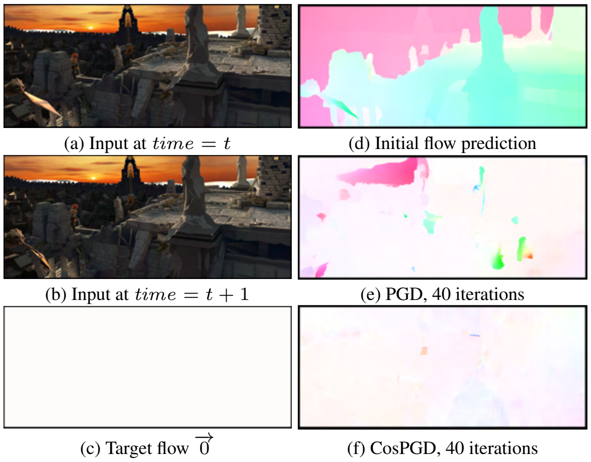
CosPGD: an efficient white-box adversarial attack for pixel-wise prediction tasks

Shashank Agnihotri [†1] Steffen Jung [†21] Margret Keuper [12]

Abstract

While neural networks allow highly accurate predictions in many tasks, their lack of robustness towards even slight input perturbations often hampers their deployment. Adversarial attacks such as the seminal *projected gradient descent* (PGD) offer an effective means to evaluate a model's robustness and dedicated solutions have been proposed for attacks on semantic segmentation or optical flow estimation. While they attempt to increase the attack's efficiency, a further objective is to balance its effect, so that it acts on the entire image domain instead of isolated point-wise predictions. This often comes at the cost of optimization stability and thus efficiency. Here, we propose CosPGD, an attack that encourages more balanced errors over the entire image domain while increasing the attack's overall efficiency. To this end, CosPGD leverages a simple alignment score computed from any pixel-wise prediction and its target to scale the loss in a smooth and fully differentiable way. It leads to efficient evaluations of a model's robustness for semantic segmentation as well as regression models (such as optical flow, disparity estimation, or image restoration), and it allows it to outperform the previous SotA attack on semantic segmentation. We provide code for the CosPGD algorithm and example usage at `https://github.com/shashankskagnihotri/cospgd`.


(a) Input at $time = t$

(d) Initial flow prediction

(b) Input at $time = t + 1$

(e) PGD, 40 iterations

(c) Target flow $\overrightarrow{0}$

(f) CosPGD, 40 iterations

Figure 1: Optical flow predictions using RAFT (Teed & Deng, 2020) on Sintel (Butler et al., 2012; Wulff et al., 2012) validation. (a) and (b) show two consecutive frames for which the initial optical flow in (d) was predicted. The results of attacking the model with target $\overrightarrow{0}$ (c) are depicted in (e) for PGD and (f) for CosPGD. For the same perturbation magnitude and number of iterations, the proposed CosPGD alters the estimated optical flow more strongly and brings it closer to target (c).

1. Introduction

Deep Neural Networks (DNNs) have been gaining popularity for estimating solutions to various complex tasks including numerous vision tasks like classification (Krizhevsky et al., 2012; He et al., 2015; Xie et al., 2016; Liu et al., 2022; Lukasik et al., 2023a), generative models (Jung & Keuper, 2020; 2021; Lukasik et al., 2022; Jung et al., 2023b), image segmentation (Ronneberger et al., 2015; Zhao et al., 2017; Jung et al., 2022; Sommerhoff et al., 2023), or disparity (Li et al., 2021) and optical flow (Fischer et al., 2015; Ilg et al., 2016; Teed & Deng, 2020; Schmalfuss et al., 2023) estimation, due to their overall precise predictions. However, DNNs are inherently black-box function approximators (Buhrmester et al., 2019), known to find shortcuts to map the input to a target (Geirhos et al., 2020), to learn biases (Geirhos et al., 2018; Gavrikov et al., 2024) and to lack robustness (Szegedy et al., 2014; Hoffmann et al., 2021).

An adversarial attack adds a crafted, small (epsilon-sized) perturbation to the input of a neural network that aims to alter the prediction, thus assessing a network's robustness as in the benchmarks by Croce et al. (2021); Jung et al. (2023a). Due to the practical relevance to evaluating and

[†] Authors contributed theoretically and practically to this paper. Please refer to the Author Contribution section for details.

[1] Data and Web Science Group, University of Mannheim, Germany [2] Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany. Correspondence to: Shashank Agnihotri <shashank.agnihotri@uni-mannheim.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

analyzing DNN models, such attacks have been extensively studied (Goodfellow et al., 2014; Kurakin et al., 2017; Wong et al., 2020b; Madry et al., 2017; Moosavi-Dezfooli et al., 2015; Kurakin et al., 2016; Schrodi et al., 2022; Agnihotri et al., 2023b; Grabinski et al., 2022; 2023; Lukasik et al., 2023b).

Existing approaches predominantly focus on attacking image classification models. However, arguably, the robustness of models for pixel-wise prediction tasks is highly relevant for many safety-critical applications such as motion estimation in autonomous driving or semantic segmentation. The application of existing attacks to pixel-wise prediction tasks such as semantic segmentation or optical flow estimation is possible in principle (e.g. as in Arnab et al. (2017)), albeit carrying only limited information since the pixel-specific loss information is not fully leveraged. In Figure 1, we illustrate this effect for a targeted attack on optical flow estimation and show that classical classification attacks such as PGD (see Figure 1(e)) only fool the network predictions to some extent: PGD tends to only fit the target (all zeros, i.e. white) in parts of the optical flow, while a few predictions remain intact.

For semantic segmentation, Gu et al. (2022) showed that harnessing pixel-wise information for adversarial attacks leads to much stronger attacks. They argue that, during the attack, the loss to be backpropagated needs to be altered such that already flipped pixel predictions are less important for the gradient computation. Thus, SegPGD (Gu et al., 2022) makes a binary decision for each pixel based on the classification result at this location, to weigh the attack loss for incorrect and correct model predictions individually. While this is intuitive for semantic segmentation, it can not extend to pixel-wise regression tasks by definition. Furthermore, due to the discrete nature of the loss scaling, SegPGD faces stability issues and has to fade back in the loss of already incorrectly predicted pixels over time (Gu et al., 2022).

In this work, we propose CosPGD, an efficient white-box adversarial attack that considers the cosine-alignment between the prediction and target for each pixel, leading to a smooth and fully differentiable attack objective. Due to its principled formulation, CosPGD can be used for a wide range of pixel-wise prediction tasks beyond semantic segmentation. Figure 1(f) shows its effect on optical flow estimation, where, in contrast to PGD, it can fit the target at almost all locations. Since it leverages the (continuous) posterior distribution of the prediction to allow for a smooth and differentiable loss computation, it can significantly outperform SegPGD on semantic segmentation. The main contributions of this work are as follows:

- We propose CosPGD, an efficient white-box adversarial attack, that can be applied to any pixel-wise prediction task, and thus allows for an efficient evaluation of their robustness in a unified setting.
- We provide theoretical and empirical proofs for the stability and spatial balancing of CosPGD during attack optimization.
- For semantic segmentation, we compare CosPGD to the recently proposed SegPGD which also uses pixel-wise information for generating attacks. CosPGD outperforms SegPGD by a significant margin.
- To demonstrate CosPGD's versatility, we also evaluate it as a *targeted* attack and as a *non-targeted* attack, for both ℓ_2 and ℓ_∞ bounds on semantic segmentation, optical flow estimation and image restoration in several settings and datasets.

2. Related work

The vulnerability of DNNs to adversarial attacks was first explored in (Goodfellow et al., 2014) for image classification, proposing the Fast Gradient Sign Method (FGSM). FGSM is a single-step (one iteration) white-box adversarial attack that perturbs the input in the direction of its gradient, generated from backpropagating the loss, with a small step size, such that the model prediction becomes incorrect. Due to its fast computation, it is still a widely used approach. Numerous subsequent works have been directed towards generating effective adversarial attacks for diverse tasks including NLP (Morris et al., 2020; Ribeiro et al., 2018; Iyyer et al., 2018), or 3D tasks (Zhang et al., 2021; Sun et al., 2021). Yet, the high input dimensionality of image classification models results in the striking effectiveness of adversarial attacks in this field (Goodfellow et al., 2014; Jia et al., 2022). A vast line of work has been dedicated to assessing the quality and robustness of representations learned by the network, including the curation of dedicated evaluation data for particular tasks (Kang et al., 2019; Hendrycks & Dietterich, 2019; Hendrycks et al., 2019) or the crafting of effective adversarial attacks. These adversarial attacks can be image-wide or localized in a small region or patch. These perturbations are in a small region of the image and are called Patch Attacks (e.g. (Brown et al., 2017; Scheurer et al., 2024)),while methods such as proposed in (Goodfellow et al., 2014; Kurakin et al., 2017; Madry et al., 2017; Wong et al., 2020b; Moosavi-Dezfooli et al., 2015; Croce & Hein, 2020; Andriushchenko et al., 2020; Carlini & Wagner, 2017; Rony et al., 2019; Dong et al., 2018) argue in a Lipschitz continuity motivated way that a robust network's prediction should not change drastically if the perturbed image is within the epsilon-ball of the original image and thus optimize attacks globally within the epsilon neighborhood of the original input. Our proposed CosPGD follows this line of work.

White-box attacks assume full access to the model and its gradients (Goodfellow et al., 2014; Kurakin et al., 2017; Madry et al., 2017; Wong et al., 2020b; Gu et al., 2022; Moosavi-Dezfooli et al., 2015; Rony et al., 2023; Dong et al., 2018; Schmalfuss et al., 2022a) while black-box attacks optimize perturbations in a randomized way (Andriushchenko et al., 2020; Ilyas et al., 2018; Qu et al., 2023). The proposed CosPGD derives its optimization from PGD (Kurakin et al., 2017) and is a white-box attack.

Further, one distinguishes between *targeted* attacks (e.g. (Wong et al., 2020a; Gajjar et al., 2022; Schmalfuss et al., 2022b)) that turn the network predictions towards a specific target and *untargeted* (or non-targeted) attacks that optimize the attack to cause any incorrect prediction. PGD (Kurakin et al., 2017), and CosPGD by extension, allows for both settings (Vo et al., 2022).

While previous attacks predominantly focus on classification tasks, only a few approaches specifically address the analysis of pixel-wise prediction tasks such as semantic segmentation, optical flow, or disparity estimation. For example, PCFA (Schmalfuss et al., 2022b) was applied to the estimation of optical flow and specifically minimizes the average end-point error (AEE) to a target flow field. A notable exception of pixel-wise white-box adversarial attack is proposed in (Gu et al., 2022). The SegPGD attack could showcase the importance of pixel-wise attacks for semantic segmentation. In this work, we propose CosPGD to provide a principled and efficient adversarial attack, that can be applied to a wide range of pixel-wise prediction tasks and provides stable optimization. CosPGD outperforms SegPGD by a significant margin when attacking semantic segmentation models while preserving its efficiency and extending it to other pixel-wise prediction tasks.

3. Preliminaries

The projected gradient descent (PGD) (Kurakin et al., 2017) attack is an iterative white box adversarial attack. It is known to be a strong attack and builds the basis for follow-up methods such as (Wong et al., 2020b). Such methods leverage the gradients of a model's loss to create strong adversarial attacks, e.g. the PGD update is given as

$$\boldsymbol{X}^{\text{adv}_{t+1}} = \boldsymbol{X}^{\text{adv}_t} + \alpha \cdot \text{sign} \nabla_{\boldsymbol{X}^{\text{adv}_t}} L(f_\theta(\boldsymbol{X}^{\text{adv}_t}), \boldsymbol{Y}) \quad (1)$$

$$\delta = \phi^\epsilon(\boldsymbol{X}^{\text{adv}_{t+1}} - \boldsymbol{X}^{\text{clean}}), \quad (2)$$

$$\boldsymbol{X}^{\text{adv}_{t+1}} = \phi^r(\boldsymbol{X}^{\text{clean}} + \delta) \quad (3)$$

Here, $L(\cdot)$ is a function (differentiable at least once) of the model prediction and the target, which defines the loss the model f_θ aims to minimize, $\boldsymbol{X}^{\text{adv}_{t+1}}$ is a new adversarial example for time step $t+1$, generated using $\boldsymbol{X}^{\text{adv}_t}$, the adversarial example at time step t and initial clean sample $\boldsymbol{X}^{\text{clean}}$. $\boldsymbol{Y}$ is the ground truth label for non-targeted attacks and the target for targeted attacks, α is the step size for the perturbation (α is multiplied by -1 for targeted attacks to take a step in the direction of the target), and the function ϕ^ϵ is clipping the δ in ϵ-ball for ℓ_∞-norm bounded attacks or the ϵ-projection in l_2-norm bounded attacks, complying with the ℓ_∞-norm or l_2-norm constraints, respectively. ϕ^r is clipping the generated example in the valid input range (usually between [0, 1]). $\nabla_{\boldsymbol{X}^{\text{adv}_t}} L(\cdot)$ denotes the gradient of $\boldsymbol{X}^{\text{adv}_t}$ generated by backpropagating the loss and is used to determine the direction of the perturbation step.

Originally, PGD has been conceived to attack image classification models. For pixel-wise prediction tasks, its update in Equation 1 considers the sum of pixel-wise losses $\bar{L}$, i.e.

$$\begin{aligned} \boldsymbol{X}^{\text{adv}_{t+1}} &= \boldsymbol{X}^{\text{adv}_t} + \\ &\alpha \cdot \text{sign} \nabla_{\boldsymbol{X}^{\text{adv}_t}} \sum_{i \in H \times W} \bar{L}\left(f_\theta(\boldsymbol{X}^{\text{adv}_t})_i, \boldsymbol{Y}_i\right) \end{aligned} \quad (4)$$

where i iterates over all positions in the prediction $f_\theta(\boldsymbol{X})$ with $f_\theta(\boldsymbol{X}), \boldsymbol{Y} \in \mathbb{R}^{H \times W \times M}$ for images of size $H \times W$ and M output dimensions (e.g. M classes for semantic segmentation). The update in PGD thus aims to increase the overall loss maximally summing over all locations. It does not take into account that the prediction in some locations might remain correct while it further increases the loss in other locations (that might already be predicted incorrectly).

4. Prediction Alignment Scaling - CosPGD

We argue that the above formulation neglects an interesting aspect: It does not facilitate inducing equally manipulated predictions in all locations. This can be disadvantageous for targeted attacks, where one wants to ensure that the target is fit at all locations equally. In particular, it is however problematic for, for example, attacks on semantic segmentation where models use cross-entropy-like losses that do not saturate. Thus, after flipping a few point-wise label predictions, PGD-based attacks might continue to increase the overall loss even without altering any further labels. Thus, we argue that the alignment between the current prediction and the target or ground truth has to be taken into account to efficiently compute strong adversaries.

In the following, we introduce CosPGD. Its goal is to employ a continuous pixel-wise measure of prediction alignment inside the computation of the attack update step so that the gradient-based CosPGD iterations smoothly converge to a strong adversary that acts on all pixel locations. The update step in CosPGD is defined as

$$\begin{aligned} \boldsymbol{X}^{\text{adv}_{t+1}} &= \boldsymbol{X}^{\text{adv}_t} + \alpha \cdot \text{sign} \nabla_{\boldsymbol{X}^{\text{adv}_t}} \\ &\sum_{i \in H \times W} \cos\left(\psi(f_\theta(\boldsymbol{X}^{\text{adv}_t})_i), \boldsymbol{Y}_i\right) \cdot \bar{L}\left(f_\theta(\boldsymbol{X}^{\text{adv}_t})_i, \boldsymbol{Y}_i\right), \end{aligned} \quad (5)$$

where ψ is a continuously differentiable, monotonous function that can be used to normalize the model output, i.e. we assume $\psi(f_\theta(\boldsymbol{X})) = 1 \quad \forall f_\theta(\boldsymbol{X})$, and

$$\cos(\boldsymbol{P}, \boldsymbol{Y}) = \frac{\boldsymbol{P} \cdot \boldsymbol{Y}}{\|\boldsymbol{P}\| \cdot \|\boldsymbol{Y}\|} \tag{6}$$

is the cosine similarity between two vectors, in this case a (normalized) network prediction $\boldsymbol{P}$ and the target or ground truth $\boldsymbol{Y} \in \mathbb{R}^M$. For the example of semantic segmentation, $\boldsymbol{Y}$ is usually one-hot encoded and therefore normalized. Cosine similarity provides a measure of similarity between the direction of two vectors and should therefore be well-suited to represent the alignment of the prediction with the target at the posterior level. It scales in a fixed range [-1, 1], such that no further normalization of the scaling is needed.

As the loss in CosPGD is scaled with a pixel-wise measure of alignment between the current prediction and the target in Equation 5, the resulting gradient update emphasizes on changing those pixel-wise predictions that are correct in the current prediction.

This yields several desirable properties. First, it facilitates to optimize adversaries to pixel-wise tasks so that the prediction in all pixels is affected. As such, it is a stronger attack than PGD on tasks such as semantic segmentation. Further, it can be applied to pixel-wise classification and regression tasks in a principled way. Second, the loss is scaled with a smooth scaling function, i.e. if the prediction changes only a little, the change in the proposed alignment score will also be small, specifically

Proposition 4.1. *For any two pixel-wise network predictions $f_\theta(\boldsymbol{X})_i$ and $f_\theta(\bar{\boldsymbol{X}})_i \in \mathbb{R}^M$, a target $\boldsymbol{Y}_i \in \mathbb{R}^M$ and a continuously differentiable function $\psi : \mathbb{R}^M \to \mathbb{R}^M$ with $\psi(f_\theta(\boldsymbol{X})) = 1 \quad \forall f_\theta(\boldsymbol{X})$, it is*

$$\begin{aligned} d \cdot \|f_\theta(\boldsymbol{X})_i - f_\theta(\bar{\boldsymbol{X}})_i\| \geq \\ \|\cos\left(\psi(f_\theta(\boldsymbol{X})_i), \boldsymbol{Y}_i\right) - \cos\left(\psi(f_\theta(\bar{\boldsymbol{X}})_i), \boldsymbol{Y}_i\right)\| \end{aligned}$$

for a real, constant $d \geq 0$.

The proof is given in the appendix. As a result of the above proposition, the gradient in Equation 5 will change smoothly over the attack iterations for a sufficiently small step-size α and allow for fast convergence properties, i.e. CosPGD should provide strong adversaries with relatively few iterations while providing a balance over the pixel locations.

Untargeted versus Targeted Attacks. Untargeted attacks intend to drive the model's predictions away from the model's intended target (ground truth). Specifically, for non-targeted attacks, CosPGD, therefore, scales the loss pixel-wise in proportion to the pixel-wise predictions' similarity to the ground truth, while also accounting for the decrease in similarity over iterations. Using cosine similarity as an alignment measure, pixels at which the network predictions are closer to the intended target (ground truth), have a higher similarity (approaching 1) and thus higher loss. Pixels with lower similarity, have a lower loss but are not rendered benign. In contrast, for the targeted setting, the attack aims to drive predictions towards the target at all locations, such that pixels at which the network predictions are closer to the target and have higher similarity should have a lower loss that pixels with lower similarity.

