Aug-NeRF: Training Stronger Neural Radiance Fields with Triple-Level Physically-Grounded Augmentations

Tianlong Chen1*, Peihao Wang1*, Zhiwen Fan1, Zhangyang Wang1
1The University of Texas at Austin
{tianlong.chen, peihaowang, zhiwenfan, atlaswang}@utexas.edu

Abstract

Neural Radiance Field (NeRF) regresses a neural parameterized scene by differentially rendering multi-view images with ground-truth supervision. However, when interpolating novel views, NeRF often yields inconsistent and visually non-smooth geometric results, which we consider as a generalization gap between seen and unseen views. Recent advances in convolutional neural networks have demonstrated the promise of advanced robust data augmentations, either random or learned, in enhancing both in-distribution and out-of-distribution generalization. Inspired by that, we propose Augmented NeRF (Aug-NeRF), which for the first time brings the power of robust data augmentations into regularizing the NeRF training. Particularly, our proposal learns to seamlessly blend worst-case perturbations into three distinct levels of the NeRF pipeline with physical grounds, including (1) the input coordinates, to simulate imprecise camera parameters at image capture; (2) intermediate features, to smoothen the intrinsic feature manifold; and (3) pre-rendering output, to account for the potential degradation factors in the multi-view image supervision. Extensive results demonstrate that Aug-NeRF effectively boosts NeRF performance in both novel view synthesis (up to 1.5dB PSNR gain) and underlying geometry reconstruction. Furthermore, thanks to the implicit smooth prior injected by the triple-level augmentations, Aug-NeRF can even recover scenes from heavily corrupted images, a highly challenging setting untackled before. Our codes are available in https://github.com/VITA-Group/Aug-NeRF.

1. Introduction

Neural radiance fields (NeRF) [29] and its variants have demonstrated impressive progresses in learning to represent 3D objects and scenes from images towards photo-realistic novel view synthesis. NeRF leverages a multi-layer perceptron (MLP) to implicitly modeling the mapping from an input 5D coordinates (i.e., 3D coordinates (x, y, z) and 2D viewing directions (θ, φ)) to volume density σ and view-dependent emitted radiance color (r, g, b) at the corresponding position in the scene. Then, the obtained continuous 5D function (i.e., MLP) can be utilized to generate novel views with traditional volume rendering mechanisms.

Although NeRF is capable of producing novel views, it unfortunately suffers from inconsistent and non-smooth geometries since the vanilla MLP lacks geometry-awareness. For example, as shown in Fig. 1, the depth maps and 3D geometries of the scene generated by NeRF show obvious discontinuity and outliers, especially around the edge of objects. Considering that the quality of reconstructed geometry plays a central role in view rendering, that might account for NeRF’s limited generalization to unseen views.

To fill in this research gap, a straightforward solution is introducing explicit geometric regularizers like Laplacian [17, 45] or total variation (TV) [84] to enhance the continuity. However, these explicit regularizers are often

*Equal Contribution.
found to constrain the representation flexibility of MLP too aggressively, resulting in inferior performance. Recent advances in robust data augmentations [72] establish promising successes in image recognition in terms of both improved functional smoothness and generalization.

Motivated by that, we design an Augmented NeRF (Aug-NeRF) training framework, which injects worst-case perturbations [23] to implicitly regularize the NeRF pipeline with physical foundations. Specifically, Aug-NeRF considers to regularize three different levels, including (i) the input coordinates, where perturbations can imitate the inaccurate camera poses during collecting images; (ii) the intermediate features, in order for a smooth/flat model loss landscape [5,8,70] when fitting objects’ 3D geometries that is believed to enhance generalization; (iii) the pre-rendering output, to model potential degradation factors in the image supervision. As presented in Fig. 1, our Aug-NeRF achieves smoother and more consistency reconstructed geometry and improved unseen view synthesis. Additionally, we find Aug-NeRF to show surprising resilience towards severely corrupted supervision images. The main contributions of this paper can be summarized as follows:

- We reveal the existence of highly non-smooth geometries in representing scenes as neural radiance fields (NeRF), which we regard as a crucial bottleneck of NeRF’s generalization ability to unseen views.

