DepthSplat: Connecting Gaussian Splatting and Depth

Haofei Xu[1,2] Songyou Peng[1] Fangjinhua Wang[1] Hermann Blum[1,4] Daniel Barath[1]
Andreas Geiger[2] Marc Pollefeys[1,3]
[1]ETH Zurich [2]University of Tübingen, Tübingen AI Center [3]Microsoft [4]University of Bonn
haofeixu.github.io/depthsplat

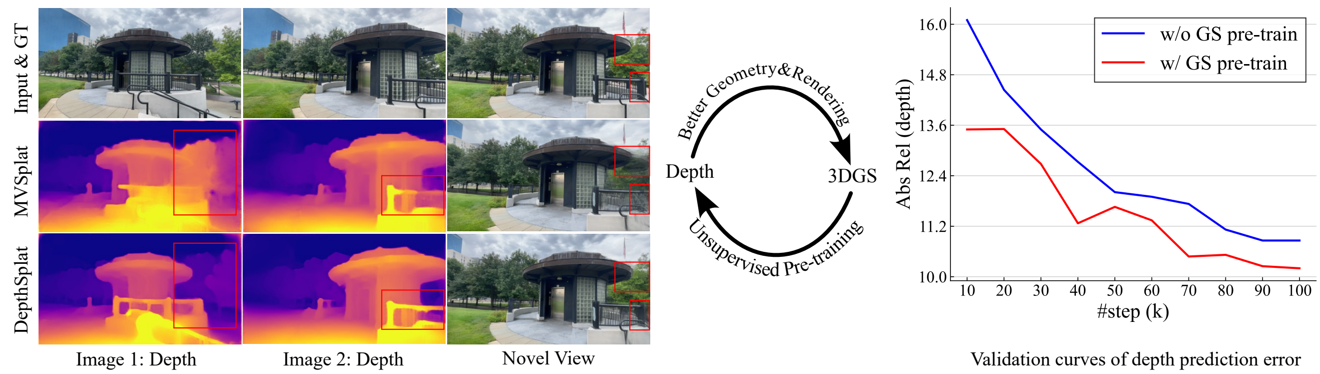

Figure 1. **DepthSplat enables cross-task interactions between Gaussian splatting and depth**. *Left:* Better depth leads to improved Gaussian splatting reconstruction. *Right:* Unsupervised depth pre-training with Gaussian splatting leads to reduced depth prediction error.

Abstract

Gaussian splatting and single-view depth estimation are typically studied in isolation. In this paper, we present DepthSplat to connect Gaussian splatting and depth estimation and study their interactions. More specifically, we first contribute a robust multi-view depth model by leveraging pre-trained monocular depth features, leading to high-quality feed-forward 3D Gaussian splatting reconstructions. We also show that Gaussian splatting can serve as an unsupervised pre-training objective for learning powerful depth models from large-scale multi-view posed datasets. We validate the synergy between Gaussian splatting and depth estimation through extensive ablation and cross-task transfer experiments. Our DepthSplat achieves state-of-the-art performance on ScanNet, RealEstate10K and DL3DV datasets in terms of both depth estimation and novel view synthesis, demonstrating the mutual benefits of connecting both tasks. In addition, DepthSplat enables feed-forward reconstruction from 12 input views (512×960 *resolutions) in 0.6 seconds.*

1. Introduction

Novel view synthesis [4, 82] and depth prediction [20, 44] are two fundamental tasks in computer vision, serving as the driving force behind numerous applications ranging from augmented reality to robotics and autonomous driving. There have been notable advancements in both areas recently.

For novel view synthesis, 3D Gaussian Splatting (3DGS) [29] has emerged as a popular technique due to its impressive real-time performance while attaining high visual fidelity. Recently, advances in feed-forward 3DGS models [7, 9, 48] have been introduced to alleviate the need for tedious per-scene optimization, also enabling few-view 3D reconstruction. The state-of-the-art sparse-view method MVSplat [9] relies on feature matching-based multi-view depth estimation [65] to localize the 3D Gaussian positions, which makes it suffer from similar limitations (*e.g.*, occlusions, texture-less regions, and reflective surfaces) as other multi-view depth methods [18, 25, 44, 56, 71].

On the other hand, significant progress has been made in monocular depth estimation, with recent models [19, 22, 28, 40, 69, 74] achieving robust predictions on diverse in-the-wild data. However, these depths typically lack consistent scales across multiple views, constraining their performance in downstream tasks like 3D reconstruction [59, 73].

The integration of 3DGS with single-view depth estimation presents a compelling solution to overcome the individual limitations of each technique while at the same time enhancing their strengths. To this end, we introduce

DepthSplat, which exploits the complementary nature of sparse-view feed-forward 3DGS and robust monocular depth estimation to improve the performance for both tasks.

Specifically, we first contribute a robust multi-view depth model by integrating pre-trained monocular depth features [70] to the multi-view feature matching branch, which not only maintains the consistency of multi-view depth models but also leads to more robust results in situations that are hard to match (*e.g*., occlusions, texture-less regions and reflective surfaces). The predicted multi-view depth maps are then unprojected to 3D as the Gaussian centers, and we use an additional lightweight network to predict other remaining Gaussian parameters. They are combined together to achieve novel view synthesis with the splatting operation [29].

While previous methods [1, 11, 31] also try to fuse monocular and multi-view depths, they usually rely on sophisticated architectures. In contrast, we identify the power of *off-the-shelf* pre-trained monocular depth models and propose to augment multi-view cost volumes with monocular features, leading to a simpler model and stronger performance.

Thanks to our improved multi-view depth model, the quality of novel view synthesis with Gaussian splatting is also significantly enhanced (see Fig. 1 left). In addition, our Gaussian splatting module is fully differentiable, which requires only photometric supervision to optimize all model components. This provides a new, unsupervised way to pre-train depth prediction models on large-scale multi-view posed datasets without requiring ground truth geometry information. The pre-trained depth model can be further fine-tuned for specific depth tasks and achieves superior results over training from scratch (see Fig. 1 right, where unsupervised pre-training leads to improved performance).

