Geometry-Free View Synthesis: Transformers and no 3D Priors

Robin Rombach* Patrick Esser* Björn Ommer
Ludwig Maximilian University of Munich & Heidelberg University, Germany
*Both authors contributed equally to this work. Code is available at https://git.io/JRPPs.

Figure 1. We present a probabilistic approach to Novel View Synthesis based on transformers, which does not require explicit 3D priors. Given a single source frame and a camera transformation (center), we synthesize plausible novel views that exhibit high fidelity (right). For comparison, SynSin [72] (left) yields uniform surfaces and unrealistic warps for large camera transformations.

Abstract

Is a geometric model required to synthesize novel views from a single image? Being bound to local convolutions, CNNs need explicit 3D biases to model geometric transformations. In contrast, we demonstrate that a transformer-based model can synthesize entirely novel views without any hand-engineered 3D biases. This is achieved by (i) a global attention mechanism for implicitly learning long-range 3D correspondences between source and target views, and (ii) a probabilistic formulation necessary to capture the ambiguity inherent in predicting novel views from a single image, thereby overcoming the limitations of previous approaches that are restricted to relatively small viewpoint changes. We evaluate various ways to integrate 3D priors into a transformer architecture. However, our experiments show that no such geometric priors are required and that the transformer is capable of implicitly learning 3D relationships between images. Furthermore, this approach outperforms the state of the art in terms of visual quality while covering the full distribution of possible realizations.

1. Introduction

Imagine looking through an open doorway. Most of the room on the other side is invisible. Nevertheless, we can estimate how the room *likely* looks. The few visible features enable an informed guess about the height of the ceiling, the position of walls and lighting etc. Given this limited information, we can then imagine several plausible realizations of the room on the other side. This 3D geometric reasoning and the ability to predict what the world will look like *before* we move is critical to orient ourselves in a world with three spatial dimensions. Therefore, we address the problem of novel view synthesis (NVS) [35, 23, 9] based on a single initial image and a desired change in viewpoint. In particular, we aim at specifically modeling *large* camera transformations, *e.g.* rotating the camera by 90° and looking at previously unseen scenery. As this is an underdetermined problem, we present a probabilistic generative model that learns the distribution of possible target images and synthesizes them at high fidelity. Solving this task has the potential to transform the passive experience of viewing images into an interactive, 3D exploration of the depicted scene. This requires an approach that both understands the geometry of the scene and, when rendering novel views of an input, considers their semantic relationships to the visible content.

Interpolation vs. Extrapolation Recently, impressive synthesis results have been obtained with geometry-focused approaches in the multi-view setting [54, 55, 44], where not just a single but a large number of images or a video of a scene are available such that the task is closer to a view interpolation than a synthesis of genuinely novel views. In contrast, if only a single image is available, the synthesis of novel views is always an extrapolation task. Solving this task is appealing because it allows a 3D exploration of a scene starting from only a single picture.

While existing approaches for single-view synthesis make small camera transformations, such as a rotation by a few degrees, possible, we aim at expanding the possible camera changes to include *large* transformations. The latter necessitates a probabilistic framework: Especially when applying large transformation, the problem is underdetermined because there are many possible target images which are consistent with the source image and camera pose. This task cannot be solved with a reconstruction objective alone, as it will either lead to averaging, and hence blurry synthesis results, or, when combined with an adversarial objective, cause a significant mode-dropping when modeling the target distribution. To remedy these issues, we propose to model this task with a powerful, autoregressive transformer, trained to maximize the likelihood of the target data.

Explicit vs. Implicit Geometry The success of transformers is often attributed to the fact that they enforce less inductive biases compared to convolutional neural networks (CNNs), which are biased towards local context. Relying mainly on CNNs, this locality-bias required previous approaches for NVS to explicitly model the overall geometric transformation, thereby enforcing yet another inductive bias regarding the three dimensional structure. In contrast, by modeling interactions between far-flung regions of source and target images, transformers have the potential to learn to represent the required geometric transformation implicitly without requiring such hand engineered operations. This raises the question whether it is at all necessary to explicitly include such biases in a transformer model. To address this question, we perform several experiments with varying degrees of inductive bias and find that our autoregressively trained transformer model is indeed capable of learning this transformation completely without built-in priors and can even learn to predict depth in an unsupervised fashion.

To summarize our contributions, we *(i)* propose to learn a probabilistic model for single view synthesis that properly takes into account the uncertainties inherent in the task and show that this leads to significant benefits over previous state-of-the-art approaches when modeling large camera transformations; see Fig. 1. We *(ii)* also analyze the need for explicit 3D inductive biases in transformer architectures for the task of NVS with large viewpoint changes and find that transformers make it obsolete to explicitly code 3D transformations into the model and instead can learn the required transformation implicitly themselves. We also *(iii)* find that the benefits of providing them geometric information in the form of explicit *depth maps* are relatively small, and investigate the ability to recover an explicit depth representation from the layers of a transformer which has learned to represent the geometric transformation implicitly and without any depth supervision.

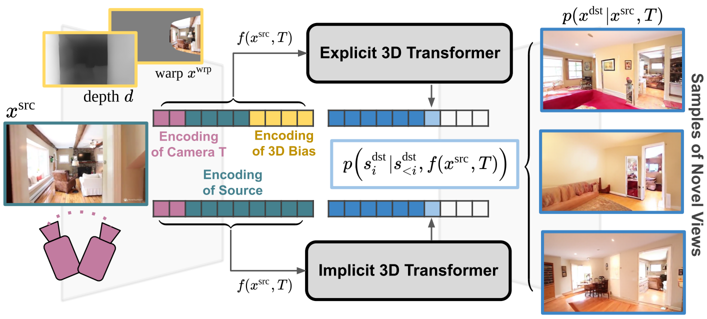
2. Related Work

Novel View Synthesis (NVS) We can identify three seminal works which illustrate different levels of reliance on geometry to synthesize novel views. [35] describes an approach which requires no geometric model, but requires a large number of structured input views. [23] describes a similar approach but shows that unstructured input views suffice if geometric information in the form of a coarse volumetric estimate is employed. [9] can work with a sparse set of views but requires an accurate photogrammetric model. Subsequent work also analyzed the commonalities and trade-offs of these approaches [5]. Ideally, an approach could synthesize novel views from a single image without having to rely on accurate geometric models of the scene and early works on deep learning for NVS explored the possibility to directly predict novel views [13, 14, 34, 75, 65] or their appearance flows [79, 48, 63] with convolutional neural networks (CNNs). However, results of these methods were limited to simple or synthetic data and subsequent works combined geometric approaches with CNNs.

