Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image

Stanislaw Szymanowicz[1*] Eldar Insafutdinov[1*] Chuanxia Zheng[1*]
Dylan Campbell[2] João F. Henriques[1] Christian Rupprecht[1] Andrea Vedaldi[1]
[1]VGG, University of Oxford [2]Australian National University
{stan,eldar,cxzheng,joao,chrisr,vedaldi}@robots.ox.ac.uk dylan.campbell@anu.edu.au

Abstract

We propose Flash3D, a method for scene reconstruction and novel view synthesis from a single image which is both very generalisable and efficient. For generalisability, we start from a 'foundation' model for monocular depth estimation and extend it to a full 3D shape and appearance reconstructor. For efficiency, we base this extension on feed-forward Gaussian Splatting. Specifically, we predict a first layer of 3D Gaussians at the predicted depth, and then add additional layers of Gaussians that are offset in space, allowing the model to complete the reconstruction behind occlusions and truncations. Flash3D is very efficient, trainable on a single GPU in a day, and thus accessible to most researchers. It achieves state-of-the-art results when trained and tested on RealEstate10k. When transferred to unseen *datasets like NYU it outperforms competitors by a large margin. More impressively, when transferred to KITTI, Flash3D achieves better PSNR than methods trained specifically on that dataset. In some instances, it even outperforms recent methods that use multiple views as input. Code, models, demo, and more results are available at https://www.robots.ox.ac.uk/~vgg/research/flash3d/.*

1. Introduction

We consider the problem of reconstructing photorealistic 3D scenes from a single image in just one forward pass of a network. This is a challenging task because scenes are complex and monocular reconstruction is ill-posed. Unambiguous geometric cues, such as triangulation, are unavailable in the monocular setting, and there is no direct evidence of the occluded parts of the scene.

This problem is closely related to monocular depth estimation [4, 6, 7, 16, 21, 33, 34, 46, 47, 57, 59, 90], which is a mature area. It is now possible to accurately estimate metric depth with excellent cross-domain generalisation [47, 90, 91]. However, while depth estimators predict the 3D shape of the nearest visible surfaces, they do not provide any *appearance* information, nor an estimate of the occluded or out-of-frame parts of the scene. Depth alone is insufficient to accurately solve tasks such as *novel view synthesis* (NVS), which additionally require modelling unseen regions and view-dependent appearance.

While methods for monocular scene reconstruction exist [35, 74, 84], they mostly operate in a 'closed-world' setting where they are trained anew for each considered dataset. In contrast, modern depth predictors generalise well to new datasets at inference time. Furthermore, current monocular scene reconstructors are often slow or incur a high computational memory cost due to volumetric rendering [35] and implicit representations [92].

Very recently, Szymanowicz et al. [68] introduced the Splatter Image (SI), a method for fast monocular reconstruction of individual objects that builds upon the success of Gaussian Splatting [32]. The approach is simple: predict the parameters of a coloured 3D Gaussian for each input image pixel using a standard image-to-image neural network architecture. The resulting Gaussian mixture was shown to reconstruct objects well, including unobserved surfaces. In part, this was due to the fact that SI is able to use some of the "background pixels" to model the occluded parts of the object. However, in scene reconstruction, there is not such a reservoir of background pixels, which poses a challenge for the method. In contrast, pixelSplat [9], MVSplat [11], latentSplat [82] and GS-LRM [95], which share a similar design, were designed for scene reconstruction; however, they address the binocular reconstruction problem, requiring *two* images of the scene captured from different *known* viewpoints. We instead consider the more challenging monocular setting, since it is more generally applicable and does not require camera extrinsics, which is a challenging research problem on its own [78, 80, 94].

In this work, we introduce a new, simple, efficient and performant approach for monocular scene reconstruction called

*denotes equal contribution

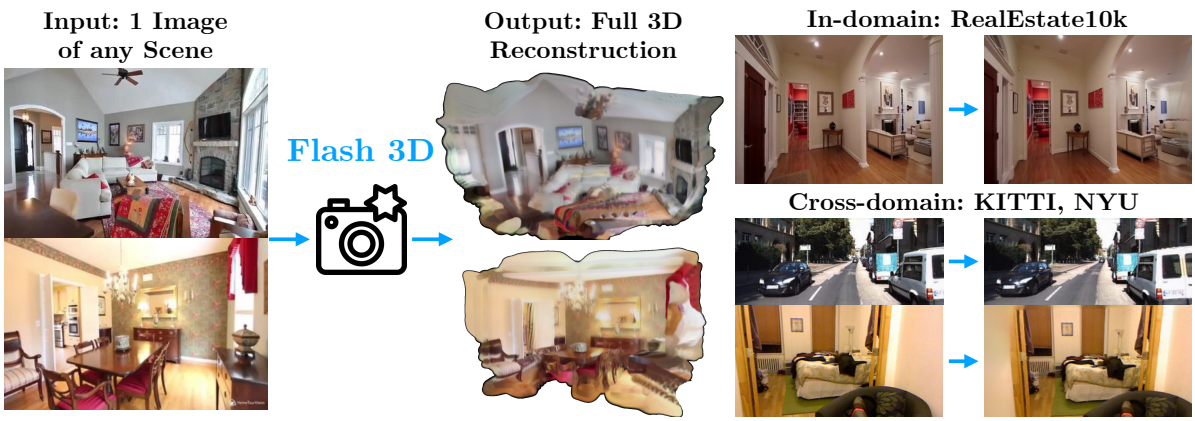

Figure 1. **Flash3D** reconstructs the 3D (*not* 2.5D) scene structure and appearance from just a single image 'in a flash', enabling accurate novel view synthesis. Trained on just one dataset, it generalises to new, different datasets and unknown scenes.

Flash3D. This is based on two key ideas. First, we address the issue of *generalisation* which limits current feed-forward monocular scene reconstructors. The aim is for Flash3D to work on any scene, not just on scenes similar to the ones in the training set. Analogous open-ended models are often called foundation models and require massive training datasets and computational resources unavailable to most research groups. A similar problem exists in 3D object reconstruction and generation [36, 37, 40, 41, 60, 97], where it is addressed by extending to 3D an existing foundation 2D image or video generator [5, 13, 20, 48, 53]. Here, we posit that scene reconstruction can also benefit from building on an existing foundation model, but opt for a monocular depth predictor as a more natural choice. We show, in particular, that by building on a high-quality depth predictor [47], we can achieve excellent generalisation to new datasets, to the point that our 3D reconstructions are more accurate than those of models trained specifically on those test domains.

