RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs

Michael Niemeyer\textsuperscript{1,2,3*}  Jonathan T. Barron\textsuperscript{3}  Ben Mildenhall\textsuperscript{3}  Mehdi S. M. Sajjadi\textsuperscript{3}  Andreas Geiger\textsuperscript{1,2}  Noha Radwan\textsuperscript{3}

\textsuperscript{1}Max Planck Institute for Intelligent Systems, Tübingen  \textsuperscript{2}University of Tübingen  \textsuperscript{3}Google Research

\{firstname.lastname\}@tue.mpg.de  \{barron, bmild, msajjadi, noharadwan\}@google.com

https://m-niemeyer.github.io/regnerf

Abstract

Neural Radiance Fields (NeRF) have emerged as a powerful representation for the task of novel view synthesis due to their simplicity and state-of-the-art performance. Though NeRF can produce photorealistic renderings of unseen viewpoints when many input views are available, its performance drops significantly when this number is reduced. We observe that the majority of artifacts in sparse input scenarios are caused by errors in the estimated scene geometry, and by divergent behavior at the start of training. We address this by regularizing the geometry and appearance of patches rendered from unobserved viewpoints, and annealing the ray sampling space during training. We additionally use a normalizing flow model to regularize the color of unobserved viewpoints. Our model outperforms not only other methods that optimize over a single scene, but in many cases also conditional models that are extensively pre-trained on large multi-view datasets.

1. Introduction

Coordinate-based neural representations [7, 34, 35, 44] have gained increasing popularity in the field of 3D vision. In particular, Neural Radiance Fields (NeRF) [37] have emerged as a powerful representation for the task of novel view synthesis, where the goal is to render unseen viewpoints of a scene from a given set of input images. Though NeRF achieves state-of-the-art performance, it requires dense coverage of the scene. However, in real-world applications such as AR/VR, autonomous driving, and robotics, the input is typically much sparser, with only few views of any particular object or region available per scene. In this sparse setting, the quality of NeRF’s rendered novel views drops significantly (see Fig. 1).

Several works have proposed conditional models to overcome these limitations [6, 8, 30, 56, 58, 62]. These models require expensive \textit{pre-training}, i.e. training the model on large-scale datasets of many scenes with multi-view images and camera pose annotations, as opposed to \textit{test-time optimization} which is done from scratch for a given test scene. At test time, novel views can be generated from only a few input images through amortized inference, optionally combined with per scene test time fine-tuning. Though these models achieve promising results, obtaining the necessary...

*The work was primarily done during an internship at Google.
NeRF optimizes the reconstruction loss for a given set of input images (blue cameras). For sparse inputs, however, this leads to degenerate solutions. In this work, we propose to sample unobserved views (red cameras) and regularize the geometry and appearance of patches rendered from those views. More specifically, we cast rays through the scene and render patches from unobserved viewpoints for a given radiance field $f_{\theta}$. We then regularize appearance by feeding the predicted RGB patches through a trained normalizing flow model $\phi$ and maximizing predicted log-likelihood. We regularize geometry by enforcing a smoothness loss on the rendered depth patches. Our approach leads to 3D-consistent representations even for sparse inputs from which realistic novel views can be rendered.

Pre-training data by capturing or rendering many different scenes can be prohibitively expensive. Moreover, these techniques may not generalize well to novel domains at test time, and may exhibit blurry artifacts as a result of the inherent ambiguity of sparse input data.

One alternate approach is to optimize the network weights from scratch for every new scene and introduce regularization to improve the performance for sparse inputs, e.g., by adding extra supervision [24] or learning embeddings representative of the input views [19]. However, existing methods either heavily rely on external supervisory signals that might not always be available, or operate on low-resolution renderings of the scene that provide only high-level information.