To scale the loss by the dissimilarity of the prediction to the target prediction, for targeted settings, the targeted CosPGD update step is given by Eqn 7 in analogy to Eqn 5.

$$\begin{aligned} &\boldsymbol{X}^{\mathrm{adv}_{t+1}} = \boldsymbol{X}^{\mathrm{adv}_t} + \alpha \cdot \mathrm{sign}\nabla_{\boldsymbol{X}^{\mathrm{adv}_t}} \\ &\sum_i \left(1 - \cos\left(\psi(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_i), \boldsymbol{Y}_i\right)\right) \cdot \bar{L}\left(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_i, \boldsymbol{Y}_i\right) \end{aligned} \tag{7}$$

Choice of ψ and Algorithm Description. In Equation 5, we require ψ to be monotonically increasing, differentiable, and, to ensure smooth convergence, smooth. To obtain a distribution over the predictions, we calculate the softmax of the predictions before taking the argmax

$$\psi(f_\theta(\boldsymbol{X})) = \mathit{softmax}(f_\theta(\boldsymbol{X})), \tag{8}$$

$$\text{where,} \quad \mathit{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}. \tag{9}$$

Thus, in Algorithm 1 (given in Appendix A.2) and Equation 5, ψ is the softmax function. In the case of semantic segmentation, we obtain the distribution of the target $\boldsymbol{Y}_i$ for every point i by generating a *one-hot encoded vector* of the label (i.e. encoding the argmax label) while we also apply softmax to compute $\boldsymbol{Y}_i$ from continuous targets, e.g. for optical flow or disparity estimation. One-hot encoding and softmax to represent $\boldsymbol{Y}_i$ are summarized by function Ψ' in Algorithm 1. $\boldsymbol{X}^{\mathrm{adv}}$ is initialized to the clean input sample $\boldsymbol{X}^{\mathrm{clean}}$ with added randomized noise in the range $[-\epsilon, +\epsilon]$, ϵ being the maximum allowed perturbation. Over attack iterations $\boldsymbol{X} = \boldsymbol{X}^{\mathrm{adv}_t}$, the adversarial example generated at iteration t, such that $t \in [0, T)$, where T is the total number of attack iterations.

Loss Scaling in Previous Approaches. When optimizing δ for an adversarial attack for semantic segmentation, Gu et al. (2022) have argued before that pixels which are already misclassified by the model are less relevant than pixels correctly classified by the model, because the intention of the attack is to make the model misclassify as many pixels as possible while perturbing the δ inside the ϵ-ball. As a consequence, they make a hard decision based on each pixels argmax prediction as of whether it is taken into account for attack computation. In (Gu et al., 2022), the PGD update from Equation 4 is thus modified to

$$\operatorname{sign}\nabla_{\boldsymbol{X}^{\mathrm{adv}_t}}\left((1-\lambda)\sum_{i\in P^T} L\left(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_i, \boldsymbol{Y}_i\right) + \lambda\sum_{k\in P^F} L\left(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_k, \boldsymbol{Y}_k\right)\right), \quad (10)$$

where P^T is the set of correctly classified pixels and P^F is the set of wrongly classified pixels, λ is a scaling factor between the two parts of the loss that is set heuristically, and $\boldsymbol{Y}$ is the one-hot encoded ground truth for semantic segmentation. See their equation (4) for details.

For positive λ and for categorical labels (i.e. $\boldsymbol{Y}$ one-hot encoded), we can rewrite the SegPGD update as

$$\operatorname{sign}\nabla_{\boldsymbol{X}^{\mathrm{adv}_t}}\left(\sum_i\left(1-\left|\lambda-\frac{|(argmax(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_i)-\boldsymbol{Y}_i|}{2}\right|\right)\cdot L\left(f_\theta(\boldsymbol{X}^{\mathrm{adv}_t})_i, \boldsymbol{Y}_i\right)\right) \quad (11)$$

for all locations $i \in P^T \cup P^F$, i.e. $|\lambda - |(argmax(f(\boldsymbol{X}^{\mathrm{adv}_t})) - \boldsymbol{Y}|/2|$ equals $1-\lambda$ for incorrect predictions, it equals λ for correct predictions.

Thus, the approach by Gu et al. (2022) resembles a discrete approximation of the proposed CosPGD. Yet, the discrete nature of this weighting scheme has several disadvantages: First, it limits SegPGD to applications where the correctness of the prediction can be evaluated in a binary way, and it disregards the actual prediction scores. For pixel-wise regression tasks (like optical flow, or image reconstruction) there is no absolute measure of correctness, so SegPGD can not be directly applied. Second, as the number of misclassified pixels increases, the attack loses effectiveness if it only focuses on correctly classified pixels in a binary way. The λ scaling in (Gu et al., 2022) has been proposed as a heuristical remedy. It scales the loss over iterations such that the impact of the proposed scheme decays over time. At the end of the attack iterations, $\lambda \approx 1/2$. This avoids the concern of the attack becoming benign after a few iterations, yet it fades out the effect of SegPGD and may reduce its efficiency. CosPGD, operating on continuous predictions, does not require such a heuristic.

Last, but maybe most importantly, the scaling based on discrete labels is not smooth, i.e. the *argmax* operation in Equation 11 is not differentiable, such that, during the iterations, the direction of the gradient update can fluctuate, potentially leading to slower convergence of the SegPGD attack, compared to the proposed CosPGD. We show empirical evidence for this issue in Figure 2 where we report the change in gradients and their directions during the attack optimization for PGD, SegPGD and the proposed CosPGD.

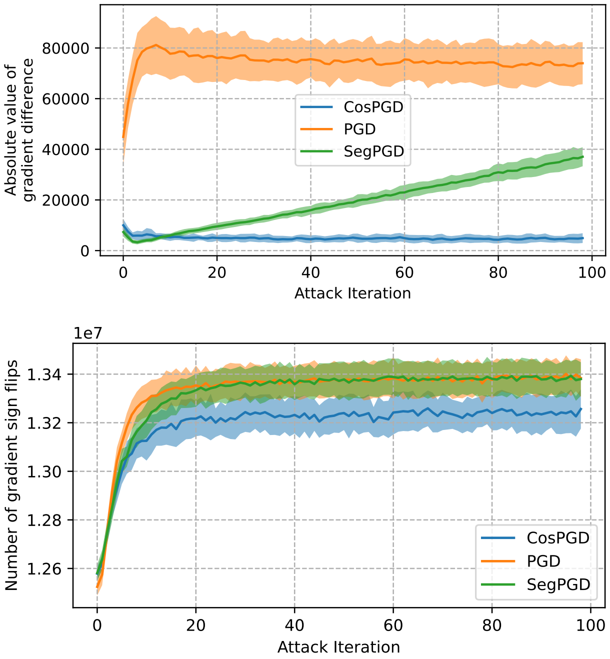


Figure 2: Change in pixel-wise image gradients over attack iterations on DeepLabV3 performing semantic segmentation on PASCAL VOC 2012 validation subset. We observe that the absolute difference between gradient values (top) is larger for PGD and increasing for SegPGD, while being stable for CosPGD. Further, CosPGD has fewer changes in gradient direction over attack iterations (bottom) compared to PGD and SegPGD. This shows CosPGD is more stable during optimization compared to PGD and SegPGD.

5. Experiments

To demonstrate the wide applicability of CosPGD, we conduct our experiments on distinct downstream tasks: semantic segmentation, optical flow estimation, and image restoration. For semantic segmentation, we compare CosPGD to SegPGD and PGD and empirically validate its improved stability over the attack iterations. Further, we verify that CosPGD indeed encourages the attack to act on the entire image domain, with quantitative and qualitative results on non-targeted attacks on semantic segmentation and targeted attacks on optical flow. For optical flow estimation and other tasks (such as image deblurring and image denoising), we compare CosPGD to PGD in the main paper. The subsequent experiments provide evidence of CosPGD being a strong adversarial attack in diverse tasks and setups. In the main paper, we report ℓ_∞-norm constrained attacks with $\epsilon \approx \frac{8}{255}$ for CosPGD, SegPGD, and PGD. For α, we follow (Gu et al., 2022) and set the step size to $\alpha = 0.01$ (please refer to Appendix B.6 for an ablation study). Further evaluations such as for different ϵ and α values for ℓ_∞ (Appendix B.1.2) and ℓ_2 bounded attacks (Appendix B.6.1), **CosPGD for Adversarial Training** (Ap-

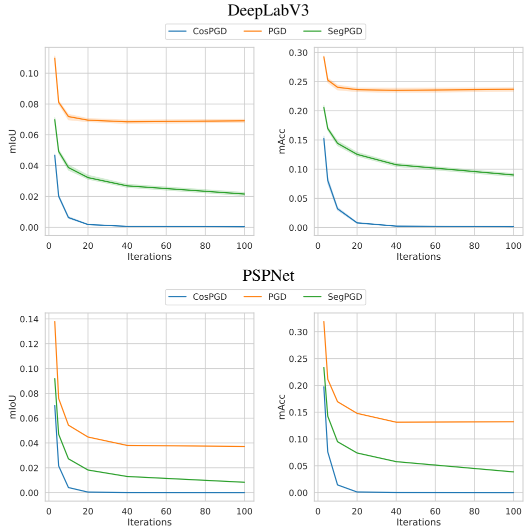


Figure 3: CosPGD versus PGD and SegPGD (ℓ_∞-norm constrained) for semantic segmentation on PASCAL VOC2012 validation set on DeepLabV3 and PSPNet. CosPGD outperforms competing attacks even in early iterations by a large margin. See also Table 11 in Appendix B.

pendix B.8), Transfer Attacks (Appendix B.2) including attacks on SAM (Kirillov et al., 2023) (Appendix B.4), Attack on Robust Models (Appendix B.3), comparison of CosPGD to recently proposed PCFA for optical flow estimation over various architectures (Appendix C.3) and Image Denoising (Appendix D), are provided in the Appendix, Table 1 provides an overview. Please also refer to the Appendix A.3 for all details on the experimental setup.

5.1. Stability during Attack Optimization

We evaluate the stability of CosPGD on semantic segmentation PASCAL VOC 2012 (Everingham et al., 2012). Figure 2(top) shows the change in gradients (i.e. the absolute distance between gradients in two subsequent iterations) due to PGD, SegPGD and CosPGD over 100 iterations. Both PGD and CosPGD gradients change constantly over time, with PGD having much stronger change. Yet, as expected, the change in gradients of SegPGD increases over the iterations, potentially leading to oscillations in the optimization. To further analyze the effect on the optimization, Figure 2 (bottom) shows the respective change in gradient direction (note that PGD, SegPGD, and CosPGD update all consider the sign of the gradient). The evaluation verifies that the CosPGD updates are more stable over the iterations, such that we can expect faster convergence, i.e. a stronger attack at fewer iterations.

An indication of the potential benefit can be seen for example in Table 11 (Appendix), where we observe that at low attack iterations (iterations=3) SegPGD implies that PSPNet is more adversarially robust than DeepLabV3. However, after more attack iterations (iterations$\geq$5), SegPGD reveals that DeepLabV3 is more robust than PSPNet. Contrary to this, CosPGD even at low attack iterations correctly predicts DeepLabV3 to be more robust than PSPNet. This is an insight that CosPGD provides with considerably fewer iterations, thus lower overall computation time, while compute costs per iteration are comparable, see Table 2 (Appendix).

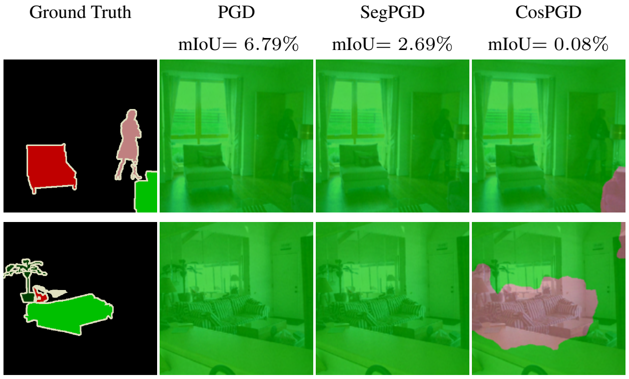


Figure 4: Example predictions of DeepLabV3 on PASCAL VOC 2012 val set after ℓ_∞ PGD, SegPGD, and CosPGD attacks with 40 iters. The ground truth segmentations are given on the left. Both PGD and SegPGD are able to successfully change most of the predicted labels to one of the ground truth labels (here in green). Yet, the region with this label is predicted correctly. Here, only CosPGD also changes the prediction in this region to a third class.

5.2. Spatial Balancing of the Attack

In the following, we show empirically that CosPGD encourages the attack to alter predictions over the entire image domain while PGD and SegPGD are weaker in this respect.

Semantic Segmentation. We first discuss the spatial balancing of CosPGD for untargeted attacks on semantic segmentation on PASCAL VOC2012, the standard setting evaluated in (Gu et al., 2022).

Therefore, we consider the mean Intersection over Union (mIoU) and mean accuracy (mACC) over the attack iterations as reported in Figure 3. The first observation is that CosPGD yields a much stronger attack compared to PGD or SegPGD for both DeepLabV3 (Chen et al., 2017) and PSPNet (Zhao et al., 2017). Second, we observe that CosPGD pushes the mIoU to values close to zero even in the first attack iterations, meaning that almost all pixel labels are flipped, while the mIoU for PGD stagnates at a high level as it decreases slowly for SegPGD, leading to significantly higher mIoUs even after 100 iterations, that for CosPGD.

For example in Figure 4 after 40 attack iterations, all attacks are considerably fooling the network into making incorrect predictions. However, once the dominant class label is changed by SegPGD or PGD, they do not further optimize over small regions of correct predictions. In contrast, CosPGD successfully fools the model into making incorrect predictions even in these small regions by either swapping the region prediction with an already existing class or forcing the model into predicting a different class.

PGD can bring down the $mIoU$ of DeepLabV3 to 6.79%. SegPGD, by naïvely utilizing the pixel-wise segmentation error, deteriorates the model performance further to 2.69%. However, CosPGD can fool the network into making incorrect predictions for almost all pixels, bringing down the model performance to almost 0% after 100 iterations.

Optical Flow. The evaluation of whether an attack alters the prediction in all regions is less trivial to conduct than for semantic segmentation, since there is no absolute measure of correctness. Therefore, in Figure 5, we evaluate CosPGD versus PGD for targeted attacks on optical flow (using RAFT (Teed & Deng, 2020)) on the KITTI-2015 validation set such that we see how many of the point-wise flow predictions have an end point error (epe) to the target that is below a certain threshold. Ideally, we would see a curve that is rising to the maximum value very quickly, indicating that all predictions are very close to the target. Figure 5 indicates that CosPGD achieves to bring more pixel-wise predictions very close to the target whereas only few predictions have larger epe. For PGD, more predictions remain with higher epe to the target. SegPGD can not directly be compared to in this regard, since it is conceived for semantic segmentation and requires an absolute measure of correctness (i.e. is the predicted label correct).

A comparison of CosPGD to PGD in terms of epe over the iterations is shown in Figure 6. Here, we quantitatively observe better performance of CosPGD compared to PGD. As this is the targeted setting, we intend to close the gap between the target prediction and the model predictions, thus a lower epe of the model prediction w.r.t. the target prediction is desired. As the attack iterations increase, across datasets, CosPGD can significantly fool the network into making predictions closer to the target, bringing down the epe to as low as 1.55 for Sintel (final) (see Appendix C).

We qualitatively observe in Figure 7 that the initial optical flow estimation by the model (which is substantially different to the target) is only moderately changed when the model is attacked with PGD. As the attack was designed for classification tasks, the model is not substantially fooled even as the intensity of the attack is increased to 40 iterations. Figure 7(b), shows qualitatively that the model predictions are not significantly different from the initial predictions. The shape of the moving car is preserved to a considerable extent. The limited effectiveness of the PGD attack is further highlighted by increasing attack iterations to 40 (see Figure 7(c)). Here, some initial predictions are still preserved, for example, the bark of the tree. This is in contrast to when the model is attacked with CosPGD, a method that utilizes pixel-wise information. In Figure 7(e), we observe that even at a small number of attack iterations (5), the model predictions are significantly different from the initial predictions, especially in the background and the shape of the moving car. The model is incorrectly predicting the motion of the pixels around the moving car. At high attack intensity, as shown in Figure 7(f) with 40 iterations, the model's optical flow predictions are significantly inaccurate and exceedingly different from the initial predictions and very close to the target of $\overrightarrow{0}$. The model fails to differentiate the moving car from its background, moreover, the bark of the tree has completely vanished. In a real-world scenario, this vulnerability of the model to a relatively small pertur-

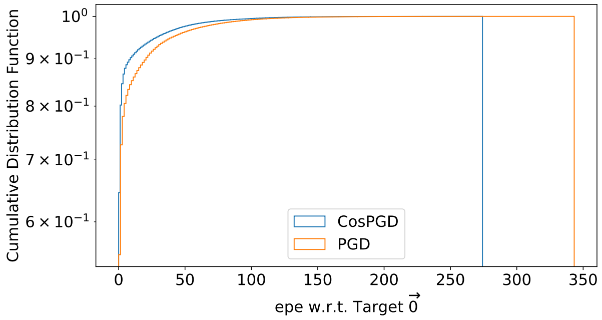


Figure 5: Comparing the distributions of epe w.r.t. Target flow $\overrightarrow{0}$ after ℓ_∞-norm constrained targeted 40 iterations CosPGD and PGD attacks on RAFT for optical flow estimation over KITTI-2015 validation dataset. A lower epe w.r.t. Target flow is desirable. We observe that CosPGD can reduce the gap to Target for more pixels than the PGD attack. Moreover, the highest epe w.r.t. Target after a CosPGD attack is significantly lower than after a PGD attack.

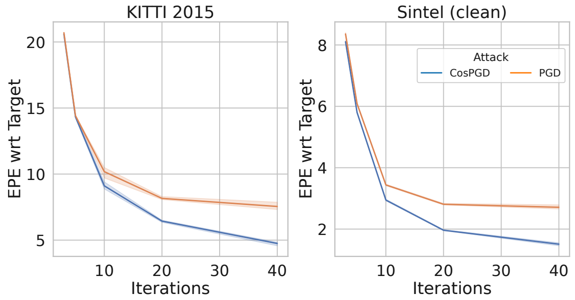
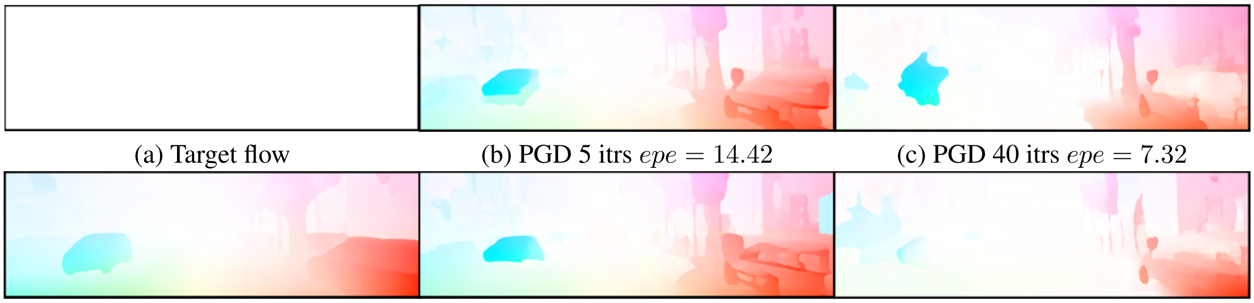


Figure 6: Comparison of performance of CosPGD to PGD for optical flow estimation over KITTI-2015 (left) and Sintel (clean → right) validation datasets as ℓ_∞-norm constrained targeted attacks using RAFT. CosPGD is a stronger targeted attack than PGD for optical flow. We also report these results in Table 13 in Appendix C.