Second, we improve feed-forward per-pixel Gaussian splatting for monocular scene reconstruction. As noted, applied to single objects, a per-pixel reconstructor can use the reservoir of background pixels to model the hidden parts of the object, which is not possible when reconstructing a full scene. Our solution is to predict multiple Gaussians per pixel, where only the first Gaussian along each ray is encouraged to conform to the depth estimate, and thus model the visible part of the scene. This is analogous to a layered representation [1, 3, 35, 58, 74, 76] and multi-Gaussian sampling in pixelSplat [9]. However, in our case Gaussians are deterministic, not limited to specific depth ranges, and the model is free to shift Gaussians off the ray to model occluded or truncated parts of the scene.

Overall, Flash3D is a simple and highly performant monocular scene reconstruction pipeline. Empirically, we find that Flash3D can (a) render high-quality images of the reconstructed 3D scene, (b) operate on a wide range of scenes, both indoor and outdoor; and (c) reconstruct occluded regions, which would not be possible with depth estimation alone or with naïve extensions of it. Flash3D achieves state-of-the-art novel view synthesis accuracy in all metrics on RealEstate10K [72]. More impressively, the same frozen model also achieves state-of-the-art accuracy when transferred to NYU [61] and KITTI [19] (in PSNR). Furthermore, in an extrapolation setting, our reconstructions can even be more accurate than those of binocular methods like pixelSplat [9] and latentSplat [82] that use two images of the scene instead of one, and are thus at a significant advantage.

In addition to the quality of the reconstructions, shown in Fig. 1, Flash3D is very efficient to evaluate and, most importantly, to train. For instance, we use 1/64th of the GPU resources of prior works such as MINE [35]. By achieving state-of-the-art results while using modest computational resources for training, this opens the research area to a wider range of researchers.

2. Related Work

Monocular feed-forward reconstruction. Like our approach, monocular feed-forward reconstructors work by passing a single image of the scene through a neural network to output a 3D reconstruction directly. For *scenes*, the works of [74–76] and MINE [35] do so by predicting multi-plane images [72], while BDS [84] uses neural radiance fields [43, 62]. Our method outperforms them in terms of speed and generalisation. Like our work, SynSin [83] uses a monocular depth predictor to reconstruct a scene; however, its reconstructions are incomplete and require a rendering network to improve the final novel views. In contrast, our approach outputs a high-quality 3D reconstruction which can be directly rendered with Gaussian Splatting [32]. For *objects*, a notable example is Large Reconstruction Model (LRM) [28], which obtains high-quality monocular reconstruction with a very large, and costly to train, model. The

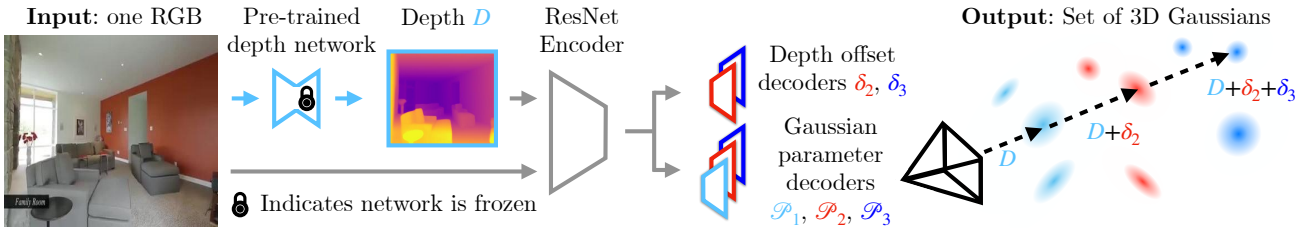

Figure 2. **Overview of Flash3D.** Given a single image I as input, Flash3D first estimates the metric depth D using a frozen off-the-shelf network [47]. Then, a ResNet50-like encoder–decoder network predicts a set of shape and appearance parameters $\mathcal{P}$ of K layers of Gaussians for every pixel $\boldsymbol{u}$, allowing unobserved and occluded surfaces to be modelled. From these predicted components, the depth can be obtained by summing the predicted (positive) offsets δ_i with the predicted monocular depth D, allowing the mean vector for every layer of Gaussians to be computed. This strategy ensures that the layers are depth-ordered, encouraging the network to model occluded surfaces.

most related work is Splatter Image [68] that uses Gaussian Splatting [32] for efficiency. Our approach also uses Gaussian Splatting as a representation, but does so for scenes rather than objects, which presents different challenges.

Few-view feed-forward reconstruction. A less challenging but still important case is few-view feed-forward reconstruction, where reconstruction requires two or more images. Early examples used NeRFs [43] as 3D representation of *objects* [12, 26, 30, 38, 51, 79, 92] and scenes [12, 14, 88]. These methods implicitly learn to match points between views; the works of [10, 93] make point matching more explicit. While many few-view reconstructors estimate the 3D shape of the object as an opacity field, an alternative is to directly predict new views [44, 55, 65, 66] of *scenes* with no explicit volumetric reconstruction, a concept pioneered by LFNs [63]. Other works use instead multi-plane images from narrow baseline stereo pairs [64, 72] and few views [31, 42]. More related to our approach, pixelSplat [9], latentSplat [82] and MVSplat [11] reconstruct scenes from a pair of images. They utilise cross-view attention to share information and predict Gaussian mixtures to represent the scene geometry. Other very recent approaches [70, 89, 95] combine LRM and Gaussian Splatting for reconstruction from a small number of images. We address *monocular* reconstruction instead, which is a much harder problem due to lack of geometric cues from triangulation.

Iterative reconstruction. Iterative or optimisation-based methods reconstruct from one or more images by iteratively fitting a 3D model to them. Due to their iterative nature, and the need to render the 3D model to fit it to the data, they are generally much slower than feed-forward approaches. DietNeRF [29] regularises reconstruction using language models, RegNeRF [45] and RefNeRF [77] use handcrafted regularisers, and SinNeRF [87] uses monocular depth. RealFusion [40] uses an image diffusion model as a prior for monocular reconstruction based on slow score distillation sampling iterations [49]. Numerous follow-ups took a similar path [69, 86]. Convergence speed and robustness can be improved by using multi-view aware generators [24, 37, 41, 73, 81, 97]. Approaches like Viewset Diffusion [67] and RenderDiffusion [2] fuse 3D reconstruction with diffusion-based generation, which can reduce but not eliminate the cost of iterative generation. More recently, ZeroNVS [56], ReconFusion [85] and Cat3D [17] utilize the large-scale diffusion model for single view or sparse view 3D reconstruction and NVS. However, they focus on "realistic" view generation, instead of "accurate" 3D reconstruction, and they still need the per-scene optimisation to achieve consistent 3D structure, which is relative expensive and slow. In contrast, our approach is feed-forward and therefore significantly faster, close to real-time (10fps). Some approaches generate novel views in a feed-forward manner, but iteratively and autoregressively, one view at a time. Examples include PixelSynth [52], GeNVS [8], and Text2Room [27]. In contrast, we generate the final 3D reconstruction in a single feed-forward pass.