- To address such limitation of NeRF, we propose Aug-NeRF, a triple-level, physically-grounded augmented training pipeline, by leveraging worst-case perturbations to implicated regularize the input coordinate, intermediate feature, and pre-rendering output levels.

- Extensive experiments validate the effectiveness of our proposal on diverse scene synthesis tasks, to endow NeRF with smoothness-aware geometry reconstruction, enhanced generalization to synthesizing unseen views, and stronger tolerance of noisy supervisions.

2. Related Work

Adversarial Training and Robust Augmentation. It is well-known that deep networks are vulnerable to imperceptible worst-case perturbations [11, 16, 23]. Numerous defense mechanisms [31,38,47,55,57,80] have been invented to address the issue, where adversarial training (AT) approaches [11, 16, 23] remains as the de-facto. Although conventional AT enhances model robustness at the price of compromising the standard accuracy [62], recent studies reveal AT can be harnessed to enhance models’ standard generalization as well [9,63,68,72,86]. Taking [72] for example, it applies adversarial perturbations to input samples as a form of data augmentation, and shows to improve image classification on the clean dataset. [9,63,86] apply worst-case perturbations to the input embedding for natural language understanding, language modeling, and vision-and-language tasks, all successfully boosting their standard generalization. [14,40,64,83] constructed more sophisticated variations of robust augmentations, including both data-driven and heuristic components, to improve model generalization further. However, such robust augmentations on inputs or intermediate features, to our best knowledge, have not been studied in the view synthesis field. This paper explores this possibility by looking into the intrinsic physical grounds.

Neural 3D Representations. Classic 3D reconstruction approaches utilizes discrete representations such as point clouds [1, 71], meshes [43, 44, 61], multi-plane images [28, 53, 54, 85], depth maps [12, 49, 74] and voxel grids [18, 21, 37, 48, 51, 58]. Neural implicit representations leverage coordinate-based neural networks to approximate visual signals [27, 34, 36]. Such ideas have been successfully applied to both 2D images [20, 50, 60] and 3D objects [6, 46, 50]. Recent advances follow differentiable rendering and end-to-end optimization to reconstruct the neural 3D scene from 2D image supervision [29, 33, 75]. Liu et al. [20] presented the first usage of neural implicit function to infer 3D representation with differentiable rendering. DVR [33] and IDR [75] adopt surface rendering to reconstruct implicit iso-surface by supervising on both images and pixel-accurate object masks.

NeRF [29] pioneered to use differentiable volumetric rendering to optimize a neural radiance field, and achieved more photorealistic and view-consistent results. Many works continue to improve its training and rendering accuracy, efficiency, and generalization. NeRF++ [81] separates two NeRFs to handle foreground and background, respectively. NeRF-W [24] tackles unstructured photos via modeling transient noises and uncertainty. MipNeRF [3] mitigates objectionable aliasing artifacts for NeRF to represent fine details. HyperNeRF [35] introduces topology-aware level-set methods to rectify NeRF geometry especially for dynamics. [10, 39, 42, 69] extend NeRF to broader scenarios with specific lighting and rendering modeling. [4, 65, 79] leverage multi-view spatial image feature or semi-reconstructed 3D information to reduce input view number and enhance the generalization to new scenes. [26, 67, 76] free NeRF from accurate camera pose estimation. Acceleration of NeRF training and inference have also been discussed in [41, 59, 78]. Despite so many exciting progresses, studying NeRF’s training stability and data robustness remains an open question.

3. Preliminaries

NeRF models the underlying 3D scene as a continuous volumetric radiance field of color and density. Formally, a typical radiance field can be written as $F : (x, \theta) \rightarrow (c, \sigma)$, where $x \in \mathbb{R}^3$ is the spatial coordinate, $\theta \in [-\pi, \pi]^2$ indicates the view direction, and $c \in \mathbb{R}^3, \sigma \in \mathbb{R}_+$ represent the RGB color and density, respectively. NeRF further parameterizes this 5D-valued function by a composition of
With this forward model, NeRF optimizes the expected distance between rendered ray colors and ground-truth pixel colors as follows:

\[ P \left[ (r, C | \Theta) \right] = \frac{1}{Z} \exp \left( -\frac{1}{2\Sigma} \left\| C(r | \Theta) - \tilde{C} \right\|_2^2 \right), \]

where \( Z \) defines a probability measure supported in the ray space \( \mathcal{R} \).

