Solving Inverse Problems with NerfGANs

Giannis Daras[†*] giannisdaras@utexas.edu
Wen-Sheng Chu[‡] wschu@google.com
Abhishek Kumar[‡] abhishk@google.com
Dmitry Lagun[‡] dlagun@google.com
Alexandros G. Dimakis[†] dimakis@austin.utexas.edu

[†]University of Texas at Austin [‡]Google Research

Abstract

We introduce a novel framework for solving inverse problems using NeRF-style generative models. We are interested in the problem of 3-D scene reconstruction given a single 2-D image and known camera parameters. We show that naively optimizing the latent space leads to artifacts and poor novel view rendering. We attribute this problem to volume obstructions that are clear in the 3-D geometry and become visible in the renderings of novel views. We propose a novel radiance field regularization method to obtain better 3-D surfaces and improved novel views given single view observations. Our method naturally extends to general inverse problems including inpainting where one observes only partially a single view. We experimentally evaluate our method, achieving visual improvements and performance boosts over the baselines in a wide range of tasks. Our method achieves $30-40\%$ *MSE reduction and* $15-25\%$ *reduction in LPIPS loss compared to the previous state of the art.*

1. Introduction

State-of-the-art generative models have become capable of generating extremely high-fidelity images of the 2-D world [2, 6, 9, 17, 18]. Despite their wide success, current generative models often fail to capture the 3-D structure of the represented scenes and offer limited control over the geometrical properties of the generated images.

NerfGANs [3, 28, 33, 34] are a new family of generative models that directly model the 3-D space by leveraging the success of Neural Radiance Fields (NeRFs) [30]. NerfGANs generate 3-D structure in the form of a Radiance Field and then output 2-D images by rendering the field from different camera views. They are not yet as competitive as state-of-the-art 2-D models for image generation [2, 5, 6, 16–18]. However, modeling directly the 3-D space offers many new possibilities, beyond generating photo-realistic images, that have yet to be explored.

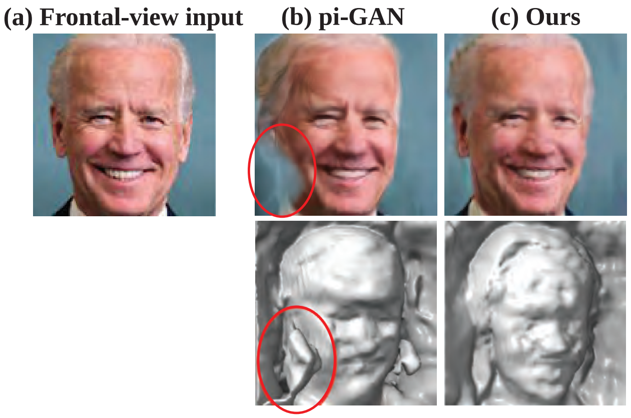


Figure 1. **NerfGAN inversion:** Given a single frontal view image, we would like to generate novel angle views and the underlying 3-D geometry. As shown, latent space optimization as proposed in pi-GAN [3] creates obstructions (*stones*) that can produce artifacts in novel views. Middle column illustrates the artifacts in the rendered image (top row) and the 3-D geometry (bottom row). Our inversion algorithm removes these issues by optimizing over both the 2-D view and the 3-D shape of the radiance field. Middle Column: Reconstructed 3-D geometry and novel view using pi-GAN direct latent space optimization. Right column: Generated 3-D structure and novel view using the proposed reconstruction algorithm. We emphasize that our algorithm also uses the same generator (pi-GAN), but recovers a better latent vector compared to direct latent-space optimization. This leads to a better 3-D geometry reconstruction and better 2-D novel views.

We study how we can use pre-trained NerfGANs as a prior to solve inverse problems. We start with the problem of single-view inversion: given a single 2-D image (*e.g.*, a photograph of a person) and known camera parameters we want to create novel views and reconstruct the 3-D geometry leveraging a pre-trained NerfGAN, e.g. the state of the

[*] This work was done during an internship at Google Research.

art pi-GAN [3]. We denote with $G(z, p)$ a 3-D NerfGAN that takes a latent vector $z \in \mathbb{R}^k$ and a 3-D space position p in $\mathbb{R}^3$ and outputs a color and a density value. For a given latent vector z, the NerfGAN scene can be rendered as a 2-D image for any camera position. Formally, for a given camera position c the produced 2-D image is denoted by $\mathcal{R}(c, G(z, \cdot))$, where $\mathcal{R}$ is the rendering operator. Given a single target image x^* and known camera parameters c, NerfGAN inversion is the problem of finding the optimal latent code z^* that creates a 3-D scene that renders to x^*.

A natural method for NerfGAN inversion (used in pi-GAN [3]) is to optimize the latent vector to match the observed target image [1]:

$$\min_z ||\mathcal{R}(c, G(z, \cdot)) - x^*||. \tag{1}$$

We unveil a major limitation of this vanilla method. The produced neural radiance field renders correctly to the given target image x^*, but even small rotations produce significant artifacts in novel views. In Figure 1, a single front view was given and a radiance field was reconstructed using latent space optimization. When rendering a novel side view, a blue shade artifact appears in the bottom left as annotated. We discover that neural fields created by solving the inverse problem often have three dimensional obstructions (we call them *stones*) that are invisible in the frontal view but create visual artifacts in the renderings of novel views.

Similar issues were raised in the pi-GAN [3] paper which observed hollow-face artifacts in generated images but not obstructions. The authors of [3] identify this as an open problem: *"In certain cases, pi-GAN can generate a radiance field that creates viable images when rendered from each direction but nonetheless fails to conform to the 3-D shape that we would expect. Further investigation may reveal insights that could resolve such ambiguities."*. We observe that solving inverse problems using direct latent space optimization, as in (1), frequently produces unrealistic 3-D obstructions that also lead to visual artifacts when rendered from novel views. To account for this problem, the authors of [3] proposed to penalize divergence between the SIREN [42] frequencies and phase shifts and their average values. We show that this approach, even though it produces smooth geometries, significantly reduces the range of the generator, leading to blurred reconstructions (see Figure 2). Instead, our method solves for

$$\min_z ||\mathcal{R}(c, G(z, \cdot)) - x^*|| + \lambda S_{3-D}(G(z, \cdot)), \tag{2}$$

where S_{3-D} imposes a high penalty for latent vectors z that create unnatural geometries. As we explain subsequently, we achieve this 3-D regularization by creating a convex combination of distances to reference geometries.

Extension to General Inverse problems: Beyond reconstructing the 3-D structure of a scene using a single view x^*, we extend our method to general inverse problems. For example, our method can be directly applied when there are missing pixels in the view (inpainting), a blurred observed view, or observations of the single view with random projections or Fourier projections arising in medical imaging and compressed sensing [1, 14].

Consider the general setting where the unknown image is x^* and we observe

$$y = \mathcal{A}[x^*] + \eta, \tag{3}$$

where $\mathcal{A}$ is the forward operator that somehow corrupts the image (e.g. pixel removal, blurring, or projections) and η is a noise vector.

Direct latent optimization in this case would correspond to finding the latent vector z that best explains the measurements, as expressed below:

$$\min_z ||\mathcal{A}[\mathcal{R}(c, G(z, \cdot)] - y||. \tag{4}$$

This natural baseline creates 3-D obstructions (thus artifacts) as in single view inversion. We propose to solve the optimization problem with the same 3-D regularization:

$$\min_z ||\mathcal{A}[\mathcal{R}(c, G(z, \cdot))] - y|| + \lambda S_{\text{3D}}(G(z, \cdot)). \tag{5}$$

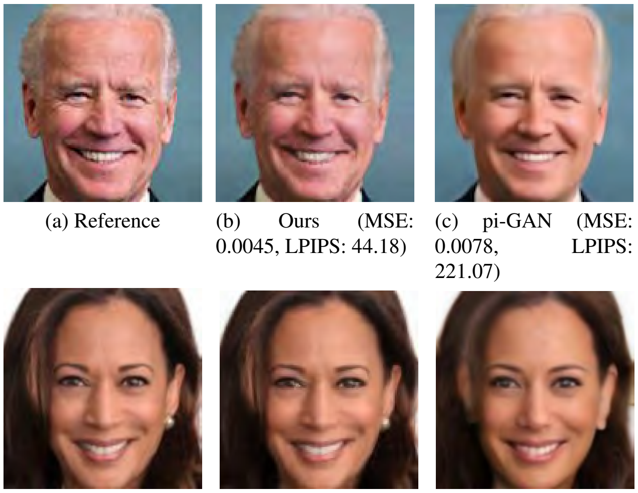
Our method can be applied to linear inverse problems (where $\mathcal{A}$ is a matrix) or even non-linear such as phase retrieval [13] as long as the $\mathcal{A}$ is differentiable almost everywhere.