Monocular depth prediction. Our method is based on monocular depth estimation [4, 6, 7, 15, 16, 21, 22, 33, 34, 46, 50, 57, 59, 90, 98], where metric or relative depth is predicted for every image pixel for a given image. By learning visual depth cues from large datasets, these approaches have demonstrated high accuracy and the capacity to generalise across datasets. While our method is agnostic to the depth predictor used, we use one of the state-of-the-art metric depth estimators, UniDepth [47], for our experiments.

3. Method

Let $I \in \mathbb{R}^{3\times H\times W}$ be an RGB image of a scene. Our goal is to learn a neural network Φ that takes as input I and predicts a representation $\mathcal{G} = \Phi(I)$ of the 3D content of the scene, both in terms of 3D geometry and photometry. We first discuss the background and baseline model in Sec. 3.1, introduce our layered multi-Gaussian predictor and discuss the use of monocular depth prediction as a prior in Sec. 3.2.

3.1. Background: Scene reconstruction from a single image

Representation: scenes as sets of 3D Gaussians. The scene representation $\mathcal{G} = \{(\sigma_i, \boldsymbol{\mu}_i, \Sigma_i, c_i)\}_{i=1}^G$ is a set of 3D Gaussians [32], where $\sigma_i \in [0,1)$ is the opacity, $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean, $\Sigma_i \in \mathbb{R}^{3\times 3}$ is the covariance matrix, and $c_i : \mathbb{S}^2 \to \mathbb{R}^3$ is the radiance function (directional colour) of each component. Let $g_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$ be the corresponding (un-normalised) Gaussian function. The colours of the Gaussians are generally represented using spherical harmonics, so that $[c_i(\boldsymbol{\nu})]_j = \sum_{l=0}^L \sum_{m=-l}^l c_{ijlm} Y_{lm}(\boldsymbol{\nu})$, where $\boldsymbol{\nu} \in \mathbb{S}^2$ is a view direction and Y_{lm} are the spherical harmonics of various orders m and degrees l. The Gaussian mixture $\mathcal{G}$ defines the opacity and colour functions of a radiance field: $\sigma(\boldsymbol{x}) = \sum_{i=1}^G \sigma_i g_i(\boldsymbol{x})$, $c(\boldsymbol{x}, \boldsymbol{\nu}) = \sum_{i=1}^G c_i(\boldsymbol{\nu})\sigma_i g_i(\boldsymbol{x}) / \sum_{i=1}^G \sigma_i g_i(\boldsymbol{x})$, where $\sigma(\boldsymbol{x})$ is the opacity at 3D location $\boldsymbol{x} \in \mathbb{R}^3$ and $c(\boldsymbol{x}, \boldsymbol{\nu})$ is the radiance at $\boldsymbol{x}$ in direction $\boldsymbol{\nu} \in \mathbb{S}^2$ towards the camera.

The field is rendered into an image J by integrating the radiance along the line of sight using the *emission–absorption* [39] equation $J(\boldsymbol{u}) = \int_0^\infty c(\boldsymbol{x}_t, \boldsymbol{\nu})\sigma(\boldsymbol{x}_t) \exp\left(-\int_0^t \sigma(\boldsymbol{x}_\tau) d\tau\right) dt$, where $\boldsymbol{x}_t = \boldsymbol{x}_0 - t\boldsymbol{\nu}$ is the ray originating at the camera centre $\boldsymbol{x}_0$ and propagating towards the pixel $\boldsymbol{u}$ in the direction $-\boldsymbol{\nu}$. The key contribution of Gaussian Splatting [32] is to approximate this integral very efficiently, implementing a differentiable rendering function $\hat{J} = \mathrm{Rend}(\mathcal{G}, \pi)$ which takes as input the Gaussian mixture $\mathcal{G}$ and viewpoint π and returns an estimate $\hat{J}$ of the corresponding image.

Monocular reconstruction. Following [68], the output $\Phi(I) \in \mathbb{R}^{C\times H\times W}$ of the neural network is a tensor that specifies, for each pixel $\boldsymbol{u} = (u_x, u_y, 1)$, the parameters of a coloured Gaussian, consisting of the opacity σ, the depth $d \in \mathbb{R}_+$, the offsets $\Delta \in \mathbb{R}^3$, the covariance $\Sigma \in \mathbb{R}^{3\times 3}$ expressed as rotation and scale (seven parameters, using quaternions for rotation), and the parameters of the colour model $c \in \mathbb{R}^{3(L+1)^2}$ where L is the order of the spherical harmonics. The mean of the Gaussian is then given by $\boldsymbol{\mu} = K^{-1}\boldsymbol{u}d + \Delta$, where $K = \mathrm{diag}(f, f, 1) \in \mathbb{R}^{3\times 3}$ is the camera calibration matrix and f its focal length. Hence, there are $C = 1+1+3+7+3(L+1)^2 = 12+3(L+1)^2$ parameters predicted for each pixel. The model Φ is trained using triplets (I, J, π) where I is an input image, J is a target image, and π is the relative camera pose. To learn the network parameters, one simply minimises the rendering loss $\mathcal{L}(\mathcal{G}, \pi, J) = \| \mathrm{Rend}(\mathcal{G}, \pi) - J \|$.

3.2. Monocular feed-forward multi-Gaussians

For generalisation, we propose to build Flash3D on a high-quality pre-trained model trained on a large amount of data. Specifically, given the similarities between monocular scene reconstruction and monocular depth estimation, we use an off-the-shelf monocular depth predictor Ψ. This model takes as input an image I and returns a depth map $D = \Psi(I)$, where $D \in \mathbb{R}_+^{H\times W}$ is a matrix of depth values, as explained next.