4. Methodology

Overview. NeRF conducts uniform sampling along each ray and interpolates a continuous radiance field via an MLP. However, we argue that the point sampling and the MLP interpolation can never be optimal during training dynamics due to the biased sampling strategy and non-smoothness of MLP. To this end, we propose to train NeRF with a smoothing prior. Sec. 4.1 provides a probabilistic interpretation of this intuition. Different from explicit smoothness modeling, e.g., total variation penalty or low rank prior, we utilize worst-case perturbations as a data-adaptive regularization. We call this training strategy Aug-NeRF.

An overview of our Aug-NeRF is presented in Fig. 2. Following the rendering pipeline of NeRF, Aug-NeRF injects adversarial noises into the following stages: point sampling, intermediate features, and MLP outputs. Each perturbation is searched within a small range to maximize the final loss. It could be treated as a regularization to be jointly minimized with the original training loss (see Sec. 4.2).

4.1. NeRF as Maximum A Posterior

Fitting a neural radiance field to satisfy multi-view observations can be modeled as a Maximal Likelihood (ML) problem \( \Theta^* = \arg \max_{\Theta} P(\mathcal{R} | \Theta) \), which can be derived as:

\[ \Theta^* = \arg \min_{\Theta} -E_{(r, C) \sim P(\mathcal{R})} \log P(r, \tilde{C} | \Theta), \]

by assuming each ray is conditionally independent given network parameters. Optimizing NeRF by MSE loss (Eqn. 2) can be obtained by regarding the conditional distribution \( P(r, \tilde{C} | \Theta) \) as a Gaussian distribution:

\[ P(r, \tilde{C} | \Theta) = \frac{1}{Z} \exp \left( -\frac{1}{2\Sigma} \left\| C(r | \Theta) - \tilde{C} \right\|_2^2 \right), \]

where \( Z \) defines a probability measure supported in the ray space \( \mathcal{R} \).
where $Z$ is a normalization term, and $\Sigma$ is the variance.

However, the maximum likelihood does not introduce any prior on the reconstructed NeRF as MLP is a universal approximator. Instead, we consider the Maximum A Posteriori (MAP) form $P(\Theta|\mathcal{R})$ to inject the prior for robust training. By Bayesian rule, we have $P(\Theta|\mathcal{R}) = P(\mathcal{R}|\Theta)P(\Theta)$, where $P(\Theta)$ is some prior distribution of the network weights $\Theta$. Hence, maximizing the original loss (Eqn. 2) plus a penalty term:

$$\mathcal{L}(\Theta|\mathcal{R}) = E_{(r,\hat{C}) \sim P(\mathcal{R})} \left\| C(r|\Theta) - \hat{C} \right\|^2_2 + \lambda R(\Theta), \quad (3)$$

where $R(\Theta) = -\log P(\Theta)/\lambda$. Here we expect $\Theta$ to induce a geometry-aware smooth $F_\theta$.

4.2. Regularize NeRF with Robust Augmentations

Imposing smoothness onto NeRF can be done in many explicit ways, such as regularizing total variation [84], Laplacian of surface [7,52,77], etc. However, those regularizers are often not sufficiently data-adaptive, and can constrain the representation flexibility too aggressively, as evidenced in Sec. 5.3. Also, their computation also usually operates on discretized volumetric representations, and needs extra differentiation steps to be added in NeRF.

Recent works [5,9,63,68,72,86] suggest a promising alternative by integrating worst-case adversarial perturbations as data augmentations (i.e., AT). AT restricts the change of loss when its input is perturbed, leading to flattening the loss landscape [30,56]. As a result, the trained network’s intrinsic feature manifold and loss landscape become smoother. Prevailing theories [15,19,32] link the generalization ability of deep networks to the geometry of the loss landscape; in particular, a model trained to converge to wide valleys (i.e., flat basins) in loss landscape shows better generalization ability as well as robustness to distributional shifts.

