MegaSynth: Scaling Up 3D Scene Reconstruction with Synthesized Data

Hanwen Jiang[1] Zexiang Xu[2] Desai Xie[3] Ziwen Chen[4] Haian Jin[5] Fujun Luan[2] Zhixin Shu[2]
Kai Zhang[2] Sai Bi[2] Xin Sun[2] Jiuxiang Gu[2] Qixing Huang[1] Georgios Pavlakos[1] Hao Tan[2]
[1]The University of Texas at Austin [2]Adobe Research
[3]Stony Brook University [4]Oregon State University [5]Cornell University

Project & Code: https://hwjiang1510.github.io/MegaSynth/

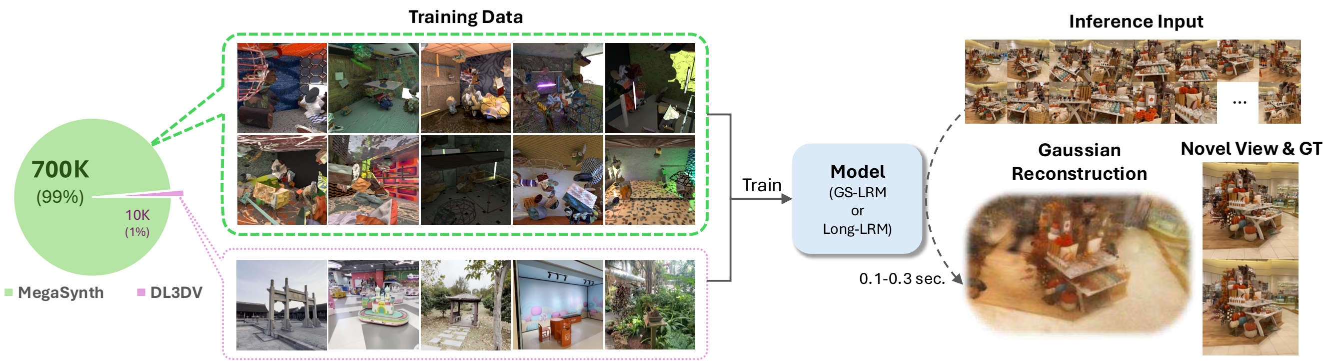

Figure 1. We introduce MegaSynth, a **non-semantic** synthesized dataset for training LRMs. MegaSynth benefits from its scalability and controllability, enabling us to generate 700K scenes in 3 days. We train LRMs with both the large-scale MegaSynth data and small-scale real data, improving LRMs for reconstructing wide-coverage scenes from dense-view images.

Abstract

We propose scaling up 3D scene reconstruction by training with synthesized data. At the core of our work is MegaSynth, a procedurally generated 3D dataset comprising 700K scenes—over 50 times larger than the prior real dataset DL3DV—dramatically scaling the training data. To enable scalable data generation, our key idea is eliminating semantic information, removing the need to model complex semantic priors such as object affordances and scene composition. Instead, we model scenes with basic spatial structures and geometry primitives, offering scalability. Besides, we control data complexity to facilitate training while loosely aligning it with real-world data distribution to benefit real-world generalization. We explore training LRMs with both MegaSynth and available real data. Experiment results show that joint training or pre-training with MegaSynth improves reconstruction quality by 1.2 to 1.8 dB PSNR across diverse image domains. Moreover, models trained solely on MegaSynth perform comparably to those trained on real data, underscoring the low-level nature of 3D reconstruction. Additionally, we provide an in-depth analysis of MegaSynth's properties for enhancing model capability, training stability, and generalization, as well as application to other tasks.

1. Introduction

The scaling law has shifted the focus of contemporary AI research toward large foundation models, which are built with scalable neural network architectures [27, 70] and trained on vast datasets [5, 61]. Following the scaling recipe seen in NLP and 2D vision [1, 3, 4], the Large Reconstruction Model (LRM) has been introduced to learn general 3D reconstruction priors [29]. For object-level reconstruction, LRMs have shown impressive reconstruction quality using either single-view or sparse-view inputs [29, 35, 72, 88], enabling a range of applications [40, 90].

Despite progress, enhancing LRM for reconstructing *wide-coverage scenes* remains challenging due to two key limitations of training data. First, scene-level datasets are significantly *smaller in scale* compared to object-level counterparts. For instance, Objaverse [17] contains 800K shape instances, whereas the largest clean scene dataset, DL3DV, includes just 10K scenes. Collecting more intentionally captured scene data is labor-intensive and difficult to scale. Second, existing scene-level datasets suffer from a *suboptimal data distribution*. They are often limited by insufficient scene diversity [15], small camera motions [49, 92], noisy content [69], and inaccurate annotations [41]. However, given the inherent complexity of 3D scenes, effective training requires clean and diverse data, especially multi-view images captured by widely spaced cameras with precise camera annotations [50].

In this work, we propose scaling up training data for scene-level reconstruction by using **synthesized data**. Our key idea is to **eliminate the reliance on semantic information in data generation**, by constructing scenes with non-semantic shape primitives arranged within basic spatial structures. This approach is motivated by our insight that *scene semantics play a minimal role in multi-view reconstruction*, as evidenced by the success of traditional non-semantic methods such as COLMAP [60], MVS [62], NeRF [50], and the emerging non-semantic properties of recent feed-forward models [9, 31, 79]. Unlike prior scene generation methods, which aim to replicate real-world scene distributions [20, 55, 56, 66, 75, 91] and are thus constrained by the complexity of modeling semantics, e.g. object affordances, our approach bypasses these challenges. This simplification enables highly scalable and efficient data generation.