Baseline architecture. Given an image I and estimated depth map D, our baseline model consists of an additional network $\Phi(I, D)$ that takes as input the image and the depth map and returns the required per-pixel Gaussian parameters. In more detail, for each pixel $\boldsymbol{u}$, the entry $[\Phi(I, D)]_{\boldsymbol{u}} = (\sigma, \Delta, s, \theta, c)$ consists of the opacity $\sigma \in \mathbb{R}_+$, the displacement $\Delta \in \mathbb{R}^3$, the scale $s \in \mathbb{R}^3$, the quaternion $\theta \in \mathbb{R}^4$ parametrising the rotation $R(\theta)$, and the colour parameters c. The covariance of each Gaussian is given by $\Sigma = R(\theta)^\top \mathrm{diag}(s) R(\theta)$ and the mean is given [68] by $\boldsymbol{\mu} = (u_x\, d/f,\ u_y\, d/f,\ d) + \Delta$, where f is the focal length of the camera (either known or also estimated by Ψ) and the depth $d = D(\boldsymbol{u})$ is from the depth map. The network Φ is a U-Net [54] utilising ResNet blocks [25] for encoding and decoding, denoted Φ_{enc} and Φ_{dec} respectively. The decoder network thus outputs a tensor $\Phi_{\text{dec}}(\Phi_{\text{enc}}(I, D)) \in \mathbb{R}^{(C-1)\times H\times W}$. Note that the network output has $C-1$ channels only, as depth is taken directly from Ψ. Please see the supplement for full details.

Multi-Gaussian prediction. While the Gaussians in the model above have the ability to be offset from the corresponding pixel's ray, each Gaussian tends naturally to model the portion of the object that projects onto that pixel. Szymanowicz et al. [68] note that, for individual objects, there is a large number of background pixels that are not associated with any object surface, and these can be repurposed by the model to capture the unobserved parts of the 3D object. However, this is not the case for scenes, where the goal is to reconstruct every input pixel, and beyond.

Since there are no "idle" pixels, it is difficult for the model to repurpose some of the Gaussians to model the 3D scene around occlusions and beyond the image field-of-view. Hence, we propose to predict a small number $K > 1$ of different Gaussians for each pixel.

Conceptually, given an image I and an estimated depth map D, our network predicts a set of shape, location and appearance parameters $\mathcal{P} = \{(\sigma_i, \delta_i, \Delta_i, \Sigma_i, c_i)\}_{i=1}^K$ for every pixel $\boldsymbol{u}$, where the depth of the ith Gaussian is given by

$$d_i = d + \sum_{j=1}^{i} \delta_j, \tag{1}$$

where $d = D(\boldsymbol{u})$ is the predicted depth at pixel $\boldsymbol{u}$ in depth map D and $\delta_1 = 0$ is a constant. Note that since the depth offset δ_i cannot be negative, this ensures that subsequent

Gaussian layers are "behind" previous ones and encourages the network to model occluded surfaces. The mean of the ith Gaussian is then given by $\boldsymbol{\mu}_i = (u_x\, d_i/f,\, u_y\, d_i/f,\, d_i) + \Delta_i$. In practice, we find $K = 2$ to be a sufficiently expressive representation.

Reconstructing beyond the border with padding. As we show empirically, it is important for the network to be able to model 3D content just outside its field-of-view. Although multiple Gaussian layers help in this regard, there is a particular need for additional Gaussians near the image border (*e.g.*, for good new view synthesis when the camera retracts). To facilitate obtaining such Gaussians, the encoder Φ_{enc} starts with *padding* the input image and depth (I, D) with $P > 0$ pixels on each side, so that the outputs $\Phi_k(I, D) \in \mathbb{R}^{(C-1)\times(H+2P)\times(W+2P)}$ are larger than the inputs.

4. Experiments

We design our experiments to support four key findings, with each section dedicated to one finding. We begin with the most important result: cross-dataset generalisation—leveraging a monocular depth prediction network and training on a single dataset results in good reconstruction quality on other datasets (Sec. 4.2). Second, we establish that Flash3D serves as an effective representation for single-view 3D reconstruction by comparing against methods specifically designed for this task (Sec. 4.3). Third, we go as far as to show that the prior learned by single-view Flash3D is as strong as that learned by two-view methods (Sec. 4.4). Finally, we show via ablation studies how each design choice contributes to performance Flash3D (Sec. 4.5) and analyse outputs from Flash3D to gain insight into its inner workings.

4.1. Experiment settings

Datasets. Flash3D is trained only on the large-scale RealEstate10k [72] dataset, containing real estate videos from YouTube. We follow the default training/testing split with 67,477 scenes for training and 7,289 for testing. Once Flash3D is trained, we assess its effectiveness across various datasets, discussed in detail in each section. For details of the evaluation protocols, see the appendix.

Metrics. For quantitative results, we report the standard image quality metrics, including pixel-level PSNR, patch-level SSIM, and feature-level LPIPS.

Compared methods. We compare with several state-of-the-art single-view 3D scene reconstruction models, including LDI [76], Single-View MPI [74], SynSin [83], BTS [84] and MINE [35]. We include a comparison to our adaptation of Splatter Image [68] to show that our method is much better suited to general scenes, as well as a comparison to ray-wise unprojection $\mathcal{U}$ of the input colours to the locations predicted by the depth network. Finally, while not a fair comparison, we also compare with state-of-the-art two-view novel view synthesis methods, including [14], pixelSplat [9], MVSplat [11], and latentSplat [82].

Table 1. **Cross-Domain Novel View Synthesis**. We evaluate Novel View Synthesis accuracy on datasets not used in training of our method. We outperform baselines which were trained on KITTI specifically. Here, cross-domain (CD) denotes that the method was not trained on the dataset being evaluated.

	KITTI				NYU			
Method	CD	PSNR↑	SSIM↑	LPIPS ↓	CD	PSNR ↑	SSIM ↑	LPIPS ↓
LDI [76]	✗	16.50	0.572	-	-	-	-	
SV-MPI [74]	✗	19.50	0.733	-	-	-	-	-
BTS [84]	✗	20.10	0.761	0.144	-	-	-	-
MINE [35]	✗	21.90	**0.828**	**0.112**	✓	24.33	0.745	0.202
UniDepth w/ $\mathcal{U}$	✓	20.86	0.774	0.154	✓	22.54	0.732	0.212
Flash3D (Ours)	✓	**21.96**	0.826	0.132	✓	**25.45**	**0.774**	**0.196**

Implementation details. Flash3D comprises a pre-trained UniDepth [47] model, a ResNet50 [25] encoder, alongside multiple depth offset decoders and Gaussian decoders. The entire model is trained on a single A6000 GPU for 40,000 iterations with batch size 16. The training is remarkably efficient, completed in one day on a single A6000 GPU. Given that UniDepth remains frozen during training, we can expedite the training by pre-extracting depth maps for the entire dataset. With this, Flash3D can be trained to achieve state-of-the-art quality on a *single A6000 GPU in 16 hours*.

