LRM-Zero: Training Large Reconstruction Models with Synthesized Data

Desai Xie[† 1, 2]   **Sai Bi**[1]   **Zhixin Shu**[1]   **Kai Zhang**[1]   **Zexiang Xu**[1]
Yi Zhou[1]   **Sören Pirk**[3]   **Arie Kaufman**[2]   **Xin Sun**[1]   **Hao Tan**[1]

[1]Adobe Research   [2]Stony Brook University   [3]Kiel University
`dexxie,ari@cs.stonybrook.edu`
`sbi,zshu,kaiz,zexu,yizho,xinsun,hatan@adobe.com`
`soeren.pirk@gmail.com`

Abstract

We present *LRM-Zero*, a Large Reconstruction Model (LRM) trained entirely on synthesized 3D data, achieving high-quality sparse-view 3D reconstruction. The core of *LRM-Zero* is our procedural 3D dataset, *Zeroverse*, which is automatically synthesized from simple primitive shapes with random texturing and augmentations (e.g., height fields, boolean differences, and wireframes). Unlike previous 3D datasets (e.g., Objaverse) which are often captured or crafted by humans to approximate real 3D data, *Zeroverse* completely ignores realistic global semantics but is rich in complex geometric and texture details that are locally similar to or even more intricate than real objects. We demonstrate that our *LRM-Zero*, trained with our fully synthesized *Zeroverse*, can achieve high visual quality in the reconstruction of real-world objects, competitive with models trained on Objaverse. We also analyze several critical design choices of *Zeroverse* that contribute to *LRM-Zero*'s capability and training stability. Our work demonstrates that 3D reconstruction, one of the core tasks in 3D vision, can potentially be addressed without the semantics of real-world objects. The *Zeroverse*'s procedural synthesis code and interactive visualization are available at: https://desaixie.github.io/lrm-zero/.

1 Introduction

The current wave of rapid development in foundation models [10] has been empowered by two key components: scalable model architectures [86, 40, 67] and massive datasets [19, 68, 79]. Foundation models in text [65, 3, 85], image [70, 29] and video [48, 5, 35] domains have capitalized on both factors. As there is a unified trend of adopting transformers [86] as the model architecture, many recent works have identified the training data as the most crucial factor [8, 20, 37, 2, 11].

Following the progress in other modalities, transformer-based [86] 3D Large Reconstruction Models [41, 49, 90, 98, 84] (LRM) has emerged as a potential foundation model for the 3D domain, which uses a scalable model architecture. However, 3D data is still difficult to acquire, considering that both manually capturing and hand-crafting 3D objects are expensive, time-consuming and require special expertise. Also, 3D data is more sensitive for its license safety, different types of bias, and identity leakage.

---

†This work is done while Desai is an intern at Adobe Research.
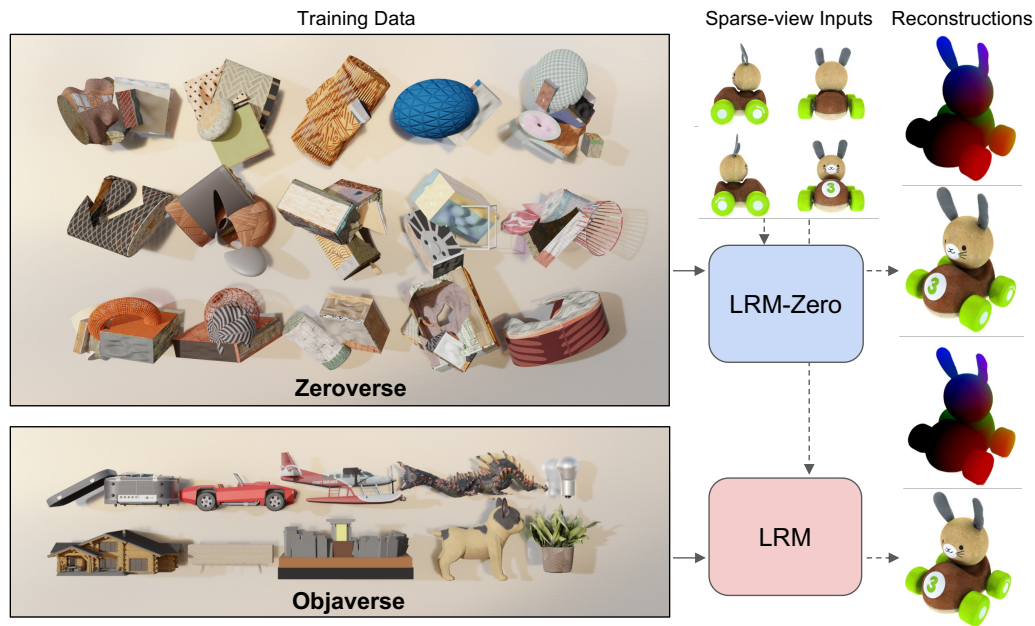
Figure 1: We present our *LRM-Zero* framework trained with synthesized procedural data *Zeroverse*. *Zeroverse* (top left) is created from random primitives with textures and augmentations, thus it does not contain semantical information as in Objaverse (bottom left). Nevertheless, when training with the same large reconstruction model architecture [107] on both datasets, *LRM-Zero* can match objaverse-trained LRM's (denoted as 'LRM') visual quality (right part) of reconstructions. A possible explanation is that 3D reconstruction, although serves as a core task in 3D vision, rely mostly on local information instead of global semantics. Reconstruction is visualized with RGB and position-based renderings, and interactive viewers can be found on our website.