The key issue is to devise a 3-D regularizer, S_{3-D}, that does not lead to measurements overfitting, i.e. big real error $||\mathcal{R}(c, G(z, \cdot)) - x||$ with small measurements error (first term of (5)). We use a set of reference geometries and an annealing mechanism in gradient descent to lock-in on better fitting geometries as we subsequently explain.

To the best of our knowledge, this paper shows for the first time the challenges involved in using NerfGANs for solving inverse problems and proposes a principled framework to address them.

Our Contributions:

- We identify a key limitation on reconstructing 3-D geometries and novel views from a single reference image. We demonstrate that existing algorithms are not sufficient, because either: i) they overfit to the measurements and produce artifacts to novel views (Figure 1), or ii) they significantly limit the expressive power of the generator (Figure 2).
- We trace the problem back to unrealistic 3-D geometries that need to be avoided in the course of the optimization. We propose a principled framework for regularizing the radiance field itself, without limiting the range of the generator. Our framework drives the network to generate realistic geometries by penalizing distance from a set of realistic geometries under a novel 3-D loss.



(a) Reference

(b) Ours (MSE: 0.0045, LPIPS: 44.18)

(c) pi-GAN (MSE: 0.0078, LPIPS: 221.07)

(d) Reference

(e) Ours (MSE: 0.0029, LPIPS: 28.11)

(f) pi-GAN (MSE: 0.0060, LPIPS: 28.90)

Figure 2. **Regularization impact on frontal-view reconstruction:** Given the frontal-view images (a), (d), we show reconstructions with our method in (b), (e) and pi-GAN in (e), (f). Our method leverages a more expressive 3-D consistency loss, yielding better MSE and LPIPS results than pi-GAN that uses distance from average frequencies. Note that the pi-GAN images are directly taken from the original paper [3].

- We show how to obtain a candidates set of realistic geometries using CLIP [37] and a pre-trained NerfGAN.
- We experimentally evaluate our method achieving visual improvements and performance boosts over the baselines in a wide range of tasks. Our method achieves $30-40\%$ MSE and $15-25\%$ reduction in LPIPS [45] compared to the previous state of the art.

2. Related Work

NerfGANs NerfGANs are NeRF [30] style generators that are trained adversarially [11]. Recently, there has been tremendous progress in making NeRFs more efficient [10, 23, 24, 31, 39, 40] and extending them beyond static scenes [7,21,22,44]. As larger and more realistic 3-D Neural radiance field generators are made possible, we expect that solving inverse problems with them to become increasingly relevant for numerous applications.

In this paper, we investigate the challenges in solving inverse problems with NerfGANs that are stemming from unrealistic 3-D structures. Our work can in principle be used with any generator from the growing family of NerfGANs [3, 28, 33, 34, 41]. For the purposes of this paper, we use the pi-GAN [3] generator since, to the best of our knowledge, there is no other NerfGAN paper that discusses inverse problems. A concurrent work [36] proposes architectural changes to the generator to obtain smoother geometries. Potentially, our method could yield even better results with this new generator, but this remains as future work.

A new line of research [12,46] uses a small NerfGAN to produce a coarse image representation at low-resolution and then a 2-D network to transform it to photorealistic, higher-resolution image. These approaches side-step the computational cost of training a traditional NerfGAN at high resolutions and present promising results in solving inverse problems. These techniques are orthogonal to our method for obtaining better 3-D geometries from single image-views and can be combined as a post-processing step to improve the quality of our generated views.

Inverse Problems Our method relies on significant prior work in unsupervised methods for inverse problems using pre-trained generative models. We note that our method leverages a pre-trained generator, hence avoiding the computational overhead of training an end-to-end supervised network to go from observations to 3-D geometry [32,47]. There are various benefits of using unsupervised methods for solving inverse problems including robustness to data structure shifts and unknown variations in the corruption process [35]. Compared to classical approaches for 3-D reconstruction from a single view, like the seminal 3DMM algorithm [8], our method is more general since it can be applied for any corrupted image, as long as the forward operator that corrupts the image is known and differentiable.

Inversion algorithms are building on latent space optimization as proposed in the CSGM [1] algorithm and we extend it using Perceptual Loss [4, 45] and Geodesic regularization for frequencies, inspired by StyleGAN2 and PULSE [17,18,29]. Our central innovation is the 3-D regularization term which is directly applied in the neural radiance field and is, to the best of knowledge, entirely novel.

3. Method

Motivation Figure 2(c), 2(f), shows inversions that appear in the pi-GAN paper. The first thing to notice is that the reconstructed images are sufficiently different compared to the input images. The poor frontal view reconstruction could be attributed to the limited range of the pi-GAN generator. Figure 2(b), 2(e) shows that this is not true; using the exact same generator we are able to get much better reconstructions compared to pi-GAN as shown visually and by the MSE and Perceptual losses.

The pi-GAN reconstructions look like smoothed versions of the reference images. This superficial smoothness comes from the regularization term that is used in the pi-GAN paper. Specifically, the authors of [3] penalize divergence between the SIREN [42] frequencies and phase shifts and their average values. Figure 1(b) shows a novel view obtained with pi-GAN without this regularization. The face is much closer to the given input. There is a caveat though; the novel view shows artifacts (circled with red color). To

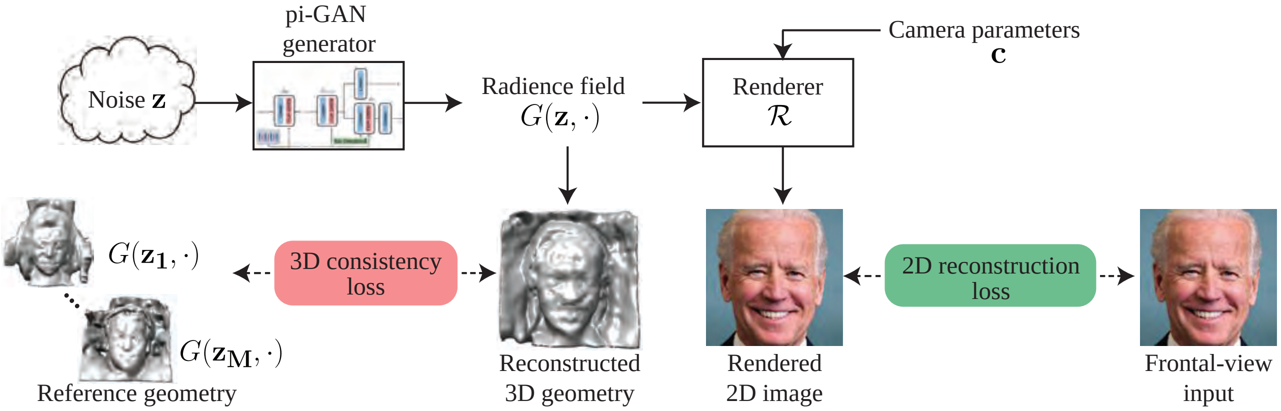


Figure 3. **An overview of the proposed method:** We jointly optimize over 2-D re-construction of the input image and 3-D consistency of the radiance field. Given a randomly sampled noise vector $\mathbf{z}$ from a Gaussian distribution, we obtain radiance field $G(\mathbf{z}, \cdot)$ via the pi-GAN generator G, and rendered 2-D images using a conventional volumetric renderer $\mathcal{R}$ for given camera parameters $\mathbf{c}$. Our loss minimizes at the same time the distance of the rendered image to the given view and a 3-D consistency term that penalizes unrealistic geometries.

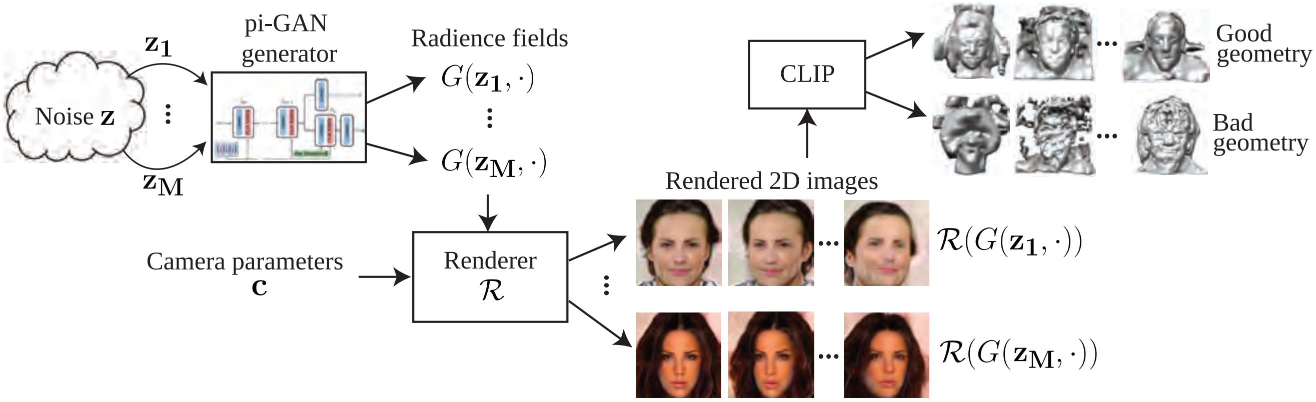


Figure 4. **Candidate selection for reference geometries:** We use CLIP [37], a pre-trained free-text image classifier, to identify whether a radiance field is realistic and to form a set of reference geometries. The idea is that for bad geometries, certain attributes (such as CLIP's belief on how realistic an image is) should change, even for small changes in the camera parameters. Given rendered 2-D images from randomly sampled noise vectors $\{\mathbf{z_1}, \ldots, \mathbf{z_M}\}$, we observe differences in CLIP's logits to identify realistic geometries.

understand where these artifacts are coming from, in Figure 1(b), second row, we visualize the 3-D geometries that correspond to the obtained frequencies and phase shifts. We see that there are three dimensional obstructions (we call them stones) that are invisible in the frontal view. This experiment shows an important trade-off, i.e. matching better the measurements but with poor generalization vs inferior reconstruction of the image but smoother novel views.

