NeRF-VAE: A Geometry Aware 3D Scene Generative Model

Adam R. Kosiorek [*1] **Heiko Strathmann** [*1] **Daniel Zoran** [1] **Pol Moreno** [1] **Rosalia Schneider** [1] **Soňa Mokrá** [1] **Danilo J. Rezende** [1]

Abstract

We propose NeRF-VAE, a 3D scene generative model that incorporates geometric structure via Neural Radiance Fields (NeRF) and differentiable volume rendering. In contrast to NeRF, our model takes into account shared structure across scenes, and is able to infer the structure of a novel scene—without the need to re-train—using amortized inference. NeRF-VAE's explicit 3D rendering process further contrasts previous generative models with convolution-based rendering which lacks geometric structure. Our model is a VAE that learns a distribution over radiance fields by conditioning them on a latent scene representation. We show that, once trained, NeRF-VAE is able to infer and render geometrically-consistent scenes from previously unseen 3D environments of synthetic scenes using very few input images. We further demonstrate that NeRF-VAE generalizes well to out-of-distribution cameras, while convolutional models do not. Finally, we introduce and study an attention-based conditioning mechanism of NeRF-VAE's decoder, which improves model performance.

1 Introduction

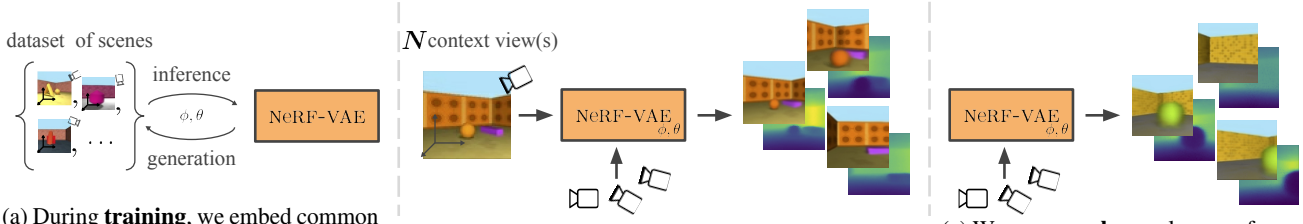
The ability to infer the structure of scenes from visual inputs, and to render high quality images from different viewpoints, has vast implications for computer graphics and virtual reality.

This problem has traditionally been tackled with 3D reconstruction-based on matching visual keypoints, e. g. Lowe (2004); Schönberger & Frahm (2016). While these methods incorporate structure through multi-view geometry (Hartley & Zisserman, 2003), very few existing methods adopt learned scene-priors such as types of objects or statistics of the background. The resulting representations are usually discrete: they consist of meshes (Dai et al., 2017), point clouds (Engel et al., 2014), or discretized volumes (Ulusoy et al., 2016), and are difficult to integrate with neural networks. An alternative approach to scene generation uses light-field rendering (Levoy & Hanrahan, 1996; Srinivasan et al., 2017) and multiplane image representations (Zhou et al., 2018; Mildenhall et al., 2019). While these methods allow image-based rendering, they do not estimate scene geometry. Hence, they are not directly useful for the purpose of 3D scene understanding. More recent deep-learning-based novel view synthesis methods have the advantages of end-to-end training, and producing distributed representations that can be easily used in other neural-net-based downstream tasks. These models, however, often have little embedded geometrical knowledge (Dosovitskiy et al., 2014) and are either geometrically inconsistent (Nguyen-Phuoc et al., 2020) or provide insufficient visual quality (Tatarchenko et al., 2015). Moreover, many of these methods are deterministic and cannot manage uncertainty in the inputs (Sitzmann et al., 2019; Trevithick & Yang, 2021).

This work attempts to address these shortcomings. We introduce NeRF-VAE—a deep generative model with the knowledge of 3D geometry as well as complex scene priors. Our work builds on Neural Radiance Fields (NeRF, Mildenhall et al. (2020)). NeRF combines implicit neural network representations of radiance fields, or *scene functions*, with a volumetric rendering process. NeRF needs to undergo a lengthy optimization process on many views of each single scene separately and does not generalize to novel scenes. In contrast, NeRF-VAE—a variational auto-encoder (VAE)—models multiple scenes, while its constant-time amortized inference allows reasoning about novel scenes. Unlike NeRF, our model is also generative, and therefore capable of handling missing data and imagining completely new scenes. In that, our work closely follows Generative Query Networks (GQN, Eslami et al. (2018)). Like GQN, NeRF-VAE defines a distribution over scene functions. Once sampled, a scene function allows rendering arbitrary views of the underlying scene. Where GQN relies on convolutional neural networks (CNNs) with no knowledge of 3D geometry, leading to geometrical inconsistencies, NeRF-VAE achieves consistency by leveraging NeRF's implicit representations and volumetric

[*]Equal contribution [1]DeepMind, London. Correspondence to: ARK <adamrk@deepmind.com>, HS <strathmann@deepmind.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.



(a) During **training**, we embed common scene properties (across the dataset) in the parameters ϕ, θ of NeRF-VAE.

(b) Once the model is trained, we can **infer** parameters of a scene in constant time, even from a single view.

(c) We can **sample** novel scenes from NeRF-VAE generative model and render them from various viewpoints.

Figure 1: An overview of NeRF-VAE: a geometry-aware 3D scene generative model. NeRF-VAE is trained on a dataset of several views (images and camera positions/orientations) from **multiple** synthetic scenes. Once trained, it allows for efficient inference of scene parameters (colours and geometry, including depth-maps) and sampling novel scenes from the prior.

rendering. We note that while these representations do not model the scene geometry explicitly (as e. g. mesh-based representations), the subsequent volumetric rendering process explicitly uses 3D rays and 3D geometry to produce output images.

NeRF represents a scene by the values of multilayer perceptron (MLP) parameters. Being extremely high-dimensional, this representation precludes using amortized inference. We change NeRF's formulation to a scene function shared between scenes, and conditioned on a **per-scene** latent variable. Intuitively, the latent variable captures scene-specific information (e. g. position and kind of objects, colours, lighting, etc.), while shared information (e. g. available textures and shapes, properties of common elements, sky) is stored in the parameters of the scene function. A prior over latent variables (and therefore scene functions) allows sampling novel scenes from the model, and rendering arbitrary viewpoints within them. NeRF-VAE parameters are learned using a collection of images from different scenes, with known camera position and orientation. Since these parameters are shared between scenes (unlike in NeRF), NeRF-VAE can infer scene structure from very few views of an unseen scene. This is in contrast to NeRF, where using few views results in poor performance. A high level overview of NeRF-VAE is depicted in Fig. 1.

