DASH: A Meta-Attack Framework for Synthesizing Effective and Stealthy Adversarial Examples (Supplementary)

A. Appendix

Below, we provide additional details on the base attacks, defense strategies, and evaluation metrics used in our study (Appendix B). We then present the hyperparameters for each base attack and detail how they are selected across different stages of **DASH** (Appendix C). The following sections demonstrate the extension of the **DASH** framework to black-box settings (Appendix D) and examine the performance of **DASH** when optimizing with various base attacks and imperceptibility metrics (Appendix E).

B. Additional Background

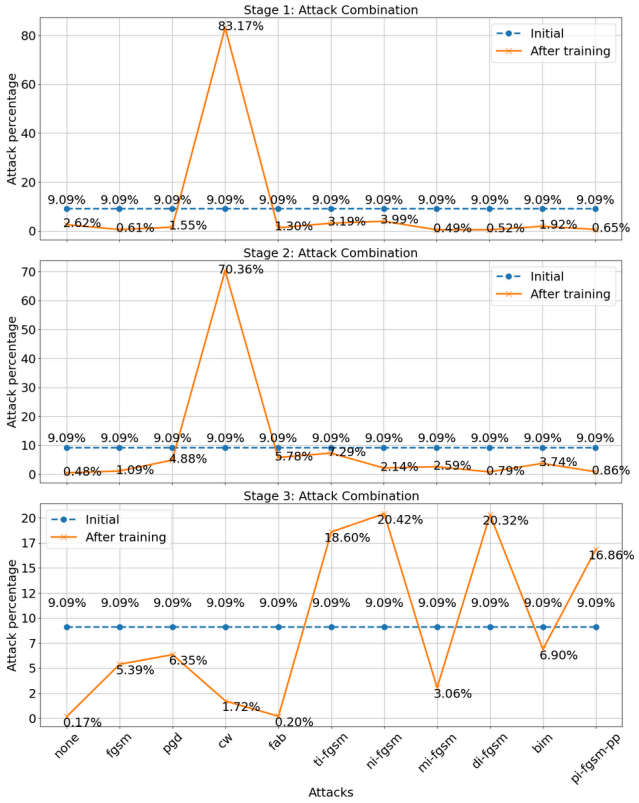

Figure 1. Learned **DASH** weights (represented as % of contribution) for different base attacks across stages.

B.1. Norm-Constrained Attacks

Classical adversarial methods constrain perturbations using ℓ_p norms by either enforcing them directly [21] or indirectly [5].

PGD Attack The PGD attack seeks an adversarial example $\boldsymbol{x}_a$ such that $\|\boldsymbol{x}_a - \boldsymbol{x}\|_\infty \leq \epsilon$, where ϵ denotes the *hard* norm constraint and we denote the norm-constraint as B_ϵ. and $\boldsymbol{x}_a$ maximizes a loss $\mathcal{L}$ such as the cross-entropy loss. The update process is iterative. The first iteration is initialized as $\boldsymbol{x}_a^{(0)} = x + \delta_0, \;\; \delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)$ and is later updated in step t as

$$\boldsymbol{x}_a = \Pi_{B_\epsilon}\left(\boldsymbol{x}_a^{(t)} + \alpha \cdot \text{sign}\left(\nabla_x \mathcal{L}(\boldsymbol{x}_a^{(t)}, y; \theta)\right)\right),$$

where α denotes the step size in each iteration, Π denotes the projection operation onto the ball B_ϵ, centered at $\boldsymbol{x}_a$. The attack runs a total of T iterations. FGSM is a single step update:

$$\boldsymbol{x}_a = \Pi_{B_\epsilon}\left(\boldsymbol{x} + \epsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(\boldsymbol{x}_a^{(t)}, y; \theta)\right)\right).$$

Both the FGSM and PGD attack can be further enhanced, mostly for the purpose of improved transferability to unknown models without white-box access, by providing enhanced inputs to the optimization process [14, 26].