A natural question is why the optimization trajectory finds such geometries that match almost perfectly the given 2-D image but produce artifacts in novel views. The short answer is that these unrealistic geometries are fairly common, even in pure image generation with Gaussian inputs. We perform an experiment to validate this.

For this experiment, we are using CLIP [37], a free text image classifier. CLIP can take as input an image and a text, and it produces text and image embeddings that have big inner product if the text describes well what is portrayed in the image. The idea of the experiment is that for bad geometries, certain attributes of the object change as you slightly move the camera, so that should be reflected in the CLIP logits. CLIP has been used before as a cherry-picking tool for image generation [38], but we show a natural extension for identifying realistic 3-D structure.

We first collect latent vectors, sampled from a Gaussian distribution, and form the set $\mathcal{S} = \{z_1, ..., z_M\}$. For each of the elements in the set, we render images from camera positions $c_1, ..., c_K$, and form the set $\mathcal{S}_z = \{\mathcal{R}(c_1, G(z, \cdot)), ..., \mathcal{R}(c_K, G(z, \cdot))\}$. Each of these sets is assigned a cost given by the maximum difference of CLIP logits between the images in the set and the text prompt T =`"A non-corrupted, non-noisy image of a person."`. We use this prompt because we expect that for side views, bad geometries will show more artifacts due to 3-D obstructions (stones) outside of the face.

The cost w_z of $\mathcal{S}_z$ is given by:

$$w_z = \max_{s_1, s_2 \in \mathcal{S}_z} |\text{CLIP}(s_1, T) - \text{CLIP}(s_2, T)| . \tag{6}$$

After assigning the costs, the set of latents that correspond to unrealistic geometries is given by:

$$\text{Bad}_{z,\epsilon} = \{z \in S | w_z \geq \epsilon\}. \tag{7}$$

The fraction of unrealistic geometries, *i.e.*, $\frac{|\text{Bad}_{z,\epsilon}|}{|S|}$, is a measure of how often do bad geometries occur in the range of a NerfGAN. We found that such geometries are common in the range of pi-GAN [3] – approximately 40% of geometries are classified as "bad" by CLIP. The top-right portion of Figure 4 illustrates geometries classified as bad by CLIP. More examples of bad geometries, their corresponding 2-D views and more details are provide in the Appendix.

Obtaining realistic reference geometries Similarly, one can use CLIP to identify realistic geometries. For reasons that will be explained in Section 3, we are interested in collecting a set of realistic geometries that render to visually plausible 2-D images. For each Gaussian sampled z, we assign two costs: i) the *consistency cost* we defined in (6) and ii) the plausibility cost:

$$c_z = \min_{s \in \mathcal{S}_z} \text{CLIP}(s, T). \tag{8}$$

We finally collect the set:

$$\text{Good}_{z,\epsilon_1,\epsilon_2} = \{z \in S | w_z \leq \epsilon_1, c_z \leq \epsilon_2\}. \tag{9}$$

The whole procedure is illustrated in Figure 4, with examples of geometries and renderings.

Optimization problem Without regularization, the optimization trajectory often reaches points of minimum loss but with poor generalization to novel views. A natural idea is to regularize towards realistic geometries. We will use the notation $G_\sigma(z, \cdot)$ to denote the density part of the radiance field and $P \in \mathbb{R}^{k\times 3}$ to denote its discrete representation.

A straightforward way to constrain the 3-D shape is to force it to be close to a 3-D geometry that is known to be good. We collect a set of latent vectors S that correspond to realistic geometries $\{G_\sigma(z, \cdot) | z \in S\}$ and try to regularize the inferred geometry towards the *most suitable* geometry in our realistic set. This can be made more concrete in the form of the following optimization problem:

$$\begin{aligned} &\min_{z \in \mathbb{R}^d} \frac{1}{2} ||\mathcal{R}(c, G_\sigma(z, \cdot)) - x^*||^2 \\ &\text{s.t.} \min_{z_{\text{ref}} \in \mathcal{S}} \mathcal{L}\left(G_\sigma(z, P), G_\sigma(z_{\text{ref}}, P)\right) \leq \epsilon. \end{aligned} \tag{10}$$

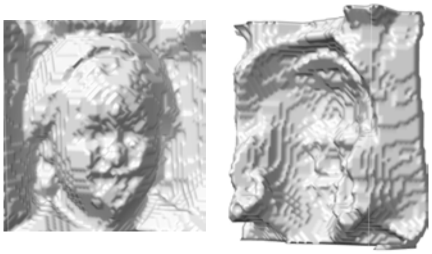


Figure 5. **Reference geometry:** Illustration of a reference geometry (front and back) extracted with the Marching Cubes algorithm used in our set of reference geometries $\{G(z_i, \cdot)\}$.

To solve this problem with Gradient Descent, we write down the penalized version and solve for:

$$\begin{aligned} \min_{z \in \mathbb{R}^d, z_{\text{ref}} \in \mathcal{S}} &\frac{1}{2} ||\mathcal{R}(c, G(z, \cdot)) - x^*||^2 \\ &+ \lambda \cdot \mathcal{L}\left(G_\sigma(z, P), G_\sigma(z_{\text{ref}}, P)\right). \end{aligned} \tag{11}$$

There are two issues with the formulation of (11). First, it is a min-min problem where the inner minimum is over a discrete set. Gradient Descent (GD) is likely to get stuck to a local minima: the reference geometry that happens to be closer to the initialization is likely to be the active constraint of (10), even though it might not be the one that minimizes the total objective. We observe this problem experimentally, for more see Figure 8. The second issue is that the two terms in our loss function might be incompatible. For example, if the reference set $\mathcal{S}$ is small, there might be no geometry that renders to the measurements.

We propose a relaxation of the objective of (11), where the $\min$ is replaced with a soft-operator that allows all reference geometries to contribute to the gradients based on how close they are, under $\mathcal{L}$, to the current radiance field. We consider the following optimization problem:

$$\begin{aligned} &\min_z \frac{1}{2} ||\mathcal{R}(c, G(z, \cdot)) - x^*||^2 + \\ &+ \lambda \sum_{z_{\text{ref}} \in S} \left(\frac{e^{-\delta \mathcal{L}_{z, z_{\text{ref}}}}}{\sum_{z'_{\text{ref}} \in S} e^{-\delta \mathcal{L}_{z, z_{\text{ref}'}}}} \right) \mathcal{L}_{z, z_{\text{ref}}}. \end{aligned} \tag{12}$$

This can be interpreted as finding a nonnegative weighting of the losses $\mathcal{L}_{z, z_{\text{ref}}}$ that solves $\min_{w \geq 0} \sum_i w_i \mathcal{L}_{z, z_{\text{ref}_i}}$, such that $D(u, w) \leq \gamma(\delta)$ for some divergence measure D, uniform distribution u, and a radius parameter γ. The softmax weighting emerges when D is taken to be KL-divergence [20]. Observe that for $\delta \to \infty$ (corresponding to a large enough γ), the optimization problems of Equations (11), (12) have the same solution. However, this formulation is more powerful since it allows blending of the reference geometries in case we cannot match the measurements otherwise. If the distribution of the contributions of each loss is close to a Dirac, then we have 3-D consistency.

In all our experiments, we use GD to solve the optimization problem of (12). We gradually anneal the temperature parameter δ during the course of the optimization to encourage convergence to a single reference geometry. We choose the z that minimizes the total loss. If the loss curve is flattened, we prefer the z that corresponds to lower temperature because it correlates with better 3-D consistency.

Choice of 3-D loss function Figure 1 shows that when solving inverse problems without 3-D regularization, the reconstructed 3-D geometries have what we call "stones", i.e. regions of high density outside of the face that create artifacts when we render the field with different camera parameters. To regularize for the stones, for each of the reference radiance fields, we obtain a face surface mask on the 3-D space and constrain the reconstructed voxel grid to match the reference outside of this 3-D mask. The intuition is that voxels outside of the facial surface should have low values (as in the reference geometries). Matching only these voxels gives our method enough flexibility to adjust the facial 3-D structure to match the measurements without having high density clusters (stones) outside of the face.