In summary, NeRF-VAE introduces four key benefits compared to existing models. First, due to its amortized inference, it removes the need for costly optimization from scratch for every new scene. Second, as it learns shared information between multiple scenes, it is able reconstruct unobserved scenes from a much smaller number of input views. Third, compared to existing convolutional generative models for view synthesis, such as GQN, it generalizes much better when evaluated on out-of-distribution camera views. Finally, NeRF-VAE is the only amortized NeRF variant that uses compact scene representation in the form of a latent variable and that can handle uncertainty in the inputs.

2 Neural Radiance Fields (NeRF)

NeRF's scene function is represented as a 6D continuous vector-valued function whose inputs are ray coordinates $(\boldsymbol{x}, \boldsymbol{d})$ partitioned into position $\boldsymbol{x} \in \mathbb{R}^3$ and orientation[1] $\boldsymbol{d} \in \mathbb{R}^3$. Its outputs are an emitted colour $(r, g, b) \in \mathbb{R}^3$ and a volume density $\sigma \geq 0$. The scene function is approximated by a neural network $F_\theta : (\boldsymbol{x}, \boldsymbol{d}) \mapsto ((r, g, b), \sigma)$ with weights θ. In order to encourage multi-view consistency, the architecture of this network is such that volume density σ only depends on position while the emitted colour (r, g, b) depends on both position and ray orientation.

Volumetric image rendering works by casting rays from the camera's image plane into the scene, one ray per pixel. The colour that each ray produces is the weighted average of colours along the ray, with weights given by their accumulated volume densities. NeRF's renderer uses a differentiable approximation to this accumulation process. For details, we refer to Mildenhall et al. (2020); Curless & Levoy (1996).

We denote the image rendering process whose inputs are a **camera** $\boldsymbol{c}$ (position and orientation) and a scene function F_θ, and which outputs the rendered image[2] $\hat{\boldsymbol{I}}$, by

$$\hat{\boldsymbol{I}} = \texttt{render}(F_\theta(\cdot), \boldsymbol{c}) \,. \tag{1}$$

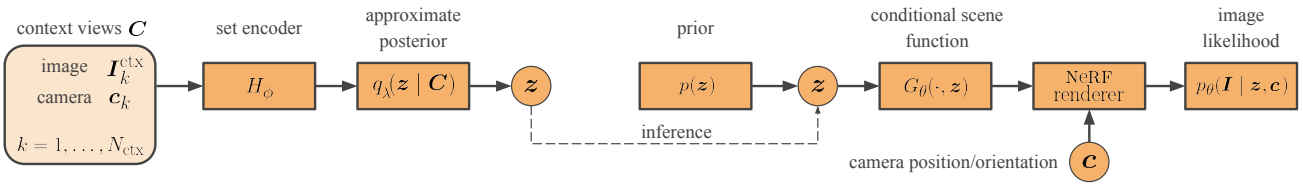
Note that computation of the camera's image plane and corresponding rays for each pixel requires camera parameters (e. g. field of view, focal length, etc).

3 NeRF-VAE

We build a generative model over scenes by introducing a view-independent latent variable $\boldsymbol{z}$ with prior $p(\boldsymbol{z})$ and a **conditional scene function** $G_\theta(\cdot, \boldsymbol{z}) : (\boldsymbol{x}, \boldsymbol{d}) \mapsto ((r, g, b), \sigma)$. G_θ is much like NeRF's F_θ, but is additionally

[1] It is also possible to parameterize orientation using two angles, as e. g. done by Mildenhall et al. (2020), but we follow their implementation in using a vector.

[2] Note that it is easily possible to compute depth estimates for each rendered ray. We use this for visualization and evaluation purposes.



(a) Inference in NeRF-VAE. (b) The generative model of NeRF-VAE.

Figure 2: Inference and generative model of NeRF-VAE. For inference, a set $\boldsymbol{C}$ of context images $\boldsymbol{I}_k^{\text{ctx}}$ and cameras $\boldsymbol{c}_k$ from a scene are encoded into an approximate posterior distribution over the latent variable $\boldsymbol{z}$. This conditions a scene function $G_\theta(\cdot, \boldsymbol{z})$, which is used by the NeRF renderer to reconstruct images from arbitrary cameras within the scene. We can sample novel scenes by sampling the latent $\boldsymbol{z}$ from the prior. During training, the reconstruction MSE and the KL divergence are used in a variational optimization objective to learn the parameters θ of the conditional scene function and the parameters ϕ of the encoder.

conditioned on $\boldsymbol{z}$, as will be detailed in Section 3.2.

It is now the latent variable $\boldsymbol{z}$ that defines a **specific** scene, out of all the scenes that G_θ is able to represent, where θ are model parameters that capture **shared structure** across scenes. The generative process of NeRF-VAE involves sampling $\boldsymbol{z} \sim p(\boldsymbol{z})$ and using the resulting conditional scene function $G_\theta(\cdot, \boldsymbol{z})$ to render an image $\hat{\boldsymbol{I}} = \texttt{render}(G_\theta(\cdot, \boldsymbol{z}), \boldsymbol{c})$ from a camera $\boldsymbol{c}$ with volumetric rendering from Eq. (1).

The individual pixel colours $\hat{\boldsymbol{I}}(i, j)$ are used to define the mean parameter of a Gaussian image likelihood

$$p_\theta(\boldsymbol{I} \mid \boldsymbol{z}, \boldsymbol{c}) = \prod_{i,j} \mathcal{N}\left(\boldsymbol{I}(i, j) \mid \hat{\boldsymbol{I}}(i, j), \sigma_{\text{lik}}^2\right)$$

with fixed or learned variance σ_{lik}^2, and we assume conditional independence of individual pixels $\boldsymbol{I}(i, j)$ given $\boldsymbol{z}$.