NeRF is trained by given 2D image views (often with known camera poses) and is tested to synthesize novel views from unseen angles. Intuitively, the unsatisfactory novel view synthesis could be seen as a training-test “generalization gap” issue. This inspires us to incorporate robust augmentations into NeRF to induce a data-adaptive smoothness prior that enhances generalization.

Designing dedicated perturbations for NeRF is far from trivial due to its inherent physics. Unlike conventional deep models, the forward pass of NeRF consists of two white-box simulating stages (point sampling, volumetric rendering) and one black-box network mapping stage. We propose to inject worst-case perturbations into all three levels: coordinates, intermediate features of MLP, and pre-rendering MLP output: all with clear physical meanings. Formally, our approach can be formulated as a min-max game:

$$\min_{\Theta} E_{(r,\hat{C}) \sim P(\mathcal{R})} \max_{\delta} \left\| C(r|\Theta, \delta) - \hat{C} \right\|^2_2,$$

where $\delta = (\delta_p, \delta_f, \delta_r) \in \mathcal{S}_p \times \mathcal{S}_f \times \mathcal{S}_r$, (4)

where $\delta_p$, $\delta_f$, and $\delta_r$ are the perturbations to be learned and injected to the input coordinate, intermediate MLP feature, and pre-rendering RGB-σ output, respectively, where $\mathcal{S}_p \subseteq \mathbb{R}^6$, $\mathcal{S}_f \subseteq \mathbb{R}^D$, and $\mathcal{S}_r \subseteq \mathbb{R}^4$ are the corresponding perturbation search range, $D$ is the hidden dimension of the MLP. We elaborate on each perturbation as below.

Input Coordinate Perturbation. The original NeRF first randomly samples point along each ray and then conducts importance sampling to simulate the quadrature of the integration. This strategy also mitigates overfitting and produces smoother scene representation [29]. Arandjelovic et al. [2] further proposes an attention-guided sampling scheme to refine this process. However, our insight is that using either coarse-to-fine or learning-based sampling will cause the sampling to overfit the density distribution of the currently rendered ray, which might hold back NeRF when the density field is biased or cannot generalize.

To this end, we propose to produce a worst-case point sampling during training, to simulate a test-time “distributional shift” for NeRF to handle. To be specific, we search a coordinate perturbation $\delta_{xyz} = (\delta_x, \delta_y, \delta_z)$ following Eqn. 4. The coordinate perturbation $\delta_p = (\delta_t, \delta_{xyz}, \delta_\theta)$ consists of three parts: 1) the along-ray perturbation $\delta_t \in \mathbb{R}$ shifts point samples along the ray, 2) the point position perturbation $\delta_{xyz} \in \mathbb{R}^3$ is added to the direct input of the NeRF MLP, 3) in addition, we also inject the perturbation $\delta_\theta \in \mathbb{S}^3$ to the view direction. Formally, given the perturbation $\delta_p$, the input of MLP turns out to be:

$$t_k^t = t_k + \delta_t, \quad \theta^t = \theta + \delta_\theta, \quad p_k^t = o + t_k^t d + \delta_{xyz}.$$

The constraint set for $\delta_\theta$ is defined as $\delta_\theta \leq [\alpha_{\epsilon} (t_k+1 - t_k)]$, where $\alpha_{\epsilon}$ is a hyperparameter. The coordinate perturbation $\delta_{xyz}$ lies in a ball $B(0, \epsilon_p)$ to constrain points with a cylinder along the ray. View direction perturbation $\delta_\theta$ is restricted within the conical frustum $[-\epsilon_p/2f, \epsilon_p/2f]^2$, where $f$ is the focal length, and $\epsilon_p$ is the pixel size.