Formally, let $\mathcal{M}(p) : \mathbb{R}^3 \to \{0, 1\}$ be the operator that gives the facial mask. We define our loss function as:

$$\mathcal{L}_{z,z_{\text{ref}}} = ||\,(G_\sigma(z,P) - G_\sigma(z_{\text{ref}},P)) \odot (1-\mathcal{M}(P))\,||_F \tag{13}$$

where $||\cdot||_F$ denotes the Frobenius norm.

In all our experiments, we use the vertices of the generated polygons of the Marching Cubes algorithm [27] to get our facial mask. Figure 5 shows an example geometry and the corresponding mask for a latent in our reference set.

4. Experimental Results

In all our experiments, we use the pi-GAN [3] generator, pre-trained on faces from CelebA [26]. At each optimization step, our method generates a voxel grid using the current latent z. One natural question is how coarse the voxel grid representation should be in order for the regularization to be effective. We perform an extensive analysis on the Appendix. In short, our finding is that a voxel grid at resolution $32 \times 32 \times 32$ works well across all the considered tasks. We use a reference set of 64 geometries, automatically selected with CLIP and manually inspected to ensure it has the desired quality and diversity. Our method runs in approximately 3 minutes per image on a workstation of 4 V100 GPUs. For more experimental details, including all our hyperparameters, we refer the reader to the Appendix.

4.1. Inversion

Our first comparison is with the regularization proposed in the pi-GAN paper (ℓ_2 penalty for divergence from the average frequencies and phase shifts). For a fair comparison, we use the reconstructions directly from the pi-GAN paper. Figure 2 shows the results for the frontal view. Our method significantly outperforms pi-GAN on the frontal view – we observe 42% reduction in MSE and 80% reduction in LPIPS for the first image. As shown in the pi-GAN paper, their method indeed gives novel views without artifacts, but since the distance to the ground truth is very large, we do not consider it further in our experiments in the main paper. We refer the interested reader to the Appendix for a more extensive quantitative comparison with this method.

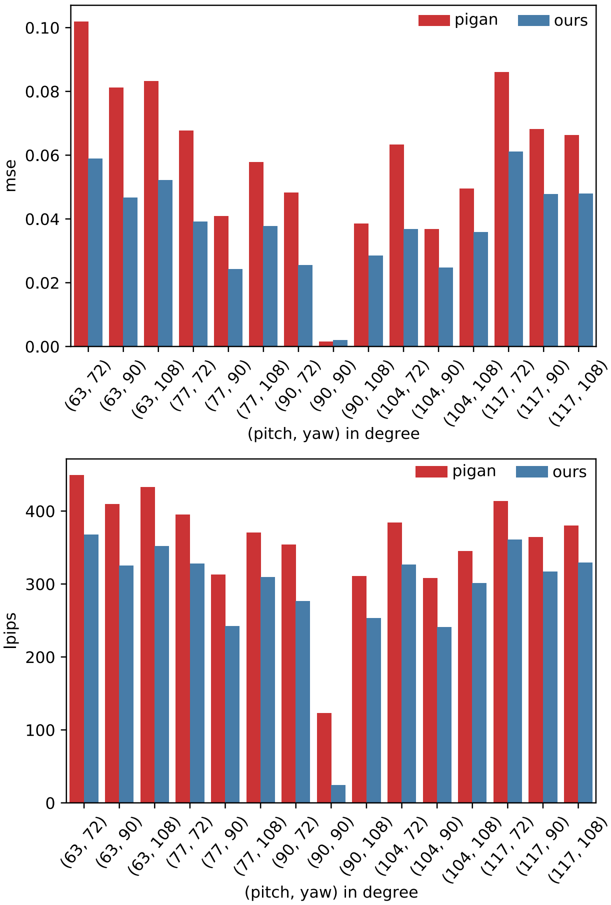


Figure 6. **MSE vs LPIPS:** Comparisons between pi-GAN [3] and our method on different camera angles on MSE and LPIPS metrics. MSE (top) indicates 2-D pixel reconstruction, where our method shows comparable loss in the given frontal view at (90,90), and noticeably lower loss in all other novel views. LPIPS (bottom) describes the perceptual loss, suggesting that our method has consistently less perceptual differences than pi-GAN.

In the rest of the paper, we compare with the unregularized baseline that follows the CSGM [1] approach to match the measurements, i.e. solves the problem defined in (1). We run our method and the CSGM baseline on the images from the pi-GAN paper and for some images in the range of

NerfGAN. For all our experiments, input is the frontal view (one can run our method for any view, as long as the camera parameters are known).

Figure 7 illustrates novel views rendered with our method and the baseline (for video results, see the Supplementary). For the images in the range, we have ground truth novel views that we also show in the Figure (obtained by inputting the corresponding viewing angle into the NerfGAN generator). As shown, our method produces less artifacts (e.g. the blurry blobs shown in Row (b) is removed, together with the unrealistic artifacts in the eyes in Row (e). For the synthetic image, our method (Row (d)) almost matches exactly the ground truth novel views presented in Row (c). For the real image, we do not have ground truth novel views, and we do see some blurriness (even in our method) that can be attributed to the limited expressivity of the generator.

In Figure 6 we show a quantitative comparison between our method and standard pi-GAN inversion for different views, measured in a set of 100 synthetic images for which we have ground truth for all views. Our method achieves $30-40\%$ MSE reduction and $15-25\%$ reduction in LPIPS compared to latent space optimization without the 3-D regularization in the novel views. In the frontal view, the methods perform on par, suggesting overfitting of the baseline to the measurements.

4.2. Inpainting

We now consider the problem of inpainting where one does not observe a full view x^* but rather a known subset of pixels is missing.

In Figure 9 we plot the Mean Squared Error (MSE) versus the ratio of observed pixels for a novel view for the task of the randomized inpainting, i.e. a fraction of the pixels, selected at random, is missing each time. As shown, latent-space optimization baseline has an increasing MSE as the number of observations increases. This happens because the baseline is overfitting to the frontal view and fails to reconstruct the novel views correctly. In contrast, our method consistently produces lower MSE for all the novel views.

Our MSE is almost constant in the considered range since our method gets optimal reconstruction in the considered range. For the frontal view, we observe that the MSE drops for both methods in the same way as the number of measurements increases. Our method gets 0.022 MSE for 10% inpainted pixels and 0.001 for 100% observed, while the baseline performs on par: 0.023 observed for 10% and 0.001 for 100% observed.

4.3. Ablation Studies

Temperature annealing In the method section, we motivated the need to converge as much as possible to a single geometry. However, in the early stages of the optimization, we want to allow all reference geometries to contribute to the gradients since otherwise gradient descent might get trapped in a local minima. To achieve this, we do temperature annealing: in the early steps of gradient descent we have a high temperature (i.e. we allow our distribution over the references to look like uniform) and as we progress we decrease the temperature (increase δ) to converge to a single radiance field. In all our experiments, we do step annealing, increasing δ by 50 every 100 optimization steps. Figure 8 shows how δ, the entropy of the distribution over the references and its maximum weight evolve over time, with and without annealing. As shown, without annealing, the model converges to a single geometry immediately (maximum weight of the blue line becomes 1 after one step), and won't change throughout the optimization. This leads to suboptimal results, as shown in the Table of Figure 8. With annealing, in the early stages, the distribution has high entropy (close to uniform) and as time progresses we converge to a single radiance field. Our annealing scheme allows the modle to discover the truly best geometry to match the measurements.