Contribution: In this paper, we present RegNeRF, a novel method for regularizing NeRF models for sparse input scenarios. Our main contributions are the following:

- A patch-based regularizer for depth maps rendered from unobserved viewpoints, which reduces floating artifacts and improves scene geometry.
- A normalizing flow model to regularize the colors predicted at unseen viewpoints by maximizing the log-likelihood of the rendered patches and thereby avoid color shifts between different views.
- An annealing strategy for sampling points along the ray, where we first sample scene content within a small range before expanding to the full scene bounds which prevents divergence early during training.

2. Related Work

Neural Representations: In 3D vision, coordinate-based neural representations [7, 34, 35, 44] have become a popular representation for various tasks such as 3D reconstruction [1, 7, 13, 14, 34, 40, 43–45, 48, 51, 55, 57], 3D-aware generative modelling [5, 9, 15, 16, 33, 38, 39, 42, 49, 64], and novel-view synthesis [2, 3, 12, 22, 25, 28, 32, 37, 41, 52, 60, 61]. In contrast to traditional representations like point clouds, meshes, or voxels, this paradigm represents 3D geometry and color information in the weights of a neural network, leading to a compact representation. Several works [29, 37, 41, 52, 61] proposed differentiable rendering approaches to learn neural representations from only multi-view image supervision. Among these, Neural Radiance Fields (NeRF) [37] have emerged as a powerful method for novel-view synthesis due to its simplicity and state-of-the-art performance. In mip-NeRF [2], point-based ray tracing is replaced using cone tracing to combat aliasing. As this is a more robust representation for scenes with various camera distances and reduces NeRF’s coarse and fine MLP networks to a single multiscale MLP, we adopt mip-NeRF as our scene representation. However, compared to previous works [2, 37], we consider a much sparser input scenario in which neither NeRF nor mip-NeRF are able to produce realistic novel views. By regularizing scene geometry and appearance, we are able to synthesize high-quality renderings despite only using as few as 3 wide-baseline input images.

Sparse Input Novel-View Synthesis: One approach for circumventing the requirement of dense inputs is to aggregate prior knowledge by pre-training a conditional model of radiance fields [6, 8, 20, 27, 30, 47, 56, 58, 62]. We restrict the following discussion and experimental comparison to [6, 8, 62] as they, in contrast to the other works, explicitly consider sparse input scenarios. PixelNeRF [62] and Stereo Radiance Fields [8] use local CNN features extracted from the input images, whereas MVSNeRF [6] obtains a 3D cost volume via image warping which is then processed by a 3D CNN. Though they achieve compelling results, these methods require a multi-view image dataset of many different scenes for pre-training, which is not always readily available and may be expensive to obtain. Further, most approaches require fine-tuning the network weights at test time despite the long pre-training phase, and the quality of
novel views is prone to drop when the data domain changes at test time. Tancik et al. [54] learn network initializations from which test time optimization on a new scene converges faster. This approach assumes that the training and test data are taken from the same domain, and results may degrade if the domain changes at test time.

In this work, we explore an alternative approach which avoids expensive pre-training by regularizing appearance and geometry in novel (virtual) views. Previous works in this direction include DS-NeRF [24] and DietNeRF [19]. DS-NeRF improves reconstruction accuracy by adding additional depth supervision. In contrast, our approach only uses RGB images and does not require depth input. DietNeRF [19] compares CLIP [11, 46] embeddings of unseen viewpoints rendered at low resolutions. This semantic consistency loss can only provide high-level information and does not improve scene geometry for sparse inputs. Our approach instead regularizes scene geometry and appearance based on rendered patches and applies a scene space annealing strategy. We find that our approach leads to more realistic scene geometry and more accurate novel views.