Since the posterior over $\boldsymbol{z}$ is intractable, we approximate it—we frame NeRF-VAE as a variational auto-encoder (VAE, Kingma & Welling (2014); Rezende et al. (2014)) with **conditional** NeRF as its decoder.

3.1 Amortized Inference for NeRF-VAE

Estimating the scene function parameters in NeRF is done separately for every scene, with no information sharing between scenes, which is time-consuming, compute-intensive, and data-hungry.

In contrast, NeRF-VAE introduces an encoder network E_ϕ with parameters ϕ, which amortizes inference of the latent variable $\boldsymbol{z}$. Input for the encoder is a collection of N_{ctx} **context views**, where each view consists of an image $\boldsymbol{I}^{\text{ctx}} \in \mathbb{R}^{H \times W \times 3}$ and corresponding camera position and orientation $\boldsymbol{c}$. These views correspond to different viewpoints of a particular scene, forming a context set $\boldsymbol{C} := \{\boldsymbol{I}_k^{\text{ctx}}, \boldsymbol{c}_k\}_{k=1}^{N_{\text{ctx}}}$. Each context element (concatenated camera $\boldsymbol{c}_k$ with image $\boldsymbol{I}_k^{\text{ctx}}$) is separately encoded using a shared encoder—we use a ResNet adapted from Vahdat & Kautz (2020), details in Appendix C. The resulting N_{ctx} outputs are subsequently averaged[3] and mapped to parameters λ of the approximate posterior distribution $\mathrm{q}_\lambda(\boldsymbol{z} \mid \boldsymbol{C})$ over the latent variable $\boldsymbol{z}$. We use diagonal Gaussian posteriors.

We fit the parameters $\{\theta, \phi\}$ of the NeRF-VAE by maximizing the following evidence lower bound (ELBO) on images,

$$\begin{aligned}&\mathcal{L}_{\text{NeRF-VAE}}(\boldsymbol{I}, \boldsymbol{c}, \boldsymbol{C}; \theta, \phi) = \\ &\mathbb{E}_{\boldsymbol{z} \sim q}\left[\log p_\theta\left(\boldsymbol{I} \mid \boldsymbol{z}, \boldsymbol{c}\right)\right] - \mathrm{KL}(\mathrm{q}_\lambda(\boldsymbol{z} \mid \boldsymbol{C}) \mid\mid p(\boldsymbol{z}))\,.\end{aligned} \tag{2}$$

In practice, we approximate the ELBO by uniformly subsampling pixels from each image, see Appendix E for details.

Iterative Amortized Inference Amortized inference suffers from *amortization gap* (Cremer et al., 2018)—contrasting the gradient-based learning in NeRF. To bridge this gap, we employ iterative (amortized) inference (Kim et al., 2018; Marino et al., 2018), which trades-off additional compute for improved inference.

Iterative inference starts with an arbitrary guess for the posterior parameters, e. g. $\lambda_0 = \mathbf{0}$, and iteratively refines them. At each step t, a latent is sampled from the current posterior $\boldsymbol{z} \sim q_{\lambda_t}(\boldsymbol{z} \mid \boldsymbol{C})$. The sample is then used to render an image and to evaluate the gradient of the ELBO in Eq. (2) w. r. t. λ_t. The gradient is passed to a recurrent refinement network (LSTM followed by a linear layer)[4] f_ϕ, which updates the posterior parameters λ_t for a given $\boldsymbol{C}$:

$$\begin{aligned}\boldsymbol{z}_t &\sim q_{\lambda_t}(\boldsymbol{z} \mid \boldsymbol{C})\ , \\ \lambda_{t+1} &\leftarrow \lambda_t + f_\phi\left(E_\phi(\boldsymbol{C}), \nabla_{\lambda_t} \mathcal{L}_{\text{NeRF-VAE}}\right)\ .\end{aligned}$$

See Appendix C for further details.

We emphasize that NeRF-VAE's decoder uses explicit geometric structure, which consequently is used by iterative inference. While this is only in an implicit manner (through ELBO gradients), it differs from a geometry-agnostic feed-forward encoder typically used with VAEs.

[3] We choose this method for its conceptual simplicity and note that more sophisticated options, such as attention-based mechanisms, might lead to better results as in Trevithick & Yang (2021)—exploration of which we leave for future work.

[4] Note that in this case, ϕ only contains parameters of the context encoder E_ϕ as well as the refinement network.

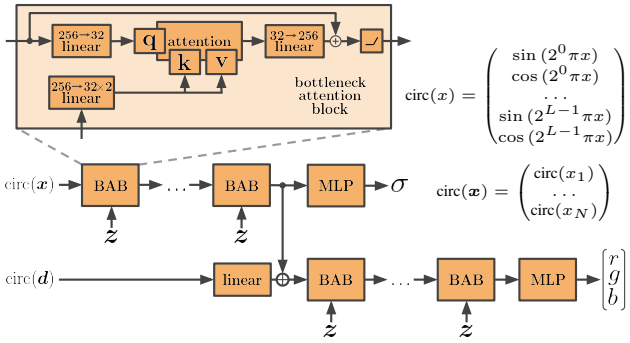


Figure 3: Attention-based scene function. Input points $(\boldsymbol{x}, \boldsymbol{d})$, along rays corresponding to each pixel in the camera plane, **attend** to different locations of the spatial latent variable $\boldsymbol{z}$ of size $[H_z \times W_z \times D_z]$ using multi-head attention. The density part (σ) depends only on the position $\boldsymbol{x}$ as in NeRF. Each attention block receives a different slice of the latent along its channel dimension. The inputs are projected to a lower-dimension space to save computation and memory; $\oplus$ is concatenation.

3.2 Conditioning the Scene Function

We now describe NeRF-VAE's conditional scene function $G_\theta(\cdot, \boldsymbol{z})$, which is conditioned on the per-scene latent variable $\boldsymbol{z}$ and has additional across-scene parameters θ.

A simple way to condition NeRF's scene function MLP is to use $\boldsymbol{z}$ to shift and scale the inputs and activations at different layers—this mechanism resembles AIN of Dumoulin et al. (2017); Brock et al. (2019) and is referred to as an MLP scene function henceforth.