Thus, in this paper, we propose *LRM-Zero*, trained on purely synthesized data, to explore another route which can potentially resolve the 3D data scarcity, licensing, and bias issues. The name 'Zero' highlights our synthesized and non-semantic training data, which we named as *Zeroverse*. *Zeroverse* is a procedural, amorphic alternative to Objaverse [23] in training reconstruction models. The comparison between *Zeroverse* and Objaverse is illustrated in Fig. 1, and more visual comparisons can be found in Appendix. The data in *Zeroverse* is procedurally created by randomly composing primitive shapes with textures and applying shape augmentations. The process resembles the previous work Xu et al. [100]. We select five primitive shapes: cube, sphere, cylinder, cone, and torus to cover different types of surfaces and topological characteristics. The textures are randomly applied, which is realistic at low-level but do not contain high-level semantics. The three different augmentation methods, i.e., height-field, boolean difference, and, wireframes, help increase the data diversity and add more curvatures, concavity, and thin structures, respectively. The primitive shapes, textures, and an illustration of the augmentations are shown in Fig. 2. In this work, we experiment with 400K *Zeroverse* data, which roughly matches the number of meaningful data in Objaverse (i.e., excluding rendering failures, flatten 3D data, point clouds, unsafe data from the overall 800K data). The initial experiments indicate that further increasing the amount of data is not effective, and we refer the reader to Appendix for our early results on scaling the data size.

We validate our *Zeroverse* by training GS-LRM [107] over it, and we denote this model as *LRM-Zero*. Surprisingly, we found that *LRM-Zero* can achieve a reconstruction quality similar to that of GS-LRM trained on Objaverse, seeing Fig. 1. More comparisons are provided in the Appendix. We also quantitatively evaluate the model on two standard 3D reconstruction benchmark ABO [18] and GSO [28]. For sparse-view reconstruction (i.e., 4 views and 8 views), *LRM-Zero* reaches competitive results against GS-LRM, and the best results gap is as low as 1.12 PSNR, 0.09 SSIM, and 0.006 LPIPS. A plausible reason for such "zero"-shot data generalization is that 3D reconstruction (with poses) relies more on the local visual clues instead of the global semantics. This is more obvious for dense-view reconstruction (e.g. 100 input views) where single-shape optimization without any data prior can reach good results [6, 46]. For the sparse-view reconstruction that we focused, *LRM-Zero*

can possibly rely on the local details (such as cross-view patch correspondence) to infer the shape, where *Zeroverse* supports *LRM-Zero* to learn such knowledge.

We analyze the effectiveness of *Zeroverse*'s design, especially for shape augmentations. We find that each type of augmentation provides visible structural improvements for the reconstructions, and most of the improvements are reflected in the metrics of our benchmarks. We also study the impact of different dataset designs on another critical property of *LRM-Zero*: training stability. Training stability is crucial for large-scale training as large models are more prone to diverge after training for a significantly long time [72, 17, 22]. We empirically found that careless design of *Zeroverse* can introduce significant instability during the training of *LRM-Zero*. As both data complexity and model hyperparameters can affect the training stability, a model-data co-design is helpful in our experiments, i.e., the model's hyperparameters and data properties are tuned jointly. Lastly, we show the generalizability of both *Zeroverse* and *LRM-Zero*. For *Zeroverse*, we show that the dataset can also enable training a NeRF-based reconstruction model and reaches competitive results to Objaverse-trained models. For *LRM-Zero*, we demonstrate that the model can generalize across different datasets including realistic 3D data, such as OmniObject3D [95] and OpenIllumination [55]. We also show that *LRM-Zero* can be combined with off-the-shelf multi-view diffusion models to support both text-to-3D generation and image-to-3D generation.

The key contribution of this paper is to demonstrate that purely synthesized data can be utilized to learn generic 3D priors for sparse-view 3D reconstruction, a core task of 3D vision. While our work may appear straightforward, it provides a minimal, yet generalizable proof-of-concept which can inspire the community to exploit procedural 3D data for 3D tasks in the future. We also provide carefully crafted studies on the co-design of data and model, as well as their effect on training stability and generalization.

Lastly, we provide the interactive *Zeroverse* data visualization and *LRM-Zero* reconstruction results in our website `https://desaixie.github.io/lrm-zero/`. We recommend the readers to have a check. The *Zeroverse* data synthesis script is released at `https://github.com/desaixie/zeroverse`, and we hope that it can facilitate future research.

2 Background: feed-forward reconstruction model

Feed-forward 3D reconstruction targets to learn a model that can regress the 3D shapes from multi-view images. The sparse-view version of this task is illustrated in the right part of Fig. 1, where multiple input views are presented, and the output is a 3D representation. To solve this task, LRM [41] introduces a pure-transformer based method which allows scalable training. Original LRM uses NeRF [61, 12] as 3D representation, and a bunch of later works [13, 84, 16, 99, 107] extend it to Gaussian Splatting [46], which is another 3D representation proposed recently. This paper is mostly experimented with the GS-LRM [107] architecture given its simplicity in model design (i.e., a pure-transformer architecture) and the SotA reconstruction quality.