3. Method

We propose a novel optimization procedure for neural radiance fields from sparse inputs. More specifically, our approach builds upon mip-NeRF [2], which uses a multiscale radiance field model to represent scenes (Sec. 3.1). For sparse views, we find the quality of mip-NeRF’s view synthesis drops mainly due to incorrect scene geometry and training divergence. To overcome this, we propose a patch-based approach to regularize the predicted color and geometry from unseen viewpoints (Sec. 3.2). We also provide a strategy for annealing the scene sampling bounds to avoid divergence at the beginning of training (Sec. 3.3). Finally, we use higher learning rates in combination with gradient clipping to speed up the optimization process (Sec. 3.4). Fig. 2 shows an overview of our method.

3.1. Background

Neural Radiance Fields A radiance field is a continuous function \( f \) mapping a 3D location \( x \in \mathbb{R}^3 \) and viewing direction \( d \in S^2 \) to a volume density \( \sigma \in [0, \infty) \) and color value \( c \in [0, 1]^3 \). Mildenhall et al. [37] parameterize this function using a multi-layer perceptron (MLP), where the weights of the MLP are optimized to reconstruct a set of input images of a particular scene:

\[
(\gamma(x), \gamma(d)) \mapsto (c, \sigma).
\]

Here, \( \theta \) indicates the network weights and \( \gamma \) a predefined positional encoding [37, 55] applied to \( x \) and \( d \).

Volume Rendering: Given a neural radiance field \( f_\theta \), a pixel is rendered by casting a ray \( r(t) = o + td \) from the camera center \( o \) through the pixel along direction \( d \). For given near and far bounds \( t_n \) and \( t_f \), the pixel’s predicted color value \( \hat{c}_\theta \) is computed using alpha compositing:

\[
\hat{c}_\theta(r) = \int_{t_n}^{t_f} T(t) \sigma_\theta(r(t)) c_\theta(r(t), d) \, dt
\]

where \( T(t) = \exp \left( - \int_{t_n}^{t} \sigma_\theta(r(s)) \, ds \right) \), and \( \sigma_\theta(\cdot) \) and \( c_\theta(\cdot, \cdot) \) indicate the density and color prediction of radiance field \( f_\theta \), respectively. In practice, these integrals are approximated using quadrature [37]. A neural radiance field is optimized over a set of input images and their camera poses by minimizing the mean squared error

\[
\mathcal{L}_{\text{MSE}}(\theta, R_i) = \sum_{r \in R_i} \| \hat{c}_\theta(r) - c_{\text{GT}}(r) \|^2,
\]

where \( R_i \) indicates a set of input rays and \( c_{\text{GT}} \) its GT color.

mip-NeRF: While NeRF only casts a single ray per pixel, mip-NeRF [2] instead casts a cone. The positional encoding changes from representing an infinitesimal point to an integration over a volume covered by a conical frustum. This is a more appropriate representation for scenes with varying camera distances and allows NeRF’s coarse and fine MLPs to be combined into a single multiscale MLP, thereby increasing training speed and reducing model size. We adopt the mip-NeRF representation in this work.

3.2. Patch-based Regularization

NeRF’s performance drops significantly if the number of input views is sparse. Why is this the case? Analyzing its optimization procedure, the model is only supervised from these sparse viewpoints by the reconstruction loss in (3). While it learns to reconstruct the input views perfectly, novel views may be degenerate because the model is not biased towards learning a 3D consistent solution in such a sparse input scenario (see Fig. 1). To overcome this limitation, we regularize unseen viewpoints. More specifically, we define a space of unseen but relevant viewpoints and render small patches randomly sampled from these cameras. Our key idea is that these patches can be regularized to yield smooth geometry and high-likelihood colors.

Unobserved Viewpoint Selection: To apply regularization techniques for unobserved viewpoints, we must first define the sample space of unobserved camera poses. We assume a known set of target poses \( \{ \mathbf{P}_{\text{target}} \} \), where

\[
\mathbf{P}_{\text{target}} = [\mathbf{R}_{\text{target}}^i | \mathbf{t}_{\text{target}}^i] \in SE(3).
\]

These target poses can be thought of bounding the set of poses from which we would like to render novel views at
test time. We define the space of possible camera locations as the bounding box of all given target camera locations
\[ S_t = \{ t \in \mathbb{R}^3 \mid t_{\text{min}} \leq t \leq t_{\text{max}} \} \tag{5} \]
where \( t_{\text{min}} \) and \( t_{\text{max}} \) are the elementwise minimum and maximum values of \( \{ t^i_{\text{target}} \} \), respectively.