4.2. Cross-domain novel view synthesis

Datasets. To evaluate the cross-domain generalisation ability, we directly evaluate performance on unseen outdoor (KITTI [18]) and indoor (NYU [61]) datasets. For KITTI, we follow standard benchmarks with a well-established protocol for evaluation, with 1,079 images for testing. For NYU, we formed a new protocol, with 250 source images for testing (see supp. mat for details). We evaluate all methods in the same manner. We verified that Unidepth [47] was not trained on these datasets.

To the best of our knowledge, we are the first to report performance on feed-forward cross-domain monocular reconstruction. We consider two challenging comparisons. First, we evaluate Flash3D and the current state-of-the-art [35] on NYU, an indoor dataset that is similar in nature to RE10k, yet unseen in training. In Tab. 1, we observe that our method performs significantly better on this transfer experiment, despite the domain gap being relatively small. This suggests that prior works do not generalise as well as our method. Second, we compare our method on KITTI in Tab. 1, where it performs on par with the state-of-the-art that was trained on this dataset. Indeed, Flash3D outperforms the others with respect to PSNR, despite being trained only on an indoor dataset. This suggests that leveraging a pretrained depth net-

Table 2. **In-domain Novel View Synthesis.** Our model shows state-of-the-art in-domain performance on RealEstate10k on small, medium and large baseline ranges.

Model	5 frames			10 frames			$\mathcal{U}[-30, 30]$ frames		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Syn-Sin [83]	-	-	-	-	-	-	22.30	0.740	-
SV-MPI [74]	27.10	0.870	-	24.40	0.812	-	23.52	0.785	-
BTS [84]	-	-	-	-	-	-	24.00	0.755	0.194
Splatter Image [68]	28.15	0.894	0.110	25.34	0.842	0.144	24.15	0.810	0.177
MINE [35]	28.45	0.897	0.111	25.89	0.850	0.150	24.75	0.820	0.179
Flash3D (Ours)	**28.46**	**0.899**	**0.100**	**25.94**	**0.857**	**0.133**	**24.93**	**0.833**	**0.160**

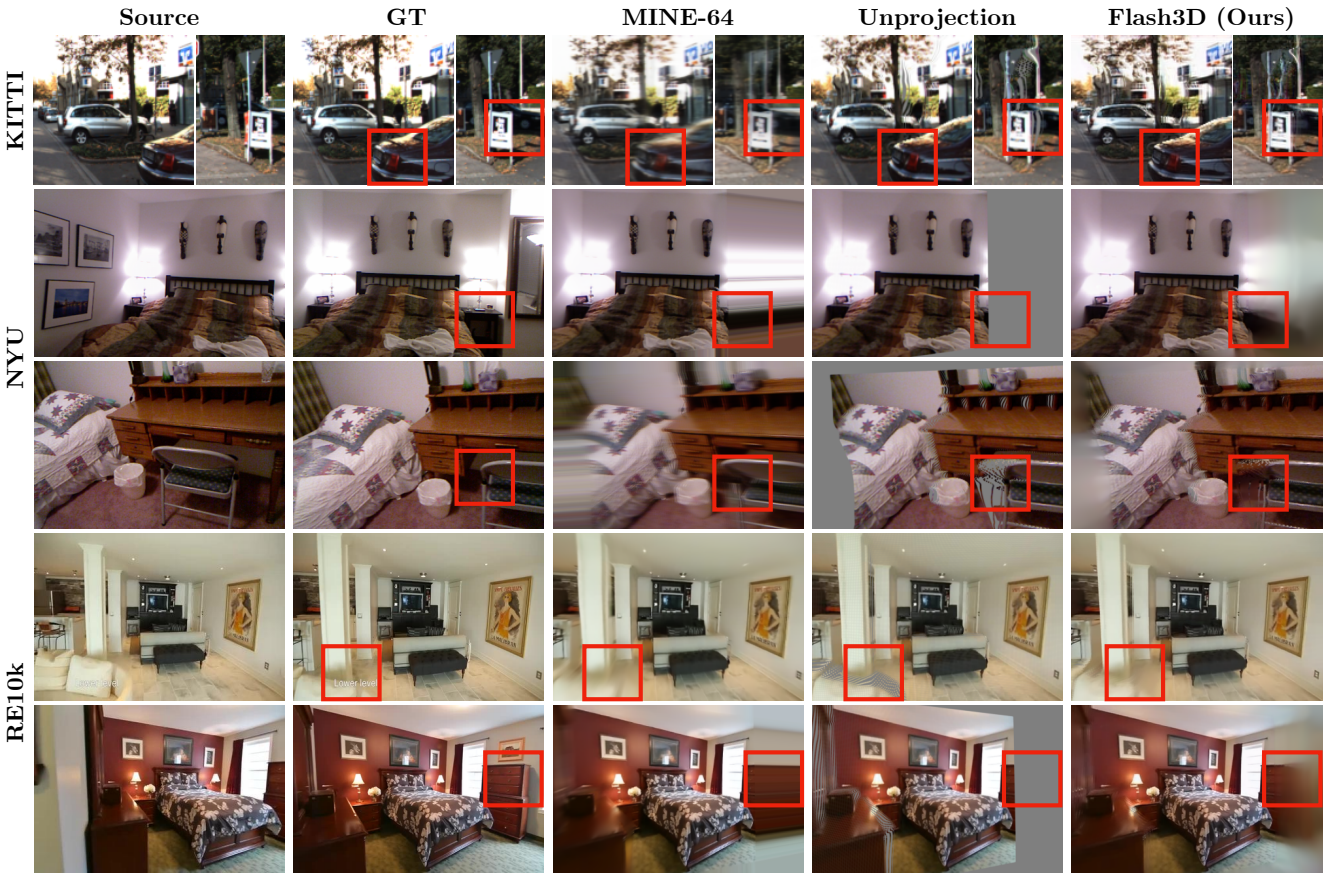

Figure 3. **Qualitative comparison of monocular reconstruction** on all datasets. Flash3D (Ours, right column) is sharper (top row, car's back) than state-of-the-art MINE [35] despite Flash3D not training on KITTI. This is thanks to leveraging a depth predictor which, when used on its own (fourth column), cannot represent occluded regions (third row, fourth row). As well as representing occluded regions better than MINE (first row, third row and fourth row), Flash3D also fills in better explanations of regions outside the source camera frustrum (second row, fifth row).

work has allowed our network to learn an extremely strong shape and appearance prior that is even more accurate than learning on this dataset directly.

4.3. In-domain novel view synthesis

We perform an in-domain evaluation on RealEstate10k [72], following the same protocol as prior works [35]. We evaluate the quality of zero-shot reconstruction and compare performance on an in-domain dataset, RealEstate10k. We evaluate the quality of reconstruction through novel view synthesis metrics as that is the only ground-truth data available in this dataset. RealEstate10k evaluates the quality of reconstructions at different distances between the source and the target, as a smaller distance makes the task easier.

