Appendix

Organization. The Appendix is organized as follows: In Appendix A, we describe the hyperparameters and provide the description about evaluation for multiple perturbations. In Appendix B, we provide additional experiments including the evaluation on common corruptions and unforseen perturbations, effect of the adversarial consistency regularization on clean accuracy, and an expanded comparison of the baselines with our proposed framework on multiple adversarial perturbations. Furthermore, Appendix B demonstrates the examples generated by our input-dependent meta-noise generator for multiple datasets and visualization of loss landscape on the SVHN dataset.

A. Experimental setup

A.1. 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 (Wong et al., 2020) for fast convergence of adversarial training. We train all the models with 30 epochs on a single machine with four GeForce RTX 2080Ti using WideResNet 28-10 architecture for CIFAR-10, SVHN and ResNet-50 for Tiny-ImageNet. We use the maximum learning rate of $\lambda = 0.21$ for all our experiments. We use $\beta = 12$ (weight for adversarial consistency loss) for the reported results of MNG-AC for all the datasets.

The noise-generator is formulated as a convolutional network with four 3×3 convolutional layers with LeakyReLU activations and one residual connection from input to output following (Rusak et al., 2020). All our algorithms are implemented using Pytorch (Paszke et al., 2019). 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 only on MNIST and CIFAR-10. Moreover, we found that MSD (Maini et al., 2020) and $\text{Adv}_{\max}$ are sensitive to the learning rate on SVHN dataset; therefore, we tune the maximum learning rate and use $\lambda = 0.01$ for these baselines. We believe that this is due to the the change in optimization formulation, which involves optimization on the worst perturbation and leads to this sensitivity for larger datasets.

A.2. Evaluation setup

For CIFAR-10 and SVHN dataset, we use $\varepsilon = \{\frac{8}{255}, \frac{2000}{255}, \frac{128}{255}\}$ and $\alpha = \{0.004, 1.0, 0.1\}$ for ℓ_∞, ℓ_1, and ℓ_2 attacks respectively. For Tiny-ImageNet dataset, we use $\varepsilon = \{\frac{4}{255}, \frac{2000}{255}, \frac{80}{255}\}$ 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 (Croce & Hein, 2020) and Foolbox (Rauber et al., 2017) library for all the other attacks.

B. Additional experimental results

Robustness against common corruprions. Our sampling strategy further allows us to increase our perturbation set, which is limited in previous works due the increased computation cost. Consequently, we evaluate our method on common corruptions perturbation set (Hendrycks & Dietterich, 2019). In particular, we use the validation corruptions provided by the authors (speckle noise, gaussian blur, spatter, and saturate) during training and evaluate on other 15 types of unseen corruptions across five levels of severity. We show the results in Table 6. Note that, while the max and multi-steep descent strategies lead to an increase in the test error, MNG-AC achieves significantly better performance compared to the other multi-perturbation baselines. This demonstrates demonstrate the simplicity and effectiveness of MNG-AC on diverese perturbation sets.

Effect of β on clean accuracy. We further show the effect of β on the clean accuracy in Figure 2. Interestingly, while increasing β improves the robustness against multiple adversarial perturbations, it decreases the clean accuracy. In particular, the absolute performance of $\text{Acc}^{\text{union}}_{\text{adv}}$ improves by $\sim 5\%$, and the clean accuracy drops by $\sim 3\%$ with an increase in the weight of adversarial consistency loss for all the datasets. We report the MNG-AC results with $\beta = 12$ for all our experiments to achieve an optimal trade-off for our proposed method.

Expanded results. Due to the length limit of our paper, we provide a breakdown of all the attacks on CIFAR-10 in Table 8, SVHN on Wide ResNet 28-10 in Table 9, Tiny-ImageNet on ResNet50 in Table 10.

Results on unforseen adversaries. We further evaluate our model on various unforeseen perturbations (Kang et al., 2019) namely we evaluate on the Elastic, ℓ_∞-JPEG, ℓ_1-JPEG and ℓ_2-JPEG attacks. Note that, even though adversarial training methods do not generalize beyond the threat model, we observe that MNG-AC improves the performance on these unseen adversaries. We compare our MNG-AC to the baselines trained with multiple perturbations on the SVHN dataset in Table 7. We notice that even though, $\text{Adv}_{\max}$ achieves better performance on ℓ_p-JPEG attacks, it obtains the minimum robustness across the $\text{Acc}^{\text{union}}_{\text{adv}}$ metric. In contrast, MNG-AC generalizes better over both the baselines for the worst-attack and shows a relative gain of $+3.5\%$ over the best performing baseline.