We conduct extensive experiments on the large-scale TartanAir [58], ScanNet [14] and RealEstate10K [82] datasets for depth estimation and Gaussian splatting tasks, as well as the recently introduced DL3DV [32] dataset, which features complex real-world scenes and thus is more challenging. Under various evaluation settings, our DepthSplat achieves state-of-the-art results. We also achieve promising feed-forward reconstruction results of large-scale or 360 scenes from 12 input views (512×960 resolutions) in 0.6 seconds. The strong performance on both tasks demonstrates the mutual benefits of connecting Gaussian splatting and depth.

2. Related Work

Multi-View Depth Estimation. As a core component of classical multi-view stereo pipelines [44], multi-view depth estimation exploits multi-view photometric consistency across multiple images to perform feature matching and predict the depth map of the reference image. In recent years, many learning-based methods [6, 15, 18, 25, 56, 57, 71] have been proposed to improve depth accuracy. Though these learning-based methods significantly improve the depth quality compared to traditional methods [23, 44, 67], they cannot handle challenging situations where the multi-view photometric consistency assumption is not guaranteed, *e.g*., occlusions, low-textured areas, and non-Lambertian surfaces.

Monocular Depth Estimation. Recently, we have witnessed significant progress in depth estimation from a single image [2, 28, 40, 69, 74], and existing methods can produce surprisingly accurate results on diverse in-the-wild data. However, monocular depth methods inherently suffer from scale ambiguities, which makes it challenging to use them for downstream tasks like 3D reconstruction [73]. In this paper, we leverage the powerful features from a pre-trained monocular depth model [70] to augment feature-matching based multi-view depth estimation, which not only maintains the multi-view consistency but also leads to significantly improved robustness in challenging situations such as low-textured regions and reflective surfaces.

Feed-Forward Gaussian Splatting. Several feed-forward 3D Gaussian splatting models [7–10, 48, 51, 62, 68, 77] have been proposed in literature thanks to its efficiency and ability to handle sparse views. In particular, pixelSplat [7] and Splatter Image [48] predict 3D Gaussians from image features, while MVSplat [9] encodes the feature matching information with cost volumes and achieves better geometry. However, it inherently suffers from the limitation of feature matching in challenging situations like texture-less regions and reflective surfaces. In this paper, we propose to integrate monocular features from pre-trained monocular depth models [70] for more robust depth prediction and 3D Gaussian reconstruction. Another line of work like LGM [51], GRM [68], and GS-LRM [77] relies significantly on the training data and compute, discarding explicit feature matching cues and learning priors purely from data. This makes them expensive to train (*e.g*., GS-LRM [77] is trained with 64 A100 GPUs for 2 days), while our model can be trained in 2 days with 4 GPUs. Moreover, our Gaussian splatting module enables pre-training depth model from large-scale multi-view posed datasets without ground truth depth supervision.

Depth and Gaussian Splatting. Flash3D [49] explores a pre-trained monocular depth model [39] for single-image 3D Gaussian reconstruction. In contrast, we target large-scale scene reconstruction from multiple views. TranSplat [76] leverages monocular depth to improve Gaussian reconstruction. However, a crucial difference is that TranSplat uses monocular features to refine the coarse depth predicted from a cost volume, which makes it inherently vulnerable to the error in the coarse stage and, thus, suffers from the error propagation issue. In contrast, we avoid this by combining cost volume and monocular features early, which performs significantly better than TranSplat (Tab. 6). Another line of work [12, 53] applies an additional depth loss to the Gaussian optimization process. We note that these two approaches (feed-forward *vs.* per-scene optimization) are orthogonal.

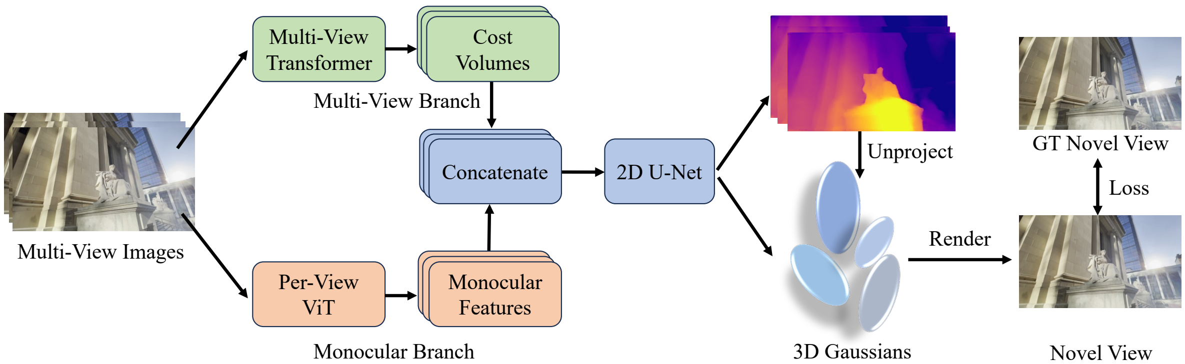

Figure 2. **DepthSplat** connects depth estimation and 3D Gaussian splatting with a shared architecture, which enables cross-task transfer. In particular, DepthSplat consists of a multi-view branch to model feature-matching information and a single-view branch to extract monocular features. The per-view cost volumes and monocular features are concatenated for depth regression with a 2D U-Net architecture. For the depth estimation task, we train the depth model with ground truth depth supervision. For the Gaussian splatting task, we first unproject all depth maps to 3D as the Gaussian centers, and in parallel, we use an additional head to predict the remaining Gaussian parameters. Novel views are rendered with the splatting operation. The full model for novel view synthesis is trained with the photometric rendering loss, which can also be used as an unsupervised pre-training stage for the depth model.