To obtain the sample space of camera rotations, we assume that all cameras roughly focus on a central scene point. We define a common "up" axis \( p_u \) by computing the normalized mean over the up axes of all target poses. Next, we calculate a mean focus point \( p_f \) by solving a least squares problem to determine the 3D point with minimum squared distance to the optical axes of all target poses. To learn more robust representations, we add random jitter to the focal point before calculating the camera rotation matrix. We define the set of all possible camera rotations (given the sampled position \( t \)) as
\[ S_R | t = \{ R(p_u, p_f + \epsilon, t) \mid \epsilon \sim N(0, 0.125) \} \tag{6} \]
where \( R(\cdot, \cdot, \cdot) \) indicates the resulting "look-at" camera rotation matrix and \( \epsilon \) is a small jitter added to the focus point. We obtain a random camera pose by sampling a position and rotation:
\[ S_P = \{|R| \mid R \sim S_R | t, t \sim S_t \} \tag{7} \]

**Geometry Regularization:** It is well-known that real-world geometry tends to be piece-wise smooth, i.e., flat surfaces are more likely than high-frequency structures \[18\]. We incorporate this prior into our model by encouraging depth smoothness from unobserved viewpoints. Similarly to how a pixel’s color is rendered in (2), we calculate the expected depth as:
\[ \hat{d}_\theta(r) = \int_{t_n}^{t_f} T(t) \sigma_\theta(r(t)) t \, dt. \tag{8} \]
We formulate our depth smoothness loss as
\[ L_{DS}(\theta, \mathcal{R}_r) = \sum_{r \in \mathcal{R}_r} \sum_{i,j=1}^{S_{\text{patch}}-1} \left( \hat{d}_\theta(r_{ij}) - \hat{d}_\theta(r_{i+1,j}) \right)^2 + \left( \hat{d}_\theta(r_{ij}) - \hat{d}_\theta(r_{ij+1}) \right)^2, \tag{9} \]
where \( \mathcal{R}_r \) indicates a set of rays sampled from camera poses \( S_P \), \( r_{ij} \) is the ray through pixel \((i, j)\) of a patch centered at \( r \), and \( S_{\text{patch}} \) is the size of the rendered patches.

**Color Regularization:** We observe that for sparse inputs, the majority of artifacts are caused by incorrect scene geometry. However, even with correct geometry, optimizing a NeRF model can still lead to color shifts or other errors in scene appearance prediction due to the sparsity of the inputs. To avoid degenerate colors and ensure stable optimization, we also regularize color prediction. Our key idea is to estimate the likelihood of rendered patches and maximize it during optimization. To this end, we make use of readily-available unstructured 2D image datasets. Note that, while datasets of posed multi-view images are expensive to collect, collections of unstructured natural images are abundant. Our only criterion for the dataset is that it contains diverse natural images, allowing us to reuse the same flow model for any type of real-world scene we reconstruct. We train a RealNVP \[10\] normalizing flow model on patches from the JFT-300M dataset \[53\]. With this trained flow model we estimate the log-likelihoods (LL) of rendered patches and maximize them during optimization. Let
\[ \phi : [0, 1]^{S_{\text{patch}} \times S_{\text{patch}} \times 3} \rightarrow \mathbb{R}^d \tag{10} \]
be the learned bijection mapping an RGB patch of size \( S_{\text{patch}} \) to \( \mathbb{R}^d \) where \( d = S_{\text{patch}} \cdot S_{\text{patch}} \cdot 3 \). We define our color regularization loss as
\[ L_{\text{NLL}}(\theta, \mathcal{R}_r) = \sum_{r \in \mathcal{R}_r} - \log p_Z \left( \phi \left( \hat{P}_r \right) \right) \tag{11} \]
where \( \hat{P}_r = \{ \hat{e}_\theta(r_{ij}) \mid 1 \leq i, j \leq S_{\text{patch}} \} \) and \( \mathcal{R}_r \) is a set of rays sampled from \( S_P \), \( \hat{P}_r \) the predicted RGB color patch with center \( r \), and \( - \log p_Z \) the negative log-likelihood (“NLL”) with Gaussian \( p_Z \).