Table 3. **Ablation Study.** Results for ablating different design choices of our method.

	RE10k – in-domain			NYU – cross-domain			KITTI – cross-domain		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Flash3D	**24.93**	**0.833**	**0.160**	**25.09**	**0.775**	**0.182**	**21.96**	**0.826**	**0.132**
w/o depth net, w/ 2nd layer	23.62	0.782	0.186	23.73	0.732	0.210	-	-	-
w/o depth net, w/o 2nd layer	24.01	0.806	0.176	23.98	0.750	0.207	-	-	-
w/ depth net, w/o 2nd layer	24.45	0.825	0.163	24.83	0.767	0.190	21.50	0.812	0.141
w/ depth net, unproject only	22.80	0.781	0.207	22.14	0.729	0.217	20.86	0.774	0.154

Table 4. **Comparison with Two-view Methods**. We compare on the split used by pixelSplat [9] for two-view interpolation and on the split used by latentSplat [82] for extrapolation. We take the view closest to the target as the source. Our method uses a *single* view and still extrapolates better.

	Input	RE10k Interpolation			RE10k Extrapolation		
Method	Views	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Du et al [14]	2	24.78	0.820	0.213	21.83	0.790	0.242
pixelSplat [9]	2	26.09	0.864	0.136	21.84	0.777	0.216
latentSplat [82]	2	23.93	0.812	0.164	22.62	0.777	0.196
MVSplat [11]	2	**26.39**	**0.869**	**0.128**	23.04	0.813	**0.185**
Flash3D (Ours)	**1**	23.87	0.811	0.185	**24.10**	**0.815**	**0.185**

In Tab. 2, we observe that we achieve state-of-the-art results on this mature benchmark across all distances between the source and the target. Further analysis in Fig. 3 reveals that our method's reconstructions are sharper and more accurate than prior state-of-the-art [35], despite being trained on an order of magnitude fewer GPUs (1 vs. 64).

4.4. Comparison to few-view novel view synthesis

Datasets. To further evaluate the effectiveness of Flash3D, we conducted assessments using the pixelSplat [9] split for interpolation and the latentSplat [82] split for extrapolation.

Unlike existing two-view methods that typically assess interpolation between two source views, Flash3D consistently performs extrapolation from a single view. The results are reported in Tab. 4. Here, Flash3D cannot outperform two-view approaches on the interpolation task, due to receiving less information. However, Flash3D surpasses all previous state-of-the-art two-view methods at view extrapolation. This highlights the utility of the multi-layer Gaussian representation of our approach at capturing and modelling unseen areas.

4.5. Ablation study and analysis

Ablation study. We ablate our method for in-domain and cross-domain settings in Tab. 3, focusing on the following questions. **Q1:** Is leveraging a monocular depth predictor useful in the task of reconstructing appearance and geometry of scenes? **Q2:** If yes, is it sufficient on its own, *i.e.*, is learning shape and appearance parameters necessary for scene reconstruction with 3D Gaussians?

Importance of depth predictor. We remove the pretrained depth network that predicts depth D, instead estimating it jointly with all other parameters. First, in Tab. 3 we observe that this leads to a significant drop in performance compared to Flash3D, indicating that the depth network contains important cues that had already been learned. Moreover, the third row of Tab. 3 indicates that without the depth network, 2 layers of Gaussians per pixel performs *worse* than using just one layer. We hypothesise that the depth network plays an important role in avoiding local optima that were reported to limit the learning capabilities of primitive-based methods [9]. Qualitatively, the fourth column in Fig. 4 illustrates that removing the depth network makes it challenging to learn accurate geometries of walls (orange wall is bent) and object boundaries (bed has an incorrect shape).

Importance of extending beyond depth prediction. Here we remove the learned parts of our network. First, we use only one layer of Gaussians and predict parameters $\mathcal{P}_1$ corresponding to the depth D predicted by the pre-trained depth network. This also results in a drop in performance in Tab. 3. We then go further and remove the network that predicts $\mathcal{P}_1$, removing learning altogether. Novel views are simply rendered from source view colours backprojected with the depths D. This understandably drops the performance even further. In Fig. 4 we observe that these drops are due to the 1-layer method not representing occluded parts of the scene. The last column in Fig. 4 illustrates that the holes are significant when using only depth unprojection, and can be partially mitigated when learning shapes $\mathcal{P}_1$ due to the network being able to stretch the Gaussians at depth discontinuities.

Analysis. Fig. 5 analyses the contribution of each Gaussian layer to a full reconstruction of the scene. We visualise the depth of each of the layers, D and $D + \delta_2$, multiplied by the opacity σ_1, σ_2 of the corresponding Gaussians, illustrating how much they are used when rendering the scene. In Fig. 5, the more saturated the colour, the more opaque the Gaussian (black is fully transparent). We observe that the first layer has the most opaque Gaussians at object boundaries (wall, cabinet) and at complicated geometries (chair), indicating that these are the regions where the depth prediction network is the most useful. This is further supported by Fig. 4 where removing the depth network impaired reconstruction in exactly the same regions. Leveraging the depth network at object boundaries results in crisp, accurate geome-