3. DepthSplat

Given N input images $\{\boldsymbol{I}^i\}_{i=1}^N$, ($\boldsymbol{I}^i \in \mathbb{R}^{H\times W\times 3}$, where H and W are the image sizes) with corresponding projection matrices $\{\mathbf{P}_i\}_{i=1}^N$, ($\mathbf{P}_i \in \mathbb{R}^{3\times 4}$, computed from the intrinsic and extrinsic matrices), our goal is to predict dense per-pixel depth $\mathbf{D}_i \in \mathbb{R}^{H\times W}$ and per-pixel Gaussian parameters $\{(\boldsymbol{\mu}_j, \alpha_j, \boldsymbol{\Sigma}_j, \boldsymbol{c}_j)\}_{j=1}^{H\times W\times N}$ for each image, where $\boldsymbol{\mu}_j$, α_j, $\boldsymbol{\Sigma}_j$ and $\boldsymbol{c}_j$ are the 3D Gaussian's position, opacity, covariance, and color. As shown in Fig. 2, key to our method is a multi-view depth model augmented with monocular depth features, where we obtain the position $\boldsymbol{\mu}_j$ of each Gaussian by unprojecting depth with camera parameters, and other Gaussian parameters are predicted by an additional head.

More specifically, our depth model consists of two branches: one for modeling feature matching using cost volumes, and another for extracting monocular features from a pre-trained monocular depth network. The cost volumes and monocular features are concatenated together for subsequent depth regression with a 2D U-Net and a softmax layer. For the depth task, we train our depth model with ground truth depth supervision. Our full model for novel view synthesis is trained with the rendering loss, which can also be used as an unsupervised pre-training stage for the depth model. We introduce the individual components below.

3.1. Multi-View Feature Matching

In this branch, we extract multi-view features with a multi-view Transformer architecture and then build multiple cost volumes that correspond to each input view.

Multi-View Feature Extraction. For N input images, we first use a lightweight weight-sharing ResNet [27] architecture to get $s\times$ downsampled features for each image independently. To handle different image resolutions, we make the downsampling factor s flexible by controlling the number of stride-2 3×3 convolutions. For example, the downsampling factor s is 4 if two stride-2 convolutions are used and 8 if three are used. To exchange information across different views, we use a multi-view Swin Transformer [34, 64, 65] which contains six stacked self- and cross-attention layers to obtain multi-view-aware features $\{\boldsymbol{F}_i\}_{i=1}^N$, $\boldsymbol{F}^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$, where C is the feature dimension. When more than two images ($N > 2$) are given as input, we perform cross-attention between each reference view and its top-2 nearest neighboring views, which are selected based on their camera distances to the reference view. This makes computation tractable with many input views.

Feature Matching. We encode the feature matching information across different views with the plane-sweep stereo approach [13, 65]. More specifically, for each view i, we first uniformly sample D depth candidates $\{d_m\}_{m=1}^D$ from the near and far depth ranges and then warp the feature $\boldsymbol{F}^j$ of view j to view i with the camera projection matrix and each depth candidate d_m. Then we obtain D warped features $\{\boldsymbol{F}_{d_m}^{j\to i}\}_{m=1}^D$ that correspond to feature $\boldsymbol{F}^i$. We then measure their feature correlations with the dot-product operation [9, 63]. The cost volume $\boldsymbol{C}_i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times D}$ for image i is obtained by stacking all correlations. Accordingly, we obtain cost volumes $\{\boldsymbol{C}_i\}_{i=1}^N$ for all input images $\{\boldsymbol{I}_i\}_{i=1}^N$. For more than two input views, similar to the strategy in cross-view attention computation, we select the top-2 nearest views for each reference view and compute feature correlations with only the selected views. This enables our cost volume construction to achieve a good speed-accuracy trade-off and scale efficiently to a larger number of input views. The correlation values for the two selected views are combined with averaging.

3.2. Monocular Depth Feature Extraction

Despite the remarkable progress in multi-view feature matching-based depth estimation [57, 65, 71] and Gaussian splatting [9], they inherently suffer from limitations in challenging situations like occlusions, texture-less regions, and reflective surfaces. Thus, we propose to integrate pre-trained monocular depth features into the cost volume to handle scenarios that are challenging or impossible to match.

More specifically, we leverage the pre-trained monocular depth backbone from the recent Depth Anything V2 [70] model thanks to its impressive performance on diverse in-the-wild data. The monocular backbone is a ViT [16, 37] model, which has a patch size of 14 and outputs a feature map that is $1/14$ spatial resolution of the original image. We simply bilinearly interpolate the spatial resolution of the monocular features to the same resolution as the cost volume in Sec. 3.1 and obtain the monocular feature $\boldsymbol{F}^i_{\text{mono}} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C_{\text{mono}}}$ for input image $\boldsymbol{I}_i$, where C_{mono} is the dimension of the monocular feature. This process is performed for all input images in parallel and we obtain monocular features $\{\boldsymbol{F}^i_{\text{mono}} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C_{\text{mono}}}\}_{i=1}^N$, which are subsequently used for per-view depth map estimations.

3.3. Feature Fusion and Depth Regression

To achieve robust and multi-view consistent depth predictions, we combine the monocular feature $\boldsymbol{F}^i_{\text{mono}} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C_{\text{mono}}}$ and cost volume $\boldsymbol{C}_i \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times D}$ via simple concatenation in the channel dimension. A subsequent 2D U-Net [42, 43] is used to regress depth from the concatenated monocular features and cost volumes. This process is performed for all the input images in parallel and for each image, it outputs a tensor of shape $\frac{H}{s} \times \frac{W}{s} \times D$, where D is the number of depth candidates. We normalize the D dimension with the softmax operation and perform a weighted average of all depth candidates to obtain the depth output.