Beyond scalability, synthesized data offers controllability. We control data complexity to facilitate training while loosely aligning it with real-world data distribution to benefit real-world generalization. Through heuristic methods, we regulate key factors, such as geometric complexity, camera pose distribution, materials, and lighting, for creating diverse scenes. Additionally, synthesized data provides precise metadata, such as camera and geometry information, further ensuring improved training stability and effectiveness.

We generate the **MegaSynth** dataset, comprising *700K* scenes. MegaSynth is *over 50 times larger* than the real dataset DL3DV and significantly scales up training data for LRMs. We utilize MegaSynth to train feed-forward LRMs [88, 93] jointly with DL3DV. Our experiments show a 1.2 to 1.8 dB PSNR gain across diverse test datasets and image resolutions. Moreover, the depth rendering quality is significantly improved, showing a better reconstruction geometry quality. These results highlight the synergy between synthesized and real data. Synthesized data excels in scale and provides rich metadata, such as geometry supervision, enabling models to develop a general geometric understanding beyond rendering supervision. Meanwhile, small-scale real data further sharpens the model. Interestingly, MegaSynth can also *benefit other 3D tasks*, where a monocular depth estimation model fine-tuned on MegaSynth demonstrates significant improvement.

2. Related Work

Scene-level 3D Reconstruction. Reconstructing scenes has been a long-standing challenge in 3D computer vision. Traditional Structure-from-Motion (SfM) and Multi-view Stereo (MVS) methods, as well as their neural counterparts, adopt a bottom-up approach [21, 22, 33, 60, 62, 64, 65, 71]. For instance, COLMAP [60] builds from low-level visual cues to more structured geometry through keypoint detection, matching, camera reconstruction, and bundle adjustment.

Learning-based methods encompass both 3D neural scene representations and feed-forward prediction models. Researchers have explored the distinct properties of explicit [51, 74], implicit [8, 44, 50, 63], and hybrid 3D representations [25, 30, 34, 37] to enhance reconstruction quality, typically optimizing the 3D representation for each scene to demonstrate capability. Meanwhile, generalizable reconstruction models have been developed, where neural networks predict 3D representation attributes in a feed-forward manner. Some approaches follow a traditional bottom-up paradigm, leveraging inductive biases from MVS [10, 73, 87], cost volume [12, 13], correspondence cues [11], and epipolar geometry [9, 19, 78]. In contrast, recent work proposes top-down frameworks [31, 39, 72, 74] that infer geometry directly and better harness the power of large models. However, some of these works rely on pairwise computations [74], which limits a global understanding of inputs. Our work, in contrast, leverages recent global-aware methods [88, 93] and focuses on scaling up training data to advance dense-view reconstruction.

Large Reconstruction Model (LRM). LRMs have been introduced to scale up generalizable 3D reconstruction methods [29], employing scalable network architectures and training on large datasets to learn generic reconstruction priors. Typically, LRMs use Transformers [29, 31, 70] or U-Nets [57, 67] as model backbones, encoding 2D image inputs into 3D representations, e.g., Triplane [29, 72] and mesh [76, 80], enabling high-quality object reconstruction. The following research has focused on enhancing object reconstruction by incorporating generative priors [81, 89] and designing more scalable training frameworks [26, 32, 79]. Additionally, novel 3D representations, such as 3D Gaussians [37], have extended LRMs to scene-level 3D reconstruction [88, 93]. However, reconstructing wide-coverage scenes remains challenging due to limited data. To address this, we propose a scalable data generation method considering the non-semantic property of multi-view reconstruction.

Training with Synthesized Data. Leveraging synthesized data for training is essential when available data is insufficient or biased. Synthesized data has been widely applied across fields such as Robotics [66], Natural Language Processing [2], Computer Vision [48], and AI for Science [68]. For example, recent depth estimation methods utilize synthesized data's accurate ground truth to enhance performance on fine structures [7, 85]. A relevant topic to our work is 3D scene generation, where generated data supports training 3D reconstruction models [16, 28, 46, 52–54, 84]. However, these methods focus on generating realistic scenes, necessitating semantic accuracy (e.g., object affordance and relationships), which constrains scalability due to the complex procedural rules required for accuracy and diversity.

While some recent methods attempt to address this limitation with language models [86], these models often lack spatial awareness and are slow in inference. In contrast, we show that semantics are not essential for multi-view reconstruction, allowing us to create a data generation pipeline free from semantic constraints and capable of generating virtually unlimited training data. Previously, non-semantic shape primitives have been used for various object reconstruction and appearance acquisition tasks [42, 43, 59, 82, 83]. Recently, LRM-Zero [79] has used primitive-based methods to generate large-scale data to train large reconstruction models, but it is limited to the object level. We focus on more challenging scene-level data synthesis, incorporating control of lighting, object composition, and camera poses for reconstructing wide-coverage scenes from dense-view images. We also present a mixed training framework to leverage the synergy between synthesized and real data. DUST3R [74] employs a pre-trained encoder from CroCo [77], which incorporates synthesized data, but its pre-training is limited to 2D image representation learning without directly learning 3D priors. In contrast, our pre-training approach directly targets 3D scenes, enhancing our model's geometric and texture understanding. We also enable joint training with both synthesized and real data.

3. Task and Preliminary

Our goal is to *reconstruct wide-coverage scenes in a feed-forward manner*. Given a set of dense-view images $\{I_i \mid i = 1, ..., n\}$ with known camera information, the model predicts the attributes of 3D representations. By default, we use $\mathbf{n} = \mathbf{32}$ views in our experiments to handle the high complexity of scenes, in contrast to previous sparse-view methods that rely on only 4 to 8 views [72, 76].

This paper primarily experiments with GS-LRM [88] and Long-LRM [93], chosen for their strong reconstruction quality. Both methods predict pixel-aligned 3D Gaussians from posed images with similar model architectures but different backbones: GS-LRM and Long-LRM employ transformer-based and Mamba-based [24, 45] backbones, respectively.