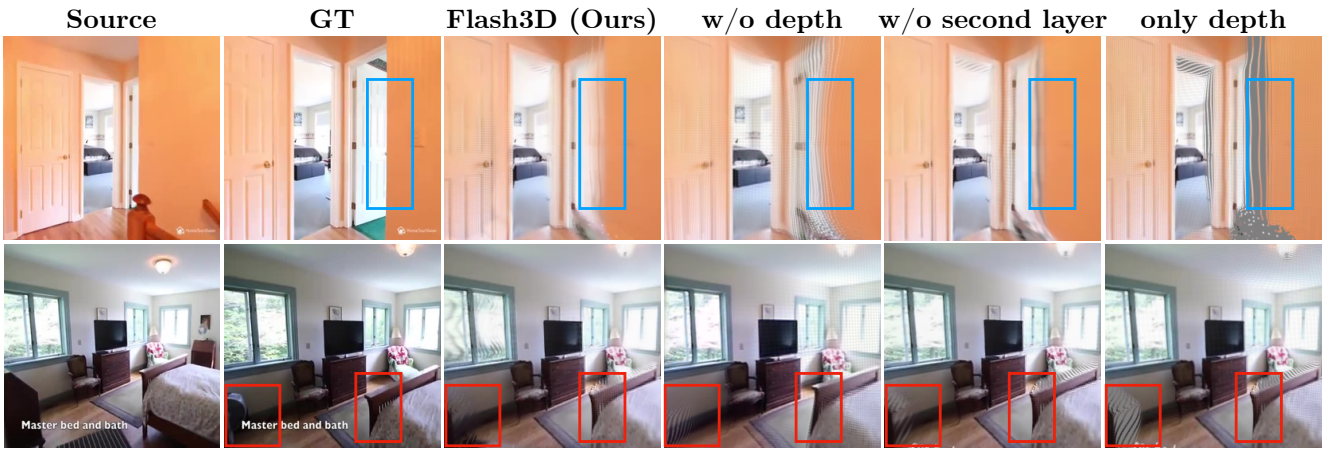

Figure 4. **Ablation.** We show how Flash3D degrades when components are removed. Removing the depth network (4th column) results in incorrect geometry (orange wall, corner of the bed). Using only one layer of Gaussians (5th) results in holes in renderings due to disocclusions (orange wall, area behind cabinet), although they are not as bad as when simply using depth unprojection (rightmost).

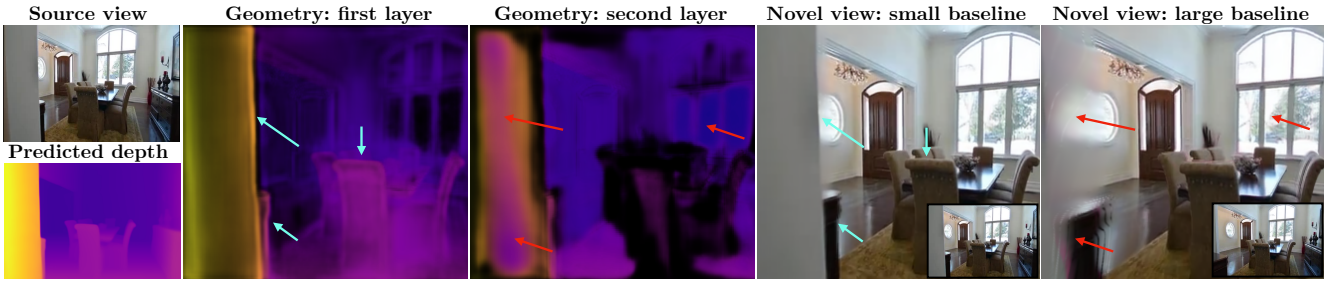

Figure 5. **Analysis of layered Gaussians.** The first layer of Gaussians (second column) represents visible parts where the depth prediction can be used (blue arrows). The second layer (third column) represents the remaining parts of the scene (red arrows): occluded regions (wall, cabinet) and regions where depth prediction is unreliable (windows). Combining the two leads to sharp geometries at small baselines (fourth column) and reasonable reconstructions at large baselines (right column).

tries at small baselines (fourth column). Interestingly, the network ignores the depth prediction for windows, which are consistently incorrect. Next, we analyse where the second layer of Gaussians places its predictions. In the third column of Fig. 5,where the network observes a wall, the second layer of Gaussians is placed at larger depth. These Gaussians are observed when the camera motion (baseline) is large, and are shown to have a reasonable appearance in Fig. 5 (right column). The last column in Fig. 5 additionally reveals a limitation of our method. It is a deterministic, regressive model of structure and appearance, and thus produces blurry renderings in presence of ambiguity: when baselines are very large, in occluded regions or when camera moves backward. Blurriness could be reduced with additional losses (perceptual [96] or adversarial [23]). Alternatively, our method could be incorporated as conditioning within a framework similar to [8] or as the reconstructor in a diffusion-based feed-forward 3D generation framework [67, 71].

5. Conclusion

We presented Flash3D, a model that can be trained in just 16h on a single GPU to achieve state-of-the-art results for monocular scene reconstruction. Our formulation allows using a monocular depth estimator as a foundation for full 3D scene reconstruction. As a consequence, the model generalizes very well: it outperforms prior works even when not trained specifically on the target dataset as them. Analyses reveal the interaction mechanisms between the pretrained network and the learned modules, and ablations verify the importance of each component.

Acknowledgements. The authors acknowledge the generous support of the Royal Academy of Engineering (RF\201819\18\163), EPSRC SYN3D Grant EP/Z001811/1, EPSRC Doctoral Training Partnerships Scholarship (DTP) EP/R513295/1 and the Oxford-Ashton Scholarship. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission (https://openaccess.ox.ac.uk/rights-retention/).

References

[1] E. Adelson. Layered representations for vision and video. In *ICCV Workshop on the Representation of Visual Scenes*, 1995. 2

[2] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In *Proc. CVPR*, 2022. 3

[3] Simon Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. CVPR*, 1998. 2

[4] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Proc. ICCV*, 1998. 1, 3

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv.cs*, abs/2311.15127, 2023. 2

[6] B. F. Buxton and H. Buxton. Monocular depth perception from optical flow by space time signal processing. *Phil. Trans. R. Soc. Lond. B*, 218, 1983. 1, 3

[7] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Trans. Circuits Syst. Video Techn.*, 28(11):3174–3182, 2018. 1, 3

[8] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3D-aware diffusion models. In *Proc. ICCV*, 2023. 3, 8

[9] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *arXiv.cs*, 2023. 1, 2, 3, 5, 7

[10] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proc. ICCV*, 2021. 3

[11] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: efficient 3d gaussian splatting from sparse multi-view images. *arXiv*, 2403.14627, 2024. 1, 3, 5, 7

[12] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (SRF): learning view synthesis for sparse views of novel scenes. In *Proc. CVPR*, 2021. 3

[13] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam S. Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, Matthew Yu, Abhishek Kadian, Filip Radenovic, Dhruv Mahajan, Kunpeng Li, Yue Zhao, Vladan Petrovic, Mitesh Kumar Singh, Simran Motwani, Yi Wen, Yiwen Song, Roshan Sumbaly, Vignesh Ramanathan, Zijian He, Peter Vajda, and Devi Parikh. Emu: Enhancing image generation models using photogenic needles in a haystack. *CoRR*, abs/2309.15807, 2023. 2

[14] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proc. CVPR*, 2023. 3, 5, 7

[15] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proc. NeurIPS*, 2014. 3

[16] Bill Freeman, Ce Liu, Forrester Cole, Noah Snavely, Richard Tucker, Tali Dekel, and Zhengqi Li. Learning the depths of moving people by watching frozen people. In *Proc. CVPR*, 2019. 1, 3

[17] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 3

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2

[20] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *CoRR*, abs/2311.10709, 2023. 2

[21] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv.cs*, abs/1609.03677, 2016. 1, 3