Among these deep learning approaches that explicitly model geometry, we can distinguish between approaches relying on a proxy geometry to perform a warping into the target view, and approaches predicting a 3D representation that can subsequently be rendered in novel views. For the proxy geometry, [43] relies on point clouds obtained from structure from motion (SfM) [1, 57] and multi-view stereo (MVS) [58, 18]. To perform the warping, [17, 74] use plane-sweep volumes, [32] estimates depth at novel views and [8, 73] a depth probability volume. [54, 55] post-process MVS results to a global mesh and [26] relies on per-view meshes [27]. Other approaches learn 3D features per scene, which are associated with a point cloud [2] or UV maps [67], and decoded to the target image using a CNN. However, all of these approaches rely on multi-view inputs to obtain an estimate for the proxy geometry.

Approaches which predict 3D representations mainly utilize layered representations such as layered depth images (LDIs) [59, 24, 25], multi-plane images (MPIs) [64, 78, 61, 16] and variants thereof [49, 36]. While this allows an efficient rendering of novel views from the obtained representations, their layered nature limits the range of novel views that can be synthesized with them. Another emerging approach [44] represents a five dimensional light field directly with a multi-layer-perceptron (MLP), but still requires a large number of input views to correctly learn this MLP.

In the case of NVS from a single view, SfM approaches cannot be used to estimate proxy geometries and early works relied on human interaction to obtain a scene model [29]. [62] uses a large scale, scene-specific light field dataset to learn CNNs which predict light fields from a single image. [39] assumes that scenes can be represented by a fixed set of planar surfaces. To handle more general scenes, most methods rely on monocular depth estimation [52, 19, 20, 77, 21] to predict warps [46, 72, 38] or LDIs [10, 33, 60]. [68] directly predicts an MPI, and [30] a mesh. To handle disoc-



variant	explicit warp	requires depth	warped features	$f(x^{\text{src}}, T)$
expl.-img	✓	✓	x^{src}	Eq.(9)
expl.-feat	✓	✓	$E(x^{\text{src}})$	Eq.(10)
expl.-emb	✓	✓	$e(E(x^{\text{src}}))$, e^{pos}	Eq.(11)
impl.-catdepth	✗	✓	–	Eq.(12)
impl.-depth	✗	✓	–	Eq.(13)
impl.-nodepth	✗	✗	–	Eq.(14)
hybrid	✓	✓	$e(E(x^{\text{src}}))$, e^{pos}	Eq.(11)+Eq.(13)

Figure 2. We formulate novel view synthesis as sampling from the distribution $p(x^{\text{dst}}|x^{\text{src}}, T)$ of target images x^{dst} for a given source image x^{src} and camera change T. We use a VQGAN to model this distribution autoregressively with a transformer and introduce a conditioning function $f(x^{\text{src}}, T)$ to encode inductive biases into our model. We analyze explicit variants, which estimate scene depth d and warp source features into the novel view, as well as implicit variants without such a warping. The table on the right summarizes the variants for f.

clusions, most of these methods rely on adversarial losses, inspired by generative adversarial networks (GANs) [22], to perform inpainting in these regions. However, the quality of these approaches quickly degrades for larger viewpoint changes because they do not model the uncertainty of the task. While adversarial losses can remedy an averaging effect over multiple possible realizations to some degree, our empirical results demonstrate the advantages of properly modeling the probabilistic nature of NVS from a single image.

Self-Attention and Transformers The *transformer* [70] is a sequence-to-sequence model that models interactions between learned representations of sequence elements by the so-called attention mechanism [4, 47]. Importantly, this mechanism does not introduce locality biases such as those present in *e.g*. CNNs, as the importance and interactions of sequence elements are weighed regardless of their relative positioning. We build our autogressive transformer from the GPT-2 architecture [50], *i.e*. multiple blocks of multihead self-attention, layer norm [3] and position-wise MLP.

Generative Two Stage Approaches Our approach is based on work in conditional generative modeling combined with neural discrete representation learning (VQVAE) [69]. The latter aims to learn discrete, compressed representations through either vector quantization or soft relaxation of the discrete assignment [42, 31]. This training paradigm provides a suitable space [56, 12, 7] to train (conditional) autoregressive likelihood models on the latent representations and has been utilized to train generative models for hierarchical, class-conditional image synthesis [53], text-controlled image synthesis [51] and music generation [11]. Recently, [15] demonstrated that adversarial training of the VQVAE improves compression while retaining high-fidelity reconstructions, subsequently enabling efficient training of an autoregressive transformer model on the learned latent space (yielding a so-called VQGAN). We directly build on this work and use VQGANs to represent both source and target views and, when needed, depth maps.

3. Approach

To render a given image x^{src} experienceable in a 3D manner, we allow the specification of arbitrary new viewpoints, including in particular *large* camera transformations T. As a result we expect multiple plausible realizations x^{dst} for the novel view, which are all consistent with the input, since this problem is highly underdetermined. Consequently, we follow a probabilistic approach and sample novel views from the distribution

$$x^{\text{dst}} \sim p(x^{\text{dst}}|x^{\text{src}}, T). \tag{1}$$

To solve this task, a model must explicitly or implicitly learn the 3D relationship between both images and T. In contrast to most previous work that tries to solve this task with CNNs and therefore oftentimes includes an explicit 3D transformation, we want to use the expressive transformer architecture and investigate to what extent the explicit specification of such a 3D model *is necessary at all*.
Sec. 3.1 describes how to train a transformer model in the latent space of a VQGAN. Next, Sec. 3.2 shows how inductive biases can be build into the transformer and describes all bias-variants that we analyze. Finally, Sec. 3.3 presents our approach to extract geometric information from a transformer where no 3D bias has been explicitly specified.