(a) Target flow (b) PGD 5 itrs $epe = 14.42$ (c) PGD 40 itrs $epe = 7.32$

(d) Initial flow $epe = 31.1$ (e) CosPGD 5 itrs $epe = 14.28$ (f) CosPGD 40 itrs $epe = 4.84$

Figure 7: Comparing PGD and CosPGD as a targeted ℓ_∞-norm constrained attack on RAFT using KITTI15 validation set over various iterations. (a) shows the targeted prediction, a $\overrightarrow{0}$, and (d) shows the initial optical flow estimation by the network before adversarial attacks. EPEs between the target and the final prediction are reported, thus lower epe is better. (b) and (c) show flow predictions after PGD attack over 5 and 40 iterations respectively, while figures (e) and (f) show flow predictions after CosPGD attack over 5 and 40 iterations respectively. CosPGD significantly reduces the gap to target (a).

bation ($\epsilon = \frac{8}{255}$) could be hazardous. CosPGD provides us with this new insight. A similar observation is made for the Sintel dataset as shown in Figure 1. The benefit of CosPGD over PGD for optical flow can be quantitatively seen in Figure 6 and Table 13 in Appendix C.

5.3. Benchmarking on Further Tasks and Settings

Semantic Segmentation. We observed the strength of CosPGD as a ℓ_∞-norm constrained attack in Figures 3 & 4. Furthermore, we show that the improved performance of CosPGD is not limited to ℓ_∞-norm constrained attacks. Figure 10 in Appendix B.6.1 demonstrates the versatility of CosPGD as an ℓ_2-norm constrained attack.

We observe that across ℓ_p-norm constraints, the gap in performance of CosPGD w.r.t other adversarial attacks significantly increases when increasing the number of attack iterations. This demonstrates that CosPGD can utilize the increase in attack iterations best and highlights the significance of scaling the pixel-wise loss with the cosine alignment of predictions rather than using a heuristic, argmax-based scaling as in SegPGD.

Thus, we successfully demonstrate the benefit of CosPGD over existing adversarial attacks for semantic segmentation. We provide more results on ℓ_∞-norm and ℓ_2-norm constrained non-targeted adversarial attacks for semantic segmentation using UNet (Ronneberger et al., 2015) with ConvNeXt backbone on **CityScapes** (Cordts et al., 2016) in Appendix B.5, further confirming the benefit of CosPGD.

Additionally, we ablate over the attack step size α for ℓ_∞-norm constrained attacks on DeepLabV3 using PASCAL VOC2012 validation dataset in Appendix B.6.2 and over multiple attack step size α and permissible perturbation ϵ for l_2-norm constrained attacks on DeepLabV3 using PASCAL VOC2012 validation dataset in Appendix B.6. We show in Appendix B.6.1 that CosPGD outperforms both PGD and SegPGD (for segmentation) in the ℓ_2-norm constraint settings under all commonly used ϵ and α values.

Optical Flow. In addition to the results discussed in Section 5.2, we provide results comparing CosPGD to PGD as a ℓ_∞-constrained non-targeted attack for optical flow estimation in Appendix C.2. We also provide a comparison to PCFA (Schmalfuss et al., 2022b) in Appendix. C.3.

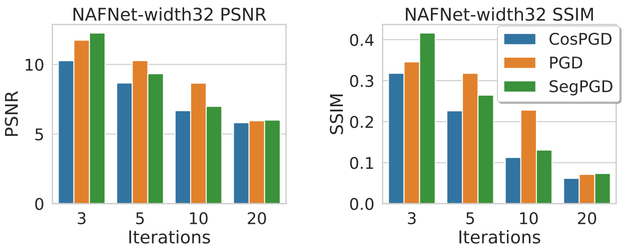


Figure 8: Non-targeted ℓ_∞-norm constrained CosPGD, PGD, and SegPGD attacks on NAFNet, recently proposed by (Chen et al., 2022) as the state-of-the-art network for image de-blurring on the GoPro dataset.CosPGD significantly outperforms the other attacks. Lower PSNR and SSIM indicate a worse restoration and thus a stronger attack.

Image Deblurring. To demonstrate CosPGD's versatility, we last consider the vision transformer-based image restoration model NAFNet (Chen et al., 2022). NAFNet outperforms Restormer (Zamir et al., 2022) for image restoration tasks like image de-blurring and image denoising on clean data, thus implying that NAFNet learns good representations. Figure 8 depicts results for NAFNet on image deblurring of the GoPro dataset images. We observe that CosPGD is a significantly stronger attack than both PGD and SegPGD on this task. We provide further discussion and results on Restormer (Zamir et al., 2022) and the “Baseline network” (Chen et al., 2022) in Appendix D.1.

6. Conclusion

In this work, we demonstrated across different downstream tasks and architectures that our proposed adversarial attack, CosPGD, is significantly more effective than other existing and commonly used adversarial attacks on several pixel-wise prediction tasks. We provide a new algorithm for evaluating the adversarial robustness of models on pixel-wise tasks. By comparing CosPGD to attacks like PGD, which were originally proposed for image classification tasks, we expanded on the work by Gu et al. (2022) and highlighted the need and effectiveness of attacks specifically designed for pixel-wise prediction tasks beyond segmentation. We illustrated the intuition behind using cosine similarity as a measure for generating stronger adversaries and leveraging more information from the model and backed it with experimental results from different downstream tasks. This further highlights the simplicity and principled formulation of CosPGD, making it applicable to a wide range of pixel-wise prediction tasks and in principle extendable to all Lipschitz continuous bounds as a targeted as well as a non-targeted attack.

Limitations. Most white-box adversarial attacks require access to ground truth labels (Goodfellow et al., 2014; Kurakin et al., 2017; Madry et al., 2017; Wong et al., 2020b; Gu et al., 2022). While this is beneficial for generating adversaries, it limits the applications of the non-targeted attacks like SegPGD as many benchmark datasets (Menze & Geiger, 2015; Butler et al., 2012; Wulff et al., 2012; Everingham et al., 2012) do not provide the ground truth for test data. The wide-applicability of CosPGD allows it to be used as a targeted attack thus mitigating this limitation to a great extent. Yet, it would be interesting to study the attack on the ground truth test images in the non-targeted setting as well, due to the potential slight distribution shifts pre-existing in the test data. We discuss additional limitations of CosPGD in Appendix E.

Acknowledgements

S.J and M.K acknowledge funding by the DFG Research Unit 5336 - Learning to Sense. The OMNI cluster of University of Siegen was used for some of the initial computations.

Impact Statement

We have carefully read the ICML 2024 Code of Ethics and confirm that we adhere to it. The proposed work is original and novel. To the best of our knowledge, all literature used in this work has been referenced correctly. Our work did not involve any human subjects and does not pose a threat to humans or the environment.

Assessing the quality of representations learned by a machine learning model is of paramount importance. This makes sure that the model is not learning shortcuts from the input distribution to the target distribution (Geirhos et al., 2020) but learning something meaningful. Adversarial attacks are a reliable tool for gauging the quality of a model’s learned representations. However adversarial attacks are time and computation exhaustive. Thus, our proposed adversarial attack, CosPGD helps in this regard as it can provide new insights into a model’s robustness and vulnerabilities with much less time and thus computation and is theoretically motivated. Thus, our work helps advance the field of machine learning.

Author Contribution

The idea for CosPGD was conceptualized by Shashank Agnihotri and improved by discussions with Steffen Jung and Margret Keuper. Shashank Agnihotri led the development, with inputs from Steffen Jung and Margret Keuper. Margret Keuper provided supervision and contributed significantly to the writing. Steffen Jung additionally made notable and significant contributions with experiments for non-targeted attacks on semantic segmentation, especially experiments with PSPNet, DeepLabV3 and Robust UPerNet. Shashank Agnihotri performed the remaining experiments.

References

Abdelhamed, A., Lin, S., and Brown, M. S. A high-quality denoising dataset for smartphone cameras. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018. doi: 10.1109/CVPR.2018.00182.

Agnihotri, S., Gandikota, K. V., Grabinski, J., Chandramouli, P., and Keuper, M. On the unreasonable vulnerability of transformers for image restoration – and an easy fix, 2023a.

Agnihotri, S., Grabinski, J., and Keuper, M. Improving stability during upsampling–on the importance of spatial context. *arXiv preprint arXiv:2311.17524*, 2023b.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pp. 484–501. Springer, 2020.

Arnab, A., Miksik, O., and Torr, P. H. S. On the robustness of semantic segmentation models to adversarial attacks, 2017. URL https://arxiv.org/abs/1711.09856.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch, 2017. URL https://arxiv.org/abs/1712.09665.

Buhrmester, V., Münch, D., and Arens, M. Analysis of explainers of black box deep neural networks for computer vision: A survey, 2019. URL https://arxiv.org/abs/1911.12116.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.) (ed.), *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pp. 611–625. Springer-Verlag, October 2012.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Chen, L., Chu, X., Zhang, X., and Sun, J. Simple baselines for image restoration, 2022.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation, 2017.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding, 2016.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *CoRR*, abs/2010.09670, 2020. URL https://arxiv.org/abs/2010.09670.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Croce, F., Singh, N. D., and Hein, M. Robust semantic segmentation: Strong adversarial attacks and fast training of robust models, 2023. URL https://arxiv.org/abs/2306.12941.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9185–9193, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00957. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00957.

Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks, 2015. URL https://arxiv.org/abs/1504.06852.

Gajjar, S., Hati, A., Bhilare, S., and Mandal, S. Generating targeted adversarial attacks and assessing their effectiveness in fooling deep neural networks. In *2022 IEEE International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5, 2022. doi: 10.1109/SPCOM55316.2022.9840784.

Gavrikov, P., Lukasik, J., Jung, S., Geirhos, R., Lamm, B., Mirza, M. J., Keuper, M., and Keuper, J. Are vision language models texture or shape biased and can we steer them? *arXiv preprint arXiv:2403.09193*, 2024.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2018. URL https://arxiv.org/abs/1811.12231.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, nov 2020. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038%2Fs42256-020-00257-z.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, 2014. URL https://arxiv.org/abs/1412.6572.

Grabinski, J., Jung, S., Keuper, J., and Keuper, M. Frequencylowcut pooling–plug & play against catastrophic overfitting. *arXiv preprint arXiv:2204.00491*, 2022.

Grabinski, J., Keuper, J., and Keuper, M. Fix your downsampling asap! be natively more robust via aliasing and spectral artifact free pooling, 2023.

Gu, J., Zhao, H., Tresp, V., and Torr, P. H. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, pp. 308–325. Springer, 2022.

Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., and Malik, J. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Hypercolumns for object segmentation and fine-grained localization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 447–456, 2015. doi: 10.1109/CVPR.2015.7298642.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations, 2019. URL https://arxiv.org/abs/1903.12261.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples, 2019. URL https://arxiv.org/abs/1907.07174.

Hoffmann, J., Agnihotri, S., Saikia, T., and Brox, T. Towards improving robustness of compressed cnns. In *ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL)*, 2021.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. URL https://arxiv.org/abs/1612.01925.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. URL https://arxiv.org/abs/1804.08598.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://aclanthology.org/N18-1170.

Jia, J., Qu, W., and Gong, N. Multiguard: Provably robust multi-label classification against adversarial examples. *Advances in Neural Information Processing Systems*, 35: 10150–10163, 2022.

Jiang, S., Campbell, D., Lu, Y., Li, H., and Hartley, R. Learning to estimate hidden motions with global motion aggregation, 2021.

Jung, S. and Keuper, M. Spectral distribution aware image generation, 2020. URL https://arxiv.org/abs/2012.03110.

Jung, S. and Keuper, M. Internalized biases in fréchet inception distance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Jung, S., Ziegler, S., Kardoost, A., and Keuper, M. Optimizing edge detection for image segmentation with multicut penalties. In *DAGM German Conference on Pattern Recognition*, pp. 182–197. Springer, 2022.

Jung, S., Lukasik, J., and Keuper, M. Neural architecture design and robustness: A dataset. *arXiv preprint arXiv:2306.06712*, 2023a.

Jung, S., Schwedhelm, J. C., Schillings, C., and Keuper, M. Happy people–image synthesis as black-box optimization problem in the discrete latent space of deep generative models. *arXiv preprint arXiv:2306.06684*, 2023b.

Kang, D., Sun, Y., Hendrycks, D., Brown, T., and Steinhardt, J. Testing robustness against unforeseen adversaries, 2019. URL https://arxiv.org/abs/1908.08016.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. *arXiv:2304.02643*, 2023.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world, 2016. URL https://arxiv.org/abs/1607.02533.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale, 2017. URL https://doi.org/10.48550/arXiv.1611.01236.

Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F. X., Taylor, R. H., and Unberath, M. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6197–6206, October 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Lukasik, J., Jung, S., and Keuper, M. Learning where to look–generative nas is surprisingly efficient. In *European Conference on Computer Vision*, pp. 257–273. Springer, 2022.

Lukasik, J., Gavrikov, P., Keuper, J., and Keuper, M. Improving native cnn robustness with filter frequency regularization. *Transactions on Machine Learning Research*, 2023a.

Lukasik, J., Moeller, M., and Keuper, M. An evaluation of zero-cost proxies-from neural architecture performance prediction to model robustness. In *DAGM German Conference on Pattern Recognition*, pp. 624–638. Springer, 2023b.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2017. URL https://arxiv.org/abs/1706.06083.

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16. arXiv:1512.02134.

Mehl, L., Schmalfuss, J., Jahedi, A., Nalivayko, Y., and Bruhn, A. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4981–4991, 2023.

Menze, M. and Geiger, A. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks, 2015. URL https://arxiv.org/abs/1511.04599.

Morris, J. X., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020. URL https://arxiv.org/abs/2005.05909.

Nah, S., Kim, T. H., and Lee, K. M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, July 2017.

Qu, W., Li, Y., and Wang, B. A certified radius-guided attack framework to image segmentation models. *arXiv preprint arXiv:2304.02693*, 2023.

Ranjan, A. and Black, M. J. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Ribeiro, M. T., Singh, S., and Guestrin, C. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://aclanthology.org/P18-1079.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses, 2019.

Rony, J., Pesquet, J., and Ayed, I. Proximal splitting adversarial attack for semantic segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20524–20533, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01966. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01966.

Scheurer, E., Schmalfuss, J., Lis, A., and Bruhn, A. Detection defenses: An empty promise against adversarial patch attacks on optical flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6489–6498, 2024.

Schmalfuss, J., Mehl, L., and Bruhn, A. Attacking motion estimation with adversarial snow. *arXiv preprint arXiv:2210.11242*, 2022a.

Schmalfuss, J., Scholze, P., and Bruhn, A. A perturbation-constrained adversarial attack for evaluating the robustness of optical flow. In *European Conference on Computer Vision*, pp. 183–200. Springer, 2022b.

Schmalfuss, J., Mehl, L., and Bruhn, A. Distracting downpour: Adversarial weather attacks for motion estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10106–10116, 2023.

Schrodi, S., Saikia, T., and Brox, T. Towards understanding adversarial robustness of optical flow networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8916–8924, 2022.

Sommerhoff, H., Agnihotri, S., Saleh, M., Moeller, M., Keuper, M., and Kolb, A. Differentiable sensor layouts for end-to-end learning of task-specific camera parameters. *arXiv preprint arXiv:2304.14736*, 2023.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, 2018.

Sun, Y., Chen, F., Chen, Z., and Wang, M. Local aggressive adversarial attacks on 3d point cloud, 2021. URL https://arxiv.org/abs/2105.09090.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks, 2014.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow, 2020. URL https://arxiv.org/abs/2003.12039.

username: mberkay0, B. M. mberkay0/pretrained-backbones-unet. https://github.com/mberkay0/pretrained-backbones-unet, 2023.

Vo, J., Xie, J., and Patel, S. Multiclass asma vs targeted pgd attack in image segmentation, 2022. URL https://arxiv.org/abs/2208.01844.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., and Yan, S. Better diffusion models further improve adversarial training, 2023.

Wong, A., Cicek, S., and Soatto, S. Targeted adversarial perturbations for monocular depth prediction. In *Advances in neural information processing systems*, 2020a.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training, 2020b. URL https://arxiv.org/abs/2001.03994.

Wulff, J., Butler, D. J., Stanley, G. B., and Black, M. J. Lessons and insights from creating a synthetic optical flow benchmark. In A. Fusiello et al. (Eds.) (ed.), *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Part II, LNCS 7584, pp. 168–177. Springer-Verlag, October 2012.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018.

Xie, C., Wu, Y., van der Maaten, L., Yuille, A., and He, K. Feature denoising for improving adversarial robustness, 2019.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks, 2016. URL https://arxiv.org/abs/1611.05431.

Xu, X., Zhao, H., and Jia, J. Dynamic divide-and-conquer adversarial training for robust semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7466–7475, 2021. doi: 10.1109/ICCV48922.2021.00739.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

Zhang, J., Chen, L., Liu, B., Ouyang, B., Xie, Q., Zhu, J., Li, W., and Meng, Y. 3d adversarial attacks beyond point cloud, 2021. URL https://arxiv.org/abs/2104.12146.