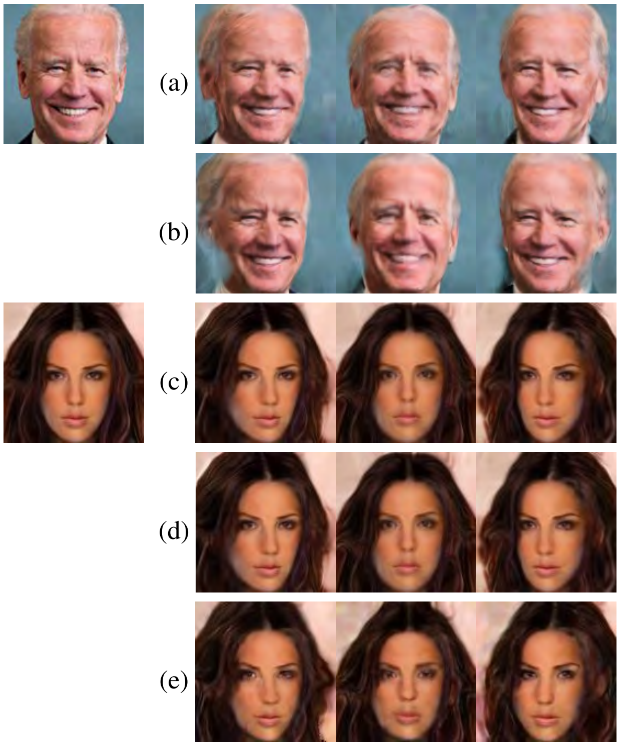
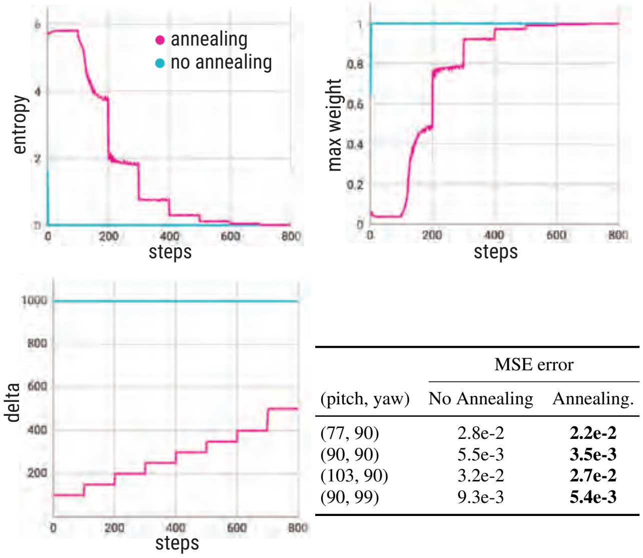


Figure 7. **Novel views**: Rows (a)(d) show novel views of our method, and rows (b)(e) show the baseline. The input image, frontal view, is shown in the left column. The second input image is synthetic, so we also show ground truth novel views on Row (c). Our method removes artifacts that appear in the baseline, such as blurry blobs in Row (b) and unrealistic eyes in Row (e).



	MSE error	
(pitch, yaw)	No Annealing	Annealing.
(77, 90)	2.8e-2	**2.2e-2**
(90, 90)	5.5e-3	**3.5e-3**
(103, 90)	3.2e-2	**2.7e-2**
(90, 99)	9.3e-3	**5.4e-3**

Figure 8. **Temperature Annealing:** We ablate the annealing component of our method. Without annealing, the optimization locks immediately in one geometry (maximum weight of the blue line becomes 1 after one step) and tries to minimize the measurements error. However, with annealing of δ, we still converge to a single distribution (max weight of the purple line becomes 1 eventually), but the model finds a geometry that gives better MSE scores consistently in all views, as illustrated in the Table.

Additional Experiments In the Appendix, we present several additional experiments on real data as well as inverse problems including compressed sensing, super-resolution and inpainting. Additionally, inspired by the work of [46], we show that we can project our renderings for novel views back to the range of a 2-D generator using Intermediate Layer Optimization [4] and achieve StyleGAN quality results for rendered views. We note that this is not part of our central innovation, but one can use it to get finer details in the rendered images. Our method can also generalize to objects beyond faces. For a relevant discussion and results, please refer to the Appendix.

5. Limitations

The success of our method relies on the quality of the collected set of voxel grids. If the set is not diverse or if it contains non-realistic 3-D structures, then our method will fail for some instances. Also, since all the geometries are in the range of the GAN, any dataset biases will be reflected in our reconstructions. We note that our method might only introduce biases in the 3-D structure of a face since we are not regularizing for color. We plan to release our dataset of reference geometries for further inspection and we refer the readers to [4, 15, 29] for a discussion on how GANs and different inversion algorithms can amplify dataset biases.

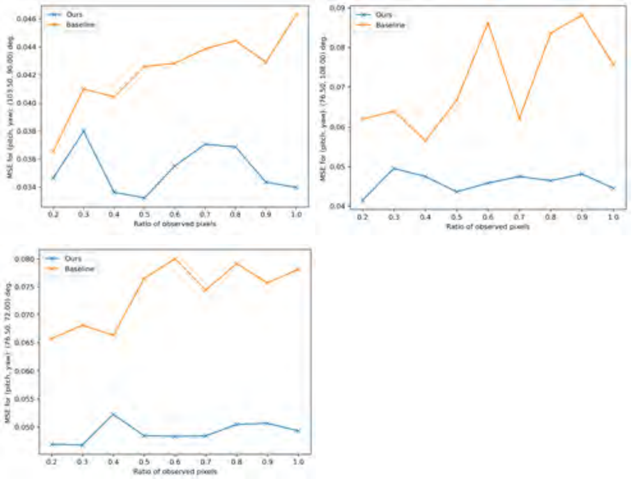


Figure 9. Inpainting plots for different views, as we change the number of observed pixels in the frontal view. Our method consistently outperforms the baseline. As the number of observed pixels increases, the baseline method overfits more and more to the measurements (frontal view) and performs worse in novel views, indicating that the reconstructed geometry gets worse.

Another concern is that the relaxation of the optimization problem allows for solutions that are blendings of the 3-D structures of the collected set. A blending of two realistic facial geometries might have artifacts. We account for this by annealing the temperature, effectively encouraging the optimization to converge to a single 3-D structure. We do not have any principled way of annealing the temperature and we leave this as a future direction.

Finally, our method regularizes only for a realistic 3-D structure, but does not add any regularization on the colors. Non-smoothness in the 3-D color signal might give undesired transitions between nearby views. We do not observe any such behavior with the pi-GAN generator, but it could happen with other models from the NerfGAN family.

6. Conclusions

In this paper, we propose a novel method for solving inverse problems for 3-D neural radiance fields given a single 2-D view. The proposed framework naturally generalizes even if a partial or corrupted 2-D view is available and extends existing work on unsupervised methods for inverse problems. The central innovation is regularizing the neural radiance field using reference geometries and we expect that better references will improve our method. We expect our method to be applicable to generative neural radiance field methods and improve in performance as more powerful pre-trained NerfGANs become available.

Supplementary Material
Solving Inverse Problems with NerfGANs

Giannis Daras
giannisdaras@google.com

Wen-Sheng Chu
wschu@google.com

Abhishek Kumar
abhishk@google.com

Dmitry Lagun
dlagun@google.com

Alexandros G. Dimakis
dimakis@austin.utexas.edu

7. Additional Experiments

This section provides additional experiments that could not fit in the paper due to the page limit. We first demonstrate that our method can be used for other objects, beyond human faces. The next step is to show how we can improve the quality of the renderings of our method by leveraging a 2-D powerful generator. We then show that our method is effective in solving general inverse problems, by obtaining realistic 3-D reconstructions given various types of image corruptions, such as compressed sensing, downsampling (super-resolution) and box inpainting. Finally, we present ablation studies that justify the design choices for our method.

7.1. Results of our method for cats

As discussed in the main paper, our method can work out-of-the-box for other objects, other than human faces. In this section, we provide evidence that supports this argument. Specifically, we use the pi-GAN generator pretrained on the cats and we show that we render realistic novel views given a single image.

One question is how we will obtain the reference geometries for the cats. We could follow the same procedure as the one mentioned in the paper, i.e. use CLIP [37] to filter out bad geometries (of cats). However, by observing the generated geometries for random latents, we see that the cats pi-GAN generator very rarely outputs unrealistic 3-D structures in the unconditional generation task. Hence, we can sidestep the CLIP procedure and use random references instead. We use 32 reference geometries for cats that correspond to latents sampled from the Gaussian distribution.

The results are shown in Figure 10. As seen, our method gives realistic renderings of novel views from a smooth 3-D geometry.

7.2. Improving the reconstruction quality with Intermediate Layer Optimization

In the main paper, we showed that our method removes 3-D obfuscations and hence gives more realistic novel views compared to the method described in pi-GAN [3]. However, there are three issues that impact the quality of the novel views, even for our method:

1. The pi-GAN generator is trained with low-resolution images. As a result, it cannot capture the fine details of real images.
2. The pi-GAN generator (for faces) is trained to model only small camera movements. This is due to the fact that the CelebA [25] dataset has mostly frontal view images. Hence, when we move the camera in positions outside of the training distribution, the image quality deteriorates.
3. The pi-GAN generator is not as powerful as state-of-the-art 2-D generators like StyleGAN [16–18] and hence certain facial attributes cannot be modelled.

These issues affect the image quality of our reconstructions. We can mitigate these issues using a 2-stage approach. At the very first stage, we use our method (described in the main paper) to obtain a coarse reconstruction and a smooth 3-D geometry. We use the radiance field from the solution of our optimization problem to render novel views. These views are expected to have blurriness and certain artifacts because of the pi-GAN issues described above. Then, we can use a powerful 2-D generator, *i.e.*, StyleGAN, as a prior to correct these artifacts separately for every novel view.

One problem with projecting back to the range of StyleGAN is that the generator is not expressive enough to model all the images. Hence we might get a photorealistic person that is not looking very close to our input (frontal view). Intermediate Layer Optimization (ILO) [4] solves this prob-

[*] This work was done during an internship at Google Research.