We further introduce an **attention-based** scene function, see Fig. 3 for an overview. Attention (Vaswani et al., 2017) allows using a high-dimensional spatial $[H_z \times W_z \times D_z]$ with latent variable over which inputs of the scene function can attend. The spatial structure arises from removing the final average pooling across locations in the ResNet encoder.[5] Since the scene function is evaluated many times (e. g. 256 in Mildenhall et al. (2020); 128 in our experiments) for every pixel, it has to be computationally cheap with low memory footprint. Consequently, we use only one linear layer per attention block and no layer norm. Additionally, our attention blocks are bottlenecks, i.e. keys, queries and values are low dimensional c. f. Srinivas et al. (2021). We find that these choices are a good trade-off between computation, memory and capacity, and used throughout our experiments.

In order to model correlations between spatial locations, otherwise unaccounted for by an independent prior $p(\boldsymbol{z})$, we apply a small CNN to the latent before it is fed into the scene function, see Appendix C for further details.

[5]Since the feature maps are averaged over context elements, locations in the latent do not necessarily correspond to parts of the scene; we use $H_z = W_z = 8$

4 Related Work

NeRF-VAE is closely related to amortized neural rendering approaches. Trevithick & Yang (2021); Yu et al. (2020) use NeRF as their decoder but require projecting all rendered points into the input space and thus do not formulate any compact scene representation. Tancik et al. (2020) suffers from a similar issue: it meta-learns good initializations for NeRF, but requires updating all parameters before it can render target observations. Sitzmann et al. (2019) uses sequential ray-marching instead of volumetric rendering, and represents scenes using a hyper-network, which is in contrast to NeRF-VAE's compact representation. NeRF-VAE has many similarities with GQN of Eslami et al. (2018), with the difference that GQN uses a CNN for rendering and is therefore not necessarily consistent across views. Mescheder et al. (2019) show impressive single-image 3D reconstructions by modelling space occupancy with an implicit representation, though not modeling colours. Henderson & Lampert (2020) describe an object-centric, voxel-based generative model for videos that is able to segment scenes in 3D and produce samples of novel scenes.

The above approaches work only when camera poses for input images are known. This is not true for GANs, where it is enough to approximate the marginal distribution of poses in the training set if appropriate inductive biases are in place (Nguyen-Phuoc et al., 2019)—an approach that also scales to an explicit multi-object setting (Nguyen-Phuoc et al., 2020). GRAF (Schwarz et al., 2020) and GIRAFFE (Niemeyer & Geiger, 2020) reuse the same idea for single- and multi-object settings, respectively, but use NeRF as the generator, which improves multi-view consistency. Both approaches are related to NeRF-VAE in the sense that they use NeRF as a decoder which is conditioned on a latent variable—although NeRF-VAE introduces a more advanced conditioning mechanism. Finally, in contrast to the GAN approaches, NeRF-VAE has an associated inference procedure.

A limitation of NeRF is that it can only model static scenes and does not support varying lighting conditions nor transient effects (e. g. moving objects) often visible in the real world. NeRF-W (Martin-Brualla et al., 2020) addresses these shortcomings by adding per-view latent variables. While our model makes latent variables explicit, it maintains a single latent per scene; an extension towards NeRF-W is an interesting research direction. Pumarola et al. (2020); Park et al. (2020); Du et al. (2020); Li et al. (2020); Xian et al. (2020) further extend NeRF to dynamic scenes and videos: they formulate flow-fields that augment the original radiance field with a temporal component. This complicated approach is necessary due to the high-dimensional nature of NeRF's representation. Since NeRF-VAE introduce a latent variable, it is possible to extend it to videos by simply adding a latent dynamics model (e. g. Hafner et al. (2020)).

We provide a tabular comparison of NeRF-VAE and discussed models in terms of desirable scene model properties in Appendix A.

5 Experiments

To evaluate NeRF-VAE, we first analyze its ability to reconstruct novel viewpoints given a small number of input views, and contrast that with NeRF. Second, we compare our model with a Generative Query Network-like autoregressive convolutional model, (Eslami et al., 2018, GQN) and show that while NeRF-VAE achieves comparably low reconstructions errors, it has a much improved generalization ability, in particular when being evaluated on camera views not seen during training. Third, we provide an ablation study of NeRF-VAE variants, with a focus on the conditioning mechanisms of the scene function described in Section 3.2. Finally, we showcase samples of NeRF-VAE.

5.1 Datasets

We use three datasets, each consisting of 64×64 coloured images, along with camera position and orientation for each image, and camera parameters used to extract ray position and orientation for each pixel.

GQN (Eslami et al., 2018)[6], consists of 200k scenes each with 10 images of rooms with a variable number of objects. Camera positions and orientations are randomly distributed along a plane within the rooms, always facing the horizon. Note that this dataset does not contain reflections or specularities, which we discuss in Section 5.4

CLEVR We created a custom CLEVR dataset (Johnson et al., 2017) with 100k scenes, each with 10 views. Each scene consist of one to three randomly coloured and shaped objects on a plane. Camera positions are randomly sampled from a dome. Orientations are such that all objects are always present, making inference from a limited number of views easier than in the GQN data. The dataset is of higher visual quality than GQN, e. g. it includes reflections.

Jaytracer In order to have more control over individual aspects of the data (e. g. complexity, distribution of camera views, etc.), we also created a custom raytraced dataset which consists of 200k randomly generated scenes, each with 10 views, with a fixed number of objects in random orientations on a plane. Shapes, colours, textures, and light position are random, an example can be seen in Fig. 4.

For both CLEVR and Jaytracer, we generate 10 additional scenes each with 200 views and ground-truth depth maps for evaluation purposes and for training NeRF. See Appendix B for more details.

[6]We use the `rooms_free_camera_no_object_rotations` variant publicly available at `https://github.com/deepmind/gqn-datasets`

5.2 Implementation Details

The conditional scene functions' architecture follows NeRF, first processing position $\boldsymbol{x}$ to produce volume density and then additionally receiving orientation $\boldsymbol{d}$ to produce the output colours. Both position and orientation use circular encoding whereby we augment the network input values with a Fourier basis, c. f. Fig. 3.