Table 6. Average test error (%) of different corruptions on CIFAR-10 dataset on WideResNet 28-10 over 5 levels of severities of common corruptions. We report the results averaged across 3 runs and five levels of severity for each corruption.

		Noise			Blur				Weather				Digital			
Model	All	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Adv_∞	22.7	17.1	16.3	23.8	18.8	20.0	22.8	19.4	19.5	22.9	38.1	16.5	54.1	19.8	16.1	16.1
Adv_1	16.6	20.0	17.3	18.1	16.0	18.5	20.7	17.9	13.4	12.6	17.7	8.2	27.6	15.0	12.7	12.3
Adv_2	18.7	13.5	12.8	16.7	15.7	16.9	19.0	16.7	17.0	17.6	34.8	12.6	45.0	16.7	12.9	12.7
Adv_{avg}	21.8	16.5	15.8	16.6	18.8	19.0	21.9	19.8	20.3	22.0	38.5	16.2	49.5	19.9	16.2	15.6
Adv_{max}	23.8	18.2	17.5	18.3	20.4	21.0	23.5	21.4	22.0	25.0	39.5	18.6	53.0	21.6	18.2	17.6
MSD	25.1	19.9	19.1	19.9	21.4	22.1	24.4	22.3	23.7	27.3	40.0	20.5	53.9	22.8	19.4	19.0
MNG-AC	16.6	15.3	13.8	16.4	13.2	22.0	17.0	14.5	16.2	17.3	18.6	12.7	27.1	15.8	15.7	14.2

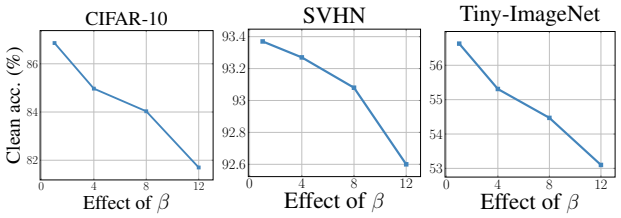

Figure 4. Ablation study on the impact of $\mathcal{L}_{\text{ac}}$ on clean accuracy on various datasets. With an increase in β, the clean accuracy decreases, due to the inherent accuracy-robustness on all the datasets.

Table 7. Performance of MNG-AC against unforseen adversaries on SVHN dataset.

Model	Elastic	ℓ_∞-JPEG	ℓ_1-JPEG	ℓ_2-JPEG	$\text{Acc}_{\text{adv}}^{\text{union}}$
Adv_{avg}	77.1± 1.1	86.6± 0.28	81.5± 2.1	81.2± 1.7	62.1 ± 0.5
Adv_{max}	60.2± 2.3	**89.9± 1.9**	**87.9± 2.1**	**87.0± 2.5**	58.5± 1.5
MNG-AC	**79.6± 1.7**	87.5± 0.9	75.9± 0.2	81.4± 1.2	**64.3± 0.5**

Visualization of generated examples. We visualize the generated examples by our generator during training by randomly selecting samples projected on various ℓ_p norms and datasets in Figure 5. From the figure, we can observe that our meta-noise generator incorporates the features by different attacks and learns diverse input-dependent noise distributions across multiple adversarial perturbations by explicitly minimizing the adversarial loss across multiple perturbations during meta-training. Overall, it combines two complementary approaches and leads to a novel input-dependent learner for generalization across diverse attacks.

Visuaization of loss landscape on SVHN dataset. Figure 6 shows the visualization of loss landscape of various methods against ℓ_∞, ℓ_1, and ℓ_2 norm attack for SVHN dataset on Wide ResNet 28-10 architecture. Similar to the CIFAR-10 dataset, we can observe that the loss is highly curved for multiple perturbations in the vicinity of the data point x for the adversarial training trained with a single perturbation, which reflects that the gradient poorly models the global landscape. In contrast, MNG-AC achieves smoother loss surface across all types of ℓ_p norm attacks.

Table 8. Summary of adversarial accuracy results for CIFAR-10 on Wide ResNet 28-10 architecture. The best and second-best results are highlighted in bold and underline respectively.