Zhao, H. semseg. https://github.com/hszhao/semseg, 2019.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *CVPR*, 2017.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

CosPGD: an efficient and unified white-box adversarial attack for pixel-wise prediction tasks

Supplementary Material

We include the following information in the supplementary material:

- Section A Additional Details:
  - Section A.1: We provide the proof for proposition 4.1.
  - Section A.2: Algorithm of CosPGD.
  - Section A.3: Hardware details
  - Section A.3.1: Implementation details including code and example usage.
  - Section A.3.3: We provide additional experimental details for the image deblurring experiments.
  - Section A.3.4: We compare the time taken by different adversarial attacks for different tasks.
  - Section A.3.2: Details on calculating *epe-f1-all*.
- Section B: Semantic Segmentation Additional Results:
  - Section B.1: We provide additional experimental results using SegFormer (Xie et al., 2021) on ADE20K (Zhou et al., 2017; 2019).
    - Section B.1.2: We report an ablation study over multiple ϵ values for ℓ_∞-norm bounded attacks
  - Section B.2: We provide evaluations on transferring adversarial attacks between a DeepLabV3 and a PSPNet model on PASCALVOC2012 dataset.
  - Section B.3: We report the performance of adversarial attacks against some SotA defense methods.
  - Section B.4: Here we report transfer attacks from a DeepLabV3 to Segment Anything Model (SAM) (Kirillov et al., 2023).
  - Section B.5: We provide extra l_∞-norm and l_2-norm constrained non-targeted adversarial attack results from Semantic Segmentation using the UNet architecture with ConvNeXt backbone on the **CityScapes** dataset (Cordts et al., 2016).
  - Section B.6: We provide an ablation study on attack step size α and ϵ for l_2-norm bounded for non-targeted adversarial attack results from Semantic Segmentation using DeepLabV3 on the PASCAL VOC 2012 dataset.
  - Section B.6.2: We provide an ablation study on attack step size α for l_∞-norm bounded for non-targeted adversarial attack results from Semantic Segmentation using DeepLabV3 on the PASCAL VOC 2012 dataset.
  - Section B.7: We report results from Figure 3 in a tabular form.
  - Section B.8: We report the results of adversarial training for semantic segmentation.
- Section C: Optical Flow Additional Results:
  - Section C.1: We report results from Figure 6 in a tabular form.
  - Section C.2: We provide extra results comparing CosPGD to PGD as a l_∞-norm constrained non-targeted adversarial attack for optical flow estimation.
  - Section C.3: We provide a comparison to the l_2-constrained PCFA (Schmalfuss et al., 2022b), which is a dedicated attack for optical flow.
- Section D: Image Restoration Results:
  - Section D.1: We report the findings on the adversarial robustness of many recently proposed transformer-based image deblurring models.
  - Section D.2: We report the results on many recently proposed transformer-based image denoising models.
- Section E: A detailed discussion on limitations of CosPGD

In Table 1, we provide a look-up table for all experiments considered in this supplementary material. We provide details on the downstream tasks, models, targeted and non-targeted attack settings, and l_∞-norm constrained and l_2-norm constrained settings considered respectively do demonstrate the wide-applicability of CosPGD.

A. Appendix

Table 1: Look-up table for considered experiments in this appendix.

Downstream Task	Networks	Dataset	Study	Non-targeted Attack		Targeted Attack	
				l_∞-norm constraint	l_2-norm constraint	l_∞-norm constraint	l_2-norm constraint
Semantic Segmentation	DeepLabV3	PASCAL VOC 2012, Cityscapes	various ϵ and α values		Sec. B.6.1		
	PSPNet		Non-targeted Attacks	Sec. B.6.2			
	UNet		Non-targeted Attacks				
	SegFormer	ADE20K	various ϵ values	Sec. B.1.2			
	Robust UPerNet (Croce et al., 2023)	PASCAL VOC 2012	Performance against Defense Methods	Sec. B.3			
	Robust PSPNet (Xu et al., 2021)	PASCAL VOC 2012	Performance against Robust Models	Sec. B.3			
	DeepLabV3 → SAM	PASCAL VOC 2012	Transfer Attack on SAM	Sec. B.4			
	DeepLabV3 → PSPNet	PASCAL VOC 2012	Transfer Attacks	Sec. B.2			
	PSPNet → DeepLabV3	PASCAL VOC 2012	Transfer Attacks	Sec. B.2			
Optical Flow Estimation	RAFT	KITTI 2015, Sintel (clean and final)	Targeted Attacks	Sec. C.2		Sec. C	Sec. C.3
	PWCNet, GMA, SpyNet		Comparison to PCFA				
Image Deblurring	Restormer, Baseline net, NAFNet	GoPro	Non-targeted Attacks	Sec. D.1			
Image Denoising	Baseline net, NAFNet	SSID	Non-targeted Attacks	Sec. D.2			

A.1. Proof of Proposition 4.1

We are to show that, for any two pixel-wise network predictions $f_\theta(\boldsymbol{X})_i$ and $f_\theta(\bar{\boldsymbol{X}})_i \in \mathbb{R}^M$, a target $\boldsymbol{Y}_i \in \mathbb{R}^M$ and a continuously differentiable function $\psi : \mathbb{R}^M \to \mathbb{R}^M$ with $\psi(f_\theta(\boldsymbol{X})) = 1 \quad \forall f_\theta(\boldsymbol{X})$, there exists a real, constant $d \geq 0$ so that

$$
\begin{aligned}
& d \cdot \| f_\theta(\boldsymbol{X})_i - f_\theta(\bar{\boldsymbol{X}})_i \| \geq \\
& \qquad \| \cos\left(\psi(f_\theta(\boldsymbol{X})_i), \boldsymbol{Y}_i\right) - \cos\left(\psi(f_\theta(\bar{\boldsymbol{X}})_i), \boldsymbol{Y}_i\right) \|.
\end{aligned}
$$

Proof. The function $\psi : \mathbb{R}^M \to \mathbb{R}^M$ as well as the cosine similarity $\cos : \mathbb{R}^M \times \mathbb{R}^M \to [-1, 1]$ are both continuously differentiable functions. From the continuous differentiability of ψ, it follows that is it Lipschitz continuous, i.e. there exists a real constant $d_1 \geq 0$ so that

$$
d_1 \cdot \| f_\theta(\boldsymbol{X})_i - f_\theta(\bar{\boldsymbol{X}})_i \| \geq \| \psi(f_\theta(\boldsymbol{X})_i) - \psi(f_\theta(\bar{\boldsymbol{X}})_i) \|
$$

for any $f_\theta(\boldsymbol{X})_i$ and $f_\theta(\bar{\boldsymbol{X}})_i \in \mathbb{R}^M$. Further, the cosine similarity effectively computes the norm of the projection of the normalized model predictions onto the target vector, which is again a continuously differentiable operation, i.e. is again Lipschitz continuous

$$
\begin{aligned}
& d_2 \cdot \| \psi(f_\theta(\boldsymbol{X})_i) - \psi(f_\theta(\bar{\boldsymbol{X}})_i) \| \geq \\
& \qquad \| \cos\left(\psi(f_\theta(\boldsymbol{X})_i), \boldsymbol{Y}_i\right) - \cos\left(\psi(f_\theta(\bar{\boldsymbol{X}})_i), \boldsymbol{Y}_i\right) \|.
\end{aligned}
$$

for a real constant $d_2 \geq 0$. □

A.2. Algorithm for CosPGD

Following we present the algorithm for CosPGD. Algorithm 1 provides a general overview of the implementation of CosPGD. It demonstrates that CosPGD is downstream-task agnostic, l_p-norm agnostic, and agnostic to targeted or non-targeted application.

A.3. Further Experimental Details on Hardware and Metrics

Semantic Segmentation We use PASCAL VOC 2012 (Everingham et al., 2012), which contains 20 object classes and one background class, with 1464 training images, and 1449 validation images. We follow common practice (Hariharan et al., 2015; Gu et al., 2022; Zhao, 2019; Zhao et al., 2017), and use work by Hariharan et al. (2011), augmenting the training set to 10,582 images. We evaluate on the validation set. Architectures used for our evaluations are PSPNet (Zhao et al., 2017) and DeepLabV3 (Chen et al., 2017), both with ResNet50 (He et al., 2015) encoders, and UNet (Ronneberger et al., 2015) with a ConvNeXt tiny encoder (Liu et al., 2022). Results are reported in Appendix B.5. We report mean Intersection over Union (mIoU) and mean pixel accuracy (mAcc).

Hardware. For the experiments on DeepLabV3, we used NVIDIA Quadro RTX 8000 GPUs. For PSPNet, we used NVIDIA A100 GPUs. For the experiments with UNet, we used NVIDIA GeForce RTX 3090 GPUs.

Algorithm 1 Algorithm for generating adversarial examples using CosPGD.

```
Require: model $f_{\text{net}}(\cdot)$, clean samples $\boldsymbol{X}^{\text{clean}}$, perturbation range $\epsilon$, step size $\alpha$, attack iterations $T$, ground truth/target $\boldsymbol{Y}$
  $\boldsymbol{X}^{\text{adv}_0} = \boldsymbol{X}^{\text{clean}} + \mathcal{U}(-\epsilon, +\epsilon)$    ▷ initialize adversarial example and clip to valid $\ell_\infty$ or $l_2$ bound
  for t ← 0 to T-1 do    ▷ loop over attack iterations
      $P = f_{\text{net}}(\boldsymbol{X}^{\text{adv}_t})$    ▷ make predictions
      $\text{cossim} \leftarrow \mathit{CosineSimilarity}(\psi(P), \Psi^{'}(\boldsymbol{Y}))$    ▷ compute cosine similarity
      if targeted attack:
              $\text{cossim} \leftarrow 1 - \text{cossim}$    ▷ punish dissimilarity to target
              $\alpha \leftarrow -\alpha$    ▷ opposite direction for targeted attack
      $L_{\text{cos}} \leftarrow \text{cossim} \cdot L(P, \boldsymbol{Y})$    ▷ scaling the pixel-wise loss for sample updates
      $\boldsymbol{X}^{\text{adv}_{t+1}} \leftarrow \boldsymbol{X}^{\text{adv}_t} + \alpha \cdot \mathit{sign}(\nabla_{\boldsymbol{X}^{\text{adv}_t}} L_{\text{cos}})$    ▷ update adversarial examples
      $\delta \leftarrow \phi^{\epsilon}(\boldsymbol{X}^{\text{adv}_{t+1}} - \boldsymbol{X}^{\text{clean}})$    ▷ clip $\delta$ to valid $\ell_\infty$ or $l_2$ bound
      $\boldsymbol{X}^{\text{adv}_{t+1}} = \phi^{\epsilon}(\boldsymbol{X}^{\text{clean}} + \delta)$    ▷ add $\delta$ to $\boldsymbol{X}^{\text{clean}}$ and clip into valid image range
  end for
  $P = f_{\text{net}}(\boldsymbol{X}^{\text{adv}_T})$    ▷ make predictions on adversarial examples
```

Optical Flow We use RAFT (Teed & Deng, 2020) and follow the evaluation procedure used therein. Evaluations are performed on KITTI2015 (Menze & Geiger, 2015) and MPI Sintel (Butler et al., 2012; Wulff et al., 2012) validation sets. We use the networks pre-trained on FlyingChairs (Dosovitskiy et al., 2015) and FlyingThings (Mayer et al., 2016) and fine-tuned on training datasets of the specific evaluation, as provided by Teed & Deng (2020). For Sintel we report the end-point error (epe) on both clean and final subsets, while for KITTI15 we report the epe and epe-$f1$-all. In Appendix C.3 we compare CosPGD to PCFA across different networks.

Hardware. We used NVIDIA V100 GPUs, a single GPU was used for each run.

Image Restoration Following the regime of (Chen et al., 2022; Zamir et al., 2022; Agnihotri et al., 2023a), for the image de-blurring task we use the GoPro dataset (Nah et al., 2017) as in (Chen et al., 2022). The images are split into 2103 training images and 1111 test images. We consider the "Baseline network" and NAFNet as proposed by (Chen et al., 2022). For the image restoration tasks we report the $PSNR$ and $SSIM$ scores of the reconstructed images w.r.t. to the ground truth images, averaged over all images. We provide further details in Appendix D.1.

Hardware. For the experiments on Image de-blurring tasks, we used NVIDIA GeForce RTX 3090 GPUs. A single GPU was used for each run.

A.3.1. Code for the attack

The code for the functions used for generating adversarial samples using CosPGD and other considered adversarial attacks in the main paper is available at `https://github.com/shashankskagnihotri/cospgd`.

Additionally, we provide sample code demonstrating the usage of the packages for a UNet-like architecture with detailed instructions at `https://github.com/shashankskagnihotri/cospgd`.

A.3.2. Calculating epe-f1-all

Following the work by Teed & Deng (2020), $f1 - all$ is calculated by averaging out over all the predicted optical flows. out is calculated using Equation (12),

$$out = epe > 3.0 \cup \frac{epe}{mag} > 0.05 \tag{12}$$

Where, mag = $\sqrt{flow\ ground\ truth^2}$ and epe is the Euclidean distance between the two vectors.

A.3.3. Image Deblurring Experimental Details

Chen et al. (2022) simplify a transformer-based architecture Restormer (Zamir et al., 2022) for image restoration tasks and first propose a simplified architecture as a Baseline network, and then improve upon it with intuitions backed by reasoning and ablation studies to propose Non-linear Activation Free Networks abbreviated as NAFNet. In this work, we perform adversarial attacks on both the Baseline network and NAFNet.

Dataset. Similar to (Chen et al., 2022), for the image de-blurring task, we use the GoPro dataset (Nah et al., 2017) which consists of 3124 realistically blurry images of resolution 1280×720 and corresponding ground truth sharp images obtained using a high-speed camera. The images are split into 2103 training images and 1111 test images. For the image denoising task, we use the Smartphone Image Denoising Dataset (SSID) (Abdelhamed et al., 2018). This dataset consists of 160 noisy images taken from 5 different smartphones and their corresponding high-quality ground truth images.

Metrics. For both the image restoration tasks, we report the $PSNR$ and $SSIM$ scores of the reconstructed images w.r.t. to the ground truth images, averaged over all images. $PSNR$ stands for Peak Signal-to-Noise ratio, a higher $PSNR$ indicates a better quality image or an image closer to the image to which it is being compared. $SSIM$ stands for Structural similarity (Wang et al., 2004).

A.3.4. Comparing Time taken by different Adversarial Attacks

Following, we report the approximate time taken by each attack in minutes. Please note, this time includes time taken for data-loading and saving of experimental results including images. For a given task, network, and dataset, the time taken by different attacks is comparable and representative of the time taken by the attacks as they followed the same attack procedures. We observe in Table 2 that the difference in time taken by the different attacks at the same number of iterations is negligible. This is because operations like one-hot encoding and softmax take negligible time.

Thus, the ability of CosPGD to provide valuable insights into model robustness with significantly less iterations than other methods, as discussed in Section 5.2 and Section 5.3 is a compelling advantage.

Table 2: Comparison of time taken in minutes by different attacks on different downstream tasks for different amount of iterations. The computation times are comparable.

				Attack iterations				
Task	Network	Dataset	Attack method	3	5	10	20	40
				Time (mins)	Time (mins)	Time (mins)	Time (mins)	Time (mins)
Semantic Segmenation	UNet	PASCAL VOC 2012	SegPGD	28.73	36.33	58.72	88.93	163.15
			CosPGD	26.67	36.75	54.45	97.08	165.35
Optical Flow	RAFT	KITTI2012	PGD	5.90	7.73	12.23	20.98	37.45
			CosPGD	6.00	7.85	12.15	21.03	38.28
		Sintel (clean + final)	PGD	69.87	97.47	158.28	297.40	557.97
			CosPGD	73.68	102.77	160.40	287.82	602.08

B. Semantic Segmentation

Following we provide additional Semantic Segmentaion evaluations, including study on different ϵ values, different α values, using different tasks and transfer attacks on SAM using a DeepLabV3.

B.1. Semantic Segmentation with SegFormer on ADE20k

B.1.1. Implementation Details

For experiments with SegFormer (Xie et al., 2021) with MIT-B0 backbone, we use the ADE20k dataset (Zhou et al., 2019). This dataset has 150 classes and is split into 25,574 training images and 2,000 validation images.

We perform ℓ_∞-bounded PGD, SegPGD and CosPGD with various ϵ values $\in \{\frac{2}{255}, \frac{4}{255}, \frac{6}{255}, \frac{8}{255}, \frac{10}{255}, \frac{12}{255}\}$, over various attack iterations $\in$ {3, 5, 10, 20, 40, 100}.

B.1.2. Ablation over multiple ϵ values for ℓ_∞-norm bounded attacks

Since ADE20K has 150 classes, making it a more difficult distribution to learn, it is not usually considered to evaluate attack methods. We expect CosPGD to be a significantly stronger attack than SegPGD or the simple PGD on this data because it can smoothly align the loss to the posterior distribution. In Table 3 we confirm this by providing additional experiments

Table 3: Attacking SegFormer with a MIT-B0 backbone using ADE20K with different ℓ_∞ bounded ϵ values and with different adversarial attacks.

Attack Method	$\frac{\epsilon}{255}$ value	Attack Iterations											
		3		5		10		20		40		100	
		mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)
PGD	2	8.45	14.44	6.62	11.49	5.36	9.45	4.21	7.51	3.8	6.73	3.3	6.12
SegPGD		5.80	10.15	4.88	8.68	3.69	6.56	2.91	5.18	2.41	4.49	2.19	4.02
CosPGD		**5.37**	**10.06**	**3.75**	**7.26**	**2.18**	**4.3**	**1.87**	**3.55**	**1.68**	**3.01**	**1.37**	**2.46**
PGD	4	5.11	9.48	2.94	5.63	1.66	3.34	1.01	2.21	0.79	1.79	0.6	1.38
SegPGD		3.29	6.15	1.83	3.7	0.89	1.9	0.47	1.18	0.3	0.86	0.26	0.68
CosPGD		**1.66**	**3.45**	**0.55**	**1.28**	**0.09**	**0.22**	**0.05**	**0.09**	**0.05**	**0.09**	**0.04**	**0.06**
PGD	6	3.97	7.5	2.05	4.1	1.07	2.28	0.67	1.57	0.41	1.14	0.36	0.88
SegPGD		2.64	5.10	1.22	2.71	0.47	1.24	0.21	0.7	0.13	0.49	0.09	0.35
CosPGD		**1.11**	**2.39**	**0.18**	**0.52**	**0.01**	**0.04**	**0.0**	**0.01**	**0.0**	**0.0**	**0.0**	**0.0**
PGD	8	3.38	6.48	1.76	3.63	0.82	1.95	0.46	1.28	0.37	1.04	0.2	0.7
SegPGD		2.31	4.54	0.90	2.06	0.33	1.03	0.15	0.61	0.09	0.35	0.05	0.28
CosPGD		**0.98**	**2.21**	**0.08**	**0.25**	**0.00**	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
PGD	10	3.29	6.28	1.74	3.58	0.79	1.99	0.47	1.27	0.34	1.01	0.24	0.74
SegPGD		1.91	3.88	0.89	2.09	0.32	0.96	0.18	0.65	0.08	0.38	0.05	0.27
CosPGD		**0.81**	**1.82**	**0.11**	**0.41**	**0.00**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
PGD	12	3.16	5.95	1.49	2.98	0.72	1.79	0.45	1.27	0.31	0.93	0.24	0.69
SegPGD		1.83	3.77	1.83	3.77	0.26	0.83	0.14	0.6	0.1	0.44	0.04	0.26
CosPGD		**0.72**	**1.68**	**0.08**	**0.22**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

using SegFormer with ℓ_∞-norm bounded $\epsilon = \frac{8}{255}$ attacks with α=0.01 for Untargeted Attacks. Note that the chosen attack settings are the default values proposed in SegPGD.

We observe that CosPGD is a significantly stronger attack than SegPGD for ADE20K and SegFormer. Please also note that white-box attacks are extremely useful in exposing a model's vulnerabilities, however, they are very expensive to run, and thus 40 or more attack iterations are generally considered to be a very high number of attack iterations in white-box attack literature (please refer to PGD, APGD, PCFA, SegPGD, AutoAttack, MI-FGSM). Here, CosPGD required merely 10 attack iterations to bring the model mIoU to absolute 0.00, whereas SegPGD is not able to achieve this even when using 100 iterations (increasing the attack cost by a factor of 10). Our current understanding is that given a reasonable perturbation attack, and step size smaller than this budget (so that the perturbations are not clipped away by the budget), all attacks should optimize the adversary in the best possible way. We have shown that CosPGD is better at this optimization than the other white-box attacks for various step-sizes(α) and various ϵ values.