Given the input views, the models first patchify each image using non-overlapping convolutions, encoding them into feature tokens $\{T_i \mid i = 1, ..., n\}$ as in ViT [18]. The feature tokens from all images are flattened and concatenated into a feature set, $\mathbf{F}$, which is later processed by the model $\mathcal{M}$. Finally, an MLP decodes Gaussian parameters $\mathbf{G}$ to represent the scene. The process is formulated as follows:

$$\{T_1, \ldots, T_n\} = \{\mathrm{Conv}(I_1), \ldots, \mathrm{Conv}(I_n)\}, \tag{1}$$

$$\mathbf{F} = [\mathrm{Flatten}(T_1), \ldots, \mathrm{Flatten}(T_n)], \tag{2}$$

$$\bar{\mathbf{F}} = \mathcal{M}(\mathbf{F}), \tag{3}$$

$$\mathbf{G} = \mathrm{MLP}(\bar{\mathbf{F}}), \tag{4}$$

where $[\cdot, ..., \cdot]$ denotes concatenation, and $\bar{\mathbf{F}}$ represents the updated feature tokens produced by the backbone.

In the next section, we introduce our approach to synthesize data for training these models.

4. Synthesizing the MegaSynth Dataset

In this section, we first give an overview of our data synthesis method and then dive deeper to introduce how we control complexity, diversity, and alignment with real data.

Overview. We synthesize MegaSynth using a procedural generation method, as illustrated in Fig. 2. The process involves: i) generating a scene floor plan, including scene size and object instance box locations, ii) instantiating object geometries with random textures, and iii) randomizing the lighting. During the process, we eliminate high-level scene semantics. We only keep the low-level structural and geometric features of scenes.

4.1. Scene Floor Plan

Without loss of generality, we plan the scene as a cube box and populate it with objects represented by 3D bounding boxes. We randomize the 3D aspect ratio and size of scenes. We design multiple object box categories to simulate real-world scene geometry structures (visualized as boxes in Fig. 2 with different colors). For example, large object boxes tend to be placed near the ground while small object boxes have more flexible placement options. We parameterize the size, location, and number of each object type, specifying each parameter as a range. This allows us to introduce randomness to improve diversity. Further details of the object box categories and their attribute sampling ranges are provided in the Appendix.

4.2. Geometry and Texture

The scene floor plan constructed above divides the room space into basic units of object boxes. We then synthesize geometry and assign textures for each geometric shape.

Geometry of general objects. For each object box, we generate geometry by combining non-semantic shape primitives [79, 82], including cubes, spheres, cylinders, and cones. These primitives incorporate diverse geometry patterns, such as flat and curved surfaces, straight and curved lines, and sharp edges. Composing these shapes further increases geometric and topological complexity. Additionally, we apply random height-field augmentations [82] to the primitives, producing surfaces with both concave and convex details.

Different object categories (defined in Sec. 4.1) utilize varying numbers of shape primitives; for instance, large objects are typically composed of more primitives than small ones, loosely reflecting the complexity distribution of real-world objects. The geometry is instantiated in a canonical space, then rescaled and translated to fit the object box.

Geometry for increasing complexity. To enhance diversity and alignment with real data, we incorporate two additional

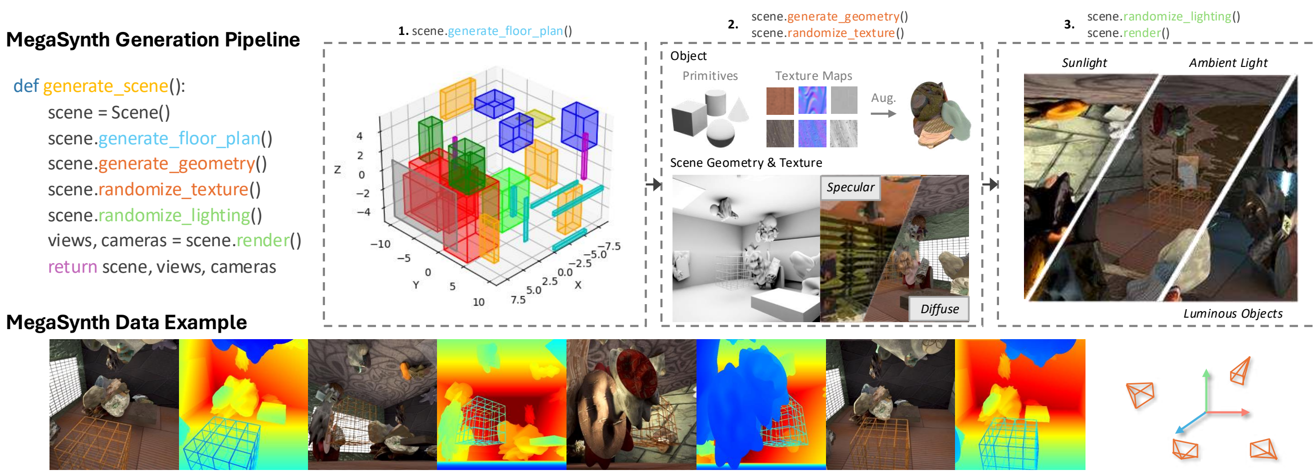

Figure 2. **MegaSynth generation pipeline.** We first generate the scene floor plan, where each 3D box represents a shape and different colors represent different object types. We compose shape primitives into objects with geometry augmentations, where these objects further compose the scene. We randomize the texture and lighting, and generate random cameras for rendering.

types of geometry. First, we add thin structures, such as wireframes of shape primitives, enabling the reconstruction of fine-grained geometries. To further increase diversity, we randomly place solid primitives intersecting with these wireframes. Second, we introduce axis-aligned geometries, such as thin sticks and flat surfaces, to reflect real-world geometry distributions under the Manhattan assumption [14].