3.1. Probabilistic View Synthesis in Latent Space

Learning the distribution in Eq. (1) requires a model which can capture long-range interactions between source and target view to implicitly represent geometric transformations. Transformer architectures naturally meet these requirements, since they are not confined to short-range relations such as CNNs with their convolutional kernels and exhibit state-of-the-art performance [70]. Since likelihood-based models have been shown [56] to spend too much capacity on short-range interactions of pixels when modeling images directly in pixel space, we follow [15] and employ a two-stage training. The first stage performs adversarially guided discrete representation learning (VQGAN), obtaining an abstract latent space that has proved to be well-suited for efficiently training generative transformers [15].

Modeling Conditional Image Likelihoods VQGAN consists of an encoder E, a decoder G and a codebook $\mathcal{Z} = \{z_i\}_{i=1}^{|\mathcal{Z}|}$ of discrete representations $z_i \in \mathbb{R}^{d_z}$. The trained VQGAN allows to encode any $x \in \mathbb{R}^{H\times W\times 3}$ into the discrete latent space as $E(x) \in \mathbb{R}^{h\times w\times d_z}$[1]. Unrolled in raster-scan order, this latent representation corresponds to a sequence $s \in \mathbb{R}^{h\cdot w\times d_z}$ and can be equivalently expressed as a sequence of integers which index the learned codebook $\mathcal{Z}$. Following the usual designation [70] we refer to the sequence elements as "tokens". An embedding function $g = g(s) \in \mathbb{R}^{h\cdot w\times d_e}$ maps each of these tokens into the embedding space of the transformer $\mathcal{T}$ and adds learnable positional encodings. Similarly, to encode the input view x^{src} and the camera transformation T, both are mapped into the embedding space by a function f:

$$f : (x^{\text{src}}, T) \mapsto f(x^{\text{src}}, T) \in \mathbb{R}^{n\times d_e}, \tag{2}$$

where n denotes the length of the conditioning sequence. By using different functions f various inductive biases can be incorporated into the architecture as described in Sec. 3.2. The transformer $\mathcal{T}$ then processes the concatenated sequence $[f(x^{\text{src}}, T), g(s^{\text{dst}})]$ to learn the distribution of plausible novel views conditioned on x^{src} and T,

$$p_{\mathcal{T}}\Big(s^{\text{dst}}|f(x^{\text{src}}, T)\Big) = \prod_i p_{\mathcal{T}}\Big(s_i^{\text{dst}}|s_{<i}^{\text{dst}}, f(x^{\text{src}}, T)\Big). \tag{3}$$

Hence, to train an autoregressive transformer by next-token prediction $p_{\mathcal{T}}(s_i|s_{<i}, f(x^{\text{src}}, T))$ we maximize the log-likelihood of the data, leading to the training objective

$$\mathcal{L}_{\mathcal{T}} = \mathbb{E}_{x^{\text{src}},x^{\text{dst}}\sim p(x^{\text{src}},x^{\text{dst}})}\Big[-\log p_{\mathcal{T}}\big(s^{\text{dst}}|f(x^{\text{src}}, T)\big)\Big]. \tag{4}$$

3.2. Encoding Inductive Biases

Besides achieving high-quality NVS, we aim to investigate to what extent transformers depend on a 3D inductive bias. To this end, we compare approaches where a geometric transformation is built *explicitly* into the conditioning function f, and approaches where no such transformation is used. In the latter case, the transformer itself must learn the required relationship between source and target view. If successful, the transformation will be described *implicitly* by the transformer.

Geometric Image Warping We first describe how an explicit geometric transformation results from the 3D relation of source and target images. For this, pixels of the source image are back-projected to three dimensional coordinates, which can then be re-projected into the target view. We assume a pinhole camera model, such that the projection of 3D points to homoegenous pixel coordinates is determined through the intrinsic camera matrix K. The transformation between source and target coordinates is given by a rigid motion, consisting of a rotation R and a translation t. Together, these parameters specify the desired control over the novel view to be generated, *i.e.* $T = (K, R, t)$.

To project pixels back to 3D coordinates, we require information about their depth d, since this information has been discarded by their projection onto the camera plane. Since we assume access to only a single source view, we require a monocular depth estimate. Following by previous works [60, 38], we use MiDaS [52] in all of our experiments which require monocular depth information.

The transformation can now be described as a mapping of pixels $i \in \{1, \dots, H\}, j \in \{1, \dots, W\}$ in the source image $x^{\text{src}} \in \mathbb{R}^{H\times W\times 3}$ to pixels i', j' in the target image. In homogeneous coordinates, their relationship is given by

$$\begin{pmatrix} j' \\ i' \\ 1 \end{pmatrix} \simeq K\left(RK^{-1}d(i,j)\begin{pmatrix} j \\ i \\ 1 \end{pmatrix} + t\right) \tag{5}$$

This relationship defines a forward flow field $F^{\text{src}\to\text{dst}} = F^{\text{src}\to\text{dst}}(K, R, t, d) \in \mathbb{R}^{H\times W\times 3}$ from source to target as a function of depth and camera parameters. The flow field can then be used to warp the source image x^{src} into the target view with a warping operation $\mathcal{S}$:

$$x^{\text{wrp}} = \mathcal{S}(F^{\text{src}\to\text{dst}}, x^{\text{src}}). \tag{6}$$

Because the target pixels obtained from the flow are not necessarily integer valued, we follow [45] and implement $\mathcal{S}$ by bilinearly splatting features across the four closest target pixels. When multiple source pixels map to the same target pixels, we use their relative depth to give points closer to the camera more weight—a soft variant of z-buffering.

In the simplest case, we can now describe the difference between explicit and implicit approaches in the way that they receive information about the source image and the desired target view. Here, explicit approaches receive source information warped using the camera parameters, whereas implicit approaches receive the original source image and the camera parameters themselves, *i.e.*

$$\textit{explicit:} \quad \mathcal{S}(F^{\text{src}\to\text{dst}}(K, R, t, d), x^{\text{src}}) \tag{7}$$

$$\textit{implicit:} \quad (K, R, t, d, x^{\text{src}}) \tag{8}$$

Thus, in explicit approaches we enforce an inductive bias on the 3D relationship between source and target by making this relationship explicit, while implicit approaches have to learn it on their own. Next, we introduce a number of different variants for each, which are summarized in Fig. 2.