GS-LRM predict the 3D Gaussians from the n multi-view images $I_1, \ldots, I_n$. The images are first patchified to features $f_1, \ldots, f_n$ with shared non-overlapping (i.e., stride equals to kernel size) convolutions. Then features are flattened and concatenated as the input to a self-attention transformer.

$$f_1, \ldots, f_n = \mathrm{Conv}(I_1), \ldots, \mathrm{Conv}(I_n) \tag{1}$$

$$x = [\mathrm{Flatten}(f_1); \ldots; \mathrm{Flatten}(f_n)] \tag{2}$$

$$y = \mathrm{Transformer}(x) \tag{3}$$

The output y of the transformer will be directly interpreted as the Gaussian Splatting parameters, and serves as the representation of the output 3D object. These parameters can be rendered for training losses or viewed interactively. The GS-LRM model is purely trained with RGB rendering loss by minimizing the difference between ground truth image and the rendering images. For more details of the GS-LRM model [107] architecture and Gaussian Splatting representation [46], please refer to the original papers. After briefly introducing the backbone model architecture of *LRM-Zero*, we next introduce our procedural data *Zeroverse* to train such a model.
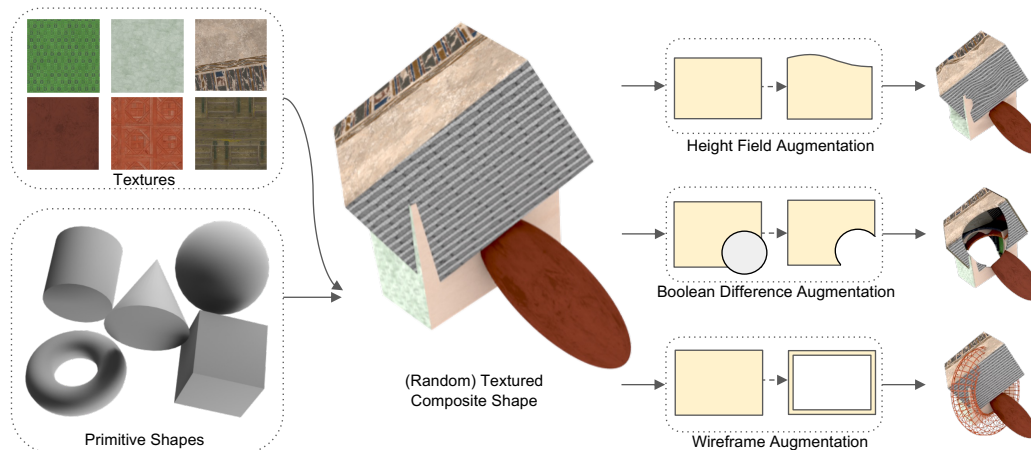
Figure 2: Illustration of the *Zeroverse* data creation process. A random textured shape is first composited from primitive shapes and textures (Sec.3.1). Then different augmentations (i.e., height field, boolean difference, wireframes in Sec . 3.2) are applied to enhance the dataset characteristics (e.g., curved surfaces, concavity, and thin structures). More visualizations in Appendix and website.

3 The *Zeroverse* dataset

In this section, we introduce the creation of *Zeroverse* that supports training a sparse-view large reconstruction model (LRM). *Zeroverse* consists of procedurally synthesized shapes with randomized parameters by revisiting the pipeline in the previous work [100], which was initially proposed for relighting and later extended for view synthesis [101] and material estimation [7, 52]. As illustrated in Fig. 2, the process first composites primitive shapes with random texturing (Sec. 3.1). Then, different augmentations are applied to enhance the diversity of the data (Sec. 3.2). As the LRM-based model only relies on multi-view rendering to train the model (i.e., do not require geometry supervision), the *Zeroverse* objects are always saved in the compact mesh format.

3.1 Composing primitives into textured shapes

Primitive shapes. Our synthetic object creation process starts with a pool of primitive shapes. The pool only consists of basic shapes for the 3D world. Specifically, in our implementation, we have 5 primitives: cube, sphere, cylinder, cone, and torus. Intuitively, cubes and spheres provides knowledge on the sharp straight lines and the purely curved shapes. Cylinders and cones contain different curved surfaces besides sphere. The specialty of torus is its hole, which is topologically different to the above shapes (i.e., a torus has genus 1). Although it is possible to create holes through combinations and augmentations (e.g., the boolean difference and wireframe in Sec. 3.2), we decide to explicitly add this capacity to our dataset. Also, the combination of multiple torus is easy to create shapes with higher genus (i.e., roughly the number of disjoint holes in a connected shape).

Compositions. With a reasonable pool of primitive shapes, we then compose them together to construct complex shapes, offering more diverse visual cues for the reconstruction task. We randomly sample 1 to 9 primitives (with replacement) from the primitive pool. The sampling probability of the numbers of primitives is configurable. Each sampled primitive will independently be scaled, translated, and rotated randomly. We simply combine these affine-transformed shapes together without special handling of the shape intersections or disconnections. Thus it is possible to have multiple disjoint shapes in one scene, which we will still refer to as one object. This satisfies the requirement for real-world reconstruction applications, where simple disjoint shapes would be considered as a single object.

Texturing. For each surface of the shape, we apply a texture randomly sampled from an internal dataset. To support the research community, in our public release version, we provide an alternative public texture dataset.

53288

Table 1: Quantitative results comparing *LRM-Zero* with GS-LRM [107] (trained on Objaverse) under the 8-input-view setting. We use GSO [28] and ABO [18] evaluation datasets and PSNR, SSIM, and LPIPS [108] metrics. *LRM-Zero* demonstrates competitive performance against GS-LRM.