Texture. Each shape primitive is assigned a random texture, including a basic color map along with normal, material, and roughness maps. We increase the probability of sampling specular and glass materials, ensuring a closer match to real-world appearances.

4.3. Lighting

Real-world images often feature complex lighting conditions. Thus, we design three lighting conditions and randomly compose them to improve the diversity and complexity. Each lighting uses a randomly sampled color and intensity.

Ambient light. We use the uniformly distributed ambient lighting with a unit brightness by default. The ambient lighting provides consistent illumination across a scene, helping to reveal scene details and stabilizing training.

Sunlight. Adding sunlight simulates true-to-life lighting effects, making the scene more complex with a higher intensity and casting shadows. We set the sunlight outside of the scene box. To enable the sunlight effect within the scene, we create windows on the walls with random sizes, under the regions that the sunlight covers. To further improve the complexity and diversity, we randomly add window bars implemented as the wireframes and window glasses.

Luminous objects and light bulbs. We randomly turn objects and axis-aligned sticks as lights and place light bulbs in the scene, simulating real-world lighting and increasing diversity. The intensity of object light can be sampled as large values to simulate lighting in dark environments.

5. Learning 3D Reconstruction on MegaSynth

In this section, we discuss how we utilize our synthesized MegaSynth (Sec. 4) to train a feed-forward reconstruction model (i.e., the LRM-based model illustrated in Sec. 3). To reach the goal, we first construct the training data by carefully sampling the camera distribution and rendering the images (Sec. 5.1). We then train our model with a mixed-data training strategy (Sec. 5.2) with rendering loss and geometry loss (Sec. 5.3). The details of the training process can be found in the Appendix.

5.1. Training Data Preparation on MegaSynth

To get training data, we render input views and target supervisions from the synthesized MegaSynth scenes. We sample cameras and then render RGB and depth images accordingly. We do not distinguish the input views and target views, i.e., they will be used interchangeably during training.

The main challenge of this data creation pipeline is the camera pose sampling. We empirically found that a careful design of camera sampling distribution can largely improve learning efficiency, model generalization, and training stability. We next detail our camera sampling process.

Basic rules. The cameras are sampled to keep a minimal distance from any objects in the scene, preventing the camera from losing context and avoiding the near-clipping issue. We randomly sample the field-of-view (FoV) of cameras, due to the diversity of lenses used in real-world image capture.

Better scene coverage. We heuristically split the scene into the inner and outer spaces, based on the distance to the scene center. The cameras sampled in outer space always look at the scene center, ensuring better view coverage. Meanwhile, the cameras in the inner space are encouraged to have more diverse poses, e.g. the orientations are randomly sampled within pre-defined ranges, increasing the diversity and matching real-world camera pose distribution.

Constrained camera baseline. The randomly sampled cameras in the outer part of the scene tend to have large baselines. To improve diversity, we choose to sample more scenes and cameras with slightly smaller baselines, aligning with real-world camera distribution. Thus, instead of sampling camera position in all free space, we first sample a distance range and then sample the camera within the constrained space.

5.2. Mixed Data Training

In training, we leverage distinct advantages from both the synthesized MegaSynth renderings and the real-world dataset (e.g., DL3DV). The synthesized data, with its diversity and scale, provide a foundation for models to learn general reconstruction priors of geometry, texture, and lighting. Moreover, easy access to accurate metadata (e.g., depth images and noise-free camera information) enhances geometric understanding and stabilizes training.

Meanwhile, real-world data offers authenticity that is hard to synthesize yet crucial for model robustness. For instance, it captures real-world imperfections like sensor noise and lighting artifacts, enhancing the model's robustness for real-world deployment. Additionally, its realistic semantics better align the model with real-world scene distributions.

We find these datasets to be complementary. Our experiments investigate two training strategies to leverage their synergy: (1) pre-training on the large-scale MegaSynth dataset followed by fine-tuning on a smaller real-world dataset; and (2) joint training on both datasets simultaneously. These approaches balance scalability and authenticity.

5.3. Rendering and Geometry Losses

We follow the standard method for training large reconstruction models using photometric image rendering losses:

$$\mathcal{L}_{img} = \mathrm{MSE}(I_i, \hat{I}_i) + \lambda \cdot \mathrm{Perceptual}(I_i, \hat{I}_i), \tag{5}$$

where λ is the weight for balancing the perceptual loss [36], I_i is ground-truth target image, and $\hat{I}_i$ is image rendered from predicted 3D Gaussians under target camera poses.

Our synthesized data naturally provides accurate geometry information, which is utilized to supervise the geometry of the 3D Gaussians predicted by the LRM models. In detail, both GS-LRM and Long-LRM (described in Sec. 3) predict pixel-aligned 3D Gaussians, where each Gaussian corresponds to a pixel in the input view. We supervise the center location of the predicted 3D Gaussians using the ground-truth geometry information. It is formulated as

$$\mathcal{L}_{loc} = M \cdot \mathrm{Smooth\text{-}L}_1(\mathbf{c}, \mathbf{G}_{loc}), \tag{6}$$

where $\mathbf{c}$ and $\mathbf{G}_{loc}$ are ground-truth and predicted 3D Gaussian location, respectively. The ground-truth Gaussian location $\mathbf{c}$ is computed from the depth maps of input views. Besides, the loss mask M masks out the pixels where the depth is larger than a threshold (e.g., 100 under the scale-normalized camera coordinate frame). This mask operation helps avoid numerical instability during training. This geometry loss proves particularly useful for scene-level reconstruction, which typically involves larger depth ranges, making it challenging to infer geometry solely from photometric cues. Additionally, it enhances the training convergence of the 3D Gaussians, as discussed in Long-LRM.