Explicit Geometric Transformations In the following, we describe all considered variants in terms of the transformer's conditioning function f. Furthermore, e denotes a

[1] This includes the vector quantization step as described in [69]

Table 1. To assess the effect of encoding different degrees of 3D prior knowledge, we evaluate all variants on RealEstate and ACID using negative log-likelihood (NLL), FID [28] and PSIM [76], PSNR and SSIM [71]. We highlight **best**, **second best** and third best scores.

	RealEstate10K					ACID				
method	FID ↓	NLL ↓	PSIM ↓	SSIM ↑	PSNR ↑	FID ↓	NLL ↓	PSIM ↓	SSIM ↑	PSNR ↑
impl.-nodepth	**48.59**	4.956	3.17 $\pm$0.43	0.42 $\pm$0.13	12.16 $\pm$2.54	**42.88**	5.365 $\pm$0.007	2.90 $\pm$0.53	0.40 $\pm$0.15	15.17 $\pm$3.40
hybrid	**48.84**	4.913	**3.09** $\pm$0.46	**0.44** $\pm$0.13	**12.51** $\pm$2.69	44.47	**5.341** $\pm$0.008	**2.83** $\pm$0.54	0.41 $\pm$0.15	**15.54** $\pm$3.52
impl.-depth	49.15	**4.836**	**3.05** $\pm$0.46	**0.44** $\pm$0.13	**12.66** $\pm$2.68	**42.93**	5.353 $\pm$0.011	**2.86** $\pm$0.52	0.41 $\pm$0.15	15.53 $\pm$3.34
expl.-img	49.63	4.924	3.18 $\pm$0.46	0.43 $\pm$0.13	12.11 $\pm$2.66	47.72	5.414 $\pm$0.006	3.00 $\pm$0.51	0.40 $\pm$0.14	14.83 $\pm$3.20
impl.-catdepth	50.04	**4.860**	3.10 $\pm$0.45	0.43 $\pm$0.13	12.43 $\pm$2.66	47.44	**5.350** $\pm$0.004	2.86 $\pm$0.55	**0.42** $\pm$0.15	**15.54** $\pm$3.57
expl.-emb	50.35	5.004	3.15 $\pm$0.45	0.43 $\pm$0.13	12.30 $\pm$2.66	47.08	5.416 $\pm$0.007	2.88 $\pm$0.54	**0.42** $\pm$0.15	15.45 $\pm$3.61
expl.-feat	54.82	5.159	3.31 $\pm$0.43	0.41 $\pm$0.13	11.75 $\pm$2.58	52.65	5.657 $\pm$0.003	3.14 $\pm$0.52	0.38 $\pm$0.15	14.06 $\pm$3.28

learnable embedding mapping the discrete VQGAN codes $E(x)$ into the embedding space of the transformer. Similarly, $e^{\text{pos}} \in \mathbb{R}^{n \times d_e}$ denotes a learnable positional encoding. The flow field $F^{\text{src}\to\text{dst}}(K, R, t, d)$ is always computed from x^{src} and, to improve readability, we omit it from the arguments of the warping operation, *i.e.* $\mathcal{S}(\cdot) = \mathcal{S}(F^{\text{src}\to\text{dst}}(K, R, t, d), \cdot)$.

(I) Our first explicit variant, *expl.-img*, warps the source image and encodes it in the same way as the target image:

$$f(x^{\text{src}}, T) = e(E(\mathcal{S}(x^{\text{src}}))) + e^{\text{pos}} \tag{9}$$

(II) Inspired by previous works [54, 2] we include a *expl.-feat* variant which first encodes the original source image, and subsequently applies the warping on top of these features. We again use the VQGAN encoder E to obtain

$$f(x^{\text{src}}, T) = e(\mathcal{S}(E(x^{\text{src}}))) + e^{\text{pos}} \tag{10}$$

(III) To account for the fact that the warped features in Eq. (10) remain fixed (due to E being frozen), we also consider a *expl.-emb* variant that warps the *learnable* embeddings and positional encodings of the transformer model. More precisely, we concatenate original embeddings with their warped variants and merge them with a learnable matrix. Doing this for both the embeddings of the codes and for the positional encodings using matrices $W^{\text{emb}}, W^{\text{pos}} \in \mathbb{R}^{d_e \times 2 \cdot d_e}$, the conditioning function f then reads:

$$\begin{aligned} f(x^{\text{src}}, T) =& W^{\text{emb}}[e(E(x^{\text{src}})), \mathcal{S}(e(E(x^{\text{src}})))] + \\ & W^{\text{pos}}[e^{\text{pos}}, \mathcal{S}(e^{\text{pos}})] \end{aligned} \tag{11}$$

Implicit Geometric Transformations Next, we describe implicit variants that we use to analyze if transformers—with their ability to attend to all positions equally well—require an explicit geometric transformation built into the model. We use the same notation as for the explicit variants.

(IV) The first variant, *impl.-catdepth*, provides the transformer with all the same components which are used in the explicit variants: Camera parameters K, R, t, estimated depth d and source image x^{src}. Camera parameters are flattened and concatenated to $\hat{T}$, which is mapped via $W^{\text{cam}} \in \mathbb{R}^{d_e \times 1}$ to the embedding space. Depth and source images are encoded by VQGAN encoders E^{d} and E to obtain

$$f(x^{\text{src}}, T) = [W^{\text{cam}}\hat{T}, e(E^{\text{d}}(d)), e(E(x^{\text{src}}))] + e^{\text{pos}} \tag{12}$$

Compared to the other variants, this sequence is roughly $\frac{3}{2}$ times longer, resulting in twice the computational costs.

(V) Therefore, we also include a *impl.-depth* variant, which concatenates the discrete codes of depth and source image, and maps them with a matrix $W \in \mathbb{R}^{d_e \times 2 \cdot d_z}$ to the embedding space to avoid an increase in sequence length:

$$f(x^{\text{src}}, T) = [W^{\text{cam}}\hat{T}, W[E^{\text{d}}(d), E(x^{\text{src}})]] + e^{\text{pos}} \tag{13}$$

(VI) Implicit approaches offer an intriguing possibility: Because they do not need an explicit estimate of the depth to perform the warping operation $\mathcal{S}$, they hold the potential to solve the task without such a depth estimate. Thus, *impl.-nodepth* uses only camera parameters and source image—the bare minimum according to our task description.

$$f(x^{\text{src}}, T) = [W^{\text{cam}}\hat{T}, e(E(x^{\text{src}}))] + e^{\text{pos}} \tag{14}$$

(VII) Finally, we analyze if explicit and implicit approaches offer complementary strengths. Thus, we add a *hybrid* variant whose conditioning function is the sum of the f's of *expl.-emb* in Eq. (11) and *impl.-depth* in Eq. (13).