CW Attack The Carlini & Wagner (CW) [5] attack enforces the norm constraints softly and formulates adversarial example generation as the following optimization without explicit projections:

$$\min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_2^2 + c \cdot \max(0, Z(\boldsymbol{x}+\delta)_y - \max_{i \neq y} Z(\boldsymbol{x}+\boldsymbol{\delta})_i + \kappa) \quad (1)$$

where $Z(x)$ represents pre-softmax logits, c balances the two objectives, and κ controls confidence margins, which is usually set 0 for common adversarial example generation.

B.2. Perceptually-Informed Attack Design

Moving beyond norm-constrained attacks, recent methods incorporate perceptual metrics directly into optimization objectives to improve their alignments with human vision.

Table 1. Performance of **DASH** with Black-Box attacks for CIFAR-10 Dataset.

Model	Attack Method	ASR (↑)						SSIM (↑)	LPIPS (↓)	FID (↓)
		Base	JPEG	TVM	Ensemble	NRP	Avg.			
ResNet-18 CIFAR-10	Sign Flip	95.27 ± 0.92	26.68 ± 1.13	75.04 ± 2.27	28.48 ± 0.47	26.83 ± 0.91	50.46 ± 0.64	93.88 ± 0.15	**0.0025 ± 0.0002**	57.58 ± 1.38
	Rays	94.80 ± 0.65	54.28 ± 1.29	51.25 ± 1.55	43.30 ± 0.84	29.43 ± 0.99	54.61 ± 0.56	94.81 ± 0.20	0.0109 ± 0.0005	30.09 ± 0.48
	Square	89.16 ± 0.92	55.43 ± 1.36	82.03 ± 0.64	46.21 ± 0.85	29.57 ± 1.54	60.48 ± 0.80	95.19 ± 0.07	0.0160 ± 0.0005	34.45 ± 0.74
	Sign Hunter	85.00 ± 1.65	**58.95 ± 1.03**	77.93 ± 1.32	**50.47 ± 0.84**	30.06 ± 1.15	60.48 ± 0.73	94.81 ± 0.15	0.0175 ± 0.0002	34.35 ± 0.39
	DASH	**97.79 ± 0.34**	50.74 ± 0.86	**88.50 ± 0.75**	42.38 ± 1.56	**32.99 ± 1.88**	**62.48 ± 0.76**	**95.85 ± 0.07**	0.0107 ± 0.0001	**30.10 ± 0.43**
ViT (base) CIFAR-10	Sign Flip	94.43 ± 0.88	34.98 ± 0.90	72.56 ± 0.77	30.37 ± 1.23	34.04 ± 1.30	53.28 ± 0.53	92.73 ± 0.13	**0.0034 ± 0.0001**	63.57 ± 2.24
	Rays	90.51 ± 1.10	63.20 ± 1.30	61.39 ± 1.23	53.85 ± 1.85	40.18 ± 1.40	61.82 ± 0.43	95.15 ± 0.18	0.0271 ± 0.0013	32.15 ± 1.12
	Square	93.89 ± 0.79	63.54 ± 2.41	85.88 ± 0.20	46.56 ± 1.84	46.52 ± 0.55	67.28 ± 0.92	96.50 ± 0.09	0.0105 ± 0.0004	31.35 ± 0.67
	Sign Hunter	95.86 ± 0.51	**70.06 ± 0.82**	90.02 ± 1.04	**59.45 ± 1.12**	39.41 ± 1.56	70.96 ± 0.54	96.19 ± 0.10	0.0177 ± 0.0003	28.70 ± 0.64
	DASH	**98.20 ± 0.36**	67.49 ± 2.28	**91.46 ± 0.91**	49.64 ± 0.76	**48.73 ± 1.27**	**71.11 ± 0.58**	**97.55 ± 0.07**	0.0081 ± 0.0001	**24.10 ± 0.43**

Table 2. Performance comparison of **DASH** with varying base attacks for CIFAR-100 Cui2024 [9] model.