The final loss function can be formulated as $\mathcal{L}^S = \mathcal{L}^S_{img} + \gamma \cdot \mathcal{L}^S_{loc}$, where γ balances the strength of geometry loss term.

6. Experiments

In this section, we describe the experimental setting and present evaluation results. Due to the space limit, implementation and training details are in the Appendix.

6.1. Datasets

Besides our MegaSynth, we use three datasets in our paper, where DL3DV is the only one we take into training, i.e., others are evaluation-only.

DL3DV [47][1] is a large-scale dataset capturing diverse real-world scenes. We split it into 6723 and 400 scenes for training and performing evaluation, respectively. The 400 testing scene is composed of the DL3DV benchmark (140 outdoor scenes) and 260 indoor scenes held out from its official training set to balance the indoor-outdoor ratio.

Hypersim [56] is a synthetic 3D indoor scene dataset with ultra photo-realistic renderings, aimed at testing the generalization capability to out-of-distribution indoor scenes. Hypersim is challenging due to its complicated geometry, extreme lighting conditions, and large camera baselines. Hypersim also provides high-quality depth ground truth. We use a test set composed of 302 scenes.

MipNeRF360, Tanks & Temples (TT) [6, 38] includes 11 scenes for further testing the out-of-domain generalization capability of models on real data.

6.2. Evaluation and Baselines

We use 32 views as input and use 32 target novel view images for evaluation. The input and target views are non-overlap and are evenly sampled. We compare with three baselines:

GS-LRM and Long-LRM trained on DL3DV. These two baselines aim at validating *the effectiveness of our proposed data*. In detail, we train the GS-LRM and Long-LRM models on the largest real scene-level dataset, DL3DV, using the same training setting as ours.

Optimization-based 3DGS [37]. This baseline aims at validating the overall performance of our method, as the

[1] We refer to the DL3DV-10K dataset. Only 7K scenes were used in this project as it was completed before the full release.

Figure 3. **Reconstruction visualization on the in-domain DL3DV data.** The results are from Long-LRM at resolution 256. We present both indoor and outdoor results in the first and second rows, respectively. With our MegaSynth (denoted as 'w. MegaSynth'), the model performs better on thin structures (e.g., bottom left), complicated lighting (e.g., top middle), and cluttered scenes (e.g., top right).

			In-Domain			**Out-of-Domain** (Zero-shot Generalization)							
		Inf. Time	*DL3DV*			*Hypersim*					*MipNeRF360 & TT*		
Model	Training Dataset		PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	AbsRel$_\downarrow$	$\delta_{1\uparrow}$	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$
RESOLUTION 128, 32 INPUT VIEWS													
3DGS [37]	N.A. (Per-scene Optimization)	5.2 min	24.27	0.817	0.166	20.67	0.672	0.293	0.320	0.715	16.46	0.458	0.405
Long-LRM [93]	DL3DV	0.12 sec	24.18	0.812	0.173	23.41	0.790	0.210	0.272	0.763	19.68	0.569	0.312
Long-LRM (ours)	DL3DV + MegaSynth		25.44	0.853	0.136	25.01	0.836	0.164	**0.258**	0.792	20.86	0.652	0.249
GS-LRM [88]	DL3DV	**0.11 sec**	24.60	0.824	0.161	23.89	0.806	0.195	0.291	0.772	19.93	0.601	0.289
GS-LRM (ours)	DL3DV + MegaSynth		**25.75**	**0.859**	**0.130**	**25.46**	**0.846**	**0.154**	**0.258**	**0.800**	**21.19**	**0.672**	**0.235**
RESOLUTION 256, 32 INPUT VIEWS													
3DGS [37]	N.A. (Per-scene Optimization)	6.4 min	23.26	0.778	0.206	21.75	0.690	0.294	0.319	0.709	16.06	0.436	0.421
Long-LRM [93]	DL3DV	**0.35 sec**	23.71	0.779	0.236	22.51	0.767	0.267	0.291	0.753	18.61	0.465	0.421
Long-LRM (ours)	DL3DV + MegaSynth		**25.14**	**0.828**	**0.186**	**24.26**	**0.817**	**0.210**	**0.255**	**0.794**	**19.84**	**0.555**	**0.339**

Table 1. **Evaluation results against baseline methods.** We report results at resolutions of 128 and 256. For resolution 256, we only report results of Long-LRM as transformer-based GS-LRM is too slow. Our models are pre-trained on MegaSynth and then tuned on DL3DV. We report NVS quality on all data and evaluate reconstruction by measuring geometry accuracy (rendered depth accuracy) on Hypersim.

optimization-based 3DGS usually demonstrates a promising reconstruction quality. We use known camera information to get point cloud initialization from the 32 input views using COLMAP. We use official training hyper-parameters.

Additionally, we note that comparing with more advanced 3DGS methods is not the focus our work, as our target is scaling up training data for improving feed-forward methods. Our contributions can be directly ablated by comparing with LRMs trained without our data.

6.3. Results

Table 1 presents our results, demonstrating that training with both DL3DV and our MegaSynth dataset significantly improves model performance, with PSNR gains ranging from 1.2 to 1.8 dB. This improvement is consistent across model architectures (GS-LRM and Long-LRM), testing data (both in-domain DL3DV and out-of-domain datasets), image resolutions, and evaluation metrics, highlighting the effectiveness of our synthesized MegaSynth in enhancing the reconstruction quality of LRMs for wide-coverage scenes. Moreover, the rendering depth quality improves significantly as evaluated on Hypersim, showing the benefit of improving geometry quality by training with MegaSynth. Fig. 3 and Fig. 4 visually compare the reconstruction results for models with and without MegaSynth. We observe remarkable improvements in scenes with complicated scene structures, geoemtry, material and lighting, aligning with data generation designs (Sec. 4). Our approach also achieves substantially better results than the optimization-based 3DGS method while offering much faster inference speeds (e.g., over 2000 times speed-up from 5 minutes to 0.1 seconds).