For ℓ_∞-norm we have shown this for $\epsilon = \frac{8}{255}$. The maximum permissible perturbation budget should not affect the relative performance of different attacks. We further solidify this claim here by providing additional experiments using SegFormer on ADE20K with ℓ_∞-norm bounded $\epsilon = \{\frac{2}{255}, \frac{4}{255}, \frac{6}{255}, \frac{8}{255}, \frac{10}{255}, \frac{12}{255}\}$ attack settings with α=0.01 for Untargeted Attacks in Table 3.

B.2. Evaluating Transfer Attacks

Table 4: Transfer Attacks on DeepLabV3 and PSPNet using 20 iterations attacks with ℓ_∞-norm bounded $\epsilon = \frac{8}{255}$ and α=0.01 using PASCAL VOC 2012 validation dataset.

Attacked Model	Attacking Model	Attack Method	mIoU (%)	mAcc (%)
DeepLabV3 ResNet50 (Clean mIoU: 76.17)	PSPNet ResNet50	**CosPGD**	**1.67**	**3.59**
		SegPGD	1.93	5.72
		PGD	5.11	12.75
PSPNet ResNet50 (Clean mIoU: 76.78)	DeepLabV3 ResNet50	**CosPGD**	**1.21**	**3.33**
		SegPGD	1.77	5.62
		PGD	4.58	12.07

CosPGD, like PGD, SegPGD, and FGSM, is a white box attack. They are designed to optimize attacks for a specific model and generalizability of the attacks to other models i.e. using them in a black-box setting is not a requirement for them at least not something they are optimized to do. However, it could be interesting to see if the adversarial examples that are optimized on a particular network, also cause a failure in the other. Thus in Table 4, we report results for the PASCAL VOC 2012 dataset when attacking PSPNet using DeepLabV3, and vice versa, both with a ResNet50 encoder. We observe that CosPGD is a significantly better attack even in this black-box setting. Here we consider ℓ_∞-norm bounded $\epsilon = \frac{8}{255}$ attacks with α=0.01. The benefit of CosPGD over previous methods becomes more significant as the number of attack iterations increases, but is measurable across attack iterations. We show this in Table 5.

Table 5: Transfer Attacks from DeepLabV3 on PSPNet over various iterations with ℓ_∞-norm bounded $\epsilon = \frac{8}{255}$ and α=0.01 using PASCAL VOC 2012 validation dataset.

Attacked Model	Attacking Model	Attack Method	Attack Iterations							
			3		10		20		40	
			mIoU (&)	mAcc (%)	mIoU (&)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (&)	mAcc (%)
PSPNet ResNet50 (Clean mIoU: 76.78)	DeepLabV3 ResNet50	**CosPGD**	**9.66**	**19.39**	**2.39**	**5.91**	**1.21**	**3.33**	**1.00**	**2.59**
		SegPGD	9.92	19.79	2.40	6.67	1.77	5.62	1.23	4.40
		PGD	14.67	27.79	5.56	13.60	4.58	12.07	4.35	11.81

B.3. Evaluating against Defense Methods

Table 6: Comparing the "Robust" PSPNet from (Xu et al., 2021) against white-box adversarial attacks over different number of iterations. Here, same as (Xu et al., 2021), $\epsilon = \frac{8}{255}$ and α=0.01. We use the model weights provided by (Xu et al., 2021) in their official GitHub repository.

Training Method	Clean Performance		Attack Method	Attack Iterations							
				2		4		6		10	
	mIoU (%)	mAcc (%)		mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)
No Defense	76.90	84.60	**CosPGD**	**9.11**	**20.77**	**1.56**	**5.02**	**0.54**	**2.03**	**0.13**	**0.40**
			SegPGD	10.39	22.14	3.86	9.69	2.62	6.97	1.88	5.36
			BIM	18.90	34.92	7.59	18.61	5.57	14.98	4.14	12.22
SAT (Xu et al., 2021)	74.78	83.36	**CosPGD**	**64.68**	**80.13**	**42.74**	**64.96**	**29.17**	**52.66**	**17.05**	**38.75**
			SegPGD	66.24	81.72	42.71	65.75	30.74	54.31	20.59	43.13
			BIM	69.89	86.68	48.62	67.34	31.54	50.80	20.67	40.05
DDC-AT (Xu et al., 2021)	75.98	84.72	**CosPGD**	**66.93**	**77.60**	**50.79**	**65.13**	**36.12**	**53.26**	**23.04**	**41.02**
			SegPGD	67.09	78.36	50.89	65.14	37.70	54.48	25.40	42.72
			BIM	74.04	83.09	51.57	65.67	39.07	55.97	26.90	45.27

In Table 6, we report the results on the evaluation of CosPGD on (Xu et al., 2021). Here we observe that defense methods as in (Xu et al., 2021) might help in reducing some effect of the attacks but not nearly strong enough to negate them and CosPGD is still the strongest adversarial attack.

Please note, we observed some errors in the white-box attack implementation in the official GitHub repository of (Xu et al., 2021). Thus, we were able to reproduce their reported clean accuracies of the three models, i.e. PSPNet with No Defense during training, PSPNet trained with SAT and PSPNet trained with DDC-AT (Xu et al., 2021). However, as their attack implementation code is wrong, specifically, the normalization done assumes the images to be in the space [0, 1], but in reality they are in [0, 255]. Thus, the performance reported by (Xu et al., 2021), under white-box adversarial attacks is incorrect. Therefore, we correct these errors and re-run their experiments and extend to them, going as far as 10 attack iterations. We correct the code from (Xu et al., 2021) and provide the corrected code here: `https://github.com/shashankskagnihotri/adv-corrected-ddcat-cospgd`.

In Table 7, we present this evaluation on (Croce et al., 2023) against their robust "UPerNet (Xiao et al., 2018) with a ConvNext-tiny backbone" encoder checkpoint that they make available in their official GitHub repository. We modify their Segmentation Ensemble Attack (SEA) (Croce et al., 2023) to only include the respective attack mentioned for the given number of attack iterations. The optimizer they used is always APGD.

We extent Table 7 in Table 8, here we report the results for $\epsilon = \frac{4}{255}$ and observe that the performance is comparable at the extremely high number of iterations i.e. 1200 attack iterations.

W.r.t. the comparison to (Croce et al., 2023) for $\epsilon = 4/255$ and very high number of iterations, we would like to highlight that, since the model is trained for this value, the differences between the attacks are actually small. Indeed, for high attack iterations, SegPGD is slightly stronger, yielding a maximum difference of 0.25% in mAcc for 300 iterations versus CosPGD, while at 10 attack iterations, CosPGD is also only slightly stronger than SegPGD in the same range. However, assuming that

Table 7: Attacking Robust UPerNet (Xiao et al., 2018) with ConvNeXt-tiny encoder from (Croce et al., 2023) with different fixed attacks in the Segmentation Ensemble Attack (SEA) over different permissible perturbation budgets (ϵ) and attack iterations. **Bold** results are the strongest attacks, while Underlined results are second strongest.

Attack Used	Optimizer Used	Attack Iterations	$\frac{\epsilon}{255}$							
			4		8		12		16	
			mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)
SEA: only **CosPGD (with Softmax) (OURS)** in (Croce et al., 2020)	APGD	10	**64.17**	**88.52**	**43.73**	**76.36**	**21.51**	**55.27**	**11.20**	**41.40**
		20	**64.15**	**88.53**	**41.94**	**74.89**	**16.27**	**45.71**	**6.54**	**24.93**
		30	**64.15**	**88.51**	**40.90**	**74.36**	**14.79**	**42.05**	**5.05**	**18.31**
		40	64.13	**88.50**	**40.61**	**74.08**	**14.01**	**39.99**	**4.80**	**16.53**
		50	64.10	**88.50**	40.77	**73.97**	**13.74**	**39.12**	**4.30**	**14.82**
		100	64.06	**88.48**	39.99	**73.29**	**12.67**	**35.97**	**3.29**	**10.69**
		300	64.05	88.48	39.52	**72.81**	**12.66**	**34.63**	**2.90**	**8.78**
SEA: only CosPGD (with Sigmoid) in (Croce et al., 2020)	APGD	10	64.48	88.60	48.60	79.47	31.92	65.45	21.59	53.70
		20	64.43	88.59	46.31	77.72	26.37	57.98	15.35	41.19
		30	64.41	88.58	45.78	77.22	24.35	54.46	13.18	34.70
		40	64.39	88.58	45.16	76.82	22.89	52.09	12.43	30.88
		50	64.39	88.58	44.95	76.57	22.54	50.91	11.59	28.78
		100	64.37	88.58	44.40	76.13	21.57	48.74	10.53	24.87
		300	64.37	88.57	44.05	75.96	21.09	47.39	10.23	22.58
SEA: only SegPGD in (Croce et al., 2020)	APGD	10	64.38	88.66	44.46	77.21	22.17	58.12	11.37	45.04
		20	64.23	88.59	42.46	75.74	17.89	51.40	8.11	33.86
		30	64.21	88.56	41.71	75.09	16.11	48.30	6.61	28.27
		40	**64.09**	88.52	40.85	74.52	45.05	14.84	5.63	23.90
		50	**64.01**	88.49	**40.46**	74.30	13.98	42.97	4.90	20.85
		100	**63.95**	88.45	**39.47**	73.54	12.78	39.34	4.04	16.26
		300	**63.80**	**88.41**	**38.69**	72.90	11.27	35.85	3.36	12.17

Table 8: Attacking Robust UPerNet with a ConvNeXt-tiny encoder from (Croce et al., 2023) with CosPGD for extremely high number of iterations i.e. 1200 iterations with $\epsilon = \frac{4}{255}$

Attack Method	Optimizer Used	Attack Iterations	ϵ=$\frac{4}{255}$	
			mIoU (%)	mAcc (%)
SEA reported by (Croce et al., 2023)	APGD	1200	63.800	88.300
SEA (Croce et al., 2023) reproduced by us			63.670	88.320
replacing SegPGD with CosPGD(softmax) in SEA (Croce et al., 2023)			63.700	88.300

(Croce et al., 2023) does not only aim for robustness w.r.t. $\epsilon = 4/255$ but aims to generalize (which we infer from their evaluation), it is fair to consider the range of improvement CosPGD reaches over SegPGD for $\epsilon = 12/255$ or $\epsilon = 16/255$ (scenarios considered in (Croce et al., 2023) as well). There, CosPGD decreases the mAcc by almost 10% more than SegPGD (for 30 iterations), and be more than 3% more for 300 iterations. The general tendency is also that with really high numbers of attack iterations (>100 iterations: not commonly considered by peer-reviewed white-box attack works), the differences between CosPGD and SegPGD become smaller, even for ϵ bounds for which the model has not been trained. This is in line with our expectation, coming from the point that CosPGD has smoother gradients and allows to compute better attacks with few iterations, as discussed in Section 4.

B.4. Evaluating Attacks against SAM

In Table 9, we show that when we attack a DeepLabV3 with a ResNet50 encoder on PASCAL VOC2012 images, and transfer the 100 iterations attack to SAM (Kirillov et al., 2023), only the CosPGD attack can cause failures in the segmentation masks. SegPGD fails to create failures in the segmentation masks of SAM, when compared to its segmentation masks on a clean image.

Note that these are just random sample results, as quantitative evaluation would be invalid. This is because the publicly available version of SAM does not perform semantic segmentation (which is segmentation with class labels). SAM merely predicts segmentation masks without assigning them any class labels, and current variants of SAM used for Semantic Segmentation, for example in this GitHub repository perform worse than the other models we considered for this task. Furthermore, the masks produced by SAM are often finer than the ground truth masks of most datasets, making the calculation of metrics like mIoU invalid.

B.5. Semantic Segmentation with UNet on Cityscapes

In the following, we provide extra results on semantic segmentation with UNet on the Cityscapes dataset.

B.5.1. Implementation Details

In this evaluation, we use a UNet architecture (Ronneberger et al., 2015) with a ConvNeXt_tiny encoder (Liu et al., 2022). We extend the implementation from (username: mberkay0, 2023)(`www.github.com`) to implement CosPGD, PGD, and SegPGD non-targeted l_∞-norm and l_2-norm attacks.

We do these evaluations on the Cityscapes dataset (Cordts et al., 2016). Cityscapes contains a total of 5000 high-quality images and pixel-wise annotations for urban scene understanding. The dataset is split into 2975, 500, and 1525 images for training, validation, and testing respectively. The model is trained on the test split and attacks are evaluated on the validation split.

B.5.2. Experimental Results and Discussion

In Figure 9, we report results from the comparison of non-targeted CosPGD to PGD and SegPGD attacks across iterations and across l_p-norm constraints: l_∞-norm and l_2-norm using UNet architecture with a ConvNeXt tiny encoder on Cityscapes validation dataset. For the l_∞-norm constraint, we use the same $\alpha = 0.01$ and $\epsilon \approx \frac{8}{255}$ as in all previous evaluations. For the l_2-norm constraint we follow common work (Croce et al., 2020; Wang et al., 2023) and use the same ϵ for CosPGD, SegPGD, and PGD i.e. $\epsilon \approx \{\frac{64}{255}, \frac{128}{255}\}$ and $\alpha =\{0.1, 0.2\}$.

Note, SegPGD has been proposed as an l_∞-norm constrained attack. We extend it to the l_2-norm constraint merely for complete comparison and curiosity.

We observe in Figure 9 that CosPGD is a significantly stronger attack than both PGD and SegPGD, across iterations and l_p-norm constraints, and α and ϵ values. Even at low attack iterations, it outperforms previous methods significantly, making it particularly efficient. Especially as an l_2-norm constrained attack, as shown before in Figure 10 for DeepLabV3 on PASCAL VOC 2012 dataset and discussed before in Section 5.2, as attack iterations increase, CosPGD can increase the performance gap quite significantly.

B.6. Ablation on Attack Step Size α

Further, we provide additional experimental results and ablation studies using DeepLabV3 for semantic segmentation on the PASCAL VOC 2012 validation dataset.

B.6.1. l_2-norm constrained adversarial attacks

Further in Figure 10, we report l_2-norm constrained attack evaluations on commonly used (Croce et al., 2020; Wang et al., 2023) values of $\epsilon \approx \{\frac{64}{255}, \frac{128}{255}\}$ and $\alpha =\{0.1, 0.2\}$.

Additionally, in Table 10 we provide comparison to C&W (Carlini & Wagner, 2017) and other l_2-norm constrained adversarial attacks with α=0.2 and $epsilon \approx \frac{128}{255}$ on PASCAL VOC 2012 validation dataset using DeepLabV3 with a ResNet50 backbone.

B.6.2. l_∞-norm constrained adversarial attacks

Following, we ablate over the attack step size α for the l_∞-norm constrained adversarial attacks and report the findings in Figure 11. We consider $\alpha \in \{0.005, 0.01, 0.02, 0.04, 0.1\}$. We can observe that the scaling in CosPGD ensures less susceptibility to the choice of step size given that it is set small enough ($\alpha \leq \epsilon$). In our work, we use step size α=0.01 to maintain consistency with previous work (Kurakin et al., 2017; Gu et al., 2022).

B.7. Tabular Results

Here we report the quantitative results that have already been presented in the main paper in Figures 3in tabular form. For the results reported in Figure 3, we report the results in tables 11. Here we observe that at low attack iterations (iterations=3) SegPGD implies that PSPNet is more adversarially robust than both DeepLabV3. However, after more attack iterations (iterations $\geq$ 5), SegPGD correctly implies that DeepLabV3 is more robust than PSPNet. Contrary to this, CosPGD even at

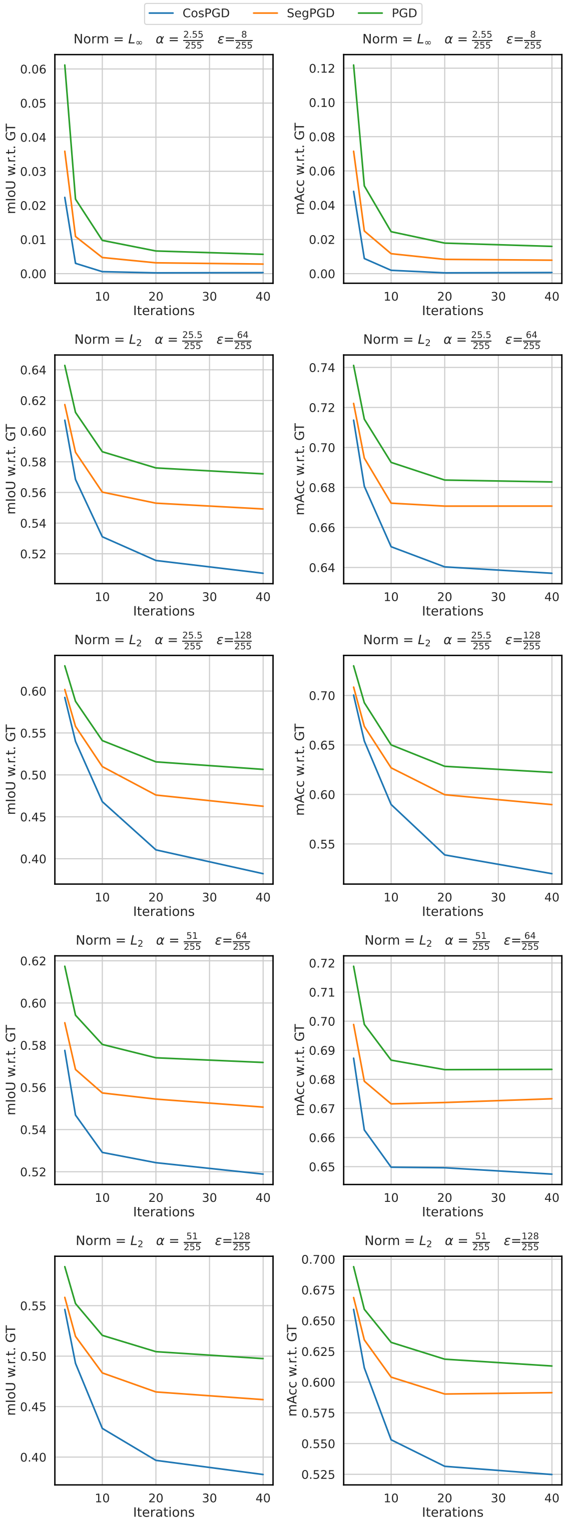


Figure 9: Comparing non-targeted CosPGD to PGD and SegPGD attacks across iterations and l_p-norm constraints, and α and ϵ values using UNet architecture with a ConvNeXt tiny encoder on Cityscapes validation dataset. CosPGD significantly outperforms previous methods by a large margin, even at few attack iterations.