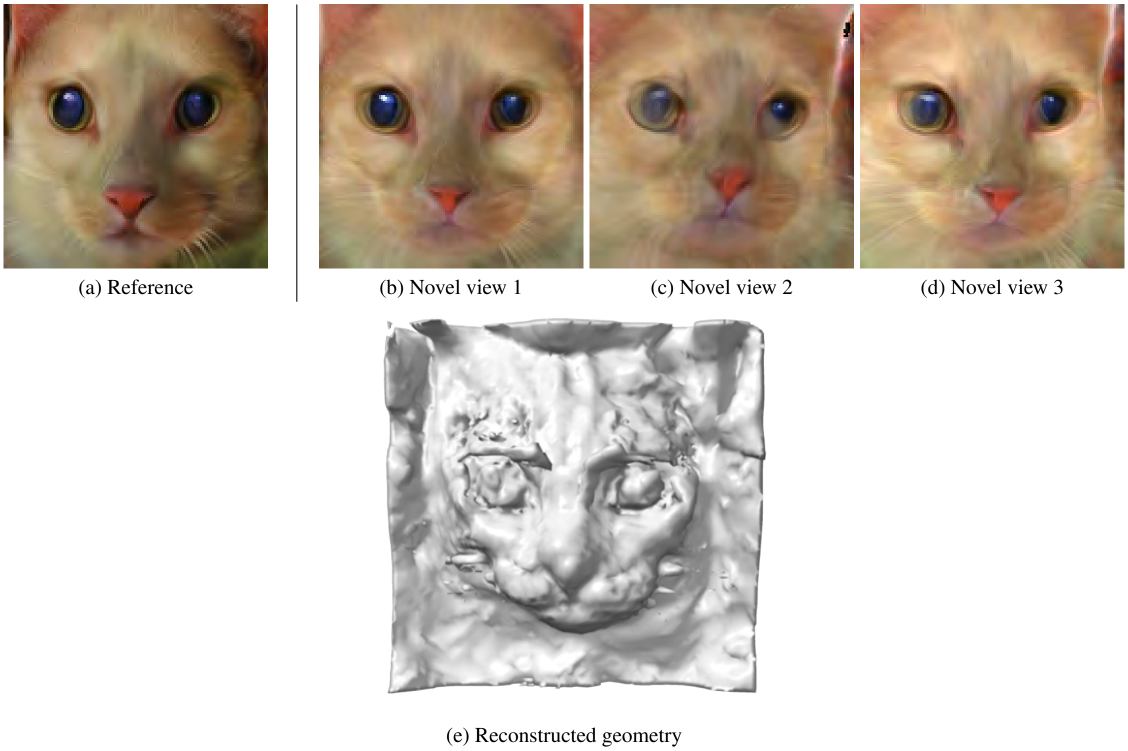

(a) Reference
(b) Novel view 1
(c) Novel view 2
(d) Novel view 3
(e) Reconstructed geometry

Figure 10. **Our results on cats:** We show results with a generator trained on cats. Top row: Novel views generated by our method. Second row: Reconstructed geometry.

lem by optimizing in the intermediate layers of StyleGAN to increase the expressive power of the generator.

Figure 11 shows a novel view enhancement with ILO. Unnatural characteristics in the ears and the face have been removed because of the StyleGAN face prior. It is important to clarify that using ILO alone we could not generate novel views, since we do not have any control of the pose. Our method is essential as a first step in this process to reconstruct a smooth geometry and render consistent views that can later be enhanced with ILO or some other 2-D image restoration technology.

To accelerate the process, we can first solve for one novel view and then use this StyleGAN latent as a warm start for the optimization problem for a different view. In general, ILO [4] requires 1-2 mins per view on a single GPU.

We note that using ILO to enhance images generated by our method is only a trick and not the central innovation of our method. For this reason, we choose to not include it in the main paper. However, practitioners working with our method might find it useful to obtain better quality reconstructions, which is why we include it here.

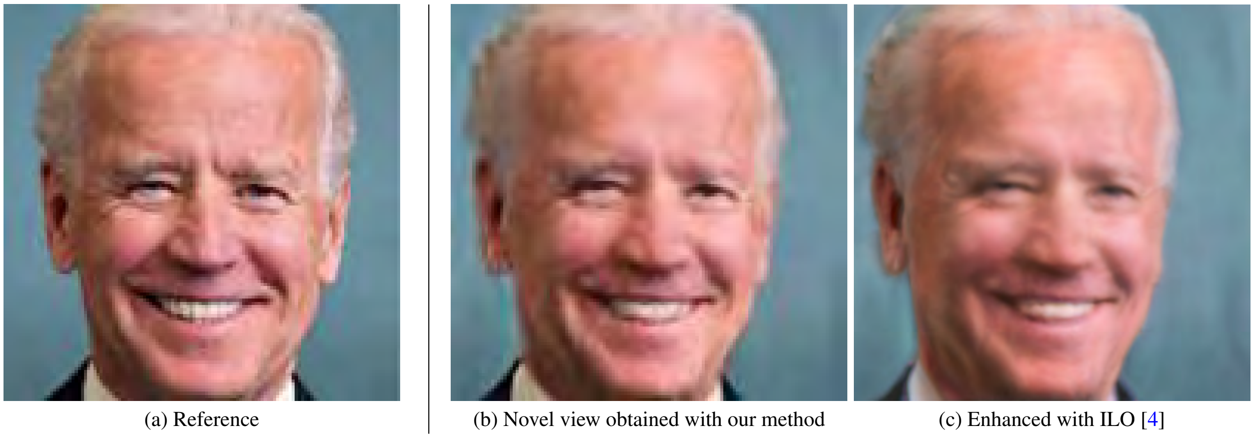
An interesting future direction that is relevant to this experiment is learning a forward operator that maps from the actual novel views to the reconstructed novel views. Essentially, it would be useful to know in which way the images that we want to fix with ILO are corrupted and then adjust the ILO algorithm to better account for this corruption. We leave this as a future direction.

7.3. Compressed Sensing

We start our additional experiments by looking at the problem of Compressed Sensing. The goal here is to reconstruct a signal $x \in \mathbb{R}^n$ (frontal view) by observing some linear measurements Ax, where $A \in \mathbb{R}^{m \times n}$ is a gaussian i.i.d matrix. Since we are working with a NerfGAN, we solve the optimization problem of 5.

Quantitative results for an arbitrarily picked novel view are shown in Figure 12. The Figure shows how the MSE error drops for both our method and the baseline as the number of measurements increase, as expected. Our method consistently outperforms the baseline for all measurements settings, which strengthens our argument that the 3D prior is useful for general inverse problems (*e.g.*, super resolution, compressed sensing), and not just inversion and inpainting as shown in the main paper.

To get a sense of how the number of measurements impacts the quality of the reconstructed image, we show examples of reconstructions for different measurement settings with our method. The images are shown in Figure 13. It is evident that as the number of measurements m increases,


(a) Reference

(b) Novel view obtained with our method

(c) Enhanced with ILO [4]

Figure 11. **Improved quality with ILO:** Novel view enhancement using StyleGAN [16–18] and Intermediate Layer Optimization (ILO) [4]. We have a two-stages approach. We first use our method to generate consistent novel views and then we use ILO to project close to the StyleGAN range and enhance the image. Unnatural characteristics in the ears and the face have been removed because of the StyleGAN face prior.

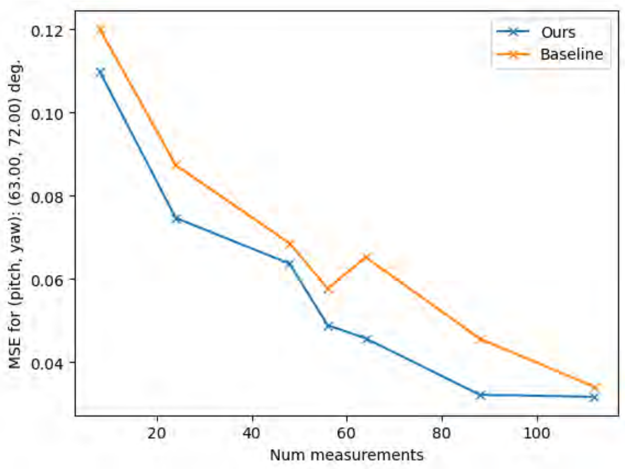


Figure 12. Quantitative results for an arbitrarily picked novel view for the task of **Compressed Sensing**. Our method consistently outperforms the baseline for all measurement settings. As the number of measurements increases, the error goes down as expected.

our reconstruction of the ground truth becomes more accurate, in agreement with the MSE plot of Figure 12.

7.4. Super-resolution

The next task we consider is the super-resolution task where the goal is to infer a higher-resolution version of a given image (and render novel views with it) by observing a very pixelated, low-resolution version of it. Quantitative results for an arbitrarily picked novel view for this task are shown in Figure 14. Our method outperforms the baseline in the low-measurements and matches its performance everywhere else. As the number of measurements increases, the error goes down as expected (similar to what happened for the Compressed Sensing case).