Base Attacks	ASR (↑)						SSIM (↑)	LPIPS (↓)	FID (↓)
	W/O Def.	JPEG	TVM	Ensemble	NRP	Avg			
PGD, MI-FGSM, CW	99.90	97.75	99.90	95.90	97.36	98.16	90.75	0.0191	60.31
NI-FGSM, CW, FGSM, DI-FGSM, None	99.41	94.82	99.32	93.16	94.73	96.29	93.33	0.0146	49.61
TI-FGSM, PGD, PI-FGSM++, CW, MI-FGSM, FGSM, None	100.00	99.90	100.00	99.51	99.71	99.82	93.90	0.0133	45.73
CW, BIM, NI-FGSM, PGD, DI-FGSM, TI-FGSM, PI-FGSM++, FGSM, MI-FGSM	97.75	94.34	97.46	92.77	94.82	95.43	92.35	0.0193	51.24
None, FGSM, PGD, CW, FAB, TI-FGSM, NI-FGSM, MI-FGSM, DI-FGSM, BIM, PI-FGSM++	100.00	99.90	100.00	99.32	99.61	99.77	94.43	0.0139	41.08

B.2.1. Frequency-Domain Approaches

Recognizing human visual system characteristics, frequency-based methods constrain perturbations to less perceptible components. Semantic Similarity Attack on High-frequency (SSAH) [20] utilizes discrete wavelet transforms to penalize low-frequency modifications. AdvDrop [13] applies perturbations in DCT space, dropping high-frequency components that humans cannot perceive.

B.2.2. Perceptual Distance Integration

PerC-AL [29] incorporates perceptual color distance to measure the perturbation magnitude and adds this metric as an additional term in the optimization process:

$$\mathcal{L}_{per} = \sum_{i,j} |\Delta E_{ij}(x, x_{adv})| \tag{2}$$

where ΔE_{ij} computes the International Commission on Illumination (CIE) color differences.

B.3. Evaluation Metrics Beyond ℓ_p Norms

LPIPS Learned Perceptual Image Patch Similarity (LPIPS) [28] compares two images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ by measuring the distance between their deep feature representations extracted from a fixed CNN (e.g. AlexNet, VGG16, SqueezeNet) and linearly calibrating those distances with learned weights.

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| \mathbf{w}_l \odot \left(\hat{y}^l_{1;h,w} - \hat{y}^l_{2;h,w} \right) \right\|_2^2 \tag{3}$$

where, $d(\mathbf{x}_1, \mathbf{x}_2)$ represents the distance between the first image $\mathbf{x}_1$ and the second image $\mathbf{x}_2$ in the embedding space. The index l denotes the layer in the network, while H_l and W_l are the height and width of the feature map at that layer. The spatial indices h and w iterate over the positions in the feature map. $\hat{y}^l_{1;h,w}$ and $\hat{y}^l_{2;h,w}$ denote the feature activations of the first and second images, respectively, at layer l and spatial location (h, w). $\mathbf{w}_l$ is a channel-wise weight vector or importance mask applied to the features, and $\odot$ indicates the element-wise (Hadamard) product. The expression $\| \cdot$

$\|_2^2$ denotes the squared ℓ_2 norm, measuring the weighted difference between the two feature representations.

FID. Fréchet Inception Distance (FID) [17] measures the distributional differences between real and generated images in the feature space of a pretrained network (typically Inception-v3). Let $\mathcal{X}_r$ be the set of real images and $\mathcal{X}_g$ the set of generated images. The FID score is computed as:

$$\mathrm{FID}(\mathcal{X}_r, \mathcal{X}_g) = \left\|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\right\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2\left(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g\right)^{\frac{1}{2}}\right) \tag{4}$$

Here, $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}_r$ are the mean and covariance of the real images' feature embeddings, while $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are those of the generated images. The operator $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix.

B.4. Defense Strategies

Below, we provide a brief overview of the defense mechanisms evaluated against our **DASH** framework, categorized into robust models and post-processing defenses.