We follow Mildenhall et al. (2020, Section 5.2) in using hierarchical volume sampling in order to approximate the colour of each pixel. This means we maintain a second instance of the conditional scene function (conditioned on the same latent $\boldsymbol{z}$), which results in an additional likelihood term in the model log-likelihood in Eq. (2).

We use Adam (Kingma & Ba, 2014) and β-annealing of the KL term in Eq. (2). Full details can be found in Appendix C.

5.3 Comparison with NeRF

NeRF-VAE, unlike NeRF, can infer scene structure without re-training—through the introduction of a per-scene latent variable and shared across-scene parameters. Our first experiment explores the following questions: 1. Can NeRF-VAE leverage parameter sharing to infer novel scenes from very few views? 2. Is NeRF-VAE's capacity affected by having a much smaller scene representation (a latent variable instead of an MLP). 3. At what number of views do both models reach comparable errors?

To focus on these conceptual differences between NeRF and NeRF-VAE, we use NeRF-VAE with the simple MLP scene function and without iterative inference. We train NeRF-VAE on the Jaytracer data using $N_{\text{ctx}} = 4$ context images (with corresponding cameras). We evaluate on 10 held-out scenes, with an increasing number of context views $N_{\text{ctx}}^{\text{test}} = 1, \ldots, 6$.

We train a separate instance of NeRF on each of these same 10 evaluation scenes, with an increasing number of training views $5, \ldots, 100$. Both models are evaluated on images of 100 unseen views from the 10 evaluation scenes. We train and evaluate both models using 10 different seeds. See Appendix F for further training details.

Results

Fig. 4 (a) shows reconstruction MSEs (mean and 95% percentiles) across the 100 test views. NeRF-VAE achieves a significantly lower MSE and uniformly lower worst case errors compared to NeRF trained on less than 100 views. This is despite NeRF-VAE's amortized inference, which is many

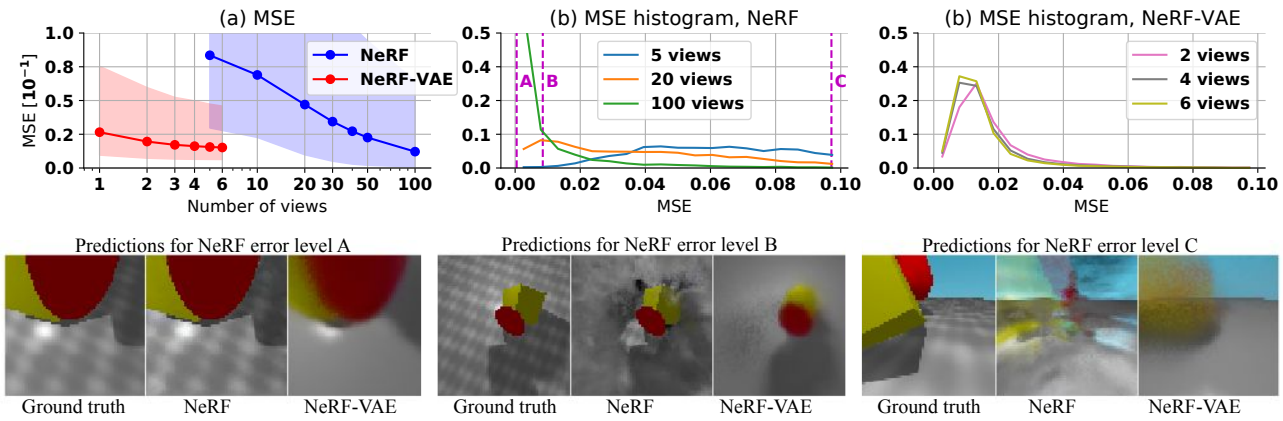


Figure 4: Error analysis of NeRF and a minimalistic version (see text) of NeRF-VAE on Jaytracer data. **(a)**: MSE decreases with increasing number of training (NeRF) and context (NeRF-VAE) views. We show the mean and 95% percentiles across 100 test views averaged over 10 scenes and 10 seeds. Despite its minimal version, NeRF-VAE performs much better for fewer (≤ 6) views; NeRF needs many more views (≥ 100) to reach comparable error. **(b, c)**: MSE histograms. Compared to NeRF-VAE, NeRF needs a large number of training views to consistently achieve low errors, and even then incurs a small number of larger errors. **Bottom**: An example scene, where views correspond to three error levels, indicated in (b), of NeRF trained on 100 views. NeRF's predictions for level A are near perfect, but the model fails catastrophically for level C—which regularly happens when training on fewer views, see (b). NeRF-VAE's predictions are more consistent, despite its simple decoder and inference.

orders of magnitude faster than running the full optimization in NeRF. NeRF achieves lower MSE than our model only when sufficient training data is available (here $= 100$ views). We emphasize that while NeRF-VAE was trained using $N_{\text{ctx}} = 4$ context views, the model generalizes well to different numbers of context views at test time.

Taking a look at the distribution of errors and the corresponding predictions in Fig. 4 (b, c), we see that NeRF's MSE distributions are extremely wide when trained on fewer than 100 views. Even for 100 views, NeRF suffers from a long tail of large errors. NeRF-VAE's errors concentrate on a small positive value for all evaluated $N_{\text{ctx}}^{\text{test}}$, with tails comparable to NeRF with 100 training views.

Fig. 4, bottom, shows predictions from both models on an example scene, for views corresponding to error levels attained by NeRF trained on 100 views: near-perfect (A), medium (B), and catastrophic (C), marked in Fig. 4 (b). NeRF's A-level predictions are near perfect, but significantly deteriorate in error levels B, C. As seen above, however, NeRF consistently (but not exclusively) attains level A only when trained with 100 views. We stress that in these simple scenes, this effect is not[7] due to incomplete scene coverage from a limited number of views—NeRF is simply not able to interpolate well between few training views. In contrast, NeRF-VAE captures the scene structure well on all error levels—though the simple version used here (MLP scene function, no iterative inference) misses high frequency details such as sharp object boundaries and textures. This suggests that while amortized inference from few views is possible, NeRF-VAE could benefit from a more expressive scene function and a better inference mechanism.

[7] Similar results, including catastrophic failures of NeRF trained with few views, can be obtained on CLEVR data where most of the scene is visible in all views, see Appendix F.1.