3.3. Depth Readout

To investigate the ability to learn an implicit model of the geometric relationship between different views, we propose to extract an explicit estimate of depth from a trained model. To do so, we use linear probing [6], which is commonly used to investigate the feature quality of unsupervised approaches. More specifically, we assume a transformer model consisting of L layers and of type *impl.-nodepth*, which is conditioned on source frame and transformation parameters only. Next, we specify a certain layer $0 \leq l \leq L$ (where $l = 0$ denotes the input) and extract its latent representation e^l, corresponding to the positions of the provided source frame x^{src}. We then train a position-wise linear classifier W to predict the discrete, latent representation of the depth-encoder E^{d} (see Sec. 3.2) via a cross-entropy objective from e^l. Note that both the weights of the transformer and the VQGANs remain fixed.

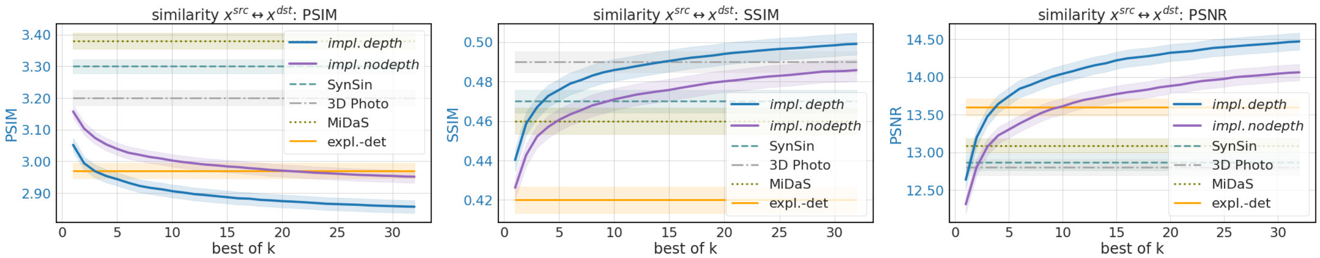


Figure 3. Average reconstruction error of the best sample as a function of the number of samples on RealEstate. With just four samples, *impl.-depth* reaches state-of-the-art performance in two out of three metrics, and with 16 samples in all three of them.

4. Experiments

First, Sec. 4.1 integrates the different explicit and implicit inductive biases into the transformer to judge if such geometric biases are needed at all. Following up, Sec. 4.2 compares implicit variants to previous work and evaluates both the visual quality and fidelity of synthesized novel views. Finally, we evaluate the ability of the least biased variant, *impl.-nodepth*, to implicitly represent scene geometry, observing that they indeed capture such 3D information.

4.1. Comparing Implicit and Explicit Transformers

To investigates if transformers need (or benefit from) an explicit warping between source and target view we first compare how well the different variants from Sec. 3.2 (see also Fig. 2) can learn a probabilistic model for NVS. We then evaluate both the quality and fidelity of their samples.

To prepare, we first train VQGANs on frames of the RealEstate10K [78] and ACID [38] datasets, whose preparation is described in the supplementary. We then train the various transformer variants on the latent space of the respective first stage models. Note that this procedure ensures comparability of different settings within a given dataset, as the space in which the likelihood is measured remains fixed.

Comparing Density Estimation Quality A basic measure for the performance of probabilistic models is the likelihood assigned to validation data. Hence, we begin our evaluation of the different variants by comparing their (minimal) negative log-likelihood (NLL) on RealEstate and ACID. Based on the results in Tab. 1, we can identify three groups with significant performance differences on ACID: The implicit variants *impl.-catdepth*, *impl.-depth*, and *impl.-nodepth* and *hybrid* achieve the best performance, which indicates an advantage over the purely explicit variants. Adding an explicit warping as in the *hybrid* model does not help significantly.

Moreover, *expl.-feat* is unfavorable, possibly due to the features $E(x^{\text{src}})$ remaining fixed while training the transformer. The *learnable* features which are warped in variant *expl.-emb* obtain a lower NLL and thereby confirm the former hypothesis. Still there are no improvements of warped features over warped pixels as in variant *expl.-img*.

The results on RealEstate look similar but in this case the implicit variant without depth, *impl.-nodepth*, performs a bit worse than *expl.-img*. Presumably, accurate depth information obtained from a supervised, monocular depth estimation model are much more beneficial in the indoor setting of RealEstate compared to the outdoor setting of ACID.

Visualizing Entropy of Predictions The NLL measures the ability of the transformer to predict target views. The entropy of the predicted distribution over the codebook entries for each position captures the prediction uncertainty of the model. See Fig. 4 for a visualization of variant *impl.-nodepth*. The model is more confident in its predictions for regions which are visible in the source image. This indicates that it is indeed able to relate source and target via their geometry instead of simply predicting an arbitrary novel view.

Measuring Image Quality and Fidelity Since NLL does not necessarily reflect the visual quality of the images [66], we evaluate the latter also directly. Comparing predictions with ground-truth helps to judge how well the model respects the geometry. However, for large camera movements, large parts of the target image are not visible in the source view. Thus, we must also evaluate the quality of the content imagined by the model, which might be fairly different from that of the ground-truth, since the latter is just one of many possible realizations of the real-world.

To evaluate the image quality without a direct comparison to the ground-truth, we report FID scores [28]. To evaluate the fidelity to the ground-truth, we report the low-level similarity metrics SSIM [71] and PSNR, and the high-level similarity metric PSIM [76], which better represents human assessments of visual similarity. Tab. 1 contains the results for RealEstate10K and ACID. In general, they reflect the findings from the NLL values: Image quality and fidelity of

Figure 4. Visualization of the entropy of the predicted target code distribution for *impl.-nodepth*. Increased confidence (darker colors) in regions which are visible in the source image indicate its ability to relate source and target geometrically, without 3D bias.