B.4.1. Robust Models

Robust models are trained to be inherently resilient to adversarial examples. Madry et al. [22] first proposed *Adversarial Training*, which injects adversarially perturbed inputs (with correct labels), typically generated via PGD attacks, into the training process to enhance robustness. TRADES [27] improves upon adversarial training by replacing the standard cross-entropy loss with a Kullback–Leibler (KL) divergence-based loss that better balances accuracy and robustness.

Subsequent methods further refine these ideas, building on adversarial training and TRADES. For example, Decoupled Adversarial Learning [9], Better Adversarial Training [25], Adversarial Weight Perturbation [4], and Efficient Robust Training [1] introduce modifications to improve training efficiency, robustness, or generalization.

B.4.2. Post-Processing Defenses

Post-processing defenses aim to increase the robustness of a given model (standard or robust) by transforming or purifying inputs at inference time.

JPEG Compression [15] applies a lossy compress–decompress operation before classification, which removes high-frequency components where adversarial noise typically resides. Total Variation Minimization (TVM) [24] similarly aims to suppress high-frequency perturbations through optimization-based denoising.

Neural Representation Purifier (NRP) [23] is a more advanced approach that learns a purification network. Given a potentially adversarial image, this network iteratively adjusts pixels until a frozen reference model confidently recognizes the input as a clean example. NRP has demonstrated strong empirical performance and is widely adopted in recent adversarial defense research.

C. Base Attack Parameters and Their Selection Across Stages

For the base attacks, we use Torchattacks [18], a PyTorch-based library that provides implementations of various adversarial attacks. The parameters for each attack used in our experiments are detailed below.

For ℓ_∞-norm attacks, we set the perturbation budget to $\epsilon = \frac{7}{255}$. For different variants of FGSM evaluated in this paper, we use a momentum factor of 1. In TI-FGSM [11] and DI-FGSM [26], we set the `resize_rate` to 0.9, `diversity_prob` to 0.7, and `random_start` to `False`. In PI-FGSM++ [14], we set the probability of using diverse inputs to 0.7 and the `project_factor` to 0.8.

For the Fast Adaptive Boundary (FAB) attack [8], we set $\alpha_{\max} = 0.1$, overshooting factor $\eta = 1.05$, and the backward step parameter $\beta = 0.9$. For BIM [19] and PGD [21], we use a step size of $\alpha = 0.01$ and number of iterations $T = 10$.

For the CW attack [5], we set the confidence parameter $c = 10$, margin parameter $\kappa = 0$, and optimize for 100 steps using a learning rate of 0.01.

C.1. Base Attack Selection Across Stages

DASH learns the parameter values during training that are used to combine multiple base attacks at each stage of the adversarial example generation process. In Figure 1, we visualize the selected attack profiles for a robust model trained on CIFAR-100 [9]. The blue dotted line represents the initial combination of base attacks, while the orange line shows the learned combination after training.

Initially, the CW attack is selected more heavily due to its strong misclassification and better perceptual preservation capability. However, as training progresses, **DASH** shifts weight away from CW, since it no longer contributes to the meta-loss, largely due to the zero-gradient problem introduced by the hinge loss (see Eq. (1)) with $\kappa = 0$. As a result, our attack strategy continues to optimize for stronger and more confident adversarial examples, enhancing its transferability and effectiveness against various unseen post-processing defenses. This comes at the cost of slightly worse perceptual similarity scores, due to the increased distortion required to maintain high confidence.

D. DASH on Black-Box Attacks

For measuring the performance of **DASH** with the Black-Box attack, we have taken 2 query-based attacks, Sign-Flip [7] and Rays [6], and 2 decision-based attacks, Sign-Hunter [2] and Square attack [3], and combined them using **DASH**. They are the current state-of-the-art black-box attacks. We have measured the metrics for 5 disjoint sets of 1,000 images and reported the mean±std. Table 1 shows the performance comparison of **DASH** with other SOTA black-box

Table 3. Effectiveness of **DASH** with varying loss for the robust CIFAR-100 Cui2024 [9] model.

