.7	39.9± 0.9	42.7± 0.2	42.8± 0.8
Brendel & Bethge	49.9± 1.1	0.0± 0.0	26.8± 0.3	52.1± 0.7	56.5± 1.8	39.6± 0.7	45.8± 0.9	48.3± 0.4	46.8± 0.9
All ℓ_∞ attacks	44.9± 0.7	0.0± 0.0	20.7± 0.3	48.9± 0.7	54.9± 1.8	33.8± 0.7	39.9± 0.9	43.7± 0.2	42.2± 0.9
PGD-ℓ_1	12.8± 0.6	91.6± 1.4	27.7± 0.7	17.9± 0.6	22.0± 0.5	49.0± 0.3	44.6± 0.2	46.8± 1.4	55.0± 1.2
PGD-Foolbox	35.2± 0.7	92.3± 1.3	53.1± 0.5	40.3± 0.7	44.6± 0.3	64.5± 0.2	60.7± 0.5	60.3± 0.4	65.5± 0.1
EAD	72.9±1.0	87.1± 3.3	75.9± 1.9	80.2± 0.7	84.5± 0.2	85.7± 0.2	83.3± 0.5	80.8± 0.1	79.3± 0.6
SAPA	71.5± 0.2	80.2± 1.8	81.9± 0.5	71.4± 0.7	76.0± 0.5	82.7± 0.1	80.0± 0.1	76.9± 0.5	76.7± 0.4
All ℓ_1 attacks	12.8± 0.6	78.1± 1.8	27.7± 0.7	17.9± 0.6	22.0± 0.5	49.0± 0.3	44.6± 0.2	43.7± 0.2	55.0± 1.2
PGD-ℓ_2	78.7± 0.3	47.6± 1.6	84.6± 0.2	77.0± 0.9	82.2± 0.2	81.5± 0.2	79.1± 0.3	76.5± 0.1	75.6± 0.4
PGD-Foolbox	74.6± 0.2	5.1± 2.1	79.8± 0.2	73.3± 0.6	78.3± 0.2	77.6± 0.2	75.8± 0.3	73.6± 0.5	73.4± 0.1
Gaussian Noise	85.2± 0.4	88.5± 1.8	90.5± 1.1	83.2± 0.3	87.8± 0.2	86.2± 0.5	83.3± 0.3	70.9± 1.1	79.3± 0.1
AutoAttack	69.9± 0.4	0.0± 0.0	76.8± 0.4	69.4± 0.3	73.7± 0.1	74.9± 0.4	73.2± 0.2	71.9± 0.4	71.5± 0.1
Brendel & Bethge	71.8± 0.9	0.0± 0.0	78.1± 0.6	70.2± 0.1	75.0± 0.3	75.9± 0.3	74.1± 0.4	80.4± 0.4	72.3± 0.1
CWL2	70.5± 0.2	0.1± 0.0	77.2± 0.5	69.7± 0.3	74.2± 0.1	74.6± 1.2	73.5± 0.2	71.1± 1.1	71.0± 0.1
All ℓ_2 attacks	69.3± 0.4	0.0± 0.0	76.8± 0.4	69.4± 0.3	73.6± 0.1	74.9± 0.4	73.2± 0.2	70.6± 1.1	71.5± 0.1
$\text{Acc}_{\text{adv}}^{\text{union}}$	12.9± 0.5	0.0± 0.0	17.9± 0.8	17.2± 0.6	21.1± 1.0	31.0± 1.4	35.7± 0.3	35.8± 0.1	**41.6± 0.8**
$\text{Acc}_{\text{adv}}^{\text{avg}}$	42.6± 0.4	25.1± 1.6	47.6± 0.4	45.4± 0.3	50.2± 0.5	52.6± 0.5	52.5± 0.3	52.0± 0.4	**56.2± 0.2**

Table A.2: Summary of adversarial accuracy results for SVHN dataset on Wide ResNet 28-10 architecture.