			GSO			ABO		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
8 input views	Res-512	GS-LRM	**33.23**	**0.971**	**0.031**	**30.92**	**0.944**	**0.067**
		LRM-Zero	31.62	0.960	0.039	28.71	0.929	0.078
	Res-256	GS-LRM	**31.90**	**0.966**	**0.030**	**30.66**	**0.949**	**0.055**
		LRM-Zero	30.78	0.957	0.036	28.82	0.934	0.065

3.2 Shape augmentations

We apply augmentation to the textured shapes to add diversity and complexity that resembles real-world objects and is not covered by the initial shape in Sec. 3.1. We implement three augmentation operators: height field, boolean difference, and wireframe conversion for better data coverage of curved shapes, concave shapes, and thin structured respectively. These diversities of the data will be reflected by the capacity of large reconstruction models with observable structural improvements (studied in Sec. 5.1). We illustrated the process and example results in the right part of Fig. 2. We do not apply the augmentation 'boolean difference' and 'wireframe' at the same time. This is for training stability (studied in Sec. 5.2) as we empirically found that an ultra-complex shape can lead the reconstruction model training to non-convergence.

Height fields. Most of the surfaces (except the torus) of our primitives have constant curvatures, and we apply height fields augmentation in Xu et al. [100] to break this constraint. An illustration of the height map can be found in Fig. 2 (top right). In detail, for each face of the primitives, we apply a height field with varying heights and curvatures to displace the surface vertices, making the surface curved and bumpy. Specifically, the magnitude of height is randomly sampled at each position in the map and we use bicubic interpolation to obtain smooth surfaces.

Boolean difference. Concave structures are common in real-world objects, for example, bowls, hats, spoons. However, the concavity is not well captured by the previous pipeline. To resolve this, we 'subtract' primitives from the shapes, which can be considering as a reversion of the 'additive' operators in the combination process. This is implemented by computing the boolean difference between the composite object in Sec. 3.1 and a basic primitive from our pool. In details, we use Blender's boolean modifier and solidify modifier to augment the initial shape. The inside faces of the resulting cut shape will have the same texture as the outside faces. Besides introducing concavity to the dataset, the boolean difference operation also expose the 'interior' of the shape (as shown in Fig. 2), which helps the reconstruction model to handle complex structures. The actual effect of the boolean operator is quite diverse, and we refer the reader to check the visualization in the Appendix.

Wireframe. Besides concavity, thin structures (especially the striped or repeated one) is another challenge in real-world reconstruction, for example, hairs, baskets, railings. To train a reconstruction model capable with thin structures, we want to explicitly add this characteristic to our *Zeroverse* dataset. And for simplicity, we use the wireframe. Wireframe is a basic augmentation from the primitive shapes, which generally converts their meshes to the skeletons. It is pre-implemented in multiple libraries, and we take the shape modifier in Blender. The results are illustrated in Fig. 2 (i.e., a wireframe of torus) and more in Appendix. The texture of the wireframe is inherited from the primitive shape but usually not distinguishable due to its thin surfaces.

4 Experiments

4.1 *LRM-Zero* experiment details

For rendering the multi-view images of *Zeroverse*, we follow [41]. For each object in *Zeroverse*, we render 32 views with randomly sampled camera rotations and random distances in the range of [2.0, 3.0]. Each image is rendered at the 512×512 resolution with uniform lighting. We use the same

network architecture and follow the hyperparameters/implementation (e.g., 80K training steps, details as GS-LRM [107]. We only decrease perceptual loss weight from 0.5 to 0.2 to improve training stability. For the results comparison, we pre-train the model with 256-resolution and fine-tuned on 512-resolution following GS-LRM. The overall training uses 64 A100 GPUs and takes 3 days. For analysis and ablation studies, we only run the 256-resolution experiments. Please refer to the original GS-LRM paper [107] for more experimental details.

Metric evaluations for results and analysis are mostly conducted on two relatively large benchmarks: Google Scanned Objects (GSO) [28] and Amazon Berkeley Objects (ABO) [18]. In our paper, we use 8 structural input-view as the standard evaluation protocol to increase view coverage. The 4 structural input-view results are provided in Appendix. In details, for 8 structural views, we render from 0 elevation with 0, 90, 180, 270 azimuth plus 40 elevation with 45, 135, 225, and 315 azimuth, while 4 structural views render from 20 elevation with 0, 90, 180, 270 azimuth. The testing views for metric calculation are randomly sampled. The generalization experiments in Sec. 5.3 use either 8 random input views for generalization test, or the fixed cameras provided by the generated models. We always assume that the camera poses are provided with input views.

4.2 Results

We evaluate *LRM-Zero* on the benchmarks and show the results in Tab. 1. The absolute PSNR values of GSO and ABO are over 30 and 28.7 respectively, which indicates that the reconstruction has high visual quality. Compared to GS-LRM [107] trained on Objaverse, the metric still shows a gap, but within a reasonable range of 1.1 PSNR on GSO and 1.9 PSNR on ABO. The gap is larger for higher resolution (i.e., Res-512) and it is possibly due to the training configuration of our 512-res fine-tuning is sub-optimal. Qualitatively, we do not observe significant visual difference between the reconstructed 3D models from *LRM-Zero* and GS-LRM. An example comparison is shown in Fig. 1 and some more comparisons in the Appendix. The interactive viewer of *LRM-Zero* reconstruction results can be found in our website.