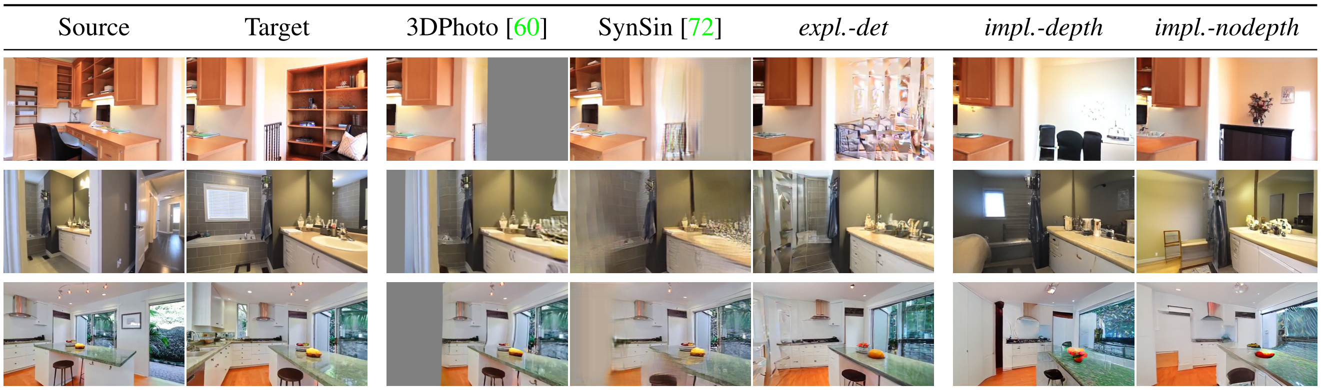


Figure 5. **Qualitative Results on RealEstate10K:** We compare three deterministic convolutional baselines (3DPhoto [60], SynSin [72], *expl.-det*) to our implicit variants *impl.-depth* and *impl.-nodepth*. Ours is able to synthesize plausible novel views, whereas others produce artifacts or blurred, uniform areas. The depicted target is only one of many possible realizations; we visualize samples in the supplement.

Table 2. Quantitative comparison on RealEstate. Reconstruction metrics are reported with 32 samples, see Fig. 3 for other values.

method	FID ↓	IS ↑	PSIM ↓	SSIM ↑	PSNR ↑
impl.-nodepth	**48.59**	4.24 ± 0.30	**2.95** ± 0.43	**0.49** ± 0.12	**14.06** ± 2.41
impl.-depth	**49.15**	4.17 ± 0.52	**2.86** ± 0.45	**0.50** ± 0.12	**14.47** ± 2.51
expl.-det	66.66	4.47 ± 0.49	2.97 ± 0.55	0.42 ± 0.15	13.60 ± 2.56
3DPhoto [60]	85.43	**5.10** ± 0.39	3.20 ± 0.54	0.49 ± 0.12	12.80 ± 2.33
SynSin [72]	113.88	3.70 ± 0.30	3.30 ± 0.51	0.47 ± 0.13	12.87 ± 2.46
MiDaS [52]	132.13	**5.63** ± 0.77	3.38 ± 0.56	0.46 ± 0.15	13.09 ± 2.16

implicit variants with access to depth are superior to explicit variants. The implicit variant without depth (*impl.-nodepth*) consistently achieves the same good FID scores as the implicit variants with depth (*impl.-catdepth* & *impl.-depth*), but cannot achieve quite the same level of performance in terms of reconstruction fidelity. However, it is on par with the explicit variants, albeit requiring no depth supervision.

4.2. Comparison to Previous Approaches

Next, we compare our best performing variants *impl.-depth* and *impl.-nodepth* to previous approaches for NVS: 3DPhoto [60], SynSin [72] and InfNat [38]. 3DPhoto [60] has been trained on MSCOCO [37] to work on arbitrary scenes, whereas SynSin [72] and InfNat [38] have been trained on RealEstate and ACID, respectively.

To assess the effect of formulating the problem probabilistically, we introduce another baseline to compare probabilistic and deterministic models with otherwise equal architectures. Specifically, we use the same VQGAN architecture as described in Sec. 3.1. However, it is not trained as an autoencoder, but instead the encoder receives the warped source image x^{wrp}, and the decoder predicts the target image x^{dst}. This model, denote by *expl.-det*, represents an *explicit and deterministic* baseline. Finally, we include the warped source image itself as a baseline denoted by MiDaS [52].

Utilizing the probabilistic nature of our model, we analyze how close we can get to a particular target image with a fixed amount of samples. Tab. 2 and 3 report the reconstruction metrics with 32 samples per target. The probabilistic variants consistently achieve the best values for the similarity metrics PSIM, SSIM and PSNR on RealEstate, and are always among the best three on ACID, where *expl.-det* achieves the best PSIM values and the second best PSNR values. We show the reconstruction metrics on RealEstate as a function of the number of samples in Fig. 3. With just four samples, the performance of *impl.-depth* is better than all other approaches except for the SSIM values of 3DPhoto [60], which are overtaken by *impl.-depth* with 16 samples, and do not saturate with 32 samples, which demonstrates the advantages of a probabilistic formulation of NVS.

Table 3. Quantitative comparison on ACID using 32 samples for reconstruction metrics. (n): number of steps used for InfNat [38].

method	FID ↓	IS ↑	PSIM ↓	SSIM ↑	PSNR ↑
impl.-nodepth	**42.88**	2.63 ± 0.14	2.77 ± 0.54	**0.46** ± 0.14	16.49 ± 3.33
impl.-depth	**42.93**	2.62 ± 0.23	**2.73** ± 0.53	**0.46** ± 0.14	**16.80** ± 3.24
expl.-det	53.77	2.60 ± 0.18	**2.72** ± 0.56	0.41 ± 0.16	**16.60** ± 3.43
InfNat [38](5)	76.07	2.44 ± 0.21	3.28 ± 0.47	0.39 ± 0.15	15.24 ± 2.87
3DPhoto [60]	76.17	**3.50** ± 0.47	3.01 ± 0.64	0.45 ± 0.14	14.87 ± 3.08
InfNat [38](1)	79.00	2.71 ± 0.23	3.11 ± 0.58	0.42 ± 0.15	15.35 ± 3.50
InfNat [38](10)	88.81	2.52 ± 0.20	3.44 ± 0.41	0.35 ± 0.14	14.32 ± 2.55
MiDaS [52]	106.10	**3.62** ± 0.36	3.11 ± 0.68	0.45 ± 0.15	14.82 ± 2.85

These results should be considered along with the competitive FID scores in Tab. 2 and 3 (where the implicit variants always constitute the best and second best value) and the qualitative results in Fig. 5 and 6, underlining the high quality of our synthesized views. It is striking that IS assigns the best scores to 3DPhoto [60] and MiDaS [52], which contain large and plain regions of gray color in regions where the source image does not provide information about the content. Where the monocular depth estimation is accurate, 3DPhoto [60] shows good results but it can only inpaint small areas. SynSin [72] and InfNat [38] can fill larger areas but, for large camera motions, their results become blurry and a similar observation holds for *expl.-det*. The probabilistic variants *impl.-depth* and *impl.-nodepth* consistently produce plausible results which are largely consistent with the source image, although small details sometimes differ. This shows that only the probabilistic variants are able to synthesize high quality images for large camera changes.