	Adv_∞	Adv_1	Adv_2	Trades_∞	RST_∞	Adv_{avg}	Adv_{max}	MNG-AC
Clean Accuracy	92.8± 0.1	92.4± 1.6	94.9± 0.0	93.9± 0.0	95.6± 0.0	92.6± 0.1	88.2± 1.6	93.4± 0.0
PGD-ℓ_∞	49.1± 0.1	3.2± 2.4	29.4± 0.1	55.5± 1.4	66.9± 0.8	22.4± 3.1	36.6± 2.0	40.5± 0.1
PGD-Foolbox	60.7± 0.4	2.5± 1.9	47.6± 0.6	66.4± 1.1	73.8± 0.3	32.5± 3.2	49.9± 0.0	57.5± 1.8
AutoAttack	46.2± 0.6	0.0± 0.0	18.9± 0.5	49.9± 1.8	61.0± 2.0	17.6± 2.6	17.5± 0.9	33.7± 0.0
Brendel & Bethge	51.6± 0.7	0.0± 0.0	22.9± 0.8	55.8± 1.5	65.6± 1.2	20.2± 2.9	6.3± 2.3	40.0± 0.3
All ℓ_∞ attacks	46.2± 0.6	0.0± 0.0	18.7± 0.6	49.9± 1.7	60.9± 2.0	17.4± 2.3	5.9± 1.2	35.1± 1.9
PGD-ℓ_1	3.1± 0.3	95.0± 1.8	30.5± 0.4	1.7± 0.3	0.7± 0.6	55.8± 2.1	48.4± 2.9	44.5± 3.2
PGD-Foolbox	19.9± 0.8	94.6± 0.4	57.5± 0.1	15.5± 0.2	11.3± 0.5	79.2± 3.4	85.4± 3.2	75.2± 2.8
EAD	65.7± 2.1	87.8± 1.9	82.3± 1.2	51.5± 2.9	60.4± 0.8	84.8± 2.4	84.5± 3.8	86.2± 2.2
SAPA	79.4± 0.8	77.3± 5.2	87.3± 0.1	73.5± 1.0	86.2± 0.5	88.5± 0.6	80.9± 4.0	89.9± 1.6
All ℓ_1 attacks	3.0± 0.3	77.9± 6.3	30.3± 0.3	1.6± 0.3	0.7± 0.6	54.2± 2.9	48.3± 4.1	47.4± 2.2
PGD-ℓ_2	81.6± 0.5	3.9± 1.4	87.8± 0.2	83.9± 0.8	85.3± 0.2	85.6± 0.6	84.3± 1.1	90.4± 0.6
PGD-Foolbox	73.2± 0.2	1.9± 1.8	82.8± 0.6	75.0± 0.7	76.0± 0.3	80.6± 0.1	60.1± 0.8	86.1± 0.1
Gaussian Noise	92.1± 0.2	16.5± 4.2	94.2± 0.2	93.3± 1.4	94.2± 0.6	92.2± 0.2	83.8± 0.6	93.2± 0.4
AutoAttack	59.0± 0.7	0.0± 0.0	79.3± 0.1	56.4± 1.3	60.7± 0.6	75.6± 0.1	40.0± 2.3	78.0± 0.8
Brendel & Bethge	68.2± 0.5	0.0± 0.0	81.0± 0.1	64.8± 0.9	68.1± 0.5	76.4± 0.4	32.7± 3.8	78.4± 0.4
CWL2	63.5± 0.8	0.1± 0.1	80.1± 1.4	61.4± 0.3	63.9± 0.2	76.8± 0.1	55.3± 5.2	80.9± 0.9
All ℓ_2 attacks	59.2± 0.7	0.0± 0.0	79.3± 0.1	56.0± 1.4	60.6± 0.6	74.7± 0.1	31.0± 5.0	77.6± 1.0
$\text{Acc}_{\text{adv}}^{\text{union}}$	3.0± 0.3	0.0± 0.0	16.4± 0.7	1.6± 0.3	0.7± 0.6	16.6± 1.3	5.8± 1.7	**30.3± 1.8**
$\text{Acc}_{\text{adv}}^{\text{avg}}$	36.2± 0.3	23.9± 2.1	42.8± 0.2	35.8± 0.6	40.7± 0.8	43.0± 1.0	26.7± 2.5	**52.6± 0.5**

Table A.3: Summary of adversarial accuracy results for Tiny-ImageNet on ResNet50 architecture.