After viewing the *LRM-Zero* visual results and the sparse-view reconstruction setup, we found that both 4-view and 8-view can not fully cover the object surfaces thus the model needs to hallucinate the invisible parts. This hallucination ability requires semantic understanding of the 3D objects while *Zeroverse* lacks by design. It might be the major reason of result gap between *Zeroverse*-trained and Objaverse-trained models in Tab. 1. The invisible regions can be mitigated by reconstructing from more views (either capturing or generating). However, more views involves more tokens and challenges the computation cost of the current fully-connected self-attention design in GS-LRM, thus beyond the scope of current paper and we leave it as future works.

5 Analysis

In this section, we analyze the properties of our *LRM-Zero* trained with the synthesized *Zeroverse*. We first conduct ablation studies in Sec. 5.1 to show the effectiveness of *Zeroverse* augmentations. Next, Sec. 5.2 explores stabilized training of *LRM-Zero* from both data and model perspective, as training stability is one of the key challenges in large-scale training [22, 17]. Last, we show the generalization of our methods by applying *LRM-Zero* over diverse data, and trained different reconstruction models on *Zeroverse*.

5.1 Ablation studies on different augmentations

We conduct ablation studies to verify the effectiveness of our height field, boolean difference, and wireframe augmentations. We show both quantative and qualitative comparisons.

Boolean difference and wireframe augmentation. As our sampling strategy does not apply boolean difference and wireframe augmentations jointly to avoid over-complex shapes. Therefore, we conduct the ablation study of these augmentations together. As shown in Tab. 2, we apply different sampling ratios to both augmentations (e.g., experiments id 1, 2, 3) and also exclude them in experiment 4. Boolean difference augmentation largely improves the metric (comparing experiment pair 2, 4 or 1, 3). Note that we use 60%/40% instead of 50%/50% because the later one has more instability (Sec. 5.2). The possible reason is visualized in Fig. 3: the lack of boolean augmentation in training data causes experiment 2 to show structural failure on concave shapes.

Table 2: Ablation studies over boolean difference and wireframe augmentations. The height-field (hf) is applied independently to each surface with prob. 0.5.

	dataset			GSO			ABO		
id	hf-only def. 40%	boolean def. 40%	wireframe def. 20%	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1	default	default	default	**30.78**	**0.957**	**0.036**	**28.82**	**0.934**	**0.065**
2	60.0%	40.0%	0%	30.75	0.957	0.036	28.76	0.931	0.066
3	66.6%	0%	33.3%	29.82	0.948	0.042	27.79	0.923	0.075
4	100.0%	0%	0%	29.88	0.949	0.042	27.39	0.919	0.077

Figure 3: Qualitative results generated by *LRM-Zero* trained on *Zeroverse* with (left two) and without boolean difference augmentation (right two). Right two *LRM-Zero*'s reconstruction results have structural failures on objects with concave shapes and complex structures.

The wireframe augmentation does not show significant improvements of the metric, but it increases the visual fidelity. As shown in Fig. 4, without wireframe augmentation in its training data, *LRM-Zero* fails to reconstruct objects with thin structures, e.g. chair and table legs, or rails.

Height-field augmentation Tab. 3 shows two experiments with and without height field augmentation. Both are trained on 120K objects consisting of 80K original compositional objects and 40K boolean augmentation objects. This setting is different from other ablation experiments in Tab. 2, because we had to synthesize and render objects with 0 height field probability, which do not exist in *Zeroverse*. We also uses the boolean-difference only augmentation to mitigate the effect of instability. These results reveal that height field augmentation can improve the results.

5.2 Training stability

As discussed in Sec. 5.1, adding augmentation substantially boosts *LRM-Zero*'s performance. However, it also makes *Zeroverse* more complex and thus introduces training instability in *LRM-Zero*. We explore various techniques to help stabilize the training from either the training side (i.e., decreasing perceptual loss weight, decreasing Guassian splatting scale clipping, decreasing view-angle threshold) or the data mixing ratio of augmentations (we found that height-filed augmentation does not introduce instability a lot thus kept it). The observations are summarized in Tab. 4. In general, we observe that shifting training hyperparameters from optimal would improve the stability. However, this would decrease the performance. Thus our final plan (as shown in experiment 6) is a more balanced augmentation mixing ratio, and only minimal change on the training side. More comprehensive experiments are in Appendix.

5.3 Generalization

We first validate the generalization of our *Zeroverse* by training a NeRF-based LRM model [49, 92] on it. NeRF-based model's architecture is different from GS-LRM. Also the 3D modeling is philosophically different: NeRF has a canonical space for Triplane (i.e., Eulerian representation) while Gaussian Splatting is pixel-aligned per-point prediction (i.e., Lagrangian representation).
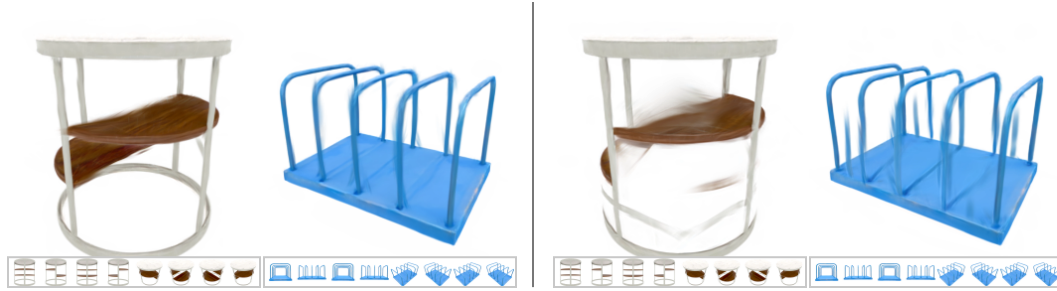
Figure 4: Qualitative results generated by *LRM-Zero* trained on default *Zeroverse* with (left two) and without wireframe augmentation (right two). Right two *LRM-Zero*'s reconstruction results have structural failures on objects with thin structures.