Pre-Rendere Output Perturbation. NeRF next maps points on a ray to the corresponding color and density, then conducts volumetric rendering to compose these point values into the 2D pixel values. As shown by Fig. 1, the reconstructed shape can be noisy and discontinuous. We attribute these artifacts to two reasons: (i) neural implicit functions represented by MLP are not necessarily smooth [60, 73]. When zooming in, we observe the function landscape to be rugged; (ii) the MLP output goes through volumetric rendering to form the RGB output. As the volumetric rendering
where the bone MLP can be written as

\[
\text{inputs or outputs. To be specific, according to [29], the back-}
\text{learned functional mappings, more than just augmenting}
\text{by [5], augmenting intermediate features can further smooth}
\text{adversarial noise into the intermediate features. As revealed}
\text{turbing per-rendering color and ray density, we also inject}
\text{In addition to per-}
\text{turb the rendering in Eqn. 1 by:}
\text{coordinates,}
\text{timated by multi-step Projected Gradient Descent (PGD).}
\text{To search for the worst-case perturbation in Eqn. 4, we}
\text{ crafting the worst-case perturbations as follows:}
\text{we}
\text{Intermediate Feature Perturbation. In addition to per-}
\text{turb per-rendering color and ray density, we also inject}
\text{adversarial noise into the intermediate features. As revealed}
\text{by [5], augmenting intermediate features can further smooth}
\text{learned functional mappings, more than just augmenting}
\text{inputs or outputs. To be specific, according to [29], the back-}
\text{bone MLP can be written as (c(p), σ(p)) = F\theta(p, θ) = (g ◦ f(p, θ), h ◦ f(p)), where f(·) (with positional encod-}
\text{ing) maps a coordinate to a D-dimension feature vector, and}
g(·), h(·) project it to RGB color and density, respectively.}
\text{We crafted the worst-case perturbations as follows:}
\text{Intermediate feature perturbation is searched over S_f =}
\text{[−ε_f, ε_f]^D with a hyperparameter ε_f. We also test various}
\text{injection points of the backbone MLP in Sec. 5.3.}

### 4.3. Optimization

To search for the worst-case perturbation in Eqn. 4, we introduce a theoretically guaranteed way to reach the maximum. We only consider additive perturbation here, and all search spaces (i.e., S_p, S_f, S_i) are defined as \(\ell_p\) norm ball with a radius \(ε > 0\). The radius \(ε\) is the maximum magnitude of the perturbation, which can roughly signify the strength of the perturbation. The perturbations can be accurately estimated by multi-step Projected Gradient Descent (PGD). Taking \(\ell_∞\) norm ball for example:

\[
δ^{(t+1)} = \Pi_{||δ||_∞ ≤ ε} \left[ δ^{(t)} + α \cdot \text{sgn}(∇_δ\mathcal{L}(Θ|R, δ)) \right]
\]

where \(α\) is the step size of the inner maximization, \(\Pi[·]\) denotes a projection operator, \(\text{sgn}(·)\) takes the sign of the input, and \(\mathcal{L}(Θ|R, δ)\) represents the MSE loss between perturbed color \(C^p\) and ground-truth color \(C\) (see Eqn. 4).

After incorporating all augmentations, the full training objective is defined as \((λ = 0.1\) as tuned by grid search):
5. Experiments

5.1. Implementation details.

Datasets. We evaluate our proposals on public representative datasets of both LLFF [28] and NeRF-Synthetic [29]. Particularly, the face-forwarding scenes (“fern”, “orchids”, “trex”) from LLFF dataset and {“drums”, “ship”, and “chair”} instances in 360° NeRF-Synthetic dataset are adopted in our experiments. To accelerate training, we down-sampled LLFF dataset by 1/8 and 360° NeRF-Synthetic dataset by 1/2.

Training. We employ the same MLP architecture and training recipe with the original NeRF. Aug-NeRF is trained for 500K iterations to guarantee convergence. All hyperparameters are carefully tuned by a grid search and the best configuration is applied to all experiments, as demonstrated in Sec. 5.3. NeRF models are trained on a NVIDIA RTX A6000 GPU with 48 GB memory.

Evaluation. We report three error metrics including peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) [66], and learned perceptual image patch similarity (LPIPS) [82]. Meanwhile, to provide a comprehensive comparison, we also follow [3] and show an “average” error metric by computing the geometric mean of MSE = 10^{−\text{PSNR}/10}, \sqrt{1 - \text{SSIM}}, and LPIPS.