We observe a notable trend of utilizing MegaSynth. The performance gains with MegaSynth on out-of-domain data are often larger than those on in-domain data. For example, Long-LRM achieves PSNR gains of 1.6 and 1.8 dB on Hypersim at resolutions of 128 and 256, respectively, surpassing the 1.3 and 1.4 dB improvements observed on the in-domain DL3DV dataset. GS-LRM results exhibits a similar pattern. The results underscore MegaSynth's effectiveness in enhancing the generalization capability of LRMs.

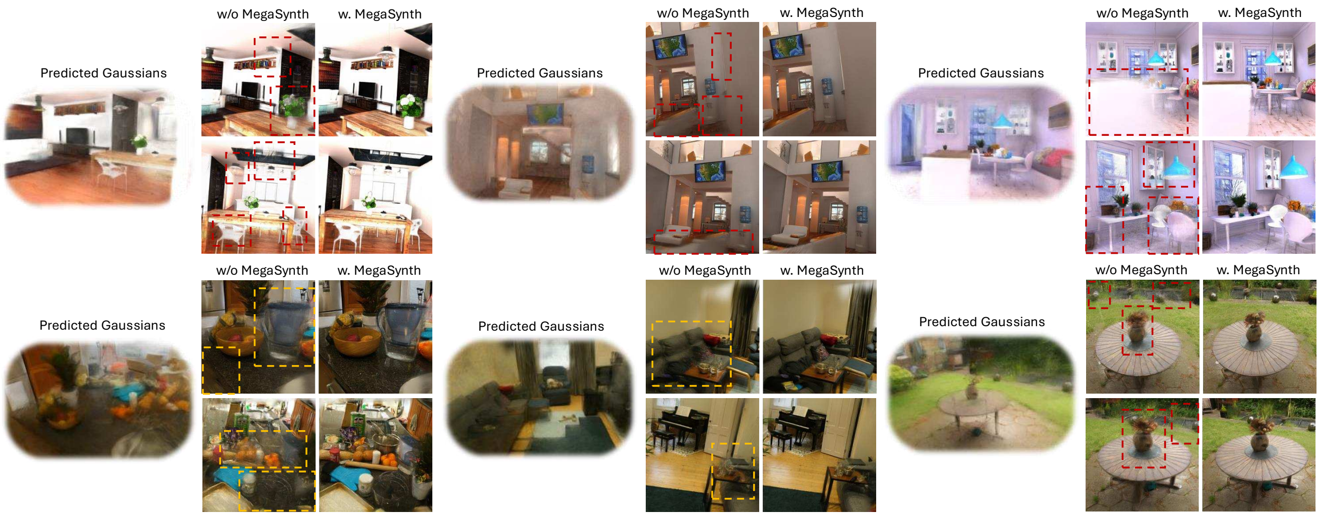

Figure 4. **Reconstruction visualization on the out-of-domain data.** The results are from Long-LRM at resolution 256. We include results for both Hypersim and MipNeRF360 are presented in the first and second rows, respectively.

	Data Control	$\mathcal{L}_{\text{loc}}^{S}$	Scale Up	MegaSynth-only Training (Trained w. only MegaSynth)				Real Data Tuning (Using DL3DV)			
				Fail. Iter.	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	Fail. Iter.	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$
(0)	✗	✗	✗	70k	17.18	0.519	0.445	80k	18.44	0.577	0.418
(1)	✓	✗	✗	45k	18.71	0.601	0.384	57k	21.87	0.738	0.266
(2)	✓	✓	✗	-	20.72	0.691	0.300	-	25.12	0.835	0.171
(3)	✓	✓	✓	-	**21.07**	**0.698**	**0.292**	-	**25.46**	**0.846**	**0.154**

Table 2. **Ablation study on data control, property and quantity.** Results are reported on the Hypersim dataset with resolution 128. We also report the number of iterations before the job fails. Please see Appendix for data control details for experiment (0). The default data is composed of 100K examples, and the scaled one contains 700K examples.

	DL3DV-Test-Indoor		
Data	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$
DL3DV	25.41	0.853	0.150
DL3DV + MegaSynth	**26.75**	**0.890**	**0.116**
	DL3DV-Test-Outdoor		
Data	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$
DL3DV	23.09	0.771	0.183
DL3DV + MegaSynth	**23.89**	**0.803**	**0.157**

Table 3. **Performance gains on indoor and outdoor test data.** Results are from 128-resolution GS-LRM. Test data split details are in Sec. 6.1.

6.4. Ablation Studies

In this section, we examine the impact of MegaSynth data quality, quantity, properties, and training paradigms for utilizing synthesized data. Without additional specification, the default experimental setup is the resolution-128 GS-LRM with pre-training + fine-tuning training protocol.

MegaSynth data quality, quantity and property. Table 2 presents our results. In general, we observe a positive correlation between performance of MegaSynth-only training and subsequent real-data fine-tuning, underscoring the value of MegaSynth in model training. Specifically, we refer MegaSynth-only training to the model trained after the pertraining stage using only MegaSynth.

In Table 2 (0), training with a basic version of MegaSynth without controlling the data diversity and complexity results in lower performance than training with real data alone (Table 4), suggesting that unregulated synthesized data fails to enhance training. Additionally, we observe training instability, with pre-training and fine-tuning failing after around 70K iterations. We hypothesize that the high data randomness contributes to this instability, impeding effective learning and negatively affecting fine-tuning of real data.

Introducing control of data distribution, as shown in Table 2 (1), improves both MegaSynth-only training and real-data fine-tuning performance, emphasizing the importance of data quality and effectiveness of our data control method. However, training instability worsens, likely due to the increased complexity that amplifies training challenges.