5.4 Comparison with a Convolution-Based Generative Model

Our next set of experiments compares NeRF-VAE to a CONV-AR-VAE—a model that is very closely related to GQN of Eslami et al. (2018), but with slight modifications to make it comparable to NeRF-VAE, see Appendix D for details.

Both NeRF-VAE and CONV-AR-VAE are able to infer novel scenes from few input images, and to subsequently synthesize novel views within those scenes. The key difference between the models is their decoder, responsible for rendering images given camera and latent scene representation: convolutional for CONV-AR-VAE and geometry informed for NeRF-VAE.

Interpolations & Generalization

We first illustrate that NeRF-VAE's explicit knowledge of 3D geometry allows it to generalize well to out-of-distribution camera positions and orientations compared to CONV-AR-VAE.

We train both NeRF-VAE and CONV-AR-VAE on the GQN dataset, using similar settings, which are detailed in Appendix F.2. For the this experiment, we restrict NeRF-VAE's scene function's access to orientations, which we will motivate below. For evaluation, we select a held-out scene and

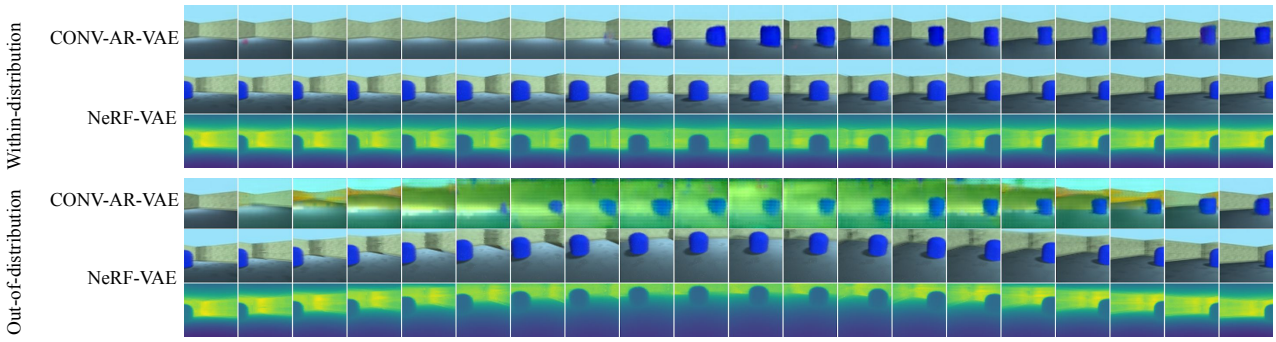


Figure 5: Generalization ability for interpolations along two trajectories. **Top**: the camera is moving parallel to the ground, facing the horizon (**within-distribution (WD)**, both models were trained using such views). While convolutional autoregressive VAE (CONV-AR-VAE) predicts the room consistently, it is inconsistent regarding the presence of the blue object. In contrast, NeRF-VAE produces fully consistent predictions, as further illustrated by visualizations of its inferred scene geometry. **Bottom**: the camera is further lifted off the ground, along a quadratic curve, facing downwards (**out-of-distribution (OOD)**). CONV-AR-VAE fails to account for downward camera orientations and outputs distorted colours, whereas NeRF-VAE produces plausible and consistent predictions and scene geometry.

let each model infer the latent representation given the same $N_{\text{ctx}}^{\text{test}} = 4$ views. We pick two arbitrary views and generate a sequence of camera positions and orientations that interpolates between them[8]. We do this twice: once with a simple linear interpolation across the plane, changing only the yaw of the camera (**within-distribution (WD) views**), and once with the camera lifting off the plane, along a quadratic curve, and looking down, changing both pitch and yaw (**out-of-distribution (OOD) views**); see Fig. 14 in Appendix F.2 for an illustration. We note that out-of-distribution here means that these camera positions lie outside the support of the training data distribution.

Fig. 5 shows each models' outputs along the within-distribution trajectories and out-of-distribution views trajectories. Both models produce plausible within-distribution interpolations, although CONV-AR-VAE has problems with object persistency. When evaluated on out-of-distribution views, CONV-AR-VAE completely fails to produce plausible outputs: ignoring the downward camera angle and instead distorting colours of the scene. In contrast, NeRF-VAE renders the inferred scene geometry properly from out-of-distribution viewpoints. Fig. 5 further shows depth estimates for NeRF-VAE's outputs, revealing the inferred scene geometry.

We now investigate generalization on CLEVR data, where we again focus on out-of-distribution camera views, and additionally investigate generalization across the number of visible objects. We compare CONV-AR-VAE to variants of NeRF-VAE, namely MLP, ATTN and II+ATTN.

We again see a clear advantage of NeRF-VAE. For instance, Fig. 6 (left) shows almost perfect predictions when using the ii+attn model as the camera gets closer to the object, compared to CONV-AR-VAE.

When evaluated on scenes that contain more objects than visible during training (here: 5 vs. 4, Fig. 6 (right)), NeRF-VAE using the attention-based scene function described in Section 3.2 correctly reconstructs all objects in the scenes, while the NeRF-VAE with the MLP scene function misses objects. Appendix F.2 contains quantitative results for this experiment.

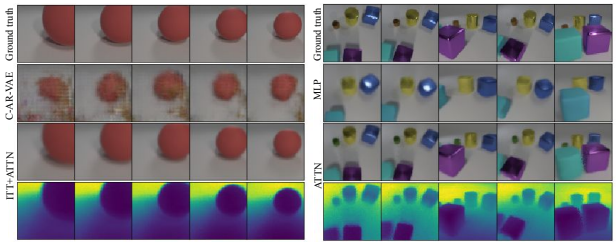


Figure 6: **Left:** Generalization to out-of-distribution camera views. While CONV-AR-VAE fails to produce plausible predictions, NeRF-VAE generalizes well. **Right:** Generalization to larger number of objects. The MLP scene function misses objects, while ATTN captures all objects.

