On the unreasonable vulnerability of transformers for image restoration – and an easy fix

Supplementary Material

Following we provide additional visual and quantitative results. Additioanally we explain the attack framework.

A. Attack Framework

Let $\mathbf{x}$ denote the ground-truth image, which is corrupted by a possibly non-linear degradation operator $\mathbf{A}$, resulting in an observation $\mathbf{y}^{\text{clean}}$, which can be expressed as

$$\mathbf{y}^{\text{clean}} = \mathbf{A}(\mathbf{x}). \tag{5}$$

Let $\mathcal{G}_\theta$ be a (Transformer-based) neural network parameterized by θ trained to recover $\mathbf{x}$ from $\mathbf{y}^{\text{clean}}$. In this work, we are interested in studying the stability of $\mathcal{G}_\theta$ to adversarial attacks that aim to degrade its performance through visually imperceptible changes to the inputs [18, 32]. We evaluate the robustness to attacks using additive perturbations δ with ℓ_p-norm constraints. We generate the adversarial perturbations based on two powerful attack methods CosPGD [1] developed for dense prediction tasks, and PGD attack [32], both of which we detail in the following. The objective of the attack is to maximize the deviation of the network output from the ground truth as measured by a loss function L, subject to ℓ_p norm constraints on the perturbation:

$$\underset{\delta}{\text{maximize}}\ L(\mathcal{G}_\theta(\mathbf{y}^{\text{clean}} + \delta),\ \mathbf{x})\ \text{ s.t. } \|\delta\|_p \leq \epsilon. \tag{6}$$

PGD. PGD is an iterative adversarial attack, where each sample is perturbed for a fixed amount of attack iterations (steps) with the intention of maximizing the loss further with each attack step. A single attack step in the PGD attack [32] is given as follows,

$$\begin{aligned}
\mathbf{y}^{\text{adv}_{t+1}} &= \mathbf{y}^{\text{adv}_t} + \alpha \cdot \text{sign}\nabla_{\mathbf{y}^{\text{adv}_t}} L(\mathcal{G}_\theta(\mathbf{y}^{\text{adv}_t}), \mathbf{x}) \quad (7)\\
\delta &= \phi^\epsilon(\mathbf{y}^{\text{adv}_{t+1}} - \mathbf{y}^{\text{clean}})\\
\mathbf{y}^{\text{adv}_{t+1}} &= \phi^r(\mathbf{y}^{\text{clean}} + \delta)
\end{aligned}$$

where the adversarial example $\mathbf{y}^{\text{adv}_{t+1}}$ at step $t+1$, is updated using the adversarial example from the previous step $\mathbf{y}^{\text{adv}_t}$, ∇ represents the gradient operation, α is the step size for the perturbation, ϕ^ϵ is denotes projection onto the appropriate ℓ_p-norm ball of radius ϵ, depending on the ℓ_p norm constraints on δ, and ϕ^r clips the adversarial example to lie in the valid intensity range of images (between [0, 1]). Prior works evaluating the adversarial robustness of image restoration networks consider L to be the reconstruction loss (MSE loss) to obtain adversarial examples maximizing the reconstruction error.

CosPGD. Instead of directly utilizing the averaged pixel-wise losses in PGD attack steps, [1] propose to weigh the pixel-wise losses using the cosine similarity between the network output and the ground truth (both scaled by softmax), to reduce the importance of the pixels which already have a large error in the previous iterations, and enable the attack to focus on the pixels with low error. For the task of restoration (a regression task), CosPGD attack steps for an untargeted attack are given as:

$$\begin{aligned}
\mathbf{x}^{\text{adv}_t} &= \mathcal{G}_\theta(\mathbf{y}^{\text{adv}_t}) \quad (8)\\
L_{\cos} &= \sum \text{cossim}(\Psi(\mathbf{x}^{\text{adv}_t}), \Psi(\mathbf{x})) \odot L(\mathbf{x}^{\text{adv}_t}, \mathbf{x})\\
\mathbf{y}^{\text{adv}_{t+1}} &= \mathbf{y}^{\text{adv}_t} + \alpha \cdot \text{sign}\nabla_{\mathbf{y}^{\text{adv}_t}} L_{\cos}\\
\delta &= \phi^\epsilon(\mathbf{y}^{\text{adv}_{t+1}} - \mathbf{y}^{\text{clean}})\\
\mathbf{y}^{\text{adv}_{t+1}} &= \phi^r(\mathbf{y}^{\text{clean}} + \delta),
\end{aligned}$$

where Ψ is the softmax function, $\odot$ denotes point-wise multiplication, and the cosine similarity (cossim) is given by

$$\text{cossim}(\vec{\mathbf{u}}, \vec{\mathbf{v}}) = \frac{\vec{\mathbf{u}} \cdot \vec{\mathbf{v}}}{||\vec{\mathbf{u}}|| \cdot ||\vec{\mathbf{v}}||} \tag{9}$$

[1] demonstrate that this approach results in a stronger attack for pixel-wise regression tasks than a PGD attack.

B. Additional Results

We provide sample reconstructed images from all considered networks under adversarial attacks. Figure A1 shows reconstructed images from GoPro test dataset [35] after the CosPGD attack [1] on the models. Whereas Figure. A2 shows reconstructed images from GoPro test dataset [35] after the PGD attack [32] on the models.

B.1. Intermediate networks

Further, we discuss some additional implementation details pertaining to the *Intermediate networks* and provide further observations and insights on their performance.

In Table A1 we report the performance of the *Intermediate network* and *Intermediate + ReLU*. Please note, the performance of the Intermediate network on the clean (unperturbed) samples is marginally lower than that reported by [7]. As [7] does not provide the code, pre-trained weights, or training configuration for this intermediate step between the Baseline network and NAFNet, our implementation is limited to the best of our understanding.

Table A1. Comparison of performance of all the considered models with α=0.01 and $\epsilon=\frac{8}{255}$.

Architecture	Clean		CosPGD						PGD					
			5 attack itrs		10 attack itrs		20 attack itrs		5 attack itrs		10 attack itrs		20 attack itrs	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Restormer	31.99	0.9635	11.36	0.3236	9.05	0.2242	7.59	0.1548	11.41	0.3256	9.04	0.2234	7.58	0.1543
+ ADV	30.25	0.9453	**24.49**	0.81	**23.48**	**0.78**	21.58	0.7317	**24.5**	0.8079	**23.5**	**0.7815**	21.58	0.7315
Baseline	32.48	0.9575	10.15	0.2745	8.71	0.2095	7.85	0.1685	10.15	0.2745	8.71	0.2094	7.85	0.1693
+ ADV	30.37	0.9355	15.47	0.5216	13.75	0.4593	12.25	0.4032	15.47	0.5215	13.75	0.4592	12.24	0.4026
NAFNet	32.87	0.9606	8.67	0.2264	6.68	0.1127	5.81	0.0617	10.27	0.3179	8.66	0.2282	5.95	0.0714
+ ADV	29.91	0.9291	17.33	0.6046	14.68	0.509	12.30	0.4046	15.76	0.5228	13.91	0.4445	12.73	0.3859
Intermediate	29.93	0.9289	6.0224	0.0509	5.8166	0.0366	5.7199	0.0315	6.0225	0.0509	5.8158	0.0365	5.7173	0.0314
+ ADV	29.00	0.9154	24.02	**0.8213**	22.01	0.7775	20.15	0.7286	24.02	**0.8213**	21.98	0.7770	20.15	0.7286
Intermediate + ReLU	30.39	0.9349	13.87	0.4093	11.63	0.3128	10.29	0.2538	13.87	0.4094	11.62	0.3127	10.29	0.2542
+ ADV	28.49	0.9072	23.90	0.8046	22.46	0.7637	**21.85**	**0.7484**	23.91	0.8046	22.47	0.7638	**21.84**	**0.7481**