Incorporating metadata during training mitigates this instability. Table 2 (2) shows that adding geometrical supervision, $\mathcal{L}_{\text{loc}}^{S}$, significantly improves stability and overall performance. This result underscores a key advantage of MegaSynth: the ability to provide additional ground-truth data. Expanding the dataset to include more scenes (i.e., 700K scenes in total), as in Table 2 (3), yields additional gains, showing the benefit of scale.

Indoor and outdoor improvements. We analyze the performance gains in Table 3, focusing on both indoor and outdoor test data from DL3DV. Although our synthesized MegaSynth data is primarily focusing on indoor scenes, we observe improvements in outdoor scenes as well, with a notably larger performance gain on indoor scenes. This suggests that the MegaSynth contributes to a generalized enhancement in geometric and appearance understanding, enabling broader generalization across diverse environments. At the same time, improving MegaSynth with outdoor characteristics would be an interesting direction.

	Hypersim		
	PSNR$_{\uparrow}$	SSIM$_{\uparrow}$	LPIPS$_{\downarrow}$
DL3DV Real-only	23.89	0.806	0.195
MegaSynth-only	21.50	0.719	0.272
Joint Training	25.33	0.844	0.157
Pre-training + Fine-tuning	**25.46**	**0.846**	**0.154**

Table 4. **Ablation study on the training framework to leverage MegaSynth.** Results are reported with GS-LRM.

Figure 5. **Visual comparison between training with only MegaSynth and only real data.** We include two failure cases of MegaSynth-only with failures highlighted.

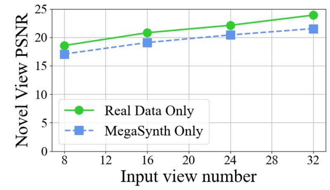

Figure 6. **Analysis of real data only and MegaSynth-only performance** with different number of input views.

Training strategies. We evaluate different strategies for utilizing MegaSynth. As shown in Table 4, training exclusively on MegaSynth (row 2) achieves performance comparable to training on real data (row 1), highlighting the effectiveness of MegaSynth and supporting our hypothesis that explicit semantics are not required for training scene reconstruction models. We visualize the results in Fig. 5. We find the model performs closely in most of the scenes but is much worse on complicated geometry patterns and large scene scales that are hard to model in synthesized data.

We further compare two approaches: (i) joint training on both synthesized MegaSynth and real data in row 3, and (ii) pre-training on MegaSynth followed by fine-tuning on real data in row 4. As shown in Table 4, the second approach yields slightly better performance, though the performance gap is minimal. This suggests that the model effectively learns the joint distribution of synthesized and real data without catastrophic forgetting during fine-tuning, indicating a degree of distribution alignment between MegaSynth and real data. Additionally, this experiment confirms that the performance gain results from the enhanced reconstruction capability acquired through MegaSynth, rather than simply from additional training iterations.

6.5. Analysis

We perform a more detailed analysis of MegaSynth, especially its effectiveness against other synthetic data and application to other 3D tasks.

Analysis on different numbers of input views. We extend our model trained with MegaSynth to scenarios with fewer input views, training GS-LRM with inputs of 8, 16, 24, and 32 views using either real-world data alone or a combination of real-world and MegaSynth data. As shown in Fig. 6, GS-LRM trained with both DL3DV and synthesized MegaSynth data demonstrates improved performance as the number of input views increases. Notably, an almost constant performance gap remains regardless of the number of views, which we attribute to the semantic gap between DL3DV and MegaSynth. These results highlight the effectiveness of MegaSynth for sparse-view reconstruction and suggest that semantic alignment is not a primary driver of 3D reconstruction performance.

Advantages over other synthetic datasets. We experiment with using other synthetic datasets for training LRMs. As shown in Table 5, Kurbic [23] (data released in SRT [58]) and Front3D [20] fail to improve LRM performance, while MegaSynth benefits the model across all test datasets consistently. In detail, Kurbic contains 1 million scenes radnomly composed by realistic 3D assets; Front3D is composed of 6,000 indoor scenes designed by artists. The results imply that realistic 3D assets or scene composition is not the gaurantee for improving reconstruction quality. Instead, reconstruction model benefits from data with better non-semantic quality, e.g. geometry difficulty and scene diversity.

	DL3DV			*Hypersim*			*MipNeRF360 & TT*		
	PSNR$_{\uparrow}$	SSIM$_{\uparrow}$	LPIPS$_{\downarrow}$	PSNR$_{\uparrow}$	SSIM$_{\uparrow}$	LPIPS$_{\downarrow}$	PSNR$_{\uparrow}$	SSIM$_{\uparrow}$	LPIPS$_{\downarrow}$
DL3DV	18.31	0.555	0.391	18.43	0.602	0.373	15.59	0.550	0.332
DL3DV+Kubric [23]	18.28	0.552	0.395	18.46	0.600	0.375	15.49	0.546	0.340
DL3DV+Front3D [20]	18.40	0.558	0.389	18.48	0.603	0.370	15.63	0.551	0.329
DL3DV+MegaSynth	**19.58**	**0.592**	**0.338**	**19.88**	**0.638**	**0.324**	**16.72**	**0.592**	**0.303**

Table 5. **Comparison with other synthetic datasets.** We report results with 8 input views and GS-LRM under resolution 128.

	AbsRel (↓)	δ_1 (↑)
Depth Anything V2	0.213	0.761
Tuned on MegaSynth	**0.158**	**0.799**

Table 6. **MegaSynth benefits monocular depth estimation.**

	DL3DV	Ours	Front3D
Geom. Difficulty (↓)	1.65	**1.35**	3.00
Diversity (↓)	**1.40**	1.60	3.00

Table 7. **User study of data difficulty and diversity.**

MegaSynth also helps other tasks. We fine-tune Depth Anything V2 ViT-B model on MegaSynth and evaluate on Hypersim. Results in Table 6 shows that MegaSynth helps improving monocular depth estimation, demonstrating the potential of MegaSynth to be used for other 3D tasks.