**Total Loss:** The total loss we optimize in each iteration is
\[ L_{\text{MSE}}(\theta, \mathcal{R}_i) + \lambda_D L_{DS}(\theta, \mathcal{R}_r) + \lambda_N L_{\text{NLL}}(\theta, \mathcal{R}_r) \tag{12} \]
where \( \mathcal{R}_i \) and \( \mathcal{R}_r \) indicate rays from input and random poses, respectively, \( \lambda_N = 10^{-6} \), and we anneal \( \lambda_D \) to 0.1.

### 3.3. Sample Space Annealing

For very sparse scenarios (e.g., 3 or 6 input views), we observe another failure mode of NeRF: divergent behavior at the start of training. This leads to high density values at ray origins. While input views are correctly reconstructed, novel views degenerate as no 3D-consistent representation is recovered. We find that annealing the sampled scene space quickly over the early iterations during optimization helps to avoid this problem. By restricting the scene sampling space to a smaller region defined for all input images, we introduce an inductive bias to explain the input images with geometric structure in the center of the scene.

Recall from (2) that \( t_n, t_f \) are the camera’s near and far plane, respectively, and let \( t_m \) be a defined center point (usually the midpoint between \( t_n \) and \( t_f \)). We define:
\[ t_n(i) = t_m + (t_n - t_m) \eta(i) \]
\[ t_f(i) = t_m + (t_f - t_m) \eta(i) \tag{13} \]
\[ \eta(i) = \min \left( \frac{i}{N_t}, p_s \right) + 1 \]
We report results on the real world multi-view datasets.

4. Experiments

- **Training Details**
  - We build our code on top of the official JAX [4] mip-NeRF codebase. We optimize with Adam [26] using an exponential learning rate decay from $2 \cdot 10^{-5}$ to $2 \cdot 10^{-7}$. We clip gradients by value at 0.1 and then by norm at 0.1. We train for 500 pixel epochs, e.g., 44K, 88K, and 132K iterations on DTU for 3/6/9 input views respectively (all fewer iterations than mip-NeRF’s default 250K steps [37]). We train all models on a TPU with 8 cores [23].

4. Experiments

**Datasets** We report results on the real world multi-view datasets DTU [21] and LLFF [36]. DTU contains images of objects placed on a table, and LLFF consists of complex forward-facing scenes. For DTU, we observe that in scenes with a white table and a black background, the model is heavily penalized for incorrect background predictions regardless of the quality of the rendered object-of-interest (see Fig. 3). To avoid this background bias, we evaluate all methods with the object masks applied to the rendered images (full image evaluations in supp. mat.). We adhere to the protocol of Yu et al. [62] and evaluate on their reported test set of 15 scenes. For LLFF, we adhere to community standards [37] and use every 8-th image as the held-out test set and select the input views evenly from the remaining images. Following previous work [62], we report results for the scenarios of 3, 6, and 9 input views.

**Metrics:** We report the mean of PSNR, structural similarity index (SSIM) [59], and the LPIPS perceptual metric [63]. To ease comparison, we also report the geometric mean of $\text{MSE} = 10^{-\text{PSNR}/10}$, $\sqrt{1 - \text{SSIM}}$, and LPIPS [2].