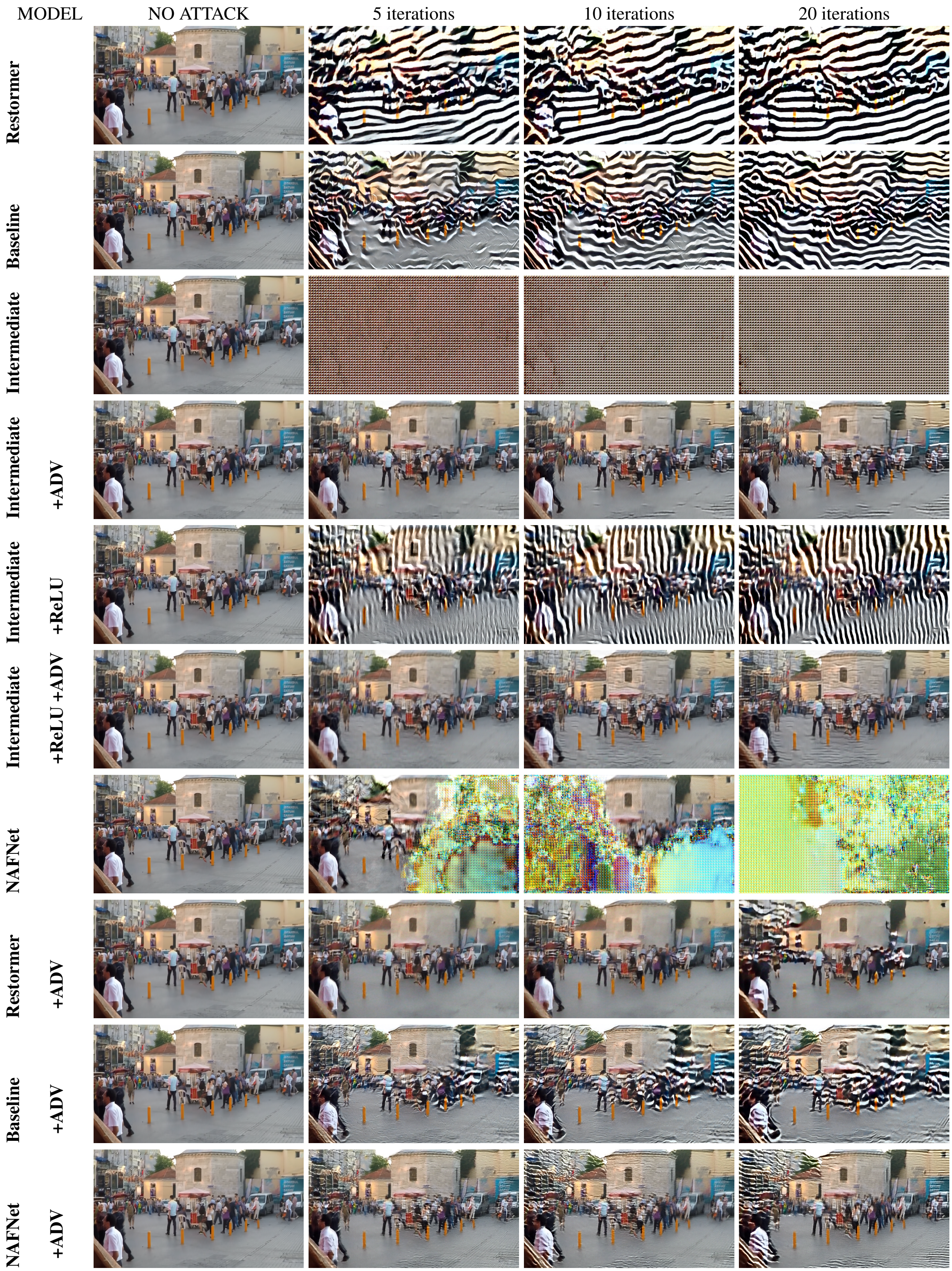

Figure A1. Comparing images reconstructed by all models after **CosPGD attack**