[22] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *Proc. ICCV*, 2019. 3

[23] 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 *Proc. NeurIPS*, 2014. 8

[24] Jiatao Gu, Alex Trevithick, Kai-En Lin, Josh M. Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. NerfDiff: Single-image view synthesis with nerf-guided distillation from 3D-aware diffusion. *arXiv.cs*, abs/2302.10109, 2023. 3

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 4, 5

[26] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[27] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In *Proc. ICCV*, 2023. 3

[28] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *Proc. ICLR*, 2024. 2

[29] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proc. ICCV*, 2021. 3

[30] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. GeoNeRF: generalizing NeRF with geometry priors. In *Proc. CVPR*, 2022. 3

[31] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Proc. NeurIPS*, 2017. 3

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *Proc. SIGGRAPH*, 42(4), 2023. 1, 2, 3, 4

[33] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*, 2016. 1, 3

[34] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv*, abs/1907.01341, 2019. 1, 3

[35] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. MINE: towards continuous depth MPI with nerf for novel view synthesis. In *Proc. ICCV*, 2021. 1, 2, 5, 6, 7

[36] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3D: Fast text-to-3D with sparse-view generation and large reconstruction model. *Proc. ICLR*, 2024. 2

[37] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *Proc. ICCV*, 2023. 2, 3

[38] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. *arXiv.cs*, abs/2107.13421, 2022. 3

[39] Nelson L. Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995. 4

[40] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. RealFusion: 360 reconstruction of any object from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[41] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, Natalia Neverova, Andrea Vedaldi, Oran Gafni, and Filippos Kokkinos. IM-3D: Iterative multiview diffusion and reconstruction for high-quality 3D generation. *arXiv preprint*, abs/2402.08682, 2024. 2, 3

[42] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *Proc. SIGGRAPH*, 38(4), 2019. 3

[43] 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 *Proc. ECCV*, 2020. 2, 3

[44] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. *arXiv.cs*, abs/2310.10375, 2023. 3

[45] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. CVPR*, 2022. 3

[46] Mertalp Ocal and Armin Mustafa. RealMonoDepth: Self-supervised monocular depth estimation for general scenes. *arXiv.cs*, abs/2004.06267, 2020. 1, 3

[47] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: universal monocular metric depth estimation. In *Proc. CVPR*, 2024. 1, 2, 3, 5

[48] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *arXiv.cs*, abs/2307.01952, 2023. 2

[49] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *Proc. ICLR*, 2023. 3

[50] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv.cs*, abs/2103.13413, 2021. 3

[51] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proc. CVPR*, 2021. 3

[52] Chris Rockwell, David F. Fouhey, and Justin Johnson. PixelSynth: Generating a 3D-consistent experience from a single image. In *Proc. ICCV*, 2021. 3

[53] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. *arXiv.cs*, abs/2104.07652, 2021. 2

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, 2015. 4

[55] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas A. Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. *CoRR*, abs/2111.13152, 2021. 3

[56] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. *arXiv.cs*, abs/2310.17994, 2023. 3

[57] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J. Fleet. Monocular depth estimation using diffusion models. *arXiv*, 2023. 1, 3

[58] J. Shade, S. Gortler, Li wei He, and R. Szeliski. Layered depth images. In *Proc. SIGGRAPH*, 1998. 2

[59] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *Proc. ICCV*, 2023. 1, 3

[60] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. In *Proc. ICLR*, 2024. 2

[61] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. ECCV*, 2012. 2, 5

[62] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *Proc. NeurIPS*, 2019. 2

[63] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021. 3

[64] 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 *Proc. CVPR*, 2019. 3

[65] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. *arXiv.cs*, abs/2112.09687, 2021. 3

[66] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable Patch-Based neural rendering. In *Proc. ECCV*, 2022. 3

[67] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion: (0-)image-conditioned 3D generative models from 2D data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 3, 8

[68] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter Image: Ultra-fast single-view 3D reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3, 4, 5, 6

[69] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-It-3D: High-fidelity 3d creation from A single image with diffusion prior. *arXiv.cs*, abs/2303.14184, 2023. 3

[70] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. *arXiv*, 2402.05054, 2024. 3

[71] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B. Tenenbaum, Frédo Durand, William T. Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv.cs*, abs/2306.11719, 2023. 8

[72] Zhou Tinghui, Tucker Richard, Flynn John, Fyffe Graham, and Snavely Noah. Stereo magnification: Learning view synthesis using multiplane images. *arxiv*, 2018. 2, 3, 5, 6

[73] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proc. CVPR*, 2023. 3

[74] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proc. CVPR*, 2020. 1, 2, 5, 6

[75] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. CVPR*, pages 209–217, 2017.

[76] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proc. ECCV*, 2018. 2, 5

[77] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd E. Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *Proc. CVPR*, 2022. 3

[78] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: solving pose estimation via diffusion-aided bundle adjustment. In *Proc. ICCV*, 2023. 1

[79] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proc. CVPR*, 2021. 3

[80] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. *arXiv*, 2023. 1

[81] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *Proc. ICLR*, 2023. 3

[82] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *arXiv*, abs/2403.16292, 2024. 1, 2, 3, 5, 7

[83] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proc. CVPR*, 2020. 2, 5, 6

[84] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 6

[85] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21551–21561, 2024. 3

[86] Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: regularizing neural radiance fields with denoising diffusion models. In *Proc. CVPR*, 2023. 3

[87] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. SinNeRF: Training neural radiance fields on complex scenes from a single image. In *Proc. ECCV*, 2022. 3

[88] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. MuRF: Multi-baseline radiance fields. *arXiv*, 2312.04565, 2023. 3

[89] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large gaussian reconstruction model for efficient 3D reconstruction and generation. *arXiv*, 2403.14621, 2024. 3

[90] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing

the power of large-scale unlabeled data. In *Proc. CVPR*, 2024. 1, 3

[91] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proc. ICCV*, 2023. 1

[92] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proc. CVPR*, 2021. 1, 3

[93] Chen Yuedong, Xu Haofei, Wu Qianyi, Zheng Chuanxia, Cham Tat-Jen, and Cai Jianfei. Explicit correspondence matching for generalizable neural radiance fields. *arXiv*, 2304.12294, 2023. 3

[94] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *Proc. ICLR*, 2024. 1

[95] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: large reconstruction model for 3D Gaussian splatting. *arXiv*, 2404.19702, 2024. 1, 3

[96] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pages 586–595, 2018. 8

[97] Chuanxia Zheng and Andrea Vedaldi. Free3D: Consistent novel view synthesis without 3D representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[98] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017. 3