**Baselines:** We compare against the state-of-the-art conditional models PixelNeRF [62], Stereo Radiance Fields (SRF) [8], and MVSNeRF [6]. We re-train PixelNeRF for the 6/9 view scenarios, leading to better results, and we similarly pre-train SRF with 3/6/9 views. We pre-train all methods on the large-scale DTU dataset. The LLFF dataset has been shown to be too small for pre-training [24] and hence serves as an out-of-distribution test for conditional models. We report the conditional models on both datasets also after additional per-scene test time optimization (“ft” for “fine-tuned”). Further, we compare against mip-NeRF [2] and DietNeRF [19] which do not require pre-training, as with our approach. As no official code is available, we reimplement DietNeRF on top of the mip-NeRF codebase (achieving better results) and train both methods for 250K iterations per scene with exponential learning rate decay from $5 \cdot 10^{-4}$ to $5 \cdot 10^{-5}$.

4.1. View Synthesis from Sparse Inputs

We first compare our model to the vanilla mip-NeRF baseline, analyzing the effect of our regularizers on scene geometry, appearance and data efficiency.
input views, but outperform conditional models for the 3-view scenario. The two unconditional baselines mip-NeRF and DietNeRF do not achieve competitive results for 3-view input views, our method achieves quantitative results comparable to conditional models (SRF, PixelNeRF, MVSNeRF) despite not requiring an expensive pre-training phase, and strongly outperforms other baselines (mip-NeRF, DietNeRF) which operate in the same setting as us. For 6 and 9 input views, our model achieves the best overall quantitative results.

### Geometry Prediction:
We observe that novel view synthesis performance is directly correlated with how accurately the scene geometry is predicted: in Fig. 4 we show expected depth maps and RGB renderings for mip-NeRF and our method on the LLFF room scene. We find that for 3 input views, mip-NeRF produces low-quality renderings and poor geometry. In contrast, our method produces an acceptable novel view and a realistic scene geometry, despite the low number of inputs. When increasing the number of input images to 6 or 9, mip-NeRF’s predicted geometry improves but still contains floating artifacts. Our method generates smooth scene geometry, which is reflected in its higher-quality novel views.

### Data Efficiency:
To evaluate our gain in data efficiency, we train mip-NeRF and our method for various numbers of input views and compare their performance. We find that for sparse inputs our method requires up to 55% fewer images than mip-NeRF [2] to achieve a similar test set performance on the DTU dataset.

### 4.2. Baseline Comparison

**DTU Dataset** For 3 input views, our method achieves quantitative results comparable to the best-performing conditional models (see Tab. 1) which are pre-trained on other DTU scenes. Compared to the other methods that also do not require pre-training, we achieve the best results. For 6 and 9 input views, our approach performs best compared to all baselines. As evidenced by Fig. 6, we see that conditional models are able to predict good overall novel views, but become blurry particularly around edges and exhibit less

---

1. Results slightly differ from Tab. 1 as a smaller test set has to be used.
consistent appearance for novel views whose cameras are far from the input views. For mip-NeRF and DietNeRF, which are not pre-trained (like our method), geometry prediction and hence synthesized novel views degrade for very sparse scenarios. Even with 6 or 9 input views, the results contain floating artifacts and incorrect geometry. In contrast, our approach performs well across all scenarios, producing sharp results with more accurate scene geometry.

**LLFF Dataset:** For conditional models, the LLFF dataset serves as an out-of-distribution scenario as the models are trained on DTU. We observe that SRF and PixelNeRF appear to overfit to the training data, which leads to low quantitative results (see Tab. 2). MVSNeRF generalizes better to novel data, and all three models benefit from additional fine-tuning. For 3 input views, mip-NeRF and DietNeRF are not able to generate competitive novel views. However, with 6 or 9 input views, they outperform the best conditional models. Despite requiring fewer optimization steps than mip-NeRF and DietNeRF and no pre-training at all, our method achieves the best results across all scenar-
Table 3. Ablation Study. For very sparse scenarios, we find that scene space annealing is crucial to avoid degenerate solutions. Further, regularizing scene geometry has a bigger impact on the performance than appearance regularization. Combining all components leads to the best performance.