Loss	ASR (↑)						SSIM (↑)	LPIPS (↓)	FID (↓)
	Base	JPEG	TVM	Ensemble	NRP	Avg.			
SSIM	100.00	99.90	100.00	99.32	99.61	99.77	94.43	0.0139	41.08
LPIPS	100.00	100.00	100.00	99.41	99.80	99.84	92.34	0.0155	45.89

Table 4. Extended Transferability Comparison on CIFAR-100 Dataset. S represents Surrogate models and T represents Target models.

S (↓) \ T (→)	Attack	[9]	[25]	[1]	[10]	Avg.(↑)	SSIM (↑)
[9]	PI-FGSM++	62.20	59.25	60.01	59.71	59.66	91.12
	CW	84.14	51.39	56.48	52.44	53.44	94.38
	AutoAttack	76.77	65.98	64.26	62.11	64.12	92.09
	DiffAttack	77.24	74.76	74.72	57.10	68.86	91.16
	AdvAD	79.14	73.84	72.70	64.32	70.29	83.18
	DASH	99.77	89.55	79.69	70.61	**79.95**	**94.43**
[25]	PI-FGSM++	59.96	63.55	61.43	59.82	60.40	91.14
	CW	52.73	83.57	56.78	52.36	53.96	94.55
	AutoAttack	63.98	78.34	64.26	61.91	63.38	91.25
	DiffAttack	75.38	76.53	74.70	57.50	69.19	90.90
	AdvAD	72.98	77.42	72.24	66.40	70.54	83.07
	DASH	94.53	99.67	80.64	73.20	**82.79**	**94.73**
[1]	PI-FGSM++	53.24	53.77	68.47	42.99	50.00	91.14
	CW	37.32	37.73	84.18	57.46	44.17	94.96
	AutoAttack	52.13	54.08	77.90	58.22	54.81	93.03
	DiffAttack	72.44	54.30	81.35	54.70	60.48	91.26
	AdvAD	64.58	64.82	81.60	61.70	**63.70**	83.92
	DASH	63.50	63.09	99.22	62.83	63.14	**95.05**
[10]	PI-FGSM++	54.86	55.25	60.29	68.03	56.80	91.30
	CW	41.56	41.58	48.03	79.90	43.72	94.30
	AutoAttack	52.32	53.57	60.53	76.25	55.47	93.22
	DiffAttack	61.46	63.20	67.22	72.24	63.96	93.87
	AdvAD	60.42	61.04	66.54	74.52	62.67	87.28
	DASH	73.59	74.47	78.87	99.57	**75.64**	**94.57**

attacks for 2 different models, ResNet-18 [16], and Vision Transformer [12], trained on the CIFAR-10 dataset. The table shows that **DASH** achieves the highest average attack success rate for the models, exceeding the strongest baselines by almost 2%, and 1%. It also maintains the best imperceptibility with the highest SSIM and FID scores. This result implies that **DASH** can also be extended to black-box attacks.

E. DASH with Varying Base Attacks

Table 2 shows the **DASH** performance with varying base attacks. In all the combinations **DASH** shows an average attack success rate over 95% and SSIM over 90%, which implies that **DASH** is compatible with diverse base attacks and consistently achieves improved performance.

In our main experiment, we used a meta-loss for optimizing attack weights in **DASH**, balancing attack success rate (ASR) and imperceptibility. Focusing on metrics aligned with the human visual system, we primarily report results using SSIM [$\mathcal{L}_{\text{total}} = \lambda_{\text{asr}} \cdot f_y(\boldsymbol{x}_a) + \lambda_{\text{ssim}} \cdot (1 - \text{SSIM}(\boldsymbol{x}, \boldsymbol{x}_a))$], but we also experimented with LPIPS [$\mathcal{L}_{\text{total}} = \lambda_{\text{asr}} \cdot f_y(\boldsymbol{x}_a) + \lambda_{\text{lpips}} \cdot \text{LPIPS}(\boldsymbol{x}, \boldsymbol{x}_a)$] and achieved similar output. Table 3 demonstrates that **DASH** maintains a high ASR with minimal perturbation when optimizing with both SSIM and LPIPS. Specifically, the ASR varies by less than 0.1% and the SSIM score varies by only 0.02 units (assuming the 0 to 1 range), confirming the robustness of the framework regardless of the chosen perceptibility metric.