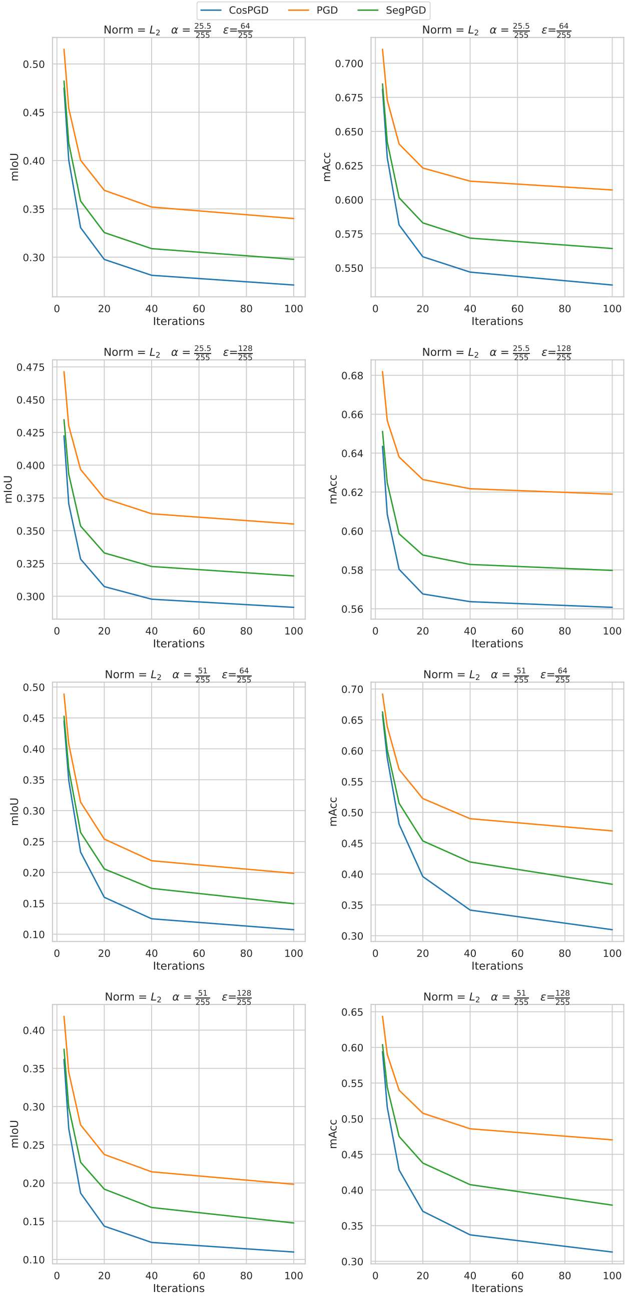


Figure 10: Comparing CosPGD to PGD and SegPGD across iterations as l_2-norm constrained attacks, and across α and ϵ values using DeepLabV3 architecture with a ResNet50 on PASCAL VOC 2012 validation dataset. Again, CosPGD outperforms previous attacks be a large margin at all attack iterations.

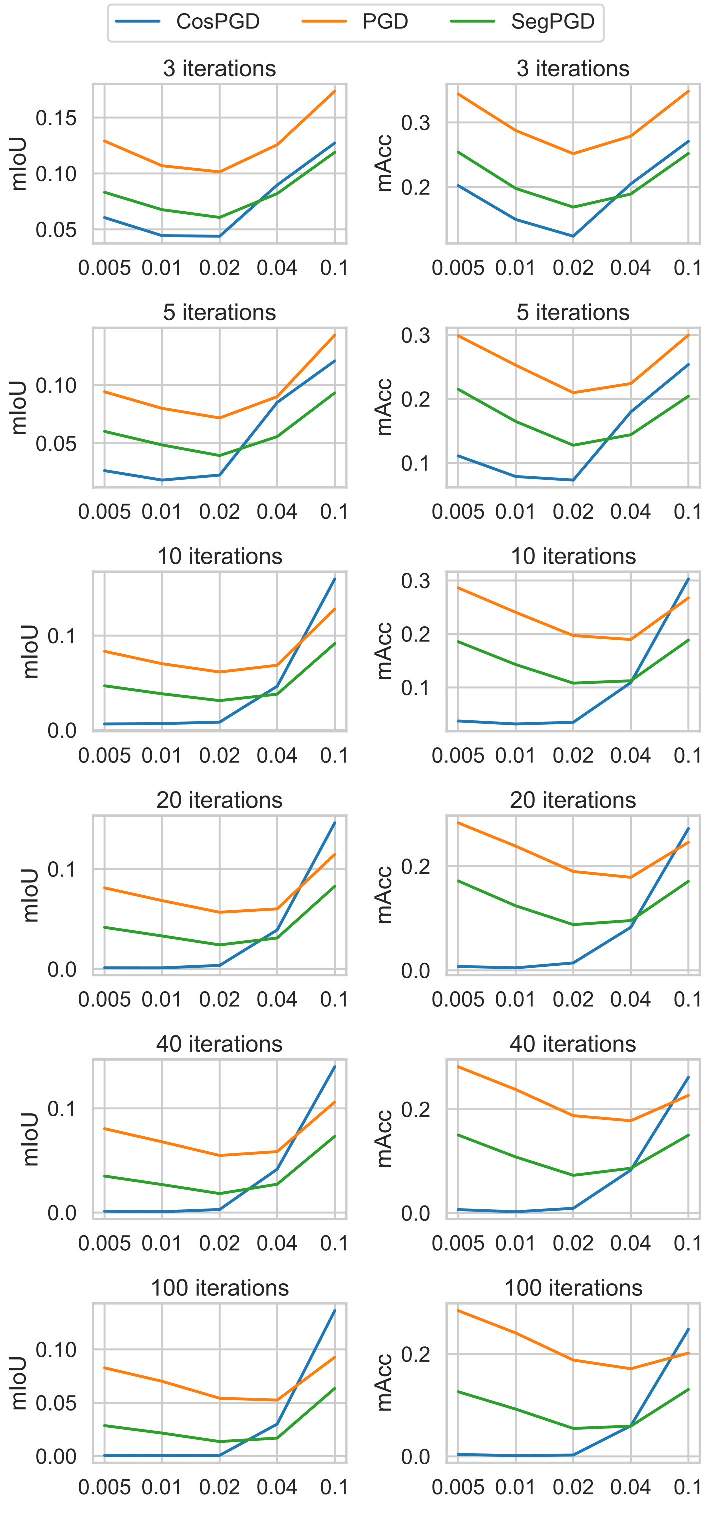


Figure 11: We ablate step sizes α for l_∞-norm constrained CosPGD, SegPGD, and PGD attacks given different number of iterations $\in \{3, 5, 10, 20, 40, 100\}$ by attacking DeepLabV3 trained on the PASCAL VOC2012 dataset with maximal perturbation of $\epsilon = 0.03$. We can observe that the scaling in CosPGD ensures less susceptibility to the choice of step size given that it is set small enough ($\alpha \leq \epsilon$).

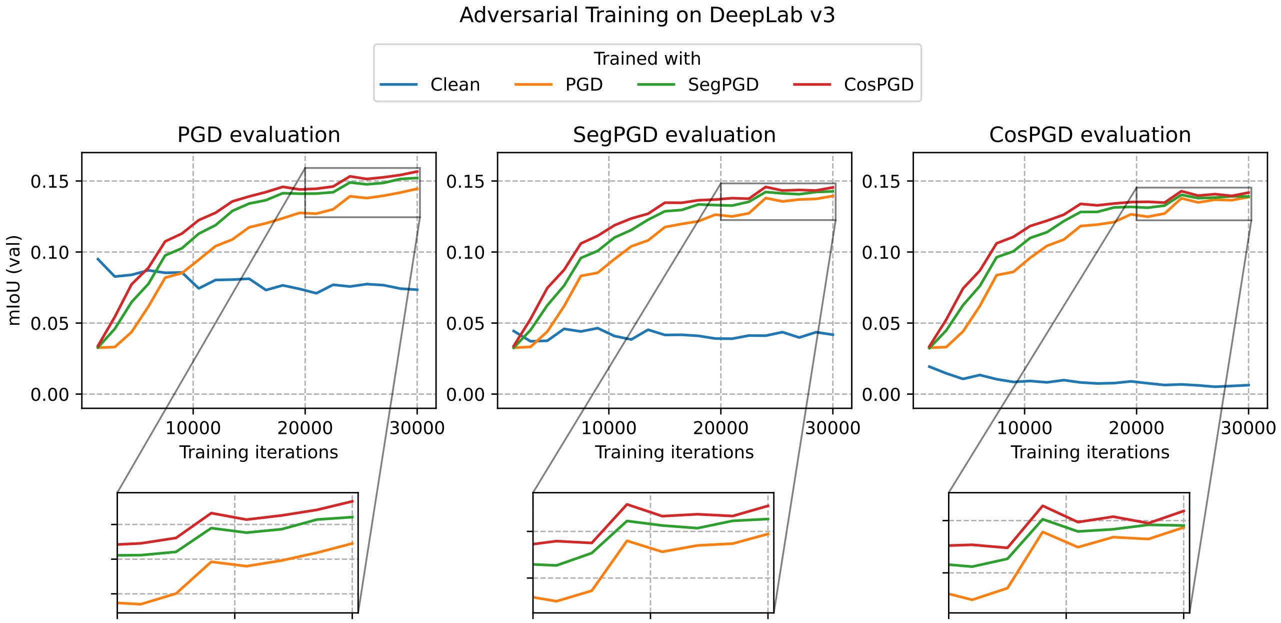


Figure 12: DeepLabV3 adversarially trained using different adversarial attacks for 3 iterations during training using 50% of the minibatch for generating adversarial samples. All checkpoints are evaluated against 10 attack iterations of the respective attacks. We observe that the model trained with CosPGD outperforms all other adversarial training methods considered against all attacks.

low attack iterations correctly predicts DeepLabV3 to be more robust than PSPNet. This is an insight that CosPGD provides with considerably less computation.

B.8. Adversarial Training

In Figure 13 we show the segmentation masks predicted by UNet after being adversarially trained. We observe that even after 100 attack iterations, the model adversarially trained using CosPGD is making reasonable predictions. However, the model trained with SegPGD is merely predicting a blob.

In Table 12 we report the performance of models trained with various adversarial attacks against different commonly used adversarial attacks across multiple attack iterations. We observe that the model trained with CosPGD performs the best against all considered adversarial attacks. The models were trained with 3 attack iterations of the respective “Training Method” attack during training.

In Figure 12 we present the training curves for training DeepLabV3 on the PASCAL VOC2012 training dataset using adversarial training with 50% minibatch being used for generating adversarial samples. All models are evaluated against 10 attack iterations of the respective attack.

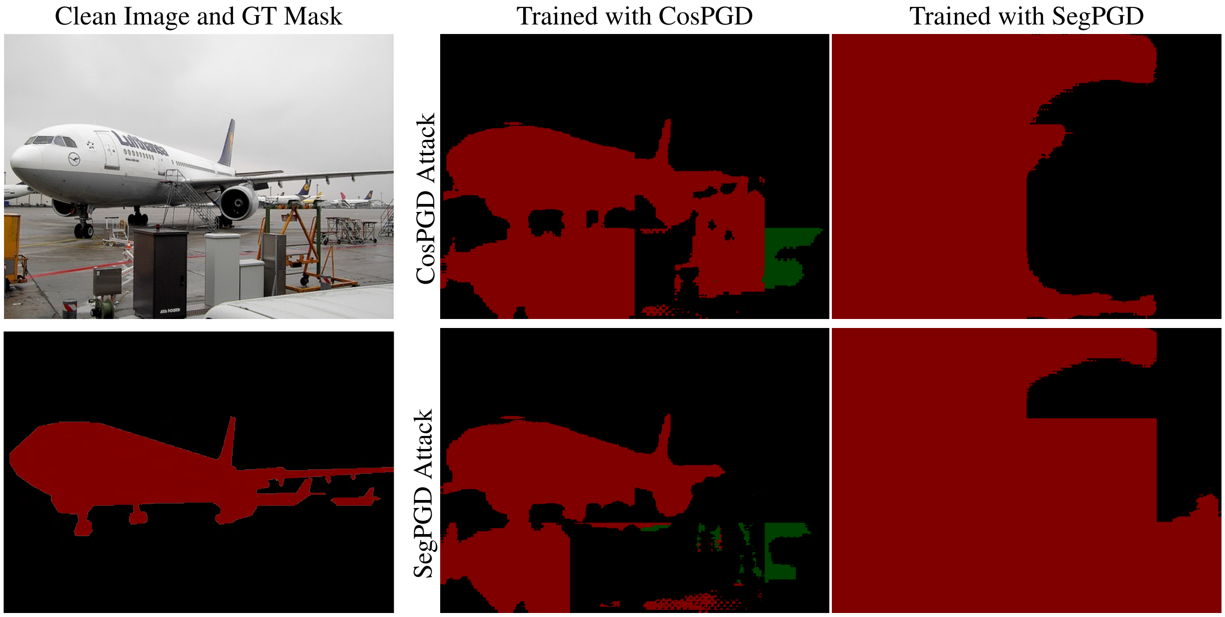


Figure 13: Predictions using UNet with ConvNeXt backbone on PASCAL VOC2012 validation dataset after 100 iterations adversarial attacks on adversarially trained models. We observe that the models adversarially trained with CosPGD are predicting reasonable masks even after 100 attack iterations, while the model trained with SegPGD is providing much worse results under both SegPGD and CosPGD attacks.

Table 9: Transfer Attack from DeepLabV3 to SAM (Kirillov et al., 2023) in a black-box setting on some random samples from PASCAL VOC2012 validation dataset. All Attacks are with $\epsilon = \frac{8}{255}$ and α=0.01 with 100 attack iterations. DeepLabV3 was trained for Semantic Segmentation using PASCAL VOC2012 train split.

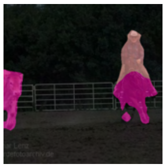
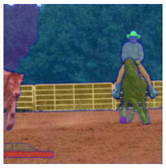
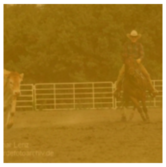
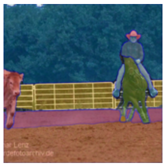
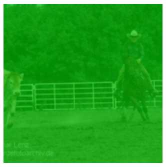
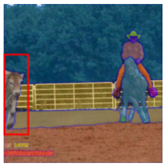
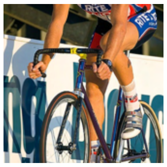
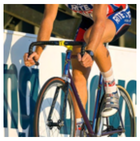
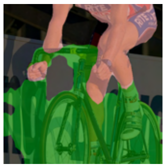
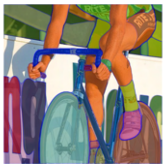
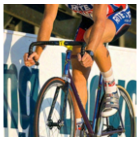
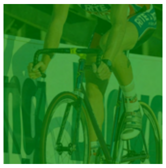
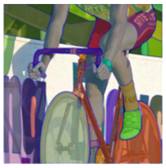
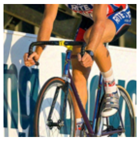
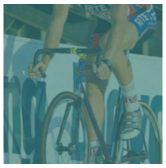
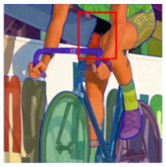
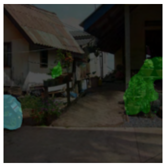
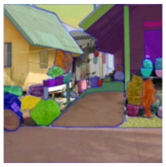
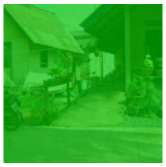
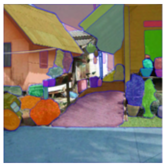
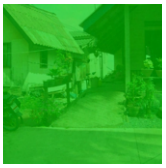
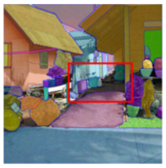
Original Image	PASCAL VOC2012 Ground Truth Mask	Attacked Image	Mask predicted by DeepLabV3 with ResNet50 Backbone	Mask Predicted by Segment Anything Model (SAM)	Observations: SAM's failure caused only under CosPGD Attack (Red Rectangle on SAM output under CosPGD attack)
		No Attack 			Under No Attack and SegPGD attack, SAM is able to detect the cow and segment it out in the image. However, **under CosPGD attack, SAM is not able to detect the animal**.
		SegPGD 			
		CosPGD 			
		No Attack 			Under No Attack and SegPGD attack, SAM is able to detect the the right thigh of the cyclist as a single object. However, **under CosPGD attack, SAM is not able to detect this** and performs an unnatural split in the segmentation masks on the right thigh.
		SegPGD 			
		CosPGD 			
		No Attack 			Under No Attack and SegPGD attack, SAM is able to detect the road in the centre of the frame and segment it out in the image. However, **under CosPGD attack, SAM is not able to detect the road**.
		SegPGD 			
		CosPGD 			

Table 10: Comparison of performance of CosPGD to SegPGD, PGD and C&W as a l_2-norm constrained attack with α=0.2 and $\epsilon \approx \frac{128}{255}$ where applicable for semantic segmentation over PASCAL VOC2012 validation dataset. We observe that CosPGD is a significantly stronger attack compared to all the other attacks for both metrics.

		Attack iterations											
Network	**Attack method**	3		5		10		20		40		100	
		mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)
DeepLabV3	**C&W** (c=1)	72.35	84.32	72.02	84.13	71.87	84.05	71.81	84.02	71.78	84.01	71.77	84.00
	PGD	41.81	64.36	34.5	59.03	27.61	54.0	23.73	50.77	21.47	48.58	19.84	47.04
	SegPGD	37.51	60.4	29.9	54.4	22.72	47.51	19.2	43.78	16.8	40.75	14.77	37.88
	CosPGD	**36.17**	**59.41**	**27.12**	**51.6**	**18.68**	**42.8**	**14.35**	**37.02**	**12.23**	**33.71**	**10.97**	**31.3**

Table 11: Comparison of performance of CosPGD to SegPGD for semantic segmentation over PASCAL VOC2012 validation dataset. We observe that CosPGD is a significantly stronger attack compared to SegPGD for both metrics and all models.

		Attack iterations											
Network	**Attack method**	3		5		10		20		40		100	
		mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)
UNet	**SegPGD**	12.38	32.41	7.75	25.27	4.46	18.36	2.98	14.24	2.20	11.66	1.55	8.66
	CosPGD	**9.67**	**29.46**	**3.71**	**15.89**	**0.61**	**3.39**	**0.06**	**0.38**	**0.03**	**0.16**	**0.01**	**0.04**
PSPNet	**PGD**	13.79	31.91	7.59	21.15	5.44	16.96	4.48	14.78	3.80	13.13	3.72	13.21
	SegPGD	9.19	23.25	4.70	14.25	2.72	9.50	1.82	7.39	1.30	5.77	0.83	3.86
	CosPGD	**7.03**	**19.73**	**2.15**	**7.60**	**0.408**	**1.44**	**0.04**	**0.11**	**0.005**	**0.021**	**0.0002**	**0.0007**
DeepLabV3	**PGD**	10.69	28.76	8.00	25.29	7.02	24.05	6.84	23.87	6.79	23.81	7.01	24.13
	BIM	10.86	29.39	7.75	24.97	6.95	24.06	6.67	23.52	6.57	23.48	–	–
	APGD	13.74	29.79	8.67	22.46	6.50	19.82	6.11	18.99	5.30	17.04	5.14	16.72
	SegPGD	6.76	19.78	4.86	16.49	3.84	14.29	3.31	12.40	2.69	10.81	2.15	9.25
	CosPGD	**4.44**	**14.97**	**1.84**	**7.89**	**0.69**	**3.18**	**0.12**	**0.48**	**0.08**	**0.25**	**0.005**	**0.16**

Table 12: Evaluating the adversarial performance of models on PASCAL VOC2012 validation dataset that are adversarially trained using PASCAL VOC2012 training dataset. "Training method" specifies the adversarial attack used during training, such that "Clean" stands for no adversarial attack being used during training. During training, 3 attack iterations were used for all adversarial attacks with α=0.01 and $\epsilon \approx \frac{8}{255}$. These models were evaluated against multiple adversarial attacks denoted by "Attack method". We observe that models trained with CosPGD substantially outperform all the other adversarial training methods.