Baseline and Comparison Variants. Our Aug-NeRF is established on the vanilla NeRF [29]. Two groups of current top-performers for view synthesis are compared, including (i) NeRF-based approaches: NeRF [29] and MipNeRF [3]; and (ii) classical methods: Neural Volume (NV) [21], Scene Representation Network (SRN) [50], and Local Light Field Fusion (LLFF) [28]. For a fair comparison, all above models are trained/tested on the same views of identical scenes.

5.2. Improved NeRF with Augmentations

Results on LLFF and 360° NeRF-Synthetic datasets. In this section, we validate our proposed Aug-NeRF on LLFF and 360° NeRF-Synthetic datasets across six representative scenes. Quantitative comparisons against vanilla NeRF and other top-performers for novel view synthesis. Performance is reported on the LLFF test set. ↑/↓ means that larger/smaller numbers denote better performance.

| Scene “fern” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|-------------|--------|--------|---------|-----------|
| SRN [50]    | 21.37  | 0.611  | 0.459   | 0.128     |
| LLFF [50]   | 22.85  | 0.753  | 0.247   | 0.086     |
| NeRF [29]   | 25.17  | 0.792  | 0.280   | 0.073     |
| MipNeRF [3] | 26.24  | 0.839  | 0.193   | 0.057     |
| Aug-NeRF (Ours) | 26.51  | 0.830  | 0.168   | 0.054     |

| Scene “orchids” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|-----------------|--------|--------|---------|-----------|
| SRN [50]        | 17.37  | 0.611  | 0.467   | 0.175     |
| LLFF [28]       | 18.52  | 0.588  | 0.313   | 0.141     |
| NeRF [29]       | 20.36  | 0.641  | 0.321   | 0.121     |
| MipNeRF [3]     | 20.87  | 0.663  | 0.262   | 0.108     |
| Aug-NeRF (Ours) | 21.60  | 0.675  | 0.243   | 0.099     |

| Scene “trex” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|--------------|--------|--------|---------|-----------|
| SRN [50]     | 22.87  | 0.761  | 0.298   | 0.091     |
| LLFF [28]    | 24.15  | 0.857  | 0.222   | 0.069     |
| NeRF [29]    | 26.80  | 0.880  | 0.249   | 0.056     |
| MipNeRF [3]  | 27.55  | 0.894  | 0.208   | 0.049     |
| Aug-NeRF (Ours) | 28.17  | 0.881  | 0.206   | 0.048     |

These results convey several observations:

1. Aug-NeRF reduces average error by 14.3% ~ 26.0% and 12.5% ~ 44.7% on the LLFF and 360° NeRF-Synthetic datasets, respectively. It consistently outperforms NeRF on all metrics by a large margin, e.g., {1.34, 1.24, 1.37, 1.33, 0.53, 0.87} PSNR improvements at scenes {“fern”, “orchids”, “trex”, “drums”, “ship”, “chair”}; showing impressive “generalization” boosts on unseen views thanks to our augmentations.

2. Compared with recent state-of-the-art MipNeRF and other classical approaches, Aug-NeRF shows a clear
Table 2. Quantitative comparison of our Aug-NeRF against NeRF and other top-performer for novel view synthesis. Performance is reported on the test set of 360° NeRF-Sythetic dataset. ↑/↓ denotes that larger/smaller numbers denote better performance.

| Scene “drums” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|---------------|---------|--------|---------|-----------|
| SRN [50]      | 17.18   | 0.766  | 0.267   | 0.135     |
| NV [21]       | 22.58   | 0.873  | 0.214   | 0.075     |
| LLFF [50]     | 21.33   | 0.890  | 0.126   | 0.069     |
| NeRF [29]     | 25.01   | 0.925  | 0.091   | 0.043     |
| MipNeRF [3]   | 26.22   | 0.939  | 0.065   | 0.034     |
| Aug-NeRF (Ours)| 26.34   | 0.941  | 0.060   | 0.032     |

| Scene “ship” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|---------------|---------|--------|---------|-----------|
| SRN [50]      | 20.60   | 0.757  | 0.299   | 0.109     |
| NV [21]       | 23.22   | 0.784  | 0.276   | 0.080     |
| LLFF [50]     | 23.22   | 0.823  | 0.218   | 0.076     |
| NeRF [29]     | 28.65   | 0.856  | 0.206   | 0.047     |
| MipNeRF [3]   | 29.30   | 0.864  | 0.190   | 0.044     |
| Aug-NeRF (Ours)| 29.18   | 0.879  | 0.173   | 0.042     |