We also apply a hierarchical matching [25] architecture where an additional refinement step at $2\times$ higher feature resolution is employed to improve the performance further. More specifically, based on the coarse depth prediction, we perform a correspondence search on the $2\times$ higher feature maps within the neighbors of the $2\times$ upsampled coarse depth prediction. Since we already have a coarse depth prediction, we only need to search a smaller range at the higher resolution, and thus, we construct a smaller cost volume compared to the coarse stage. Such a 2-scale hierarchical architecture not only leads to improved efficiency since most computation is spent on low resolution, but also leads to better results thanks to the use of higher-resolution features [25]. Similar feature fusion and depth regression procedure is used to get higher-resolution depth predictions, which are subsequently upsampled with a DPT head [41] to the full resolution.

3.4. Gaussian Parameter Prediction

For the task of 3D Gaussian splatting, we directly unproject the per-pixel depth maps to 3D with the camera parameters as the Gaussian centers $\boldsymbol{\mu}_j$. To predict other remaining Gaussian parameters α_j (opacity), $\boldsymbol{\Sigma}_j$ (covariance) and $\boldsymbol{c}_j$ (color), we use an additional DPT head [41], where it takes the concatenated image, depth and feature information as input and outputs all the remaining Gaussian parameters. With all the predicted 3D Gaussians, we can render novel views with the Gaussian splatting operation [29].

3.5. Training Loss

We study the properties of our proposed model on two tasks: depth estimation and novel view synthesis with 3D Gaussian splatting [29]. The loss functions are introduced below.

Depth estimation. We train our depth model (without the Gaussian splatting head) with ℓ_1 loss and gradient loss between the inverse depths of prediction and ground truth:

$$
\begin{aligned}
L_{\text{depth}} = \alpha \cdot |\boldsymbol{D}_{\text{pred}} - \boldsymbol{D}_{\text{gt}}| + \beta \cdot |\partial_x \boldsymbol{D}_{\text{pred}} - \partial_x \boldsymbol{D}_{\text{gt}}| \\
+ \beta \cdot |\partial_y \boldsymbol{D}_{\text{pred}} - \partial_y \boldsymbol{D}_{\text{gt}}|),
\end{aligned} \tag{1}
$$

where ∂_x and ∂_y denotes the gradients on the x and y directions, respectively. Following UniMatch [65], we use $\alpha = 20$ and $\beta = 20$.

View synthesis. We train our full model with a combination of mean squared error (MSE) and LPIPS [78] losses between rendered and ground truth image colors:

$$
L_{\text{gs}} = \sum_{m=1}^{M} \left(\text{MSE}(I^m_{\text{render}}, I^m_{\text{gt}}) + \lambda \cdot \text{LPIPS}(I^m_{\text{render}}, I^m_{\text{gt}}) \right), \tag{2}
$$

where M is the number of novel views to render in a single forward pass. The LPIPS loss weight λ is set to 0.05 [9].

4. Experiments

Implementation Details. We implement our method in PyTorch [38] and optimize our model with the AdamW [36] optimizer and cosine learning rate schedule. We adopt the xFormers [30] library for our monocular ViT backbone implementation. We use a lower learning rate 2×10^{-6} for the pre-trained Depth Anything V2 [70] backbone, and other remaining layers use a learning rate of 2×10^{-4}. The feature downsampling factor s in our multi-view branch (Sec. 3.1) is chosen based on the image resolution. More specifically, for experiments on the 256×256 resolution RealEstate10K [82] dataset, we choose $s = 4$. For higher resolution datasets (*e.g.*, TartanAir [58], ScanNet [14], KITTI [24] and DL3DV [32]), we choose $s = 8$. Our hierarchical matching models in Sec. 3.3 use 2-scale features, *i.e.*, $1/8$ and $1/4$, or $1/4$ and $1/2$ resolutions.

Table 1. **DepthSplat model variants**. We evaluate different monocular backbones and different multi-view models (1-scale and 2-scale features for hierarchical matching as described in Sec. 3.3), where both larger monocular backbones and 2-scale hierarchical models lead to consistently improved performance for both depth and view synthesis tasks. The inference time is measured on a single A100 GPU for two input views at 256×256 resolutions.

Mono	Multi-View	Depth (TartanAir)		3DGS (RealEstate10K)			Param	Time
		Abs Rel ↓	δ_1 ↑	PSNR ↑	SSIM ↑	LPIPS ↓	(M)	(s)
ViT-S	1-scale	8.46	93.02	26.84	0.878	0.122	38	0.050
ViT-B	1-scale	6.94	94.46	27.06	0.882	0.119	115	0.053
ViT-L	1-scale	**6.07**	**95.52**	**27.23**	**0.885**	**0.116**	347	0.061
ViT-S	2-scale	7.01	94.56	27.12	0.884	0.119	41	0.062
ViT-B	2-scale	6.22	95.31	27.34	0.887	0.116	120	0.070
ViT-L	2-scale	**5.57**	**96.07**	**27.47**	**0.889**	**0.114**	354	0.079

Training Details. For depth experiments on ScanNet [14], we train our depth model on $4\times$ GH200 GPUs for 100K iterations with a total batch size of 32. For the ablation experiments on depth task, we mainly use synthetic datasets (TartanAir [58] and VKITTI2 [5]) for training since they provide high-quality ground truth depths. For Gaussian splatting experiments, we consider both low-resolution and high-resolution images. In particular, for comparisons with previous methods [7, 9], we mainly use 256×256 resolution RealEstate10K [82] and 256×448 resolution DL3DV [32] datasets. We also report high-resolution (512×960) results on RealEstate10K and DL3DV datasets for qualitative evaluations. For experiments on the 256×256 RealEstate10K dataset, we train our model on $4\times$ GH200 GPUs for 150K iterations with a total batch size of 32, which takes 1 day for the small model and 2 days for the large model. For experiments on the 256×448 DL3DV [32] dataset, we fine-tune our RealEstate10K pre-trained model on $4\times$ GH200 GPUs for 100K iterations with a total batch size of 4, where the number of input views is randomly sampled from 2 to 6. We evaluate the model's performance on different number of input views (2, 4, 6). For high-resolution results, we fine-tune our low-resolution pre-trained models on high-resolution images. More training details are provided in the appendix. Our code and training scripts are available at github.com/cvg/depthsplat to ease reproducibility.