			Attack iterations											
Network	**Training method**	**Attack method**	3		5		10		20		40		100	
			mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)
UNet	Clean	PGD	23.18	46.64	14.58	35.89	8.21	24.99	5.57	18.57	4.14	14.53	3.6	11.72
	PGD		29.26	57.52	21.28	51.06	13.74	41.57	9.29	32.51	7.47	27.46	6.38	22.43
	SegPGD		31.77	63.91	22.77	57.82	14.86	48.09	11.03	40.25	8.98	34.29	7.45	28.4
	CosPGD		**47.35**	**68.67**	**43.75**	**66.34**	**38.1**	**62.85**	**34.33**	**60.06**	**32.28**	**58.64**	**30.55**	**57.51**
	Clean	SegPGD	12.38	32.41	7.75	25.27	4.46	18.36	2.98	14.24	2.20	11.66	1.55	8.66
	PGD		29.38	57.82	21.31	51.35	13.77	41.72	9.39	33.15	7.45	26.98	6.38	22.26
	SegPGD		31.69	63.94	22.47	57.07	14.82	47.94	10.9	40.32	9.09	34.68	7.33	27.99
	CosPGD		**47.16**	**68.51**	**43.85**	**66.41**	**37.64**	**62.58**	**33.99**	**59.8**	**31.91**	**58.31**	**30.48**	**57.01**
	Clean	CosPGD	9.67	29.46	3.71	15.89	0.61	3.39	0.06	0.38	0.03	0.16	0.01	0.04
	PGD		29.23	57.71	21.09	50.73	13.49	40.91	9.28	32.68	7.36	27.02	6.29	22.0
	SegPGD		31.53	63.96	22.46	57.23	14.81	48.09	10.86	40.26	9.20	35.33	7.28	28.03
	CosPGD		**47.07**	**68.39**	**43.95**	**66.52**	**37.64**	**62.38**	**34.01**	**60.03**	**32.0**	**58.47**	**30.55**	**57.28**
DeepLabV3	Clean	PGD	11.02	30.96	8.50	**27.34**	7.63	**26.35**	7.57	**26.30**	7.59	**26.19**	7.39	**25.98**
	PGD		21.05	29.07	16.74	24.61	14.45	22.19	13.82	21.56	13.58	21.32	13.42	21.17
	SegPGD		22.67	31.87	17.85	26.99	15.21	24.26	14.42	23.47	14.11	23.16	13.90	22.93
	CosPGD		**23.13**	**32.21**	**18.33**	**27.34**	**15.68**	24.60	**14.80**	23.61	**14.49**	23.29	**14.27**	23.06
	Clean	SegPGD	6.78	20.50	5.05	17.40	3.99	14.95	3.32	12.94	2.60	10.57	1.80	8.05
	PGD		20.62	28.54	16.12	23.79	13.95	21.42	13.41	20.84	13.20	20.61	13.04	20.42
	SegPGD		22.06	31.37	16.89	26.02	14.27	**23.23**	13.57	**22.50**	13.33	**22.23**	13.09	21.92
	CosPGD		**22.33**	**31.48**	**17.15**	**26.07**	**14.54**	23.18	**13.89**	22.45	**13.67**	22.22	**13.54**	**22.15**
	Clean	CosPGD	4.71	16.35	1.94	8.09	0.61	3.32	0.24	1.59	0.09	0.53	0.08	0.59
	PGD		20.56	28.48	16.05	23.75	13.87	21.45	13.38	20.92	13.18	20.72	**13.07**	20.59
	SegPGD		21.87	31.19	16.62	25.77	13.91	22.93	13.19	22.17	12.92	21.87	12.78	21.72
	CosPGD		**22.14**	**31.33**	**16.88**	**25.85**	**14.18**	**22.99**	**13.48**	**22.21**	**13.20**	**21.90**	13.05	**21.76**

C. Optical flow estimation

C.1. Tabular Results

Table 13: Comparison of performance of CosPGD to PGD as a targeted attack for optical flow estimation over KITTI15 and Sintel validation datasets using RAFT for different numbers of attack iterations. epe values are compared, with respect to both, the **Target** i.e. $\overrightarrow{0}$ where a lower epe indicates a better attack and Initial flow prediction (optical flow estimated by the model before any adversarial attack) where a higher epe indicates a better attack. CosPGD and PGD perform similarly for a low number of iterations, where CosPGD fits the target slightly better. CosPGD significantly outperforms PGD from the 10^{th} iteration onwards on both metrics.

Attack	**KITTI 2015**						**MPI Sintel**											
							clean						**final**					
Iterations	SegPGD		PGD		CosPGD		SegPGD		PGD		CosPGD		SegPGD		PGD		CosPGD	
	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑	Target↓	Initial↑
3	**20.57**	11.28	20.7	**11.4**	20.6	11.2	8.35	**6.83**	8.3	6.8	**8.1**	6.6	7.58	**7.52**	7.6	7.3	**7.5**	7.3
5	14.33	17.75	14.4	**17.8**	**14.3**	17.7	6.06	8.97	6.1	**9.0**	**5.8**	8.8	5.44	**9.43**	5.6	9.4	**5.2**	9.3
10	11.08	21.36	10.5	22.1	**9.0**	**23.4**	3.51	11.16	3.4	11.2	**2.9**	**11.4**	3.13	11.32	3.1	11.3	**2.6**	**11.5**
20	7.76	24.55	8.1	24.6	**6.5**	**25.8**	2.97	11.61	2.8	11.7	**2.0**	**12.1**	2.62	11.7	2.5	11.8	**1.6**	**12.1**
40	7.53	24.89	7.3	25.0	**4.8**	**27.4**	2.66	11.8	2.8	11.7	**1.6**	**12.4**	2.4	11.83	2.6	12.3	**1.3**	**12.3**

Here we report the extended results from Figure 6 comparing CosPGD to PGD as a targeted attack using RAFT for KITTI15 and Sintel datasets in Figure 14 and in tabular form in Table 13. We observe that CosPGD is more effective than PGD to change the predictions toward the targeted prediction. During a low number of iterations (iterations = 3 and 5), PGD is on par with CosPGD in increasing the epe values of the predictions compared to the initial predictions on non-attacked images. However, as the number of iterations increases, CosPGD outperforms PGD for this metric as well. In the following, we report further results and compare CosPGD to a recently proposed sophisticated l_2-norm constrained targeted attack PCFA.

C.2. Non-targeted attacks for optical flow estimation

For l_∞-norm constrained non-targeted attacks, CosPGD changes pixels values temperately over a larger region of the image, while PGD changes it drastically but only for a small region in the image. This can be observed in Figure 15 when CosPGD and PGD are compared as l_∞-norm constrained non-targeted attacks for optical flow estimation. We observe that both CosPGD and PGD are performing at par as both have very similar epe values across iterations. However, CosPGD across iterations has a lower epe-$f1$-all value. As shown by Equation 12 in Section A.3.2, epe-$f1$-all is the measure of average overall epe values that are above a modest threshold. Therefore, both CosPGD and PGD have very similar epe scores while CosPGD has a significantly lower epe-$f1$-all compared to PGD. This implies that CosPGD and PGD are performing at par, however, PGD is drastically changing epe values at certain pixels, while CosPGD is changing epe values temperately over considerably more pixels. Figure 16 shows this qualitatively for 4 randomly chosen samples.

C.3. Comparison to PCFA

Further, we compare CosPGD as a l_2-norm constrained targeted attack to the recently proposed *state-of-the-art* l_2-norm constrained targeted attack PCFA (Schmalfuss et al., 2022b). For comparison. we use the same settings as those used by the authors for both attacks, for 20 attack iterations (steps), generating adversarial patches for each image individually, bounded under the change of variables methods proposed by Schmalfuss et al. (2022b). Here, we observe that a sophisticated l_2-norm constrained targeted attack, PCFA that does not utilise pixel-wise information for generating adversarial patches over all considered networks and datasets, performs similar to CosPGD. We compare over the performance over RAFT, PWCNet (Sun et al., 2018), GMA (Jiang et al., 2021) and SpyNet (Ranjan & Black, 2017) We consider both targeted settings proposed by Schmalfuss et al. (2022b), i.e. target being a zero vector $\overrightarrow{0}$ and target being the negative of the initial prediction (*negative flow*). We compare the average epe over all images. A lower AEE is w.r.t. Target and higher AEE w.r.t. initial indicate a stronger attack. In Table 14(currently included at the end of the appendix to not disturb the table numbers), we compare PCFA and CosPGD on multiple datasets, multiple networks over 3 random seeds.

Figure 17, provides an overview of the comparison between the two methods, using targets as $\overrightarrow{0}$ and *negative flow*. Figures 18, 19, provide further details compares both methods when using $\overrightarrow{0}$ and *negative flow* as the target, respectively.

In Table 14, we include the results in a tabular form.

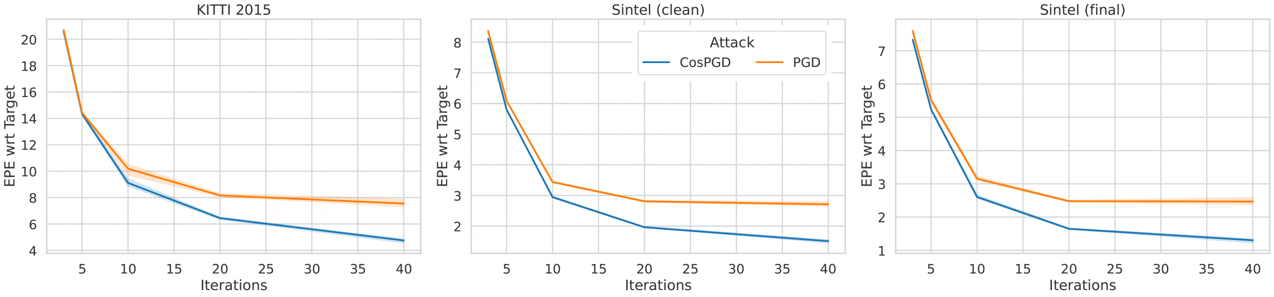


Figure 14: An extension to Figure 6. Comparison of performance of CosPGD to PGD for optical flow estimation over KITTI-2015 (left) and Sintel (clean → left and final → right) validation datasets as ℓ_∞-norm constrained targeted attacks using RAFT. CosPGD is a stronger targeted attack than PGD for optical flow. We also report these results in Table 13 in Appendix C.

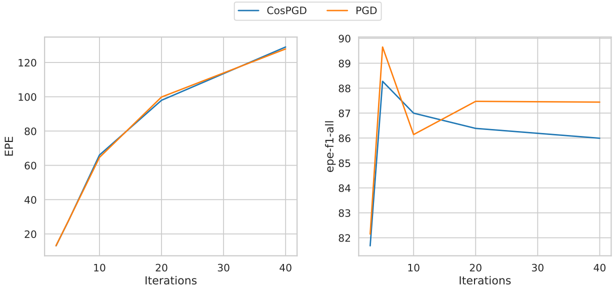


Figure 15: Comparing CosPGD and PGD as l_∞-norm constrained non-targeted attacks for optical flow estimation using RAFT on KITTI 2015 validation dataset.

It would be interesting to extend these evaluations to newer optical flow datasets such as Spring (Mehl et al., 2023).

D. Image Restoration Tasks

Following, we provide further results and discussion on the two considered image restoration tasks namely, Image Deblurring in Section D.1 and Image Denoising in Section D.2

D.1. Image Deblurring models

In Figure 20 for the Baseline network, we observe that both CosPGD and PGD are performing at par. While for the newly proposed NAFNet, PGD is still estimating NAFNet's adversarial robustness to be very similar to the Baseline network and only after 20 attack iterations it is estimating correctly that NAFNet is not as robust as the Baseline network. However, CosPGD reveals that NAFNet is not as robust as the baseline even at a low number of iterations (3 attack iterations). This valuable insight regarding model robustness of newly proposed transformer-based image restoration models is provided by CosPGD with considerably less computation.

To enable the applicability of SegPGD on this task, we implement SegPGD by comparing the equality of the pixel values to use their proposed loss for comparison. Following the discussion from Section 5.3, in Figure 8 for the Baseline network we also observe that SegPGD here is significantly weaker due to its limitation to image classification tasks as discussed in Section 4. However, for NAFNet, from 5 attack iterations onwards SegPGD is outperforming PGD, while still being weaker than CosPGD. This, interesting improvement in the performance of SegPGD as an adversarial attack can be attributed to the pixel-wise nature of the attack, similar to CosPGD further highlighting the benefits of utilizing pixel-wise information when crafting adversarial attacks for pixel-wise prediction tasks.

Additionally, we report the findings on many recently proposed state-of-the-art image restoration models using CosPGD in Table 15.

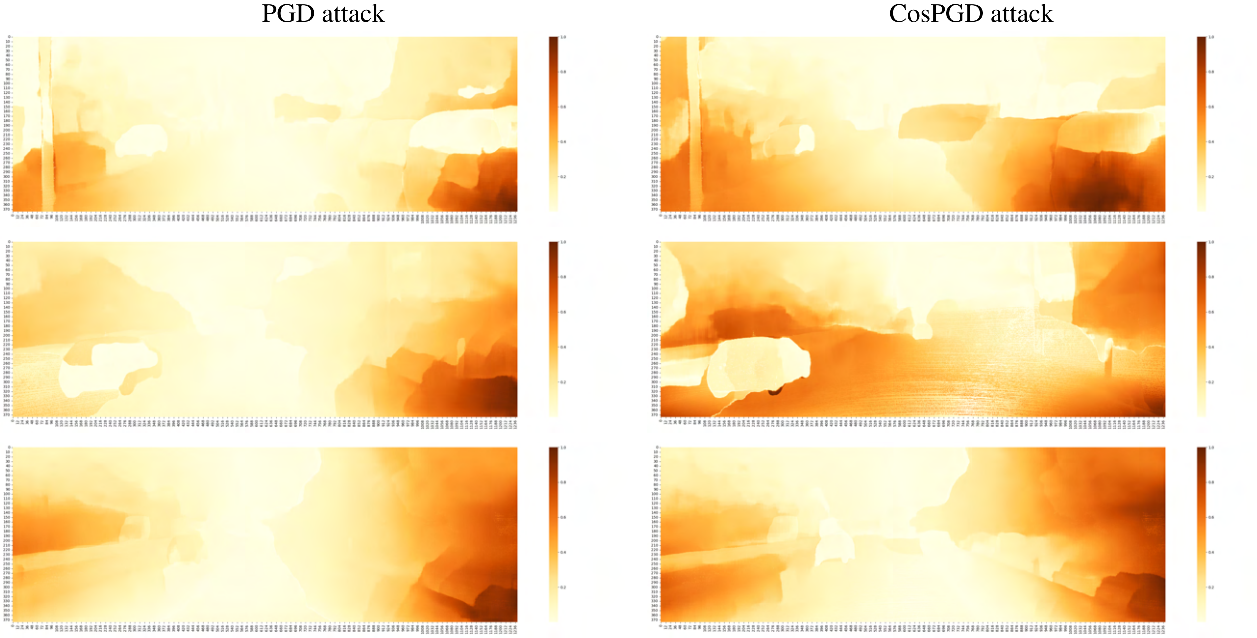


Figure 16: Comparing change in pixel-wise epe values w.r.t. initial epe values after 40 iterations of PGD and CosPGD as non-targeted ℓ_{∞}-norm constrained attacks on RAFT using KITTI15 validation set. The values for each image are: $\frac{|epe_{adv} - epe_{initial}|}{max(epe_{adv})}$ where epe_{adv} & $epe_{initial}$ are pixel-wise epe values of the final adversarial sample and the initial non-attacked image, respectively.

D.2. Non-targeted Attacks for Image Denoising Task

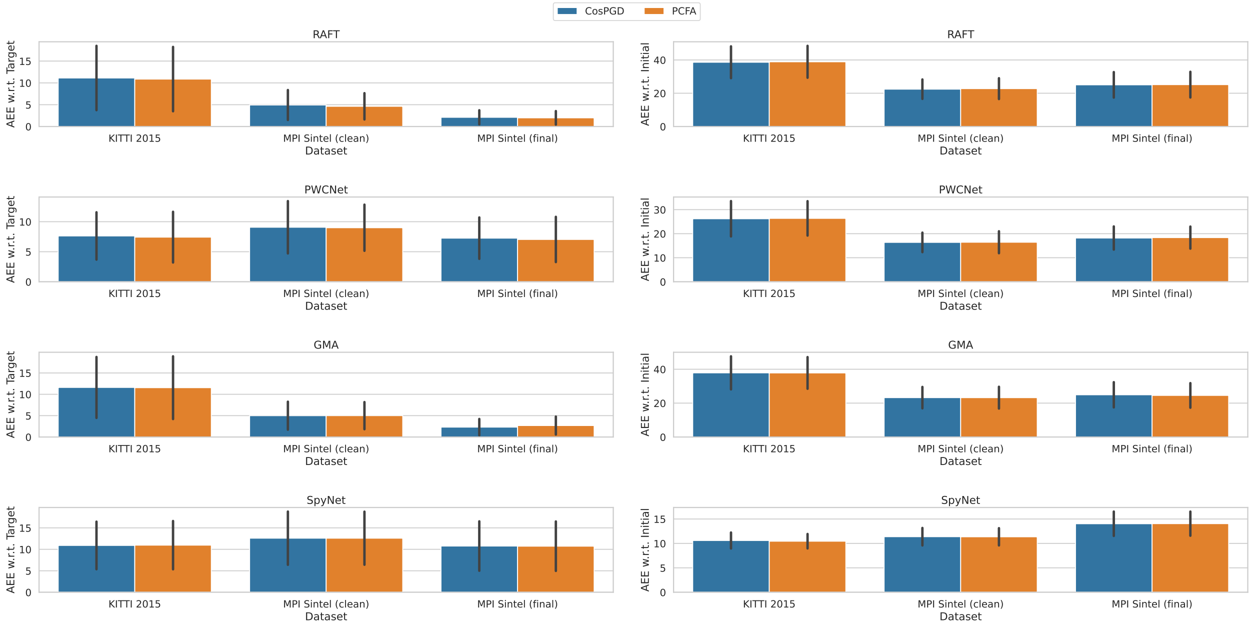
Dataset. For the image denoising task, following work from (Chen et al., 2022; Zamir et al., 2022) we use the Smartphone Image Denoising Dataset (SSID) (Abdelhamed et al., 2018). This dataset consists of 160 noisy images taken from 5 different smartphones and their corresponding high-quality ground truth images. Similar to the image deblurring task, we report the $PSNR$ and $SSIM$ values as metrics for this image restoration task as well.