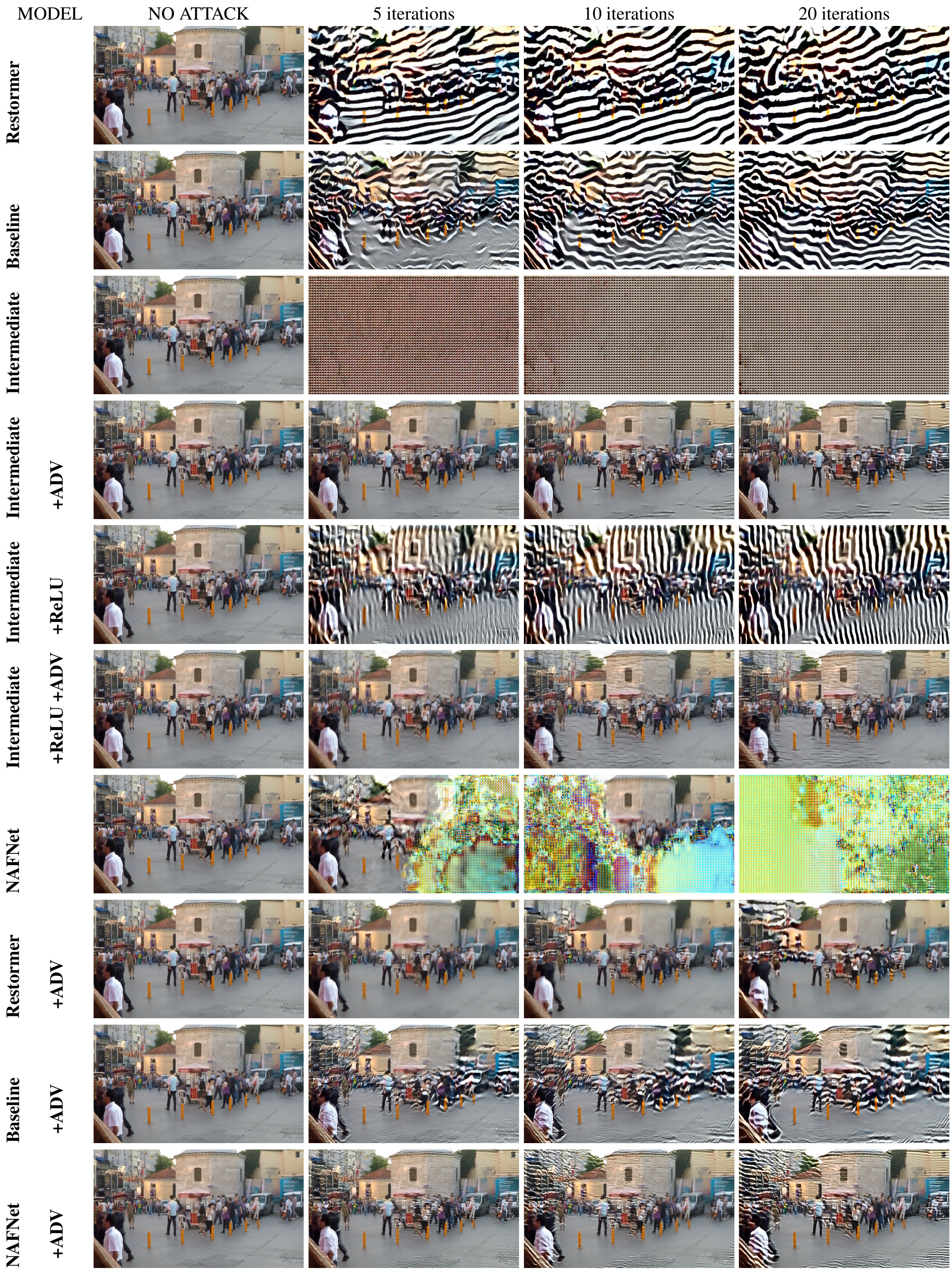

Figure A2. Comparing images reconstructed by all models after **PGD attack**