4.1. Model Variants

We first study several different model variants for both depth estimation and view synthesis tasks. In particular, we explore different monocular backbones [70] (ViT-S, ViT-B, ViT-L) and different multi-view models (1-scale and 2-scale). We conduct depth experiments on the large-scale TartanAir [58] synthetic dataset, which features both indoor and outdoor scenes and has perfect ground truth depth. The Gaussian splatting experiments are conducted on the standard RealEstate10K [82] dataset. Following community standards, we report the depth evaluation metrics [20] of Abs Rel (relative ℓ_1 error) and δ_1 (percentage of correctly estimated pixels within a threshold) and novel view synthesis metrics [29] of PSNR, SSIM, and LPIPS. The results in Tab. 1 demonstrate that both larger monocular backbones and 2-scale hierarchical models lead to consistently improved performance for both tasks. We name three different model sizes with "Small": "ViT-S monocular, 1-scale multi-view" , "Base": "ViT-B monocular, 2-scale multi-view", and "Large": "ViT-L monocular, 2-scale multi-view".

From Tab. 1, we can also observe that better depth network architecture leads to improved view synthesis. In the appendix, we conduct additional experiments to study the effect of different initializations for the depth network to the view synthesis performance. Our results show that a better depth model initialization also contributes to improved rendering quality. In summary, both better depth network architecture and better depth model initialization lead to improved novel view synthesis results.

Table 2. **Ablations**. We evaluate the contribution of the monocular feature branch and the cost volume branch (1st group of experiments), as well as different monocular features (2nd group of experiments). Our results indicate that the monocular feature and cost volume are complementary, with large performance drops when removing either one. The pre-trained Depth Anything V2 model weights achieve the best view synthesis results.

Components	Depth (TartanAir)		3DGS (RealEstate10K)		
	Abs Rel ↓	δ_1 ↑	PSNR ↑	SSIM ↑	LPIPS ↓
Full	**8.46**	**93.02**	**26.84**	**0.878**	**0.122**
w/o mono feature	12.25	88.00	26.04	0.864	0.134
w/o cost volume	11.34	90.02	23.24	0.766	0.184
ConvNeXt-T	10.50	91.13	26.11	0.865	0.134
Midas	9.53	91.61	26.46	0.872	0.128
DINO V2	8.93	92.49	26.76	0.877	0.123
Depth Anything V1	**8.38**	**93.23**	26.76	0.877	0.124
Depth Anything V2	8.46	93.02	**26.84**	**0.878**	**0.122**

4.2. Ablation and Analysis

In this section, we study the properties of our key components on the TartanAir dataset (for depth) and RealEstate10K dataset (for view synthesis with Gaussian splatting).

Monocular Features for Depth and Gaussian Splatting. In Tab. 2, we compare our full model (Full) with the model variant where the monocular depth feature branch (with a ViT-S model pre-trained by Depth Anything V2 [70]) is removed (w/o mono feature), leaving only the multi-view branch. We can observe a clear performance drop for both depth and view synthesis tasks. In Fig. 3, we visualize the depth predictions and error maps of both models on the ScanNet dataset. The pure multi-view feature matching-based approach struggles a lot at texture-less regions and reflective surfaces. At the same time, our full model achieves reliable results thanks to the powerful prior captured in monocular

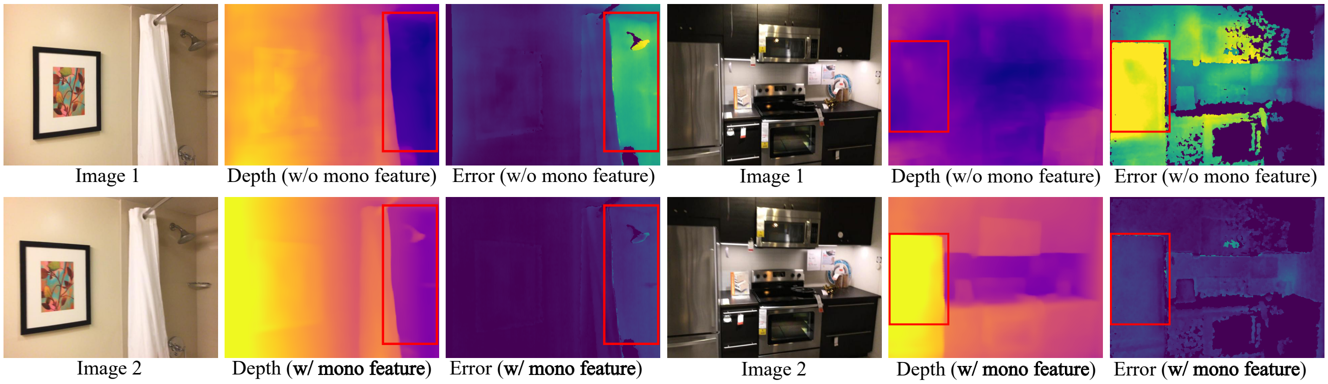

Figure 3. **Effect of monocular features for depth on ScanNet**. The monocular feature greatly improves challenging situations like texture-less regions (*e.g.*, the wall in the first example) and reflective surfaces (*e.g.*, the refrigerator in the second example).