Discussion. Further extending the findings from Section C.2 we report l_{∞}-norm constrained non-targeted attacks for the image denoising on the SSID dataset using the Baseline network and NAFNet (as proposed by (Chen et al., 2022)) in Figure. 21. We observe that both CosPGD and PGD are performing at par for both, the Baseline network and NAFNet. Additionally, similar to findings in Section 5.3, SegPGD is unable to perform at par with CosPGD and PGD.

After both CosPGD and PGD attacks it appears that the image denoising networks are relatively more robust than image deblurring networks. These findings also correlate with (Xie et al., 2019), as they report that feature denonising improves model robustness against adversarial attacks.

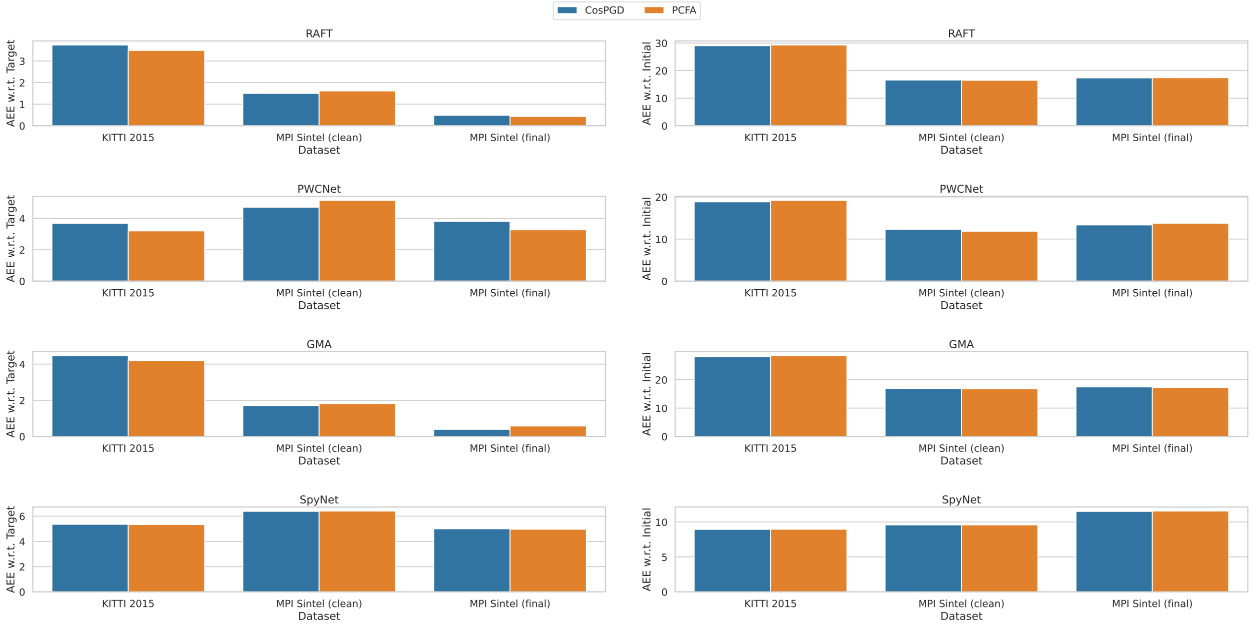
E. Discussion on limitations of CosPGD

Similar to most white-box adversarial attacks (Goodfellow et al., 2014; Kurakin et al., 2017; Madry et al., 2017; Wong et al., 2020b; Gu et al., 2022), CosPGD currently requires access to the model's gradients for generating adversarial examples. While this is beneficial for generating adversaries, it limits the applications of the non-targeted settings as many benchmark datasets (Menze & Geiger, 2015; Butler et al., 2012; Wulff et al., 2012; Everingham et al., 2012) do not provide the ground truth for test data. Evaluations of the validation datasets certainly show the merit of the attack method. CosPGD mitigates this limitation by also being applicable as an effective targeted attack. Nevertheless, it would be interesting to study the attack on test images as well in an untargeted setting, due to the potential slight distribution shifts pre-existing in the test data. While CosPGD is significantly more efficient than other existing adversarial attacks, all white-box adversarial attacks are time and memory consuming and benchmarking them across multiple downstream tasks, datasets, and networks is a very time-consuming process.



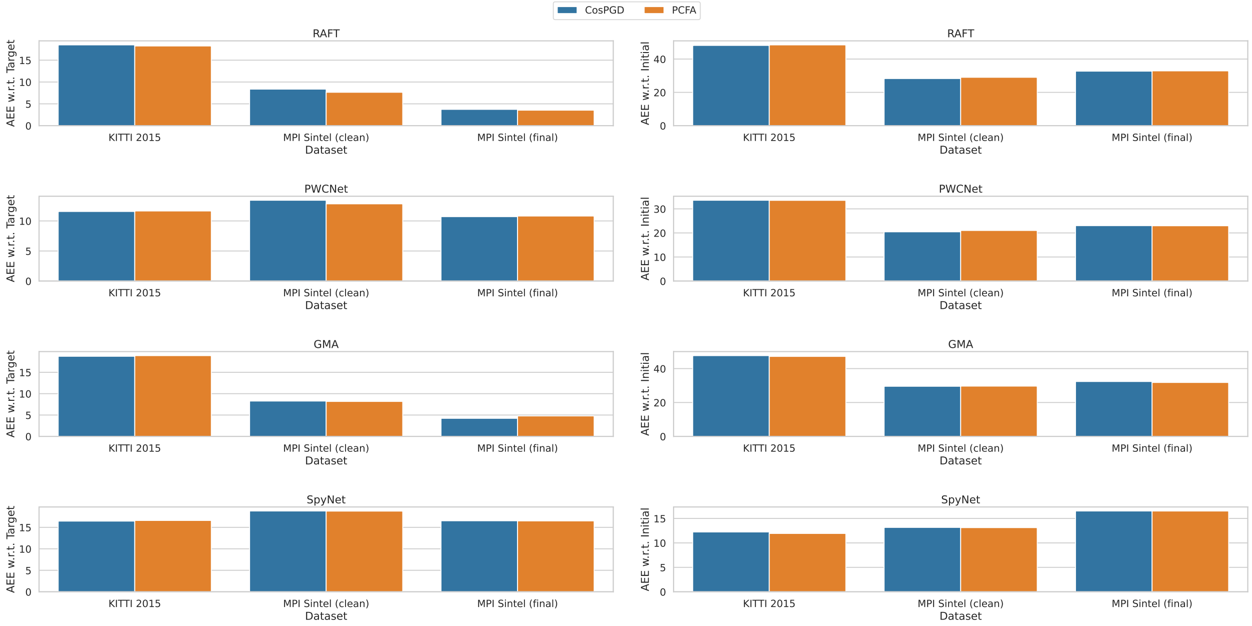
AEE w.r.t. Target, lower is better AEE w.r.t. Initial, higher is better

Figure 17: Comparison of mean and standard deviation of the results using different targets, $\overrightarrow{0}$ and *negative flow* for CosPGD and PCFA. A lower AEE is w.r.t. Target and a higher AEE w.r.t. initial indicate a stronger attack.



AEE w.r.t. Target, lower is better AEE w.r.t. Initial, higher is better

Figure 18: Comparison of PCFA and CosPGD when using $\overrightarrow{0}$ as the target. A lower AEE is w.r.t. Target and a higher AEE w.r.t. initial indicate a stronger attack.



AEE w.r.t. Target, lower is better

AEE w.r.t. Initial, higher is better

Figure 19: Comparison of PCFA and CosPGD when using *negative flow* as the target. A lower AEE is w.r.t. Target and a higher AEE w.r.t. initial indicate a stronger attack.

Additionally, there are settings, especially for non-targeted attacks, where approaches like pixel-wise PGD would work at par with CosPGD as the epe can be increased equally well by either changing all pixel-wise regression estimates slightly (sophisticated attack like CosPGD) or by changing only a few of them drastically (brute force attacks like PGD). This can also be seen in the results in C.2.

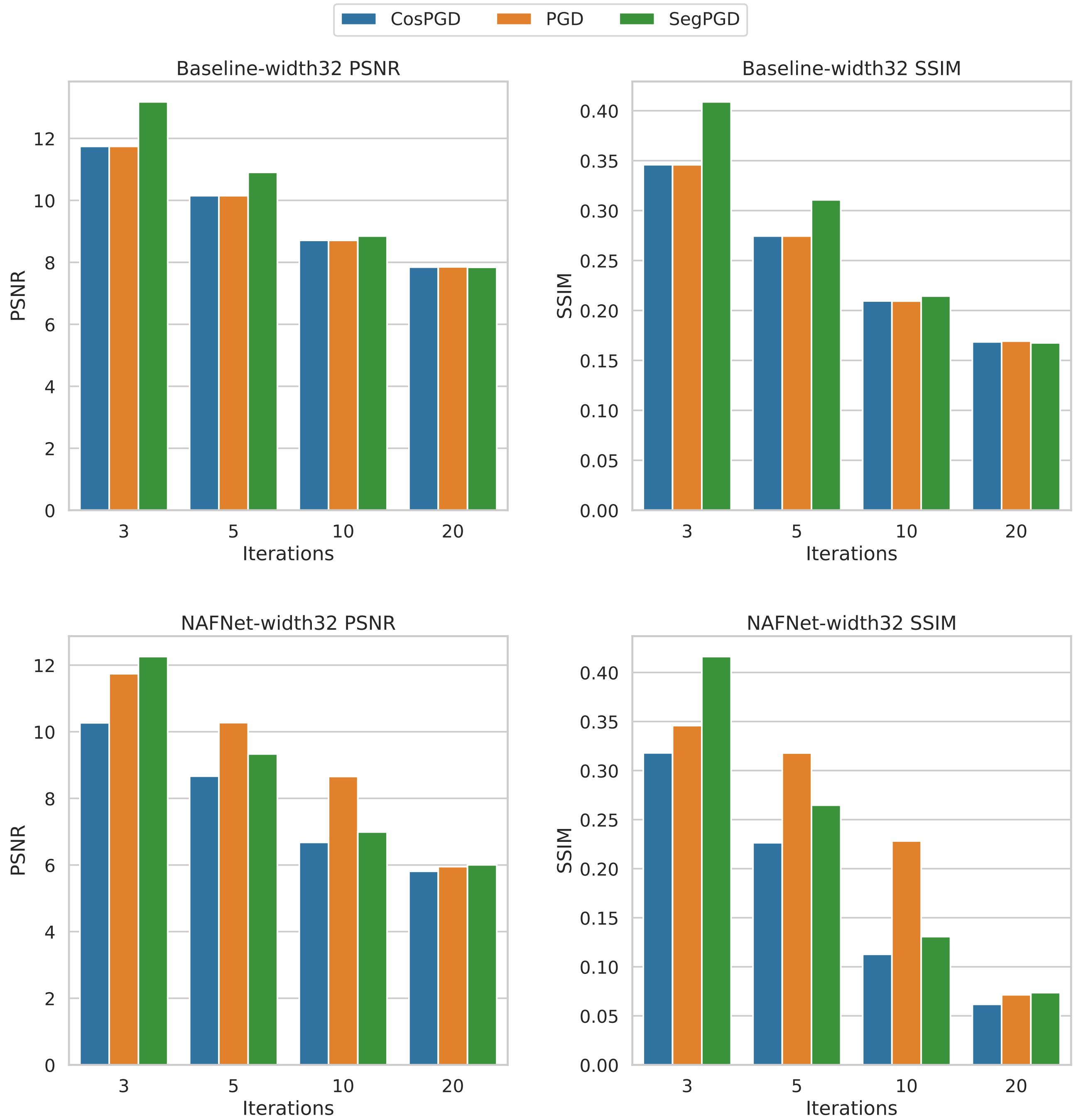


Figure 20: Non-targeted l_∞-norm constrained CosPGD, PGD, and SegPGD attacks on the "Baseline network" and NAFNet for image deblurring task on the GoPro dataset, recently proposed by (Chen et al., 2022) as the state-of-the-art networks for image restoration tasks. The "Baseline network" is significantly more robust than the NAFNet and thus the performance of the Baseline network against CosPGD attack is at par with its performance against PGD. However, PGD indicates at low attack iterations (iterations ≤ 10) that NAFNet is more robust than "Baseline network" and only after 20 attack iterations its correctly indicates that NAFNet is less robust. However, CosPGD is able to draw this conclusion at merely 3 attack iterations.

Table 14: Comparison of performance of CosPGD to PCFA as a targeted l_2-norm constrained attack for optical flow estimation over KITTI2015 and Sintel validation datasets using different optical flow models over 3 random seeds. Average epe values are compared, with respect to both, the **Target** where a lower epe indicates a better attack and **Initial flow prediction** (optical flow estimated by the model before any adversarial attack) where a higher epe indicates a better attack. We compare over both targets used by (Schmalfuss et al., 2022b), i.e. zero vector $\overrightarrow{0}$ and Negative of the Initial Flow. **CosPGD and PCFA performance is very comparable.**

Model	Target $\overrightarrow{0}$				Negative Initial Flow			
	AEE wrt Target↓		AEE wrt Initial↑		AEE wrt Target↓		AEE wrt Initial↑	
	CosPGD	PCFA	CosPGD	PCFA	CosPGD	PCFA	CosPGD	PCFA
KITTI 2015								
GMA	28.69 ± 0.12	28.67 ± 0.17	3.89 ± 0.09	3.89 ± 0.15	47.00 ± 0.40	47.08 ± 0.69	19.22 ± 0.53	19.20 ± 0.57
PWCNet	19.13 ± 0.04	18.96 ± 0.08	3.25 ± 0.08	3.47 ± 0.14	33.13 ± 0.25	33.13 ± 0.26	12.01 ± 0.20	12.02 ± 0.22
RAFT	29.09 ± 0.03	29.17 ± 0.11	3.75 ± 0.05	3.63 ± 0.10	48.83 ± 0.35	48.93 ± 0.29	17.97 ± 0.29	17.81 ± 0.27
SpyNet	9.00 ± 0.01	9.01 ± 0.03	5.31 ± 0.01	5.35 ± 0.06	12.10 ± 0.02	12.08 ± 0.05	16.47 ± 0.03	16.44 ± 0.05
MPI Sintel (clean)								
GMA	16.87 ± 0.14	16.76 ± 0.11	1.75 ± 0.15	1.85 ± 0.10	29.25 ± 0.38	29.05 ± 0.38	8.58 ± 0.34	8.82 ± 0.37
PWCNet	12.20 ± 0.21	12.18 ± 0.07	4.87 ± 0.17	4.75 ± 0.12	20.57 ± 0.21	20.43 ± 0.21	13.20 ± 0.13	13.21 ± 0.29
RAFT	16.42 ± 0.03	16.46 ± 0.05	1.69 ± 0.04	1.65 ± 0.06	29.01 ± 0.11	29.20 ± 0.01	7.67 ± 0.11	7.47 ± 0.05
SpyNet	9.69 ± 0.01	9.75 ± 0.07	6.40 ± 0.05	6.35 ± 0.00	13.08 ± 0.01	13.17 ± 0.03	18.75 ± 0.02	18.76 ± 0.06
MPI Sintel (final)								
GMA	17.34 ± 0.07	17.31 ± 0.11	0.53 ± 0.07	0.54 ± 0.11	32.11 ± 0.20	32.04 ± 0.24	4.57 ± 0.22	4.64 ± 0.24
PWCNet	13.61 ± 0.10	13.44 ± 0.14	3.52 ± 0.13	3.66 ± 0.12	23.00 ± 0.30	23.01 ± 0.06	10.84 ± 0.28	10.75 ± 0.05
RAFT	17.38 ± 0.04	17.36 ± 0.03	0.55 ± 0.09	0.50 ± 0.03	32.72 ± 0.22	32.72 ± 0.14	3.71 ± 0.21	3.75 ± 0.13
SpyNet	11.56 ± 0.01	11.59 ± 0.03	4.97 ± 0.01	4.97 ± 0.01	16.51 ± 0.01	16.55 ± 0.06	16.52 ± 0.01	16.47 ± 0.05

Table 15: Comparison of clean and adversarial performance of image reconstruction models, as considered by (Agnihotri et al., 2023a). '+ADV' denotes FGSM adversarial training with a 50-50 mini-batch split for generating an adversarial sample.

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(Zamir et al., 2022)	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(Chen et al., 2022)	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(Chen et al., 2022)	**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

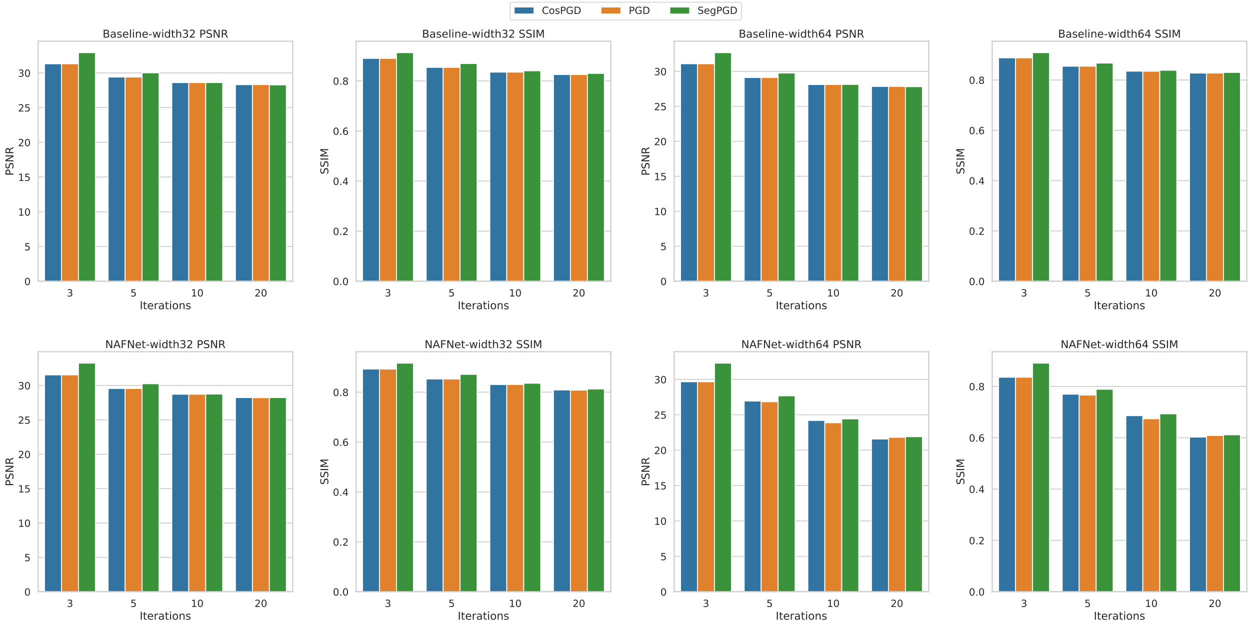


Figure 21: Comparing CosPGD to PGD and SegPGD as l_∞-norm constrained non-targeted attacks for the image denoising task using Baseline network (top row) and NAFNet (bottom row) on SSID dataset. A lower value of PSNR and SSIM indicate a stronger attack.