	Adv_∞	Adv_1	Adv_2	$Trades_\infty$	RST_∞	Adv_{avg}	Adv_{max}	MSD	MNG-AC	MNG-AC + RST
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	88.7± 0.2
PGD-ℓ_∞	46.9± 0.5	0.40± 0.7	30.4± 1.4	52.0± 0.6	56.9± 0.1	35.7± 0.5	42.5± 0.4	46.3± 0.6	45.4± 0.8	52.8± 0.9
PGD-Foolbox	54.7± 0.4	0.33± 0.6	40.9± 0.9	57.8± 0.5	62.9± 0.3	45.0± 0.4	50.4± 0.4	52.9± 0.8	52.1± 0.6	59.0± 0.7
AutoAttack	44.9± 0.7	0.0± 0.0	28.8± 1.3	48.9± 0.9	54.9± 0.3	34.2± 0.5	39.9± 0.5	43.5± 0.5	41.4± 0.7	47.2± 0.8
Brendel & Bethge	49.9± 1.1	0.0± 0.0	35.4± 1.0	52.1± 0.7	56.5± 1.8	40.2± 1.5	45.5± 0.9	49.3± 1.1	47.0± 0.9	53.4± 0.8
All ℓ_∞ attacks	44.9± 0.7	0.0± 0.0	28.8± 1.3	48.9± 0.7	54.9± 1.8	34.1± 0.5	39.9± 0.5	43.5± 0.5	41.4± 0.7	47.2± 0.8
PGD-ℓ_1	26.4± 0.5	93.9± 0.6	54.4± 0.6	32.4± 1.0	36.2± 0.6	61.4± 0.6	57.9± 0.6	54.4± 0.7	65.4± 0.2	73.8± 0.2
PGD-Foolbox	35.2± 0.7	92.3± 1.3	54.2± 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	74.9± 0.7
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	88.4± 0.5
SAPA	71.5± 0.2	80.7± 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	85.4± 0.3
All ℓ_1 attacks	26.2± 0.4	80.7± 0.7	54.2± 0.4	32.3± 1.0	36.0± 0.9	61.3± 0.6	57.9± 0.7	54.3± 0.4	65.4± 0.3	73.8± 0.7
PGD-ℓ_2	57.1± 0.4	3.0± 0.9	66.2± 0.2	60.8± 0.8	62.4± 0.2	66.5± 0.4	66.4± 0.2	65.0± 0.2	67.2± 0.2	76.7± 0.7
PGD-Foolbox	65.9± 0.7	3.4± 1.9	72.0± 0.4	66.2± 0.6	70.8± 0.3	70.1± 0.1	69.7± 0.7	68.6± 0.2	70.9± 0.3	79.0± 0.3
Gaussian Noise	84.6± 0.5	81.0± 2.3	88.5± 0.4	82.4± 0.6	87.4± 0.2	84.5± 0.6	81.9± 0.4	80.7± 0.8	79.9± 0.3	87.7± 0.4
AutoAttack	55.1± 0.8	0.0± 0.0	65.8± 0.3	57.8± 0.6	59.8± 0.2	65.7± 0.4	64.5± 0.1	63.1± 0.5	65.2± 0.5	73.7± 0.2
Brendel & Bethge	59.6± 1.2	28.1± 0.3	68.6± 0.2	60.9± 1.0	62.9± 0.7	67.5± 0.2	66.6± 0.2	66.4± 0.3	66.6± 0.7	75.3± 0.2
CWL2	57.5± 0.9	0.1± 0.0	66.7± 0.3	59.3± 0.4	60.9± 0.3	66.8± 0.2	65.4± 0.3	64.1± 0.3	66.5± 0.6	74.2± 0.5
All ℓ_2 attacks	55.0± 0.9	0.0± 0.0	65.8± 0.3	57.8± 0.6	59.5± 0.2	65.7± 0.4	64.5± 0.1	63.1± 0.5	65.2± 0.5	73.7± 0.2
Acc_{adv}^{union}	25.6± 0.6	0.0± 0.0	28.6± 1.4	31.5± 1.2	35.7± 0.6	34.1± 0.1	39.7± 0.5	<u>42.7± 0.5</u>	41.4± 0.7	**47.2± 0.7**
Acc_{adv}^{avg}	41.9± 0.6	26.8± 0.6	49.6± 0.3	46.3± 0.7	50.1± 0.8	53.7± 0.3	54.1± 0.4	53.6± 0.2	<u>57.2± 0.4</u>	**64.9± 0.3**

Table 9. Summary of adversarial accuracy results for SVHN dataset on Wide ResNet 28-10 architecture.