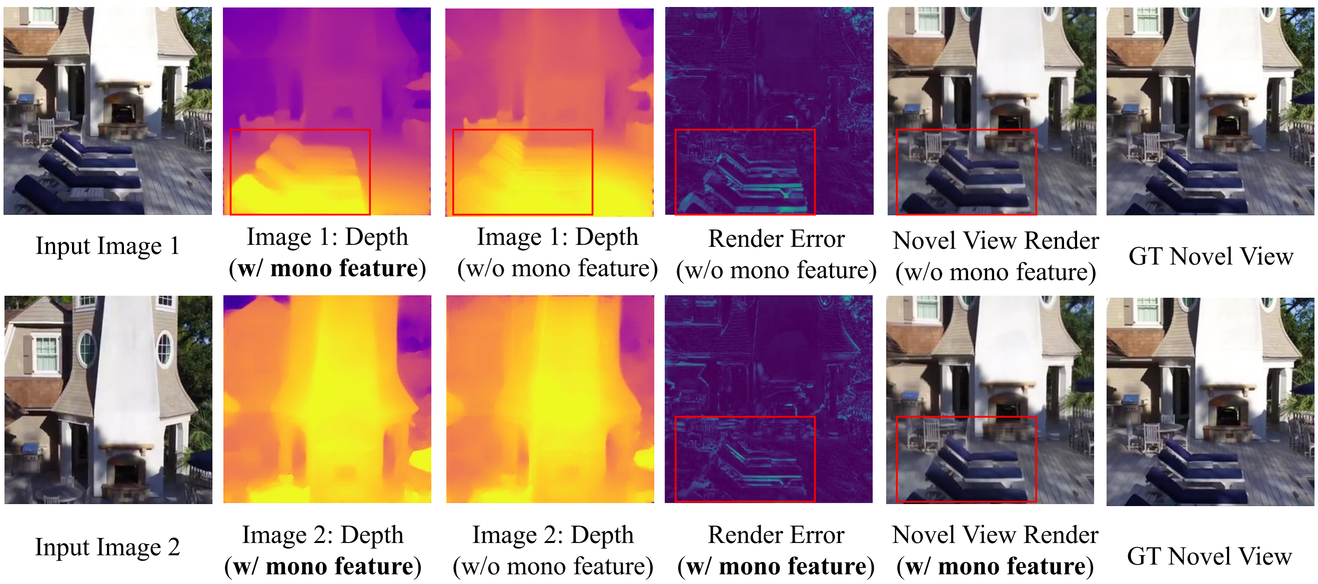

Figure 4. **Effect of monocular features for 3DGS on RealEstate10K**. Without monocular features, the model struggles at predicting reliable depth for pixels that are not able to find correspondences (*e.g.*, the lounger chair highlighted with the read rectangle), which subsequently causes misalignment in the rendered image due to the incorrect geometry.

depth features. We also show the visual comparisons for the Gaussian splatting task in Fig. 4 with two input views. For regions (*e.g.*, the lounge chairs) that only appear in a single image, the pure multi-view method is unable to find correspondences. It thus produces unreliable depth predictions, leading to misalignment in the rendered novel views due to the incorrect geometry.

We also experiment with removing the cost volume (w/o cost volume) in the multi-view branch and observed a significant performance drop. This indicates that obtaining scale- and multi-view consistent predictions with a pure monocular depth backbone is challenging, which constrains achieving high-quality results for 3D Gaussian reconstruction.

Different Monocular Features. In Tab. 2, we also evaluate different monocular features, including the ConvNeXt-T [35] features used in AFNet [11], and other popular monocular features including Midas [40] and DINOv2 [37]. The pre-trained Depth Anything V2 [70] monocular features achieve the best view synthesis results.

Table 3. **Monocular fusion strategy**. Our simple concatenation performs surprisingly good compared to other alternatives.

Method	Abs Rel ↓	δ_1 ↑
Cost volume from ViT	11.26	90.84
Explicit scale align	9.13	90.21
Attention fusion	**8.40**	92.95
Concatenation (Ours)	8.46	**93.02**

Monocular Fusion Strategy. We compare with alternative strategies for fusing the monocular features for multi-view

Table 4. **Unsupervised Depth Pre-Training with Gaussian Splatting**. We evaluate the same architecture but with different weights. The pre-training is performed on RealEstate10K, and fine-tuning is performed on TartanAir and VKITTI2. Unsupervised pre-training (2nd row) produces improved depth predictions compared to random initialization (1st). The performance can be further improved with additional fine-tuning (4th), where the benefit of pre-training is especially significant on challenging datasets like TartanAir and KITTI compared to training from random initialization (3rd).

Pre-Train	Fine-Tune	TartanAir		ScanNet		KITTI	
		Abs Rel ↓	δ_1 ↑	Abs Rel ↓	δ_1 ↑	Abs Rel ↓	δ_1 ↑
✗	✗	76.22	3.43	39.97	22.56	90.69	0.00
✓	✗	29.53	53.16	21.51	76.09	56.83	46.26
✗	✓	10.86	90.55	6.70	96.14	11.56	87.27
✓	✓	**10.20**	**91.10**	**6.60**	**96.27**	**10.68**	**89.92**

Table 5. **Two-view Depth Estimation on ScanNet.** Our DepthSplat outperforms all prior methods by clear margins.

Method	Abs Rel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓
DeMoN [54]	23.1	0.761	0.289
BA-Net [50]	16.1	0.346	0.214
DeepV2D [52]	5.7	0.168	0.080
NeuralRecon [47]	4.7	0.164	0.093
DRO [26]	5.3	0.168	0.081
UniMatch [65]	5.9	0.179	0.082
DepthSplat (w/o GS pre-train)	4.5	0.125	0.061
DepthSplat (w/ GS pre-train)	**3.8**	**0.114**	**0.055**

Table 6. **Two-view 3DGS on RealEstate10K**. Our DepthSplat achieves the best performance.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
pixelNeRF [75]	20.43	0.589	0.550
GPNR [46]	24.11	0.793	0.255
AttnRend [17]	24.78	0.820	0.213
MuRF [66]	26.10	0.858	0.143
pixelSplat [7]	25.89	0.858	0.142
MVSplat [9]	26.39	0.869	0.128
TranSplat [76]	26.69	0.875	0.125
DepthSplat	**27.47**	**0.889**	**0.114**

Figure 5. **Novel view synthesis on RealEstate10K**. Our DepthSplat performs significantly better than pixelSplat [7] and MVSplat [9] in challenging regions.

depth estimation in Sec. 4.2. In particular, we first compare with MVSFormer [6] that constructs the cost volume with monocular ViT features, which leads to a single-branch architecture. We can observe that our fusion strategy performs significantly better than the single-branch design, potentially because our two-branch design disentangles feature matching and monocular priors, which makes the learning task easier. We also compare with fusing the monocular depth maps (instead of monocular features in our model) by first aligning the monocular depth scale using the multi-view depth and then fusing the depth maps together with an additional network. Again, our simple concatenation performs better. We further compare our concatenation with the attention-based adaptive fusion and observe very similar results. Thus we opt to the simpler concatenation design.