Table 3: Ablations studies on height-field augmentation.

	dataset			Height Field	GSO			ABO		
id	hf-only def. 40%	boolean def. 40%	wireframe def. 20%	HF probability def. 0.5	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1	60.0%	40.0%	0%	default	**30.24**	**0.952**	**0.039**	**28.31**	**0.926**	**0.072**
2				0	29.22	0.941	0.045	27.70	0.916	0.076

Despite of these differences, the results in Tab. 5 are similar to what we observed in GS-LRM that *Zeroverse*-trained model is competitive to Objaverse-trained models.

Besides the standard benchmark GSO and ABO, we also evaluate our *LRM-Zero* on diverse datasets to show its generalization, such as realistic 3D objects in OpenIllumination [55] and OmniObject3D [95], cross-evaluation on Zeroverse and Objaverse, and the generative outputs by Instant3D [49] and One2345++ [57]. As these experiments are for generalization test, we use 8 randomly-sampled input for OpenIllumination, OmniObject3D, Objaverse, and Zeroverse. For Instant3D and One2345++, we use the default camera setup of the generative model's outputs, where Instant3D and One2345++ have 4 and 6 structural cameras, respectively. As shown in Tab. 6, our *LRM-Zero* is competitive. We visualize the Instant3D and One2345++ results in Fig. 5, where *LRM-Zero* still work for these truly novel generated images, showcasing that *LRM-Zero* can be used in the 3D generation pipeline.

6 Related works

3D reconstruction is an important task in 3D vision. As 3D data is usually hard to capture, 3D reconstruction gives the ability to get 3D model from other modalities (e.g., images). Traditional methods [66, 61, 56, 14] on 3D reconstruction focuses on the per-sample optimization, where the 3D
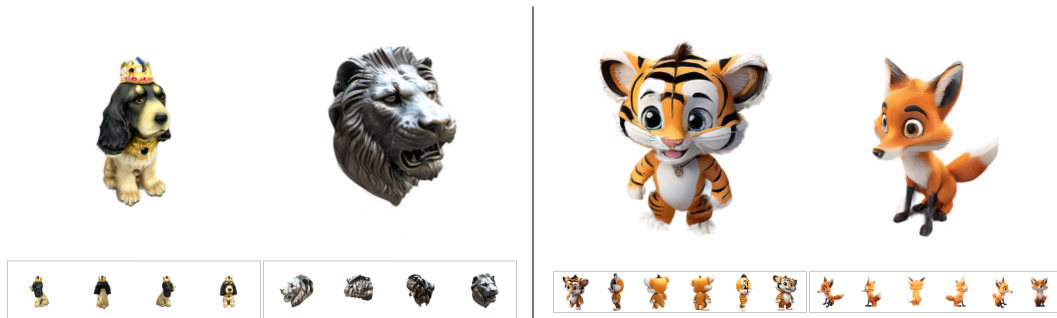

Figure 5: *LRM-Zero*'s qualitative results on Instant3D text-to-3D (left two) and One2345++ image-to-3D (right two) generated multi-view images.

Table 4: Illustrating the training stability issues when constructing the procedural *Zeroverse* dataset. The instability can be resolved either with training stabilizing techniques (e.g., reducing perceptual loss weight, Gaussian scale clipping, and view angle threshold), or with reducing the complexity of *Zeroverse*. 'failed' experiments are usually due to model divergence.

	dataset			training			result
id	hf-only	boolean	wireframe	perceptual loss weight (default 0.5)	Gaussian scale clipping (default -1.2)	view angle threshold (default 60)	GSO PSNR, if finished
1	100%	0%	0%	default	default	default	29.54
2	20%	80%	0%	default	default	default	failed
3				0.2	default	default	failed
4	40%	60%	0%	0.2	default	default	failed
5				0.2	-1.6	40	30.32
6	40%	40%	20%	0.2	default	default	30.78

Table 5: NeRF-LRM-Zero performs competitively against NeRF-LRM-Objv.

	GSO			ABO		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF-LRM-Zero	29.33	**0.936**	**0.065**	28.96	0.921	0.084
NeRF-LRM-Objv	**29.72**	**0.936**	**0.064**	**30.79**	**0.932**	**0.071**

shapes are parameterized and optimized by the rendering loss [61] or geometry loss [66, 25]. These optimization-based methods are usually slow and require adequate number of views (e.g., 100 views). Although methods are proposed [82, 30, 104, 62] to resolve these constraints for efficiency and view requirements [43, 64, 81], the speed is not largely improved.

Recent progresses advances this task with learning-based feed-forward methods [105, 94, 80, 63, 45, 50, 44]. Instead of optimization, these methods train a model from large-scale object [73, 106, 97, 23, 24] or scene [109, 71, 54] data to predict the shape directly. Besides the benefits of efficiency, these feed-forward methods can naturally support sparse-views as input (e.g., 4 to 12 input view images) because they learn data patterns from massive dataset. Some models can even go with extreme case of single-view reconstruction [105, 80, 41, 83], which needs to have data prior from realistic 3D data. Multi-view stereo methods [78, 103, 36, 87, 26, 88] are another family of feed-forward 3D reconstruction methods, but they cannot deal with sparse-view or single-view settings since they are based on local feature matching.