	Adv_∞	Adv_1	Adv_2	$Trades_\infty$	RST_∞	Adv_{avg}	Adv_{max}	MSD	MNG-AC	MNG-AC + RST
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	86.9± 0.3	81.8± 0.3	93.4± 0.0	96.3± 0.3
PGD-ℓ_∞	49.1± 0.1	3.2± 2.4	29.3± 0.4	55.5± 1.4	66.9± 0.8	24.9± 2.7	32.7± 0.6	39.7± 0.7	42.6± 0.5	58.0± 1.4
PGD-Foolbox	60.7± 0.4	2.5± 1.9	43.2± 1.3	66.4± 1.1	73.8± 0.3	37.1± 3.1	45.6± 0.2	48.5± 0.2	56.1± 0.9	66.8± 0.6
AutoAttack	46.2± 0.6	0.0± 0.0	21.8± 0.3	49.9± 1.8	61.0± 2.0	21.5± 2.8	28.8± 0.2	34.1± 0.6	34.2± 1.0	43.8± 1.5
Brendel & Bethge	51.6± 0.7	0.0± 0.0	26.5± 0.9	55.8± 1.5	65.6± 1.2	24.5± 2.9	36.4± 0.4	41.7± 0.2	42.1± 1.9	50.7± 0.9
All ℓ_∞ attacks	46.2± 0.6	0.0± 0.0	21.7± 0.4	49.9± 1.7	60.9± 2.0	21.5± 2.7	28.8± 0.2	34.1± 0.6	34.2± 1.0	43.8± 1.5
PGD-ℓ_1	10.0± 0.3	97.5± 1.3	45.2± 0.3	4.8± 0.4	3.6± 0.4	62.3± 3.9	48.9± 0.9	43.4± 0.5	71.6± 2.0	78.9± 2.0
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	52.8± 0.2	48.2± 0.2	73.3± 0.7	82.0± 0.3
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	85.7± 0.3	81.1± 0.2	92.1± 2.2	95.8± 0.3
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	81.4± 0.2	75.6± 0.3	89.9± 1.6	94.1± 0.2
All ℓ_1 attacks	8.2± 0.9	77.2± 2.9	44.7± 0.5	4.2± 0.4	3.5± 0.5	61.2± 4.1	48.9± 0.9	43.4± 0.5	71.3± 1.7	78.9± 2.0
PGD-ℓ_2	36.3± 0.9	3.4± 1.4	63.6± 0.5	34.4± 2.0	35.2± 0.7	60.5± 0.2	78.5± 0.4	73.0± 0.3	72.3± 0.3	90.6± 0.4
PGD-Foolbox	55.7± 0.1	4.2± 1.8	72.3± 0.9	56.0± 0.2	56.7± 1.0	70.3± 0.6	78.5± 0.4	73.0± 0.3	77.3± 0.2	83.3± 0.2
Gaussian Noise	91.8± 0.1	69.3± 2.5	91.8± 0.2	93.5± 0.3	92.5± 0.4	90.7± 0.9	86.3± 0.5	80.6± 0.7	92.0± 0.4	94.0± 0.5
AutoAttack	30.2± 0.5	0.0± 0.0	62.9± 0.2	28.0± 2.2	38.9± 0.8	58.0± 1.7	56.3± 0.8	54.1± 0.1	66.7± 0.9	72.6± 0.1
Brendel & Bethge	41.8± 0.8	0.0± 0.0	67.0± 0.9	39.9± 1.3	47.8± 0.6	61.4± 2.3	60.9± 0.9	57.9± 0.8	71.2± 1.0	78.0± 0.3
CWL2	39.4± 0.3	0.0± 0.0	54.8± 0.2	35.4± 1.9	45.0± 0.5	61.5± 0.6	57.8± 1.2	55.2± 0.4	69.2± 0.9	74.2± 0.5
All ℓ_2 attacks	30.2± 0.5	0.0± 0.0	62.9± 0.2	26.7± 2.0	28.8± 0.9	56.1± 2.3	56.3± 0.8	54.1± 0.1	66.7± 0.9	72.6± 0.2
Acc_{adv}^{union}	8.1± 0.9	0.0± 0.0	21.0± 0.4	4.1± 0.4	3.5± 0.5	20.4± 2.7	28.8± 0.2	34.1± 0.6	<u>34.2± 1.0</u>	**43.8± 1.5**
Acc_{adv}^{avg}	28.3± 0.1	25.7± 1.0	43.1± 0.3	26.9± 1.1	31.1± 0.6	45.9± 0.9	44.7± 0.4	44.0± 0.1	<u>57.4± 0.4</u>	**65.1± 0.3**