Comparison with real data. Tab. 7 presents a user study ranking geometry difficulty and scene diversity of datasets, showing our comparability with real data and advantage over the other synthetic data Front3D. Please see more analysis on measuring alignment with real data in Appendix.

7. Conclusion

We introduce MegaSynth, a non-semantic procedurally generated dataset, to improve LRMs for reconstructing wide-coverage scenes. MegaSynth benefits from its scalability and controllability, improving the model's understanding of geometry and appearance. Experiments show MegaSynth's capability of improving LRM reconstruction quality via both pre-training and joint training. The performance gains are consistent over different model architectures, test data domains, and input/output resolutions. Interestingly, LRMs trained sorely with MegaSynth demonstrate comparable performance with using real data, demonstrating that reconstruction is almost a non-semantic/low-level task.

Acknowledgment. QH acknowledges NSF IIS 2047677 and NSF IIS 2413161.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[4] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22861–22872, 2024.

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[7] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

[8] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer, 2020.

[9] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024.

[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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021.

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

[12] 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 preprint arXiv:2403.14627*, 2024.

[13] 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. *arXiv preprint arXiv:2411.04924*, 2024.

[14] James Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. *Advances in Neural Information Processing Systems*, 13, 2000.

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[16] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[17] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[19] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023.

[20] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[21] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[22] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[23] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022.

[24] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[25] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

[26] Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. In *European Conference on Computer Vision*, pages 333–350. Springer, 2025.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[28] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023.

[29] 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. *arXiv preprint arXiv:2311.04400*, 2023.

[30] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[31] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *arXiv preprint arXiv:2310.01410*, 2023.

[32] Hanwen Jiang, Qixing Huang, and Georgios Pavlakos. Real3d: Scaling up large reconstruction models with real-world images. *arXiv preprint arXiv:2406.08479*, 2024.

[33] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. In *2024 International Conference on 3D Vision (3DV)*, pages 31–41. IEEE, 2024.

[34] Hanwen Jiang, Haitao Yang, Georgios Pavlakos, and Qixing Huang. Cofie: Learning compact neural surface representations with coordinate fields. *arXiv preprint arXiv:2406.03417*, 2024.

[35] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024.

[36] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[37] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[38] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4): 1–13, 2017.

[39] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024.

[40] 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. *arXiv preprint arXiv:2311.06214*, 2023.

[41] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

[43] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2020.

[44] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[45] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[46] Tsung-Yi Lin, Chen-Hsuan Lin, Yin Cui, Yunhao Ge, Seungjun Nah, Arun Mallya, Zekun Hao, Yifan Ding, Hanzi Mao, Zhaoshuo Li, et al. Genusd: 3d scene generation made easy. In *ACM SIGGRAPH 2024 Real-Time Live!*, pages 1–2, 2024.

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[48] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[49] 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. *ACM Transactions on Graphics (TOG)*, 2019.

[50] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[51] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *Advances in Neural Information Processing Systems*, 36:5791–5807, 2023.

[52] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[53] NVIDIA, Maciej Bala, Yin Cui, Yifan Ding, Yunhao Ge, Zekun Hao, Jon Hasselgren, Jacob Huffman, Jingyi Jin, J.P. Lewis, Zhaoshuo Li, Chen-Hsuan Lin, Yen-Chen Lin, Tsung-Yi Lin, Ming-Yu Liu, Alice Luo, Qianli Ma, Jacob Munkberg, Stella Shi, Fangyin Wei, Donglai Xiang, Jiashu Xu, Xiaohui Zeng, and Qinsheng Zhang. Edify 3d: Scalable high-quality 3d asset generation. *arXiv preprint arXiv:2411.07135*, 2024.

[54] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, et al. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12630–12641, 2023.

[55] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21783–21794, 2024.

[56] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021.

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[58] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022.

[59] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 85–101. Springer, 2020.

[60] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[61] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[62] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 519–528. IEEE, 2006.

[63] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *ArXiv*, abs/2006.09661, 2020.

[64] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259*, 2024.

[65] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80:189–210, 2008.

[66] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on robot learning*, pages 477–490. PMLR, 2022.

[67] 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 preprint arXiv:2402.05054*, 2024.

[68] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[69] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. *arXiv preprint arXiv:2406.11819*, 2024.

[70] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[71] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21686–21697, 2024.

[72] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023.

[73] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021.

[74] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[75] 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.

[76] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024.

[77] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

[78] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *arXiv preprint arXiv:2403.16292*, 2024.

[79] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024.

[80] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[81] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model, 2023.

[82] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018.

[83] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019.

[84] Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. Scene synthesis via uncertainty-driven attribute synchronization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5630–5640, 2021.

[85] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[86] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237, 2024.

[87] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.

[88] 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 preprint arXiv:2404.19702*, 2024.

[89] Tianyuan Zhang, Zhengfei Kuang, Haian Jin, Zexiang Xu, Sai Bi, Hao Tan, He Zhang, Yiwei Hu, Milos Hasan, William T. Freeman, Kai Zhang, and Fujun Luan. Relitlrm: Generative relightable radiance for large reconstruction models, 2024.

[90] Yue Zhang, Ziqiao Ma, Jialu Li, Yanyuan Qiao, Zun Wang, Joyce Chai, Qi Wu, Mohit Bansal, and Parisa Kordjamshidi. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *arXiv preprint arXiv:2407.07035*, 2024.

[91] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023.

[92] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[93] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint 2410.12781*, 2024.