Degenerate Orientations in the GQN data

Recall that camera views in the GQN data lie within a plane parallel to the ground. Consequently, certain points in the scene are **only** observed from orientations within this plane. When trained on this degenerate data, evaluating a neural-network-based scene function on different (e. g. when looking down) orientations inputs leads to unpredictable outputs. A simple way to circumvent the resulting rendering artifacts is to remove orientations from the scene function's inputs, as done in the above interpolations. In the case of the GQN data, which does not contain viewpoint dependent colours (reflections or specularities), this restriction does not impair visual output quality. We replicate the interpolations from

[8]Note that the GQN dataset does not contain ground truth images for these interpolated views.

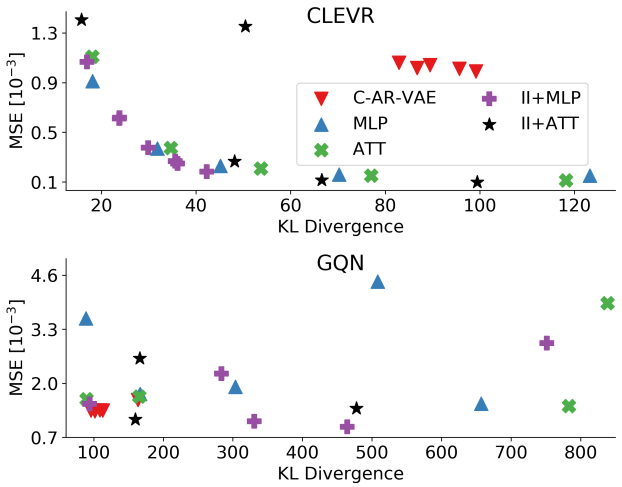


Figure 7: Variants of NeRF-VAE compared with CONV-AR-VAE on CLEVR and GQN datasets. We trained NeRF-VAE with a range of β values to investigate reconstruction/KL trade-offs. Iterative inference (II) improves MLP-based scene functions, but has a small effect on attentive (ATT) ones.

Fig. 5 with orientation inputs in Appendix F.3.

Quantitative Comparison & Model Ablations

Fig. 7 shows MSE and Kullback-Leibler (KL)-divergence of NeRF-VAE variants using amortized or iterative amortized (II) inference and MLP or attentive (ATT) scene functions, and juxtaposes them against the CONV-AR-VAE. We focus on CLEVR and GQN datasets. All NeRF-VAEs are trained with increasing values of β to trace out the available trade-offs between MSE and KL.

CLEVR contains simple objects visible from every viewpoint, and has no complicated textures. Consequently, the MLP scene function achieves good MSE and KL. Iterative inference further increases reconstruction performance while decreasing KL. It is interesting that MLP-based models tend to have lower KL values than attentive models while still maintaining good reconstruction. However, attentive models do obtain lower reconstruction errors in high-KL regimes, which suggests that they have higher capacity to model complicated data. CONV-AR-VAE is not able to model the CLEVR data well—the model attains high KL values despite manually setting a high MSE threshold.

The GQN dataset contains high-frequency textures and rooms which are not fully visible from every view, causing both a more difficult inference problem and more complicated rendering compared to CLEVR. The MLP scene function without iterative inference achieves low KL albeit generally higher MSE. ATT can achieve lower errors than MLP at similar KL levels, despite also reaching very large KL values, indicating a peaked posterior for certain values of β. Using iterative inference helps both decoders, and allows the model to achieve both lowest KL and MSE. We note that while CONV-AR-VAE achieves slightly lower MSE and KL on this within-distribution evaluation, it fails to generalize to out-of-distribution views as discussed above.

5.5 Samples & Uncertainty

We now demonstrate NeRF-VAE's capability to learn an unconditional prior over scenes (as opposed to images).

We first sample the latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, and then render a number of views of the induced scene function $G_\theta(\cdot, \boldsymbol{z})$. Samples from NeRF-VAE trained on the GQN and CLEVR datasets are shown in Fig. 8. These samples resemble the training data distribution to a high degree, both in appearance and variability. Furthermore, the depth estimates (example in last row) reveal a consistent geometric structure of the sampled scene.

Fig. 9 shows an example where NeRF-VAE infers a scene from a single context image containing a barely-visible pink object. The model is able to accurately predict a view that contains the full object. At the same time, the model maintains uncertainty about the exact shape (sphere vs. icosahedron), as can be seen in the predictive variance of depth estimate.

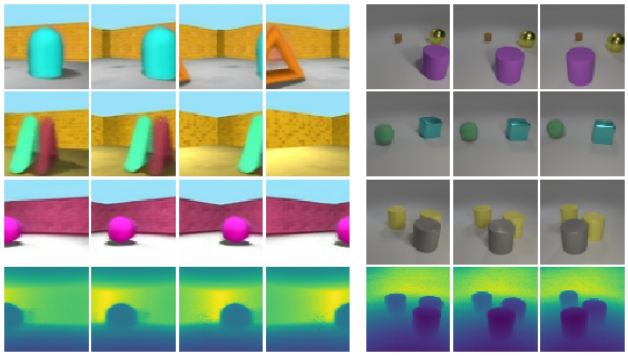


Figure 8: Sampled scenes from NeRF-VAE, trained on GQN (left) and CLEVR (right). Each row shows interpolated views in a different scene, corresponding to a sample of the latent variable. The last row shows a rendered depth map of the images in the above row (other depth maps omitted for space reasons).

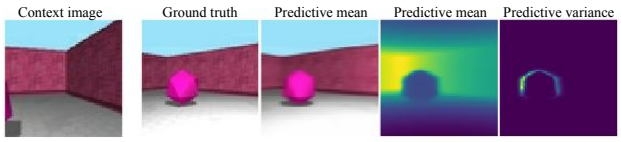


Figure 9: An example of predictions under limited information in the context view. We show colour and depth predictions. NeRF-VAE is able to reconstruct a plausible explanation of the object which is only marginally visible in the single context view. The predictive variance of depth estimates (averaged over 100 samples from the posterior over the latent variable) accounts for object boundaries not clearly visible in the context.