| Scene “chair” | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Average ↓ |
|---------------|---------|--------|---------|-----------|
| SRN [50]      | 26.96   | 0.910  | 0.106   | 0.040     |
| NV [21]       | 28.33   | 0.916  | 0.109   | 0.036     |
| LLFF [50]     | 28.72   | 0.948  | 0.064   | 0.027     |
| NeRF [29]     | 33.00   | 0.967  | 0.046   | 0.016     |
| MipNeRF [3]   | 33.82   | 0.972  | 0.042   | 0.014     |
| Aug-NeRF (Ours)| 33.87   | 0.972  | 0.040   | 0.014     |

Aug-NeRF achieves superior performance in representing fine geometry, as shown in Fig. 3 such as Fern’s and Orchid’s leaves, the skeleton ribs, and rading in T-rex. Both NeRF and MipNeRF reconstruct the low-frequency geometry and color variation, but fail to generate high-quality fine details (see zoom-in).

### Depth and Geometry visualization

The learned depth maps and fitted 3D geometries from NeRFs are provided in and 4 and Fig. 5, respectively. The 3D shapes (Fig. 5) are synthesized by MarchingCube algorithms [22]. We observe that vanilla NeRF suffers from a serrated surface (which overhelms the fine details), while traditional TV and Laplacian regularizations tend to excessively smoothen the results. Aug-NeRF reduces noises and improves surface smoothness, in a detail- and geometry-preserving manner.

Superior synthesis when trained on noisy data. As an extra study, we examine Aug-NeRF under supervision images with additive noise corruptions. From Tab. 3 and Fig. 6, compared to the vanilla NeRF, Aug-NeRF shows consistent 6.5% ~ 26.9% average error reductions for both Gaussian and Shot noises, while it substantially improves the visual quality of constructed test views (e.g., much fewer noises in the “fern”). We regard it as an additional bonus from enforcing smooth geometry in NeRF training.

### 5.3. Ablation Study

Multi-level v.s. single-level augmentation. To compare the effects of robust augmentations at different levels, we conduct step-wise evaluation as: (i) NeRF, (ii) NeRF + Feature Aug., (iii) NeRF + Feature & Output Aug., (iv) NeRF + Feature & Output & Input coordinates Aug., which is our complete Aug-NeRF. Tab. 4 shows that applying robust augmentation to each level brings extra and complementary generalization gains, among which augmenting the pre-rendering output level makes the biggest difference.

Worst-case v.s. random perturbations. One straightforward baseline for Aug-NeRF is to just use random data augmentation. Particularly, we employ random Gaussian noises to both intermediate features and pre-rendering outputs of NeRF. As in Tab. 4, Random Aug. obtains moderate performance boosts for all metrics, but are clearly less obvious than our worst-case perturbations.

Effects of augmentation strength and location. The accuracy gains from Aug-NeRF are largely determined by the strength and location of crafted worst-case perturbations. A comprehensive investigation on three levels of augmentations, i.e., input coordinate, intermediates features, and pre-rendering output, are presented in Fig. 7. When studying one of the factors, we stick to the best configuration for the rest factors. Fig. 7 reveals that: First, NeRF gains the most from {coordinate, features, pre-rendering output} augmentations with {PGD-3, PGD-1, PGD-1} and step size {10^{-2}, 10^{-3}, 10^{-5}}; Second, applying generated perturbations to the middle layer of NeRF’s MLP contributes the most significantly;
Comparison with explicit smooth regularizations. In contrast to our implicit smooth prior, there exists several explicit smooth regularizations which can be directly plugged into the NeRF pipeline, like $\ell_1$ sparsity Reg. $R_{\ell_1}(\Theta) = \int |\sigma_\Theta(u)| du$; $\Delta$ Laplacian Reg. $R_{\text{Lap}}(\Theta) = \int |\Delta \sigma_\Theta(u)| du$; $\oplus$ Total Variation (TV) Reg. $R_{\text{TV}}(\Theta) = \int \| \nabla \sigma_\Theta(u) \|_2^2 du$. As demonstrated in Tab. 4, although hyperparameters are carefully tuned by a grid search, both $\ell_1$ and Laplacian regularizers degrade the performance, as such explicit constraints are often too aggressive and limit the representation flexibility of NeRF. The TV Reg. can lead to positive gains but still largely lags behind our proposals.