4.3. Unsupervised Depth Pre-Training with Gaussian Splatting

By connecting Gaussian splatting and depth, our DepthSplat provides a way to pre-train the depth model in a fully unsupervised manner. In particular, we first train our full model on the large-scale multi-view posed RealEstate10K dataset (contains $\sim$ 67K Youtube videos) with only the Gaussian splatting rendering loss (Eq. (2)), without any direct supervision on the depth predictions. After pre-training, we take the pre-trained depth model and further fine-tune it to the depth task on the mixed TartanAir [58] and VKITTI2 [5] datasets with ground truth depth supervision. In Tab. 4, we evaluate the performance on both in-domain TartanAir test set and the zero-shot generalization on unseen ScanNet and KITTI datasets. We can observe that the benefit of pre-training is especially significant on the challenging datasets like TartanAir and KITTI. The results align well with the observation in [21] which suggests that the pre-training acts as regularization and guides the learning towards better minima and enables better generalization to out-of-distribution datasets. The visual comparison results are provided in the appendix. Given the increasing popularity of view synthesis [61, 81] and multi-view generative models [3, 45], new multi-view datasets [32] and models [55] are gradually introduced, our approach provides a way to pre-train depth models on large-scale multi-view posed image datasets.

4.4. Benchmark Comparisons

Comparisons on ScanNet and RealEstate10K. Tab. 5 and Tab. 6 compare the depth and novel view synthesis results on the standard ScanNet and RealEstate10K benchmarks, respectively. We can see clearly that our DepthSplat achieves state-of-the-art performance on both datasets for both tasks. The visual comparison with previous methods is shown in Fig. 5, where our method significantly improves the perfor-

Table 7. **Comparisons on DL3DV**. Our DepthSplat not only consistently outperforms MVSplat on different number of input views, but also scales more efficiently to more input views.

Method	#Views	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s)
MVSplat [9]	2	17.54	0.529	0.402	**0.072**
DepthSplat		**19.31**	**0.615**	**0.310**	0.083
MVSplat [9]	4	21.63	0.721	0.233	0.146
DepthSplat		**23.12**	**0.780**	**0.178**	**0.107**
MVSplat [9]	6	22.93	0.775	0.193	0.263
DepthSplat		**24.19**	**0.823**	**0.147**	**0.132**

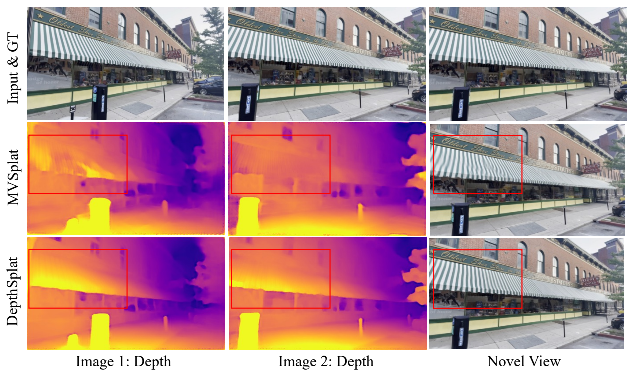

Figure 6. **Visual comparisons on DL3DV**. Our DepthSplat performs significantly better than MVSplat [9] on regions that hard to match (*e.g.*, repeated patterns).

mance on challenging scenarios like texture-less regions and occlusions. For the depth results on ScanNet, we report our base models (with ViT-B and two-scale feature matching) trained with and without Gaussian splatting pre-training on the mixed RealEstate10K and DL3DV datasets. Our model significantly outperforms previous methods and can benefit from additional Gaussian splatting pre-training.

Comparisons on DL3DV. To evaluate the performance on complex real-world scenes, we conduct comparisons with the representative model MVSplat [9] on the recently introduced DL3DV dataset [32]. We also compare the results of different numbers of input views (2, 4 and 6) on this dataset. We fine-tune MVSplat and our RealEstate10K pre-trained models on DL3DV training scenes and report the results on the benchmark test set in Tab. 7. Our DepthSplat consistently outperforms MVSplat in all metrics, and our improvements are more significant with fewer input views, which indicates that our model is more robust to sparse views. Visual comparisons with MVSplat on the DL3DV dataset are shown in Fig. 6, where MVSplat's depth quality lags behind our DepthSplat due to matching failure, which leads to blurry and distorted view synthesis results. We show more visual comparison results in the appendix. It is also worth noting

Table 8. **Cross-dataset generalization**. When generalizing to DL3DV and ACID with RealEstate10K pre-trained models, our DepthSplat is consistently better than MVSplat, especially on the more challenging DL3DV dataset.

Method	DL3DV			ACID		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MVSplat [9]	24.14	0.808	0.149	28.15	0.841	0.147
DepthSplat	**27.66**	**0.895**	**0.096**	**28.37**	**0.847**	**0.141**

that our method scales more efficiently to more input views thanks to our lightweight local feature matching approach (Sec. 3.1), which is unlike the expensive global pair-wise matching used in MVSplat.