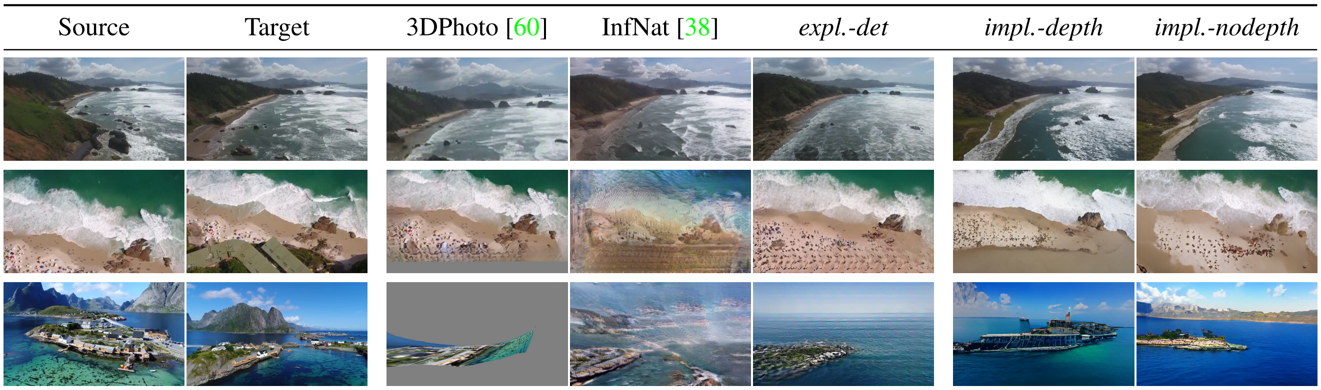


Figure 6. **Qualitative Results on ACID:** The outdoor setting of the ACID dataset yields similar results as the indoor setting in Fig. 5. Here, we evaluate against the baselines 3DPhoto [60], InfNat [38] and *expl.-det*. For InfNat [38], we use 5 steps to synthesize a novel view.

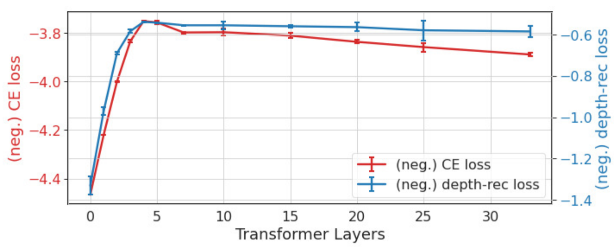


Figure 7. Minimal validation loss and reconstruction quality of depth predictions obtained from linear probing as a function of different transformer layers. The probed variant is *impl.-nodepth*.

4.3. Probing for Geometry

Based on the experiments in Sec. 4.1 and Sec. 4.2, which showed that the unbiased variant *impl.-nodepth* is mostly on-par with the others, we investigate the question whether this model is able to develop an implicit 3D "understanding" without explicit 3D supervision. To do so, we perform linear probing experiments as described in Sec. 3.3.

Fig. 7 plots the negative cross-entropy loss and the negative PSIM reconstruction error of the recovered depth maps against the layer depth of the transformer model. Both metrics are consistent and quickly increase when probing deeper representations of the transformer model. Furthermore, both curves exhibit a peak for $l = 4$ (*i.e.* after the third self-attention block) and then slowly decrease with increasing layer depth. The depth maps obtained from this linear map resemble the corresponding true depth maps qualitatively well as shown in Fig. 7. This figure demonstrates that a linear estimate of depth only becomes possible through the representation learned by the transformer ($l = 4$) but not by the representation of the VQGAN encoder ($l = 0$). We hypothesize that, in order to map an input view onto a target view, the transformer indeed develops an implicit 3D representation of the scene to solve its training task.

5. Discussion

We have introduced a probabilistic approach based on transformers for novel view synthesis from a single source image with strong changes in viewpoint. Comparing various explicit and implicit 3D inductive biases for the transformer showed that explicitly using a 3D transformation in the architecture does not help their performance significantly. However, removing inductive biases also comes at a price. Without priors on camera movements or warping layers, the architecture must be able to take relationships between arbitrary positions into account, which requires a compressed representation. In our experiments, compression artifacts dominate the error for small viewpoint changes. Avoiding them increases computational costs (see Sec. E). Synthesizing two views from the same image generally results in two incompatible realizations. However, we can run our approach iteratively. When synthesizing continuous trajectories, sampling still leads to flickering but this can be alleviated with deterministic sampling (see Sec. A).

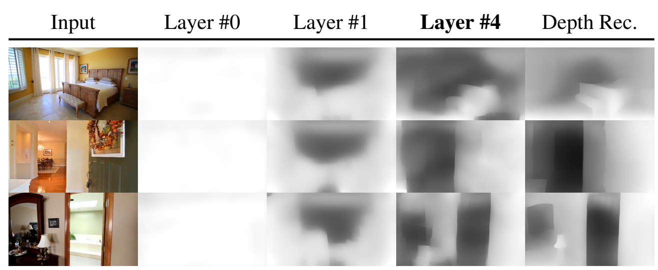


Figure 8. Linearly probed depth maps for different transformer layers. The results mirror the curve in Fig. 7: After a strong initial increase, the quality for layer 4 is best. The depth reconstructions in the right column provide an upper bound on achievable quality.