6. Conclusion and Broad Impact

In this paper, we have presented Aug-NeRF that addresses the inherent non-smooth geometries of NeRF. Specifically, based on solid physical grounds, Aug-NeRF seamlessly injects worst-case perturbations into three levels of the NeRF pipeline, leading to substantially improved geometry continuity and generalization ability. Extensive quantitative and qualitative results across diverse scenes validate the effectiveness of our proposals. Moreover, the implicit smooth prior induced by triple-level augmentation enables NeRF to recover scenes from noisy supervision images. One limitation is that we only study additive noises (e.g., Gaussian) for corrupted images. We will extend the investigation to other complicated corruptions.
References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In European Conference on Computer Vision (ECCV), 2020.

[2] Relja Arandjelović and Andrew Zisserman. Nerf in detail: Learning to sample for view synthesis. arXiv preprint arXiv:2106.05264, 2021.

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In IEEE International Conference on Computer Vision (ICCV), 2021.

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In IEEE International Conference on Computer Vision (ICCV), 2021.

[5] Tianlong Chen, Yu Cheng, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zhangyang Wang, and Jingjie Liu. Adversarial feature augmentation and normalization for visual recognition. arXiv preprint arXiv:2103.12171, 2021.

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[7] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. International Journal of Computer Vision (IJCV), 2007.

[8] Farzan Farnia, Jesse M Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In International Conference on Learning Representations (ICLR), 2019.

[9] Zhe Gan, Yan-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjie Liu. Large-scale adversarial training for vision-and-language representation learning. In Advances in Neural Information Processing Systems (NeurIPS), 2020.

[10] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. arXiv preprint arXiv:2012.05903, 2020.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR), 2015.

[12] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhao Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In International Conference on Learning Representations (ICLR), 2019.

[14] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In International Conference on Learning Representations, 2019.

[15] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In International Conference on Learning Representations (ICLR), 2020.

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In International Conference on Learning Representations Workshop (ICLRW), 2017.

[17] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[18] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. International Journal of Computer Vision (IJCV), 2000.

[19] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Advances in Neural Information Processing Systems (NeurIPS), 2018.

[20] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

[21] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. In ACM Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH), 2019.

[22] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In ACM Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH), 1987.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations (ICLR), 2018.

[24] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[25] Nelson Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics (TVCG), 1995.

[26] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In IEEE International Conference on Computer Vision (ICCV), 2021.

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[28] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and...
[56] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[57] Ke Sun, Zhanxing Zhu, and Zhouchen Lin. Towards understanding adversarial examples systematically: Exploring data size, task and model factors. *arXiv preprint arXiv:1902.11019*, 2019.

[58] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *International Conference on Computer Vision (ICCV)*, 1998.

[59] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[60] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghaban, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[61] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 2019.

[62] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[63] Dilin Wang, Chengyue Gong, and Qiang Liu. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning (ICML)*, 2019.

[64] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in Neural Information Processing Systems*, 34, 2021.

[65] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Iibnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[66] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 2004.

[67] Zirui Wang, Zhangze Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-+: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[68] Colin Wei and Tengyu Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. In *International Conference on Learning Representations (ICLR)*, 2020.

[69] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[70] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[71] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[72] Chiang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[73] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[74] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mysnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018.

[75] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[76] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. Inerf: Inverting neural radiance fields for pose estimation. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[77] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[78] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[79] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. Pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[80] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

[81] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[82] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[83] Xiaofeng Zhang, Zhangyang Wang, Dong Liu, and Qing Ling. Dada: Deep adversarial data augmentation for extremely
low data regime classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2807–2811. IEEE, 2019.

[84] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[85] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2018.

[86] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2020.