Cross-Dataset Generalization. We evaluate the cross-dataset generalization capability by directly testing on DL3DV [32] and ACID [33] datasets with RealEstate10K [82] pre-trained models. The results are shown in Tab. 8, where our DepthSplat consistently outperforms MVSplat on both datasets, especially on the more challenging DL3DV dataset which features complex scene structures. The visual comparison results are shown in the appendix.

High-Resolution Results. In addition to the comparison results on the 256 × 256 resolution RealEstate10K and 256 × 448 resolution DL3DV dataets, we show more high-resolution (512 × 960) qualitative results in the appendix. We also invite the readers to our project page haofeixu.github.io/depthsplat for high-resolution (512 × 960) video results on different number of input views (6 and 12), where our DepthSplat is able to reconstruct larger-scale or 360 scenes from more input views.

5. Conclusion

In this paper, we introduce DepthSplat, a new approach to connecting Gaussian splatting and depth to achieve state-of-the-art results on ScanNet, RealEstate10K and DL3DV datasets for both depth and view synthesis tasks. We show that our model enables unsupervised depth pre-training with Gaussian splatting rendering loss, providing a way to leverage large-scale posed multi-view image datasets for training more robust and generalizable multi-view depth models.

Limitations. Our current model requires camera poses as input along with the multi-view images, which might be challenging to obtain when the input views are extremely sparse. An interesting future direction is to remove this requiement by exploring pose-free models [72, 80]. In addition, our model predicts pixel-aligned Gaussians, where the number of Gaussians will be significantly large when processing a large number of input views [60, 79]. Further improving the geometry representation and the scalability to many input views would be exciting future work.

Acknowledgement

We thank Yuedong Chen for his generous help with the DL3DV dataset. Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533) and the DFG EXC number 2064/1 - project number 390727645. This work was supported as part of the Swiss AI Initiative by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID a03 on Alps. H. Blum is also with the Lamarr Institute and D. Barath is also with HUN-REN SZTAKI.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *CVPR*, 2022. 2

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Zoedepth: Zero-shot transfer by combining relative and metric depth. In *CVPR*, 2023. 2

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. 2023. 7

[4] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *ACM TOG*, 2001. 1

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 5, 7

[6] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer: Learning robust image representations via transformers and temperature-based depth for multi-view stereo. 2022. 2, 7

[7] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 1, 2, 5, 7

[8] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *ECCV*, 2024.

[9] 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. In *ECCV*, 2024. 1, 2, 3, 4, 5, 7, 8

[10] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In *NeurIPS*, 2024. 2

[11] JunDa Cheng, Wei Yin, Kaixuan Wang, Xiaozhi Chen, Shijie Wang, and Xin Yang. Adaptive fusion of single-view and multi-view depth for autonomous driving. In *CVPR*, 2024. 2, 6

[12] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *CVPR*, 2024. 2

[13] Robert T Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996. 3

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 4, 5

[15] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. *CVPR*, 2022. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. 4

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

[18] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *CVPR*, 2021. 1, 2

[19] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021. 1

[20] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. 2014. 1, 5

[21] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *JMLR*, 2010. 7

[22] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. 2024. 1

[23] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 2

[24] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4

[25] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020. 1, 2, 4

[26] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, Zilong Dong, and Ping Tan. Dro: Deep recurrent optimizer for video to depth. 2023. 7

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[28] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 1, 2

[29] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 1, 2, 4, 5

[30] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, et al. xformers: A modular and hackable transformer modelling library, 2022. 4

[31] Rui Li, Dong Gong, Wei Yin, Hao Chen, Yu Zhu, Kaixuan Wang, Xiaozhi Chen, Jinqiu Sun, and Yanning Zhang. Learning to fuse monocular and multi-view cues for multi-frame depth estimation in dynamic scenes. In *CVPR*, 2023. 2

[32] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 2, 4, 5, 7, 8

[33] 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. In *ICCV*, 2021. 8

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 6

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019. 4

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. 2023. 4, 6

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. 2019. 4

[39] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, 2024. 2

[40] 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 TPAMI*, 2020. 1, 2, 6

[41] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 4

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Springer, 2015. 4

[44] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 2

[45] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. 7

[46] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*. Springer, 2022. 7

[47] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, 2021. 7

[48] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 1, 2

[49] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In *3DV*, 2025. 2

[50] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment networks. In *ICLR*, 2018. 7

[51] 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. 2024. 2

[52] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2019. 7

[53] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *WACV*, 2025. 2

[54] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017. 7

[55] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. 2024. 7

[56] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, 2021. 1, 2

[57] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. In *CVPR*, 2022. 2, 4

[58] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 2, 4, 5, 7

[59] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *ICCV*, 2023. 1

[60] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. In *NeurIPS*, 2024. 8

[61] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. 2023. 7

[62] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *ECCV*, 2024. 2

[63] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, 2020. 3

[64] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 3

[65] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE TPAMI*, 2023. 1, 3, 4, 7

[66] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *CVPR*, 2024. 7

[67] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *CVPR*, 2019. 2

[68] 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. 2024. 2

[69] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1, 2

[70] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. 2024. 2, 4, 5, 6

[71] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 1, 2, 4

[72] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *ICLR*, 2025. 8

[73] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *IEEE TPAMI*, 2022. 1, 2

[74] 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 *CVPR*, 2023. 1, 2

[75] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 7

[76] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *AAAI*, 2025. 2, 7

[77] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. 2024. 2

[78] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[79] Shengjun Zhang, Xin Fei, Fangfu Liu, Haixu Song, and Yueqi Duan. Gaussian graph network: Learning efficient and generalizable gaussian representations from multi-view images. In *NeurIPS*, 2024. 8

[80] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, 2025. 8

[81] Chuanxia Zheng and Andrea Vedaldi. Free3d: Consistent novel view synthesis without 3d representation. In *CVPR*, 2024. 7

[82] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 2018. 1, 2, 4, 5, 8