Visual results for the task of super-resolution our shown in Figure 15. Given a low-resolution image (first column, resolution 16 × 16) we show novel views rendered by our method and the baseline at resolution 128×128. Both methods seem to be performing relatively well on this task, but as the MSE scores of Figure 14, our method has better performance in the very low-measurements regime.

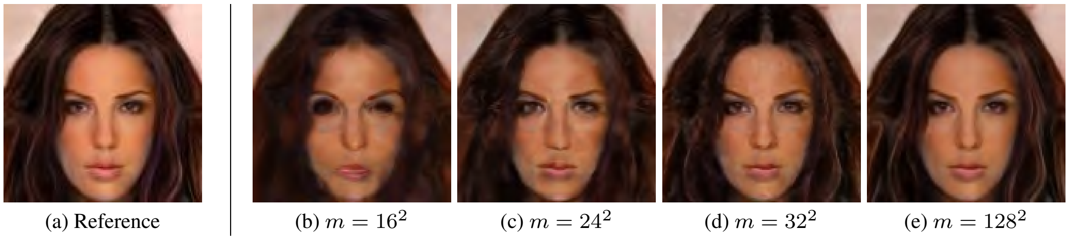
7.5. Inpainting

In the paper, we presented results for the task of randomized 2-D inpainting. Figure 16 illustrates one example of box inpainting. As shown, our method is capable of filling the missing region in a plausible way.

7.6. Ablation: number of reference geometries

We run an ablation study to examine the role of the number of reference geometries in the quality of the reconstructions. Intuitively, we expect that as we increase the number of reference geometries, the trend should be that the MSE should go down for novel views. This follows from the fact that our annealing scheme will force the optimization to converge to a single reference geometry eventually, so as the set of reference geometries becomes bigger, we expeect that we can find a geometry that matches our measurements *and* generalizes well to other views.

Our intuition is confirmed by Figure 17. The Figure shows that increasing the number of reference voxel grids leads to lower MSE, even in the frontal view. A more diverse set of reference geometries gives more flexibility to the optimization procedure to discover a geometry that matches well the measurements



(a) Reference

(b) $m = 16^2$

(c) $m = 24^2$

(d) $m = 32^2$

(e) $m = 128^2$

Figure 13. Effect of number of measurements in reconstructed images for the task of compressed sensing. We want to reconstruct the reference image $x \in \mathbb{R}^n$, by observing $y = Ax$, where A is a linear Gaussian matrix $\in \mathbb{R}^{m \times n}$. For the reference image, we have $n = 128^2 \times 3$. As shown, increasing the number of measurements m, leads to better reconstruction of the reference image.

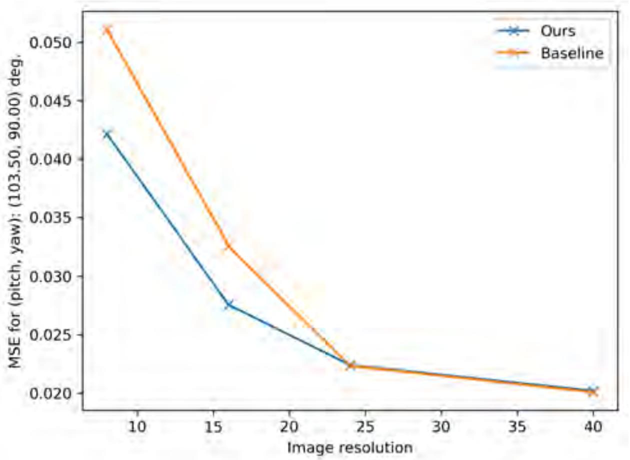


Figure 14. Quantitative results for an arbitrarily picked novel view for the task of **Super Resolution**. Our method outperforms the baseline in the low-measurements and matches its performance everywhere else. As the number of measurements increases, the error goes down as expected.

The trade-off here is that as the number of reference geometries increases, the computational cost of running the method increases as well (since we have to compute MSE between the current voxel grid and all the reference voxel grids). There are ways to mitigate the computational issue (i.e. completely removing some geometries as the inverse temperature goes up), but we will leave that as a future work.

7.7. Ablation: comparison with piGAN's regularization

pi-GAN [3] optimizes over the frequencies and the phase-shifts over the SIREN [42] network. During training, the frequencies and the phase shifts for all SIREN layers are the same. When solving inverse problems, the authors of [3] propose to let them diverge, in an attempt to make the generator more powerful. This is very similar in spirit in how StyleGAN [18] optimizes over the styles for all layers.

The main difference is that StyleGAN enforces a geodesic loss while pi-GAN penalizes frequencies and phase shifts for getting away from their average values. Figure 2 of the main paper shows that this regularization significantly limits the power of the generator (images are taken directly from the pi-GAN paper). In this section, we explore this observation a little bit more. Specifically, we run pi-GAN's proposed regularization, geodesic regularization and our method (3-D loss + geodesic) for real images from CelebA [25] and we report the average MSE scores in Table 1. As shown, Geodesic Regularization and the pi-GAN's regularization perform on par and our method consistently outperforms both of them by a large margin.

	Frontal view MSE	Frontal view LPIPS
pi-GAN	0.0070	89.95
Geodesic	0.0068	94.01
Ours	**0.0037**	**36.17**

Table 1. Comparison of different regularization schemes on real images from CelebA [25].

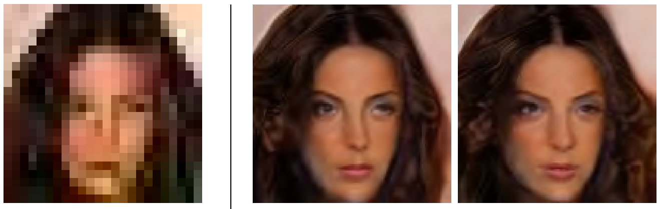
8. Discussion

8.1. Things that did not work

In this section, we share some negative results. Our goal is to make our efforts known to other researchers working in this field so they can avoid our methods, adjust them or even contradict our findings.

3-D Loss functions We spent a lot of time deriving a 3-D loss function that penalizes unrealistic voxel grids. We ended up defining something as unrealistic if it is far from a (known to be good) set of reference geometries. However, initially, we tried removing the dependency on the 3-D geometries and just regularize the 3-D voxel grid by penalizing unexpected behaviors.

Outside of the facial area, we expect the density values to be small since they correspond to empty space. Hence, a natural regularization would minimize the l_2 norm of the vectors with the densities of the non-facial surface. This

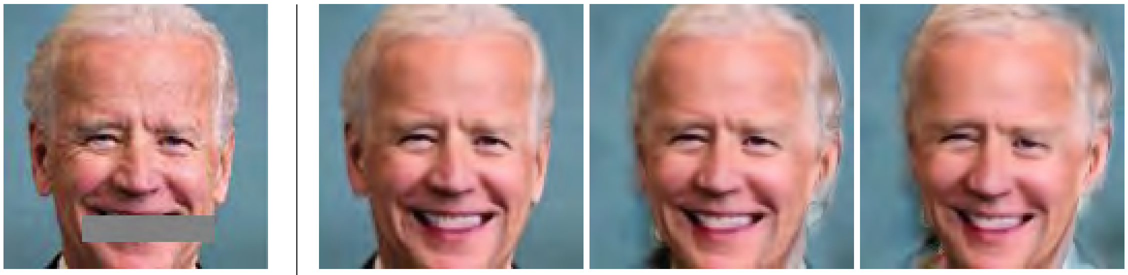


(a) Input image

(b) Novel view rendered with our method.

(c) Baseline novel view.

Figure 15. Super-resolution visual results. Given a low-resolution image (first column, resolution 16 × 16) we show novel views rendered by our method and the baseline at resolution 128 × 128.



(a) Input image

(b) Novel view

(c) Novel view

(d) Novel view

Figure 16. Inpainting of an image box. Our method is capable of filling the missing region in a plausible way.

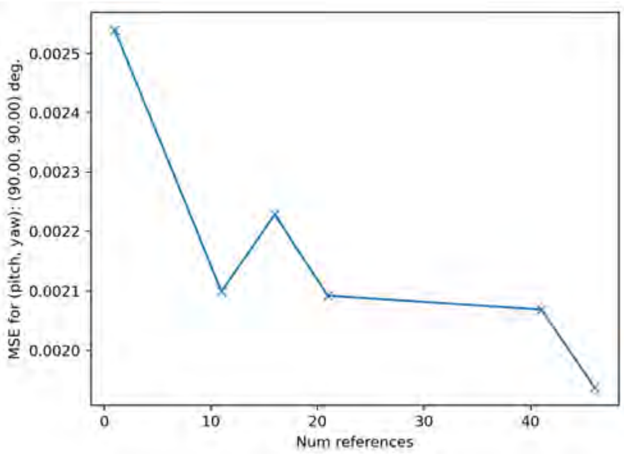


Figure 17. Ablation on how the number of reference grids affects the MSE error. The observed trend is that as the number of reference geometries goes up, the MSE error goes down, even for the frontal view. This agrees with our intuition that a more diverse set of reference geometries gives more flexibility to the optimization procedure to discover a geometry that matches well the measurements. For this experiment, we used Tempetature Annealing, as described in the paper.

method failed miserably. We attribute the failure to specific patterns that usually appear in the pi-GAN generated voxel grids. Specifically, the l_2 regularization encourages all the selected voxels to have small densities independently. pi-GAN does not have this structure; periodic patterns appear in the densities that most likely stem from the frequencies we feed to the SIREN [42] network.