Table 10. Summary of adversarial accuracy results for Tiny-ImageNet on ResNet50 architecture.

	Adv_∞	Adv_1	Adv_2	Trades_∞	Adv_{avg}	Adv_{max}	MSD	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	45.5± 0.1	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	32.2± 0.8	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	29.8± 0.1	33.9± 0.1	32.3± 0.3
AutoAttack	29.6± 0.1	10.5± 0.7	16.3± 0.3	28.7± 0.9	23.7± 0.2	30.0± 0.1	29.4± 0.3	28.1± 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	32.8± 0.1	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	29.4± 0.3	28.1± 0.7
PGD-ℓ_1	38.7± 0.6	44.6± 0.1	44.9± 1.1	36.9± 0.5	44.3± 0.1	39.9± 0.4	35.3± 0.8	45.1± 0.5
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	37.3± 0.3	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	45.5± 0.1	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	40.2± 0.1	49.3± 0.4
All ℓ_1 attacks	38.2± 0.7	44.6± 0.1	44.1± 0.4	33.2± 0.2	43.3± 0.2	39.5± 0.4	35.3± 0.8	45.1± 0.5
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	41.1± 0.1	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	41.0± 0.6	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	44.2± 0.2	52.1± 0.5
AutoAttack	42.5± 0.8	41.9± 0.0	44.9± 0.6	38.9± 0.8	44.6± 1.3	42.4± 0.9	33.9± 0.8	44.4± 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	39.2± 1.1	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	34.3± 0.8	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	33.9± 0.8	44.4± 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**	29.4± 0.3	28.1± 0.8
$\text{Acc}_{\text{adv}}^{\text{avg}}$	36.7± 0.4	32.2± 0.4	31.7± 0.5	32.8± 0.1	37.2± 0.2	33.5± 0.6	33.5± 0.6	**39.1± 0.6**

Figure 5. Visualization of the generated examples by MNG-AC along projected on ℓ_1, ℓ_2, and ℓ_∞-norm ball for CIFAR-10 and SVHN dataset.

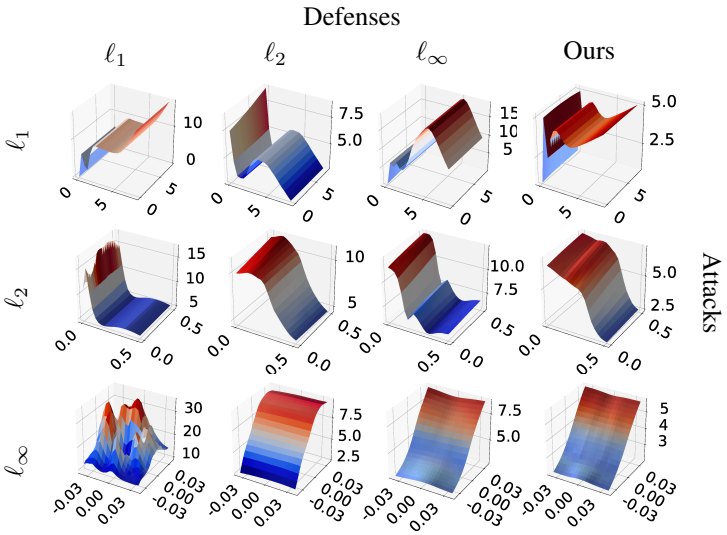

Figure 6. Visualization of the loss landscapes for the ℓ_1, ℓ_2, and ℓ_∞-norm attacks on the SVHN dataset. The rows represent the attacks and columns represent different defenses. We can observe that that MNG-AC obtains smooth loss surface across all ℓ_p-norm attacks.