F. Extended Transfer Result

Besides the transfer results among multiple versions of Wide-ResNets, we also evaluated strict architectural transferability using XCiT-M12 [10] (a Transformer), as reported in Table 4. The result shows that the adversarial examples generated with **DASH** for one model also perform well for other models. **DASH** achieves the strongest transferability, achieving the top average ASR for each model while also maintaining the best perceptual quality (SSIM). For Cui2024 [9], and Wang2023 [25] models **DASH** outperforms the best baseline AdvAD with almost 10% and 12% improvement in average ASR, keeping the best SSIM score. For Addepalli2022 [1] **DASH** shows similar performance to the best baseline AdvAD while achieving a higher SSIM score. For Debenedetti2023 [10], **DASH** outperforms the best baseline DiffAttack with almost 12% improvement in average ASR while maintaining the highest SSIM score. This indicates that **DASH** consistently achieves good result in both transferability and imperceptibility scenarios.

References

[1] Sravanti Addepalli, Samyak Jain, et al. Efficient and effective augmentation strategy for adversarial training. *Advances in Neural Information Processing Systems*, 35:1488–1501, 2022. 3, 4

[2] Abdullah Al-Dujaili and Una-May O'Reilly. Sign bits are all you need for black-box attacks. In *International conference on learning representations*. 3

[3] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020. 3

[4] Brian R Bartoldson, James Diffenderfer, Konstantinos Parasyris, and Bhavya Kailkhura. Adversarial robustness limits via scaling-law and human-alignment studies. *arXiv preprint arXiv:2404.09349*, 2024. 3

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, 2017. 1, 3

[6] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1739–1747, 2020. 3

[7] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *European Conference on Computer Vision*, pages 276–293. Springer, 2020. 3

[8] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International conference on machine learning*, pages 2196–2205. PMLR, 2020. 3

[9] Jiequan Cui, Zhuotao Tian, Zhisheng Zhong, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. Decoupled kullback-leibler divergence loss. *Advances in Neural Information Processing Systems*, 37:74461–74486, 2024. 2, 3, 4

[10] Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *2023 IEEE conference on secure and trustworthy machine learning (SaTML)*, pages 225–253. IEEE, 2023. 4

[11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4312–4321, 2019. 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[13] Ranjie Duan, Yuefeng Chen, Dantong Niu, Yun Yang, A Kai Qin, and Yuan He. Advdrop: Adversarial attack to dnns by dropping information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7506–7515, 2021. 2

[14] Lianli Gao, Qilong Zhang, Jingkuan Song, and Heng Tao Shen. Patch-wise++ perturbation for adversarial targeted attacks. *arXiv preprint arXiv:2012.15503*, 2020. 1, 3

[15] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3

[18] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 3

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. arxiv 2016. *arXiv preprint arXiv:1607.02533*, 2016. 3

[20] Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15315–15324, 2022. 2

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 3

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 3

[23] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 262–271, 2020. 3

[24] Bao Wang, Alex T Lin, Wei Zhu, Penghang Yin, Andrea L Bertozzi, and Stanley J Osher. Adversarial defense via data dependent activation function and total variation minimization. *arXiv preprint arXiv:1809.08516*, 2018. 3

[25] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International conference on machine learning*, pages 36246–36263. PMLR, 2023. 3, 4

[26] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2730–2739, 2019. 1, 3

[27] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. 3

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2

[29] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1039–1048, 2020. 2