Another property that we tried to enforce is smoothness of the 3-D signal, *i.e.*, Lipschitzness of the 3-D gradients. Our motivation was to reduce sudden changes in the facial structure that appeared when moving the camera even a little bit from the frontal view. Optimizing for this objective led to a different type of unrealistic geometries, where whole areas were smoothed out and a few spikes in the frontal view captured the information needed for a decent frontal view rendering.

Finally, we tried removing the dependency on the facial masks (obtained using the Marching Cubes algorithm). Our goal was to penalize unexpected behaviors everywhere, not just outside of the face. This method also failed; the expressivity of the generator reduced dramatically when we tried to match all the reference voxels. In retrospect, we could have expected that since this is esentially trying to match a random 3-D face from a dataset of 3-D faces.

Optimization Any method that regularizes inverse problems shows a trade-off between matching the measurements (in our case, the frontal view) and respecting the prior (in our case, maintaining a realistic 3-D structure). We explored different ways to balance this trade-off by applying optimization tricks. Specifically, we tried to do Alternating Gradient Descent [43], i.e. do one optimization step to minimize the measurements and one to respect the prior. We experimented with unbalanced Alternating Gradient Descent (e.g. taking more steps to fit the measurements) and we also tried to first fit the measurements completely and then adjust to respect the prior. All these attempts improved only marginally the results and hence they are not used in the paper to avoid the complication of tuning them for all our experiments. Finally, we tried different learning rates schedulings, but we observed no big improvement to the results over the vanilla Adam [19] optimizer with weight decay.

8.2. CLIP

In the main paper, we explained that we use CLIP [37] as a way to automatically filter out bad geometries and consequently form a set of reference geometries. In this section, we provide more details about how this is done in practice and we include examples of bad geometries and their corresponding 2-D views, as detected by CLIP.

The main idea is that we expect consistent geometries to generate views that are not very different in some semantically meaningful space, such as the space of CLIP embeddings. In all our experiments, we render 9 views with CLIP, with $\theta \in \{76.5^\circ, 90^\circ, 103.5^\circ\}$ and $\phi \in \{81^\circ, 90^\circ, 99^\circ\}$. We found this value by experimenting and manually inspecting the CLIP classifications of good and bad geometries for 10 validation images.

Examples of identified bad geometries and their corresponding 2-D images are shown in Figure 18. Surprisingly, the collected geometries are pro-duced by latents sampled from the Gaussian distribution. This in-dicates that pi-GAN has failure modes that we need to avoid whensolving inverse problems

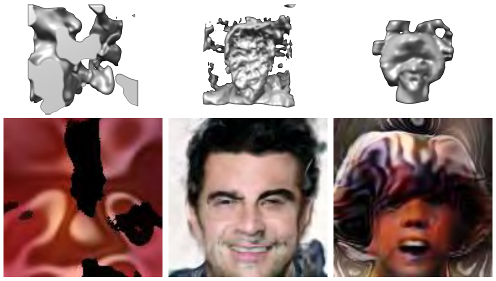


Figure 18. Automatically classified bad geometries and their corresponding 2-D views. For the classification, we used the free-text classifier, CLIP [37]. Surprisingly, the collected geometries are produced by latents sampled from the Gaussian distribution. This indicates that pi-GAN has failure modes that we need to avoid when solving inverse problems.

9. Experimental Details

9.1. Hyperparameters

In all our experiments, we tried to follow as closely as possible the hyperparameters reported in the pi-GAN paper [3]. For the baseline runs, we started with the hyperparameters reported in the paper and we reported the best result among this run and our tuning of the baseline's hyperparameters.

We use the Adam [19] optimizer with initial learning rate at 0.01 and a half step decay every 200 steps, as recommended. As in the pi-GAN paper, for all inverse problems, we optimize for the frequencies and the phase-shifts (and not the latent z itself). During training, the model had the same frequencies and phase-shifts across all layers for a single image. Inspired by the StyleGAN [16–18] papers we allow these frequencies and phase-shifts to diverge. The pi-GAN paper is using MSE from their average values as regularization that forces them to stay close. In our paper, we use Geodesic Loss, as used in [4, 17, 29].

Additionally to the MSE distance, we also use a Perceptual Distance, LPIPS [45], for the inversion problem, as in [4]. For a fair comparison with the baseline, we have deactivated LPIPS for all the plots included in the paper. However, for best visual results, we recommend adding the Perceptual Loss in addition to MSE.

Below, we provide a list of hyperparameters that one needs to tune for best results with method and a proposed set of sensible values.

- Temperature Annealing Values: $[100., 150., 200., 250., 300., 350., 400., 500., 550.]$
- Temperature Annealing Steps: $[100, 200, 300, 400, 500, 600, 700]$
- Number of reference voxel grids: $\{16, 32, 64, 128\}$
- Voxel grid resolution: $\{32^3, 64^4\}$
- Learning rate: {1e-2, 5e-3, 1e-3}
- LPIPS coefficient: $\{1e-2, 5e-3\}$
- MSE coefficient: $\{1, 5, 10\}$
- Prior coefficient: $\{1, 1e-1, 1e-2\}$
- Geodesic coefficient: $\{5e-1, 1e-1\}$

9.2. Plots

For all the plots, we are running our method and the baseline for 100 images in the range of the GAN and we average across all runs. For the inverse problems where corruption has taken place (e.g. inpainting, super-resolution) we report MSE to the ground-truth signal that is never observed during the optimization. For all these plots, we are able to report MSE for novel views because the images we test on are in the range of the GAN, i.e. we first sampled a latent z and then produced them. For real images, we can only compare the visual results between our method and the baseline as we did in the paper.

References

[1] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017. 2, 3, 6

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018. 1

[3] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 1, 2, 3, 5, 6, 9, 12, 14

[4] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G Dimakis. Intermediate layer optimization for inverse problems using deep generative models. *arXiv preprint arXiv:2102.07364*, 2021. 3, 8, 9, 10, 11, 14

[5] Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G Dimakis. Smyrf: Efficient attention using asymmetric clustering. *arXiv preprint arXiv:2010.05315*, 2020. 1

[6] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14531–14539, 2020. 1

[7] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4D view synthesis and video processing. *arXiv preprint arXiv:2012.09790*, 2020. 3

[8] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, and et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics*, 39(5):1–38, Sep 2020. 3

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 1

[10] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *https://arxiv.org/abs/2103.10380*, 2021. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis, 2021. 3

[13] Paul Hand, Oscar Leong, and Vladislav Voroninski. Phase retrieval under a generative prior. *arXiv preprint arXiv:1807.04261*, 2018. 2

[14] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jonathan I Tamir. Robust compressed sensing mri with deep generative priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[15] Ajil Jalal, Sushrut Karmalkar, Jessica Hoffmann, Alex Dimakis, and Eric Price. Fairness for image generation with uncertain sensitive attributes. In *International Conference on Machine Learning*, pages 4721–4732. PMLR, 2021. 8

[16] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks, 2021. 1, 9, 11, 14

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018. 1, 3, 9, 11, 14

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 3, 9, 11, 12, 14

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 14

[20] Abhishek Kumar and Ehsan Amid. Constrained instance and class reweighting for robust learning under label noise, 2021. 5

[21] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3d video synthesis, 2021. 3

[22] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *https://arxiv.org/abs/2011.13084*, 2020. 3

[23] David Lindell, Julien Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. *https://arxiv.org/abs/2012.01714*, 2020. 3

[24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 3

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 9, 12

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 6

[27] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 6

[28] 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. *arXiv preprint arXiv:2103.15606*, 2021. 1, 3

[29] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 3, 8, 14

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 1, 3

[31] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *arXiv preprint arXiv:2103.03231*, 2021. 3

[32] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yong-Liang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes, 2018. 3

[33] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 1, 3

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 3

[35] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020. 3

[36] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis, 2021. 3

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 3, 4, 9, 14

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 4

[39] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. DeRF: Decomposed radiance fields. *https://arxiv.org/abs/2011.12490*, 2020. 3

[40] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021. 3

[41] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis, 2021. 3

[42] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions, 2020. 2, 3, 12, 13

[43] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015. 13

[44] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *https://arxiv.org/abs/2011.12950*, 2020. 3

[45] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 3, 14

[46] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis, 2021. 3, 8

[47] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3