6 Discussion

We presented NeRF-VAE, a geometry-aware scene generative model that leverages NeRF as a decoder in a VAE-framework. Thanks to an explicit rendering procedure, NeRF-VAE is view-consistent and generalizes to out-of-distribution cameras, unlike convolutional models such as GQN. Additionally, the learned prior over scene functions allows NeRF-VAE to infer scene structure from very few views. This is in contrast to NeRF, where too few views may result in contrived explanations of a scene that work for some views but not for others. The combination of geometric structure and a prior over scene functions, however, is not a panacea: if the data is degenerate (e. g. cameras restricted to a plane) and the model is overparameterized (e. g. unnecessarily accounting for view-dependent colours), NeRF-VAE might still explain the data in implausible ways.

One of the limitations of NeRF-VAE is its limited per-scene expressivity. NeRF allocates **all** of its capacity to a single scene, and is able to capture high levels of complexity. NeRF-VAE, however, splits its capacity between shared across-scene information (conditional scene function) and per-scene information (latent). In order to allow for amortized inference, the capacity of the latent needs to be limited—resulting in reduced per-scene expressivity compared to NeRF.

At the same time, a low-dimensional latent variable opens interesting possibilities for future work, e. g. interpolation between scenes, extensions to dynamic scenes and videos, and a latent representation that dynamically grows with input complexity.

Acknowledgments

We thank Hyunjik Kim, Mehdi S. M. Sajjadi, Karl Stelzner, Adam Goliński, Alex X. Lee, Theophane Weber, and the anonymous reviewers for their comments on initial versions of this paper. We also thank Jony Hudson, Bojan Vujatovic, Yotam Doron, and Alex Goldin for their support throughout the project.

References

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Representation Learning*, 2019.

Cremer, C., Li, X., and Duvenaud, D. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, 2018. arXiv/1801.03558.

Curless, B. and Levoy, M. A volumetric method for building complex models from range images. In *Annual Conference on Computer graphics and Interactive Techniques*, pp. 303–312, 1996.

Dai, A., Nießner, M., Zollöfer, M., Izadi, S., and Theobalt, C. BundleFusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics*, 2017.

Dosovitskiy, A., Springenberg, J. T., Tatarchenko, M., and Brox, T. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

Du, Y., Zhang, Y., Yu, H.-X., Tenenbaum, J., and Wu, J. Neural radiance flow for 4d view synthesis and video processing. *arXiv/2012.09790*, 2020.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. In *International Conference on Representation Learning*, 2017. arXiv/1610.07629.

Engel, J., Schöps, T., and Cremers, D. LSD-SLAM: Large-scale direct monocular slam. In *European Conference on Computer Vision*, 2014.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. GENESIS: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2019.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, (6394):1204–1210, 2018. 360.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. arXiv/1912.01603.

Hartley, R. and Zisserman, A. *Multiple view geometry in computer vision (2. ed.)*. 2003.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016. arXiv/1603.05027.

Henderson, P. and Lampert, C. H. Unsupervised object-centric video generation and decomposition in 3d. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3106–3117. Curran Associates, Inc., 2020.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2011–2023, 2020.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Conference on Computer Vision and Pattern Recognition*, July 2017.

Kim, Y., Wiseman, S., Miller, A., Sontag, D., and Rush, A. M. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv/1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Levoy, M. and Hanrahan, P. Light field rendering. In *Annual Conference on Computer Graphics and Interactive Techniques*, pp. 31–42, 1996.

Li, Z., Niklaus, S., Snavely, N., and Wang, O. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv/2011.13084*, 2020.

Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pp. 91–110, 2004. 60.

Marino, J., Yue, Y., and Mandt, S. Iterative amortized inference. *arXiv/1807.09356*, 2018.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. NeRF in the Wild: Neural radiance fields for unconstrained photo collections. *arXiv/2008.02268*, 2020.

Mescheder, L. M., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *Conference on Computer Vision and Pattern Recognition*, pp. 4455–4465, 2019.

Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., and Kar, A. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., and Yang, Y. HoloGAN: Unsupervised learning of 3d representations from natural images. In *International Conference on Computer Vision*, pp. 7587–7596, 2019.

Nguyen-Phuoc, T., Richardt, C., Mai, L., Yang, Y., and Mitra, N. BlockGAN: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems*, 2020. arXiv/2002.08988.

Niemeyer, M. and Geiger, A. Giraffe: Representing scenes as compositional generative neural feature fields. *arXiv/2011.12100*, 2020.

Park, K., Sinha, U., Barron, J., Bouaziz, S., Goldman, D., Seitz, S., and Brualla, R.-M. Deformable neural radiance fields. *arXiv/2011.12948*, 2020.

Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F. D-NeRF: Neural radiance fields for dynamic scenes. *arXiv/ 2011.13961*, 2020.

Rezende, D. J. and Viola, F. Generalized elbo with constrained optimization, geco. In *NeurIPS Bayesian Deep Learning Workshop*, 2018.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Schönberger, J. L. and Frahm, J.-M. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. GRAF: Generative radiance fields for 3d-aware image synthesis. *arXiv/2007.02442*, 2020.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. arXiv/1906.01618.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., and Vaswani, A. Bottleneck transformers for visual recognition. 2021.

Srinivasan, P. P., Wang, T., Sreelal, A., Ramamoorthi, R., and Ng, R. Learning to synthesize a 4d RGBD light field from a single image. In *International Conference on Computer Vision*, 2017.

Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P. P., Barron, J., and Ng, R. Learned initializations for optimizing coordinate-based neural representations. *arXiv/2012.02189*, 2020.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. Single-view to multi-view: Reconstructing unseen views with a convolutional network. *arXiv/1511.06702*, 2015.

Trevithick, A. and Yang, B. GRF: Learning a general radiance field for 3d scene representation and rendering. In *International Conference on Learning Representations*, 2021. arXiv/2010.04595.

Ulusoy, A. O., Black, M. J., and Geiger, A. Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction. In *Conference on Computer Vision and Pattern Recognition*, pp. 3280–3289, 2016.

Vahdat, A. and Kautz, J. NVAE: A deep hierarchical variational autoencoder. *arXiv/2007.03898*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. 30.

Xian, W., Huang, J., Kopf, J., and Kim, C. Space-time neural irradiance fields for free-viewpoint video. *arXiv/2011.12950*, 2020.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. pixelNeRF: Neural radiance fields from one or few images. *arXiv/2012.02190*, 2020.

Zhou, T., Tucker, R., Flynn, J., Fyffe, G., and Snavely, N. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 2018.