<table>
<thead>
<tr>
<th></th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>Average ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Scene Space Ann.</td>
<td>10.17</td>
<td>0.613</td>
<td>0.332</td>
<td>0.291</td>
</tr>
<tr>
<td>w/o Geometry Reg.</td>
<td>14.34</td>
<td>0.689</td>
<td>0.246</td>
<td>0.188</td>
</tr>
<tr>
<td>w/o Appearance Reg.</td>
<td>18.34</td>
<td>0.742</td>
<td>0.191</td>
<td>0.117</td>
</tr>
<tr>
<td>Ours</td>
<td>18.89</td>
<td>0.745</td>
<td>0.190</td>
<td>0.112</td>
</tr>
</tbody>
</table>

Table 4. Geometry Regularization. We compare different choices of geometry regularization strategies on DTU (3 input views) and find that our depth smoothness prior performs best.

<table>
<thead>
<tr>
<th></th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
<th>Average ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opacity Reg. [31]</td>
<td>11.07</td>
<td>0.617</td>
<td>0.309</td>
<td>0.268</td>
</tr>
<tr>
<td>Ray Density Entropy Reg.</td>
<td>13.93</td>
<td>0.680</td>
<td>0.254</td>
<td>0.198</td>
</tr>
<tr>
<td>Normal Smooth. Reg. [41]</td>
<td>14.22</td>
<td>0.683</td>
<td>0.251</td>
<td>0.193</td>
</tr>
<tr>
<td>Density Surface Reg.</td>
<td>14.71</td>
<td>0.687</td>
<td>0.247</td>
<td>0.184</td>
</tr>
<tr>
<td>Sparsity Reg. [17]</td>
<td>16.77</td>
<td>0.711</td>
<td>0.221</td>
<td>0.145</td>
</tr>
<tr>
<td>Depth Smooth. Reg. (Ours)</td>
<td>18.89</td>
<td>0.745</td>
<td>0.190</td>
<td>0.112</td>
</tr>
</tbody>
</table>

Figure 8. Qualitative Ablation Study. Scene space annealing (-Anneal.) and geometry regularization (-Geo. Reg.) are crucial to avoid floating artifacts while the appearance regularizer (-App. Reg.) ensures stable color prediction.

Figure 9. Failure Analysis. As we do not attempt to hallucinate geometric details in this work, our model may lead to blurry predictions in unobserved regions with areas of fine geometry (9b). We identify incorporating uncertainty prediction or generative components into our model as interesting future work.

5. Conclusion

We have presented RegNeRF, a novel approach for optimizing Neural Radiance Fields (NeRF) in data-limited regimes. Our key insight is that for sparse input scenarios, NeRF’s performance drops significantly due to incorrectly optimized scene geometry and divergent behavior at the start of optimization. To overcome this limitation, we propose techniques to regularize the geometry and appearance of rendered patches from unseen viewpoints. In combination with a novel sample-space annealing strategy, our method is able to learn 3D-consistent representations from which high-quality novel views can be synthesized. Our experimental evaluation shows that our model outperforms not only methods that, similar to us, only optimize over a single scene, but in many cases also conditional models that are extensively pre-trained on large scale multi-view datasets.

Limitations and Future Work: In this work, we do not attempt to hallucinate geometric details. As a result, our model may lead to blurry predictions in unobserved areas with fine geometric structures (see Fig. 9). We identify incorporating uncertainty prediction mechanisms [50] or generative components [5, 15, 39, 49] as promising future work.

Potential Negative Impact: We identify misleading content creation, reconstruction of sensitive data, and high energy consumption of deep learning-based systems as potential negative impacts for society and environment.
References