	Adv_∞	Adv_1	Adv_2	Trades_∞	Adv_{avg}	Adv_{max}	MNG-AC
Clean Accuracy	54.2± 0.1	57.8± 0.2	59.8± 0.1	48.2± 0.2	56.0± 0.2	53.5± 0.0	53.1± 0.1
PGD-ℓ_∞	32.1± 0.0	11.5± 1.2	17.9± 1.1	32.2± 0.4	25.0± 0.6	32.0± 0.6	29.3± 0.3
PGD-Foolbox	34.6± 0.4	17.2± 0.1	5.2± 0.6	34.1± 0.2	34.0± 0.2	28.3± 0.1	32.3± 0.3
AutoAttack	29.6± 0.1	10.1± 0.7	16.3± 0.3	28.7± 0.9	23.7± 0.2	30.0± 0.1	27.7± 0.4
Brendel & Bethge	32.7± 0.1	14.6± 0.8	20.8± 0.6	31.0± 0.9	28.1± 0.2	33.2± 0.5	31.5± 0.6
All ℓ_∞ attacks	29.6± 0.1	10.5± 0.7	5.2± 0.6	28.7± 0.9	23.7± 0.2	29.8± 0.1	27.4± 0.7
PGD-ℓ_1	32.0± 1.1	39.3± 0.9	37.2± 0.2	31.1± 0.3	38.0± 0.1	33.6± 0.4	39.0± 0.9
PGD-Foolbox	40.0± 0.8	44.8± 0.2	45.2± 0.2	37.6± 0.9	44.7± 1.5	40.6± 0.1	45.0± 0.2
EAD	52.3± 1.5	56.3± 0.6	57.3± 0.0	46.7± 0.9	54.6± 0.9	51.2± 0.2	52.7± 0.3
SAPA	46.5± 0.9	52.9± 0.7	53.5± 1.2	40.8± 0.1	50.3± 1.1	46.6± 0.1	49.3± 0.4
All ℓ_1 attacks	31.8± 1.0	39.3± 1.0	37.2± 0.4	30.9± 0.2	38.0± 0.2	33.4± 0.3	39.6± 0.7
PGD-ℓ_2	48.5± 1.1	49.1± 0.1	51.8± 1.8	42.6± 0.7	49.9± 1.7	47.0± 0.3	49.1± 0.4
PGD-Foolbox	45.6± 0.4	45.2± 0.4	47.7± 0.7	41.0± 0.3	47.0± 1.3	44.9± 0.4	47.0± 0.2
Gaussian Noise	52.5± 1.3	56.1± 0.6	57.6± 0.3	46.4± 0.9	54.4± 0.8	51.1± 0.0	52.1± 0.5
AutoAttack	42.4± 0.8	41.9± 0.0	44.6± 0.6	38.9± 0.8	44.4± 1.3	42.4± 0.9	44.6± 0.4
Brendel & Bethge	43.7± 0.4	44.4± 0.1	46.6± 1.1	39.2± 0.7	45.1± 1.6	43.6± 0.4	45.4± 0.1
CWL2	43.5± 1.3	44.8± 1.1	47.5± 0.7	39.5± 0.4	46.8± 1.9	43.4± 0.1	46.0± 0.4
All ℓ_2 attacks	42.5± 0.6	41.9± 0.0	44.9± 0.1	35.8± 0.7	44.6± 0.1	42.4± 1.0	44.8± 0.1
$\text{Acc}_{\text{adv}}^{\text{union}}$	19.8± 1.1	10.1± 0.7	5.2± 0.6	26.1± 0.9	23.6± 0.3	**29.0± 0.3**	27.4± 0.8
$\text{Acc}_{\text{adv}}^{\text{avg}}$	33.8± 0.1	30.4± 0.1	29.1± 0.0	32.8± 0.1	35.4± 0.7	35.3± 0.4	**37.2± 0.6**

Summary of previous reviews

All the reviewers found our work novel [R2, R3] and tackles an important problem of robustness against multiple perturbations [R2, R3, R4]. Moreover, all reviewers appreciated the significant decrease in computation time for multiple perturbations adversarial training. However, we guess that the notations were unclear, and some crucial experiments were missing in the paper. The paper has now been rewritten, provides exhaustive experiments on benchmark datasets against state-of-the-art adversarial attacks.

Ablation studies [R1, R4]. Following R1, R4 suggestions, we now provide a comparison with Gaussian Noise, MNG without the AC loss on multiple datasets in our ablation studies. We can observe that MNG-AC significantly outperform all the baselines, which is not surprising since learning the noise is more efficient way to improve generalization.

Comparison with baselines [R1, R2]. R1, R2 suggests to compare against Maini et al. [8], however, since it was accepted to ICML 20, we did not have sufficient time to add it as a baseline for the NeurIPS submission. We have now updateed our manuscript and provide a comparison with it in the main paper.

Stronger attacks [R1, R3]. Following R1 and R3]'s suggestion, we show a comparison against Autoattack [11] on all the datasets. Furthermore, we compare against Brendel and Bethge attack [10] for various ℓ_p norms.

Scalability to larger datasets [R3, R4]. Both the reviewers mention that our paper could be made stronger with experiments on the larger datasets. We have now updated our paper and provided experiments on large scale architectures and datasets. Due to the limited time provided for rebuttal, we could not perform ImageNet-scale experiments, but we will perform this experiment and include it in the final version if the paper is accepted.

Incorrect implementation [R2]. R2 raises an issue with the implementation of our max strategy, we have fixed this in our updated manuscript.

Other comments. Reviewers also highlight some inconsistencies with our notations. We have revised our manuscript by clarifying all the notations and inconsistencies.

The changes that have been made to the paper

- We rewrote and revised throughout the paper including texts, notations, and figures to avoid confusion and for readability and clarity.
- We included the experimental results using modern deep neural networks, ResNet18 on CIFAR-10 and SVHN -> Wide ResNet 28-10 on CIFAR-10, SVHN and ResNet50 on Tiny-ImageNet. (From [R3, R4] comments)
- We provide a standard deviation for all the results. (From [R1, R4] comments)
- We include additional state-of-the-art baselines. (From [R1, R2] comments)
- We include ablation studies dissecting various components of our framework. (From [R1, R4] comments)
- We evaluate on state-of-the-art adversarial attacks. (From [R1, R3] comments)

References

[1] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[2] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

[3] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Workshop on Deep Learning and Unsupervised Feature Learning, NeurIPS*, 2011.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[6] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[8] Pratyush Maini, Eric Wong, and J Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *ICML*, 2020.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2017.

[10] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *NeurIPS*, 2019.

[11] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[12] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 2017.

[13] Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *NeurIPS*, 2019.

[14] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, ICML*, 2017.

[15] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018.