To conclude, our approach is not a final solution to novel view synthesis, but an important step towards synthesizing large camera changes and understanding the need for 3D priors. Our results demonstrate significant improvements over existing approaches, and even with no depth information as input our model learns to infer depth within its internal representations. Future works should explore how to combine these capabilities and insights with improved performance at synthesizing stable high-resolution trajectories.

This work has been supported by the German Research Foundation (DFG) projects 371923335 and 421703927.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, pages 72–79. IEEE Computer Society, 2009. 2

[2] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor S. Lempitsky. Neural point-based graphics. In *ECCV (22)*, volume 12367 of *Lecture Notes in Computer Science*, pages 696–712. Springer, 2020. 2, 5

[3] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 3

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 3

[5] Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432. ACM, 2001. 2

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 2020. 5

[7] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *ICLR (Poster)*. OpenReview.net, 2017. 3

[8] Inchang Choi, Orazio Gallo, Alejandro J. Troccoli, Min H. Kim, and Jan Kautz. Extreme view synthesis. In *ICCV*, pages 7780–7789. IEEE, 2019. 2

[9] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20. ACM, 1996. 1, 2

[10] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognit. Lett.*, 125:333–340, 2019. 2

[11] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *CoRR*, abs/2005.00341, 2020. 3

[12] Sander Dieleman. Musings on typicality, 2020. 3

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, pages 1538–1546. IEEE Computer Society, 2015. 2

[14] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):692–705, 2017. 2

[15] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020. 3, 4, 14

[16] John Flynn, Michael Broxton, Paul E. Debevec, Matthew DuVall, Graham Fyffe, Ryan S. Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, pages 2367–2376. Computer Vision Foundation / IEEE, 2019. 2

[17] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. *CoRR*, abs/1506.06825, 2015. 2

[18] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010. 2

[19] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV (8)*, volume 9912 of *Lecture Notes in Computer Science*, pages 740–756. Springer, 2016. 2

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 6602–6611. IEEE Computer Society, 2017. 2

[21] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3827–3837. IEEE, 2019. 2

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 3

[23] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54. ACM, 1996. 1, 2

[24] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3d photography. *ACM Trans. Graph.*, 36(6):234:1–234:15, 2017. 2

[25] Peter Hedman and Johannes Kopf. Instant 3d photography. *ACM Trans. Graph.*, 37(4):101:1–101:12, 2018. 2

[26] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6):257:1–257:15, 2018. 2

[27] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel J. Brostow. Scalable inside-out image-based rendering. *ACM Trans. Graph.*, 35(6):231:1–231:11, 2016. 2

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pages 6626–6637, 2017. 5, 6

[29] Youichi Horry, Ken-ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *SIGGRAPH*, pages 225–232. ACM, 1997. 2

[30] Ronghang Hu, Nikhila Ravi, Alexander C. Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[31] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR (Poster)*. OpenReview.net, 2017. 3

[32] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 35(6):193:1–193:10, 2016. 2

[33] Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, Peizhao Zhang, Zijian He, Peter Vajda, Ayush Saraf, and Michael F. Cohen. One shot 3d photography. *ACM Trans. Graph.*, 39(4):76, 2020. 2

[34] Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, pages 2539–2547, 2015. 2

[35] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42. ACM, 1996. 1, 2

[36] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 178–196. Springer, 2020. 2

[37] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 7

[38] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. *CoRR*, abs/2012.09855, 2020. 2, 4, 6, 7, 8, 15, 20

[39] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. In *CVPR*, pages 4616–4624. IEEE Computer Society, 2018. 2

[40] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR (Poster)*. OpenReview.net, 2017. 14

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019. 14

[42] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR (Poster)*. OpenReview.net, 2017. 3

[43] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *CVPR*, pages 6878–6887. Computer Vision Foundation / IEEE, 2019. 2

[44] 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 *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020. 1, 2

[45] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, pages 5436–5445. IEEE, 2020. 4

[46] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Trans. Graph.*, 38(6):184:1–184:15, 2019. 2

[47] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, pages 2249–2255. The Association for Computational Linguistics, 2016. 3

[48] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, pages 702–711. IEEE Computer Society, 2017. 2

[49] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Trans. Graph.*, 36(6):235:1–235:11, 2017. 2

[50] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 3, 14

[51] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 3

[52] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2, 4, 7, 12, 15

[53] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, pages 14837–14847, 2019. 3

[54] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV (19)*, volume 12364 of *Lecture Notes in Computer Science*, pages 623–640. Springer, 2020. 1, 2, 5

[55] Gernot Riegler and Vladlen Koltun. Stable view synthesis. *CoRR*, abs/2011.07233, 2020. 1, 2

[56] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *ICLR (Poster)*. OpenReview.net, 2017. 3

[57] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113. IEEE Computer Society, 2016. 2, 15

[58] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV (3)*, volume 9907 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2016. 2

[59] Jonathan Shade, Steven J. Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *SIGGRAPH*, pages 231–242. ACM, 1998. 2

[60] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, pages 8025–8035. IEEE, 2020. 2, 4, 7, 8, 15, 18, 20

[61] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, pages 175–184. Computer Vision Foundation / IEEE, 2019. 2

[62] Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4d RGBD light field from a single image. In *ICCV*, pages 2262–2270. IEEE Computer Society, 2017. 2

[63] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *ECCV*

(3), volume 11207 of *Lecture Notes in Computer Science*, pages 162–178. Springer, 2018. 2

[64] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. *Int. J. Comput. Vis.*, 32(1):45–61, 1999. 2

[65] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV (7)*, volume 9911 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2016. 2

[66] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *ICLR*, 2016. 6

[67] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.*, 38(4):66:1–66:12, 2019. 2

[68] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, pages 548–557. IEEE, 2020. 2

[69] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pages 6306–6315, 2017. 3, 4

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3, 4, 14

[71] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 5, 6

[72] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, pages 7465–7475. IEEE, 2020. 1, 2, 7, 15, 18

[73] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 842–857. Springer, 2016. 2

[74] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Trans. Graph.*, 38(4):76:1–76:13, 2019. 2

[75] Jimei Yang, Scott E. Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*, pages 1099–1107, 2015. 2

[76] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595. IEEE Computer Society, 2018. 5, 6

[77] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 6612–6619. IEEE Computer Society, 2017. 2

[78] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4):65:1–65:12, 2018. 2, 6, 15

[79] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 286–301. Springer, 2016. 2