Synthetic data has been popular used in computer vision [38, 58, 32, 76], such as in segmentation [15, 21], object detection [42], image classification [39], deblurry [75], face analysis [93], etc. In 3D vision, synthetic data is widely used because of 3D data is harder to harvest, e.g., in depth estimation [4, 69], in optical flow [27, 60, 59], in finding multi-view correspondence [91], and for improving the 3D consistency of multi-view diffusion models [96]. Specifically to reconstruction, the

Table 6: Generalization of *LRM-Zero* to various evaluation datasets.

	OpenIllumination			OmniObject3D			Objaverse-test			Zeroverse-test		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GS-LRM	14.02	**0.598**	0.460	**29.32**	**0.940**	**0.055**	**28.37**	**0.920**	**0.079**	26.48	0.880	0.089
LRM-Zero	**14.44**	0.591	**0.455**	25.81	0.909	0.080	25.88	0.884	0.112	**28.23**	**0.912**	**0.068**

exploration of synthetic is mainly on specific categories, for example, for face [74], for human [33], constructions [47], or for evaluation [1, 31]. Some synthetic data are template-based [9, 34, 51, 102] and injecting human's knowledge about the semantic. Xu et al. [100, 101] and their subsequent works [52, 7, 53, 77, 110] have leveraged procedurally synthesized data for relighting, view synthesis, and various appearance acquisition and rendering tasks. However, these methods are designed for captures under controlled lighting conditions or objects with specific materials; additionally, their data is created on a relatively small scale. We revisit their procedural data generation workflow, extending it with additional data augmentation techniques and scaling it up to train large reconstruction models.

7 Limitations

In this paper, we mainly focus on providing a proof of concept on using synthetic to tackle one of the key problems in 3D vision: 3D reconstructions, and here are part of the limitations.

Scalability. The scalability of such synthetic-based method is still under investigation. We have done some initial exploration and the results can be found in Appendix. From these early experiments, it seems that the convergence property and optimal training hyperpameters might be different from the standard experimental setup with real data. The scaling-up exploration would naturally involve more resources (mainly computing resources, i.e., GPU hours) which is beyond our affordability.

Also, the community also lacks a study over the scalability of reconstruction models over 'real' data. Objaverse-XL [24] brings 10 more data over Objaverse but the data is much nosier, has different formats, contains a large portion without textures, and the legal concerns are not fully resolved. All these issues exposes challenges in understanding the scalability of the feed-forward reconstruction method.

Semantics. The synthetic data created in the way of Sec. 3 lacks of semantics (e.g., the data distribution is not supposed to match the real 3D world distribution). Thus this data might not be suitable to learn semantical-rich tasks. For the simplest example, *Zeroverse* is hard to train single-view reconstructions as shown in MCC [94], Shap-E [45], LRM [41], etc, which learn semantic from Objaverse [23], MvImgNet [106], and Co3D [73]. At the same time, we can complete single-view reconstruction by chaining with multi-view generator [57, 89] as shown in [49], relying on the semantical understanding of multi-view generation. The exact boundary of semantic tasks and intrinsic tasks in 3D vision is still under debate.

8 Broader Impacts

The broader impacts of this work are overall positive. First, the proof of concept in using synthesized data would largely reduce the bias inside the real dataset. As the model has weak inductive bias (i.e., through the use of the pure-transformer architecture), the potential semantical bias is mostly from the data. Second, the 3D data are usually having license concerns, where the synthesized data can help resolve. Third, as the 3D reconstruction can be potentially learned from synthetic data without real-world semantic information, we can possibly separate the 3D generation into two problems: generation and reconstruction. The reconstruction is mostly a semantics-free task.

On the other hands, this work can potentially largely lowers the bar of 3D reconstructions, for which data is the main blocker previously. The accessible 3D generation (when chaining with generative models as shown in [49]) and 3D reconstruction ability may introduce legal concerns on 3D licensing and moral concerns on 3D identities.

9 Conclusion

We introduced the *LRM-Zero* and its training data *Zeroverse*. *Zeroverse* is constructed with procedural synthesizing, where primitive shapes are composited, textured, and then augmented. We found the LRM model trained with *Zeroverse* can be competitive with Objaverse-trained LRMs, thus illustrating a promising direction of using synthetic data in 3D reconstruction research. We released our data creation code, and hope that it can help future research.

Acknowledgments

References

[1] Pranav Acharya, Daniel Lohn, Vivian Ross, Maya Ha, Alexander Rich, Ehsan Sayyad, and Tobias Höllerer. Using synthetic data generation to probe multi-view stereo networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1583–1591, 2021. 10

[2] AI@Meta. Llama 3 model card, 2024. 1

[3] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024. 1

[4] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2810, 2018. 9

[5] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation, 2024. 1

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 2

[7] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5960–5969, 2020. 4, 10

[8] 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. *arXiv preprint arXiv:2311.15127*, 2023. 1

[9] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 561–578. Springer, 2016. 10

[10] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

[11] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. 1

[12] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 3

[13] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. *arXiv preprint arXiv:2312.12337*, 2023. 3

[14] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 8

[15] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1841–1850, 2019. 9

[16] 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. 3

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 3, 6

[18] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F. Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21126–21136, 2022. 2, 5, 6, 18

[19] Together Computer. Redpajama: an open dataset for training large language models, 2023. 1

[20] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 1

[21] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7283–7290. IEEE, 2019. 9

[22] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023. 3, 6

[23] 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. 2, 9, 10, 19, 23

[24] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 9, 10, 21

[25] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021. 9

[26] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8585–8594, 2022. 9

[27] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 9

[28] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 2, 5, 6, 18

[29] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. 1

[30] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 9

[31] Mario Fuentes Reyes, Pablo d'Angelo, and Friedrich Fraundorfer. An evaluation of stereo and multiview algorithms for 3d reconstruction with synthetic data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:1021–1028, 2023. 10

[32] Adrien Gaidon, Antonio Lopez, and Florent Perronnin. The reasonable effectiveness of synthetic visual data. *International Journal of Computer Vision*, 126(9):899–901, 2018. 9

[33] Yongtao Ge, Wenjia Wang, Yongfan Chen, Yang Liu, Hao Chen, Xuan Wang, and Chunhua Shen. 3d human reconstruction in the wild with synthetic data using generative models. *ICLR Submission*, 2024. 10

[34] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 75–82. IEEE, 2018. 10

[35] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 1

[36] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 9

[37] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023. 1

[38] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4077–4085, 2016. 9

[39] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*, 2022. 9

[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[41] 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, 2024. 1, 3, 5, 9, 10

[42] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. Sail-vos 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1418–1428, 2021. 9

[43] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 9

[44] Hanwen Jiang, Qixing Huang, and Georgios Pavlakos. Real3d: Scaling up large reconstruction models with real-world images. *arXiv preprint arXiv:2406.08479*, 2024. 9

[45] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 9, 10

[46] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 2, 3

[47] Juhyeon Kim, Jeehoon Kim, Yohan Kim, and Hyoungkwan Kim. 3d reconstruction of large-scale scaffolds with synthetic data generation and an upsampling adversarial network. *Automation in Construction*, 156: 105108, 2023. 10

[48] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. VideoPoet: a large language model for zero-shot video generation. *arXiv*, 2312.14125, 2024. 1

[49] 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. 1, 7, 8, 10

[50] Rui Li, Tobias Fischer, Mattia Segu, Marc Pollefeys, Luc Van Gool, and Federico Tombari. Know your neighbors: Improving single-view reconstruction via spatial vision-language reasoning. In *CVPR*, 2024. 9

[51] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):1–17, 2017. 10

[52] 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. 4, 10

[53] 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. 10

[54] 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. *arXiv preprint arXiv:2312.16256*, 2023. 9

[55] Isabella Liu, Linghao Chen, Ziyang Fu, Liwen Wu, Haian Jin, Zhong Li, Chin Ming Ryan Wong, Yi Xu, Ravi Ramamoorthi, Zexiang Xu, et al. Openillumination: A multi-illumination dataset for inverse rendering evaluation on real objects. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 8

[56] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 8

[57] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023. 8, 10

[58] 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. 9

[59] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126:942–960, 2018. 9

[60] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 9

[61] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 8, 9

[62] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 9

[63] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 9

[64] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 9

[65] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, and et. al. Gpt-4 technical report, 2024. 1

[66] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 8, 9

[67] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[68] Guilherme Penedo, Hynek Kydlíček, Leandro von Werra, and Thomas Wolf. Fineweb, 2024. 1

[69] Benjamin Planche, Ziyan Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Oliver Lehmann, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, et al. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. In *2017 International conference on 3d vision (3DV)*, pages 1–10. IEEE, 2017. 9

[70] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[71] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 9

[72] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 3

[73] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 9, 10

[74] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision (3DV)*, pages 460–469. IEEE, 2016. 10

[75] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *European conference on computer vision*, pages 487–503. Springer, 2022. 9

[76] 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. 9

[77] 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. 10

[78] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 9

[79] 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. 1

[80] Bokui Shen, Xinchen Yan, Charles R Qi, Mahyar Najibi, Boyang Deng, Leonidas Guibas, Yin Zhou, and Dragomir Anguelov. Gina-3d: Learning to generate implicit neural assets in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4913–4926, 2023. 9

[81] Ruoxi Shi, Xinyue Wei, Cheng Wang, and Hao Su. Zerorf: Fast sparse view 360 {\deg} reconstruction with zero pretraining. *arXiv preprint arXiv:2312.09249*, 2023. 9

[82] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *CVPR*, 2022. 9

[83] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. *arXiv preprint arXiv:2312.13150*, 2023. 9

[84] 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. 1, 3

[85] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, and et. al. Llama 2: Open foundation and fine-tuned chat models, 2023. 1

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[87] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14203, 2021. 9

[88] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8606–8615, 2022. 9

[89] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 10

[90] 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. 1

[91] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 9

[92] 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. 7

[93] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021. 9

[94] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9065–9075, 2023. 9, 10

[95] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 3, 8

[96] Desai Xie, Jiahao Li, Hao Tan, Xin Sun, Zhixin Shu, Yi Zhou, Sai Bi, Sören Pirk, and Arie E Kaufman. Carve3d: Improving multi-view reconstruction consistency for diffusion models with rl finetuning. *arXiv preprint arXiv:2312.13980*, 2023. 9

[97] Zhangyang Xiong, Chenghong Li, Kenkun Liu, Hongjie Liao, Jianqiao Hu, Junyi Zhu, Shuliang Ning, Lingteng Qiu, Chongjie Wang, Shijie Wang, et al. Mvhumannet: A large-scale dataset of multi-view daily dressing human captures. *arXiv preprint arXiv:2312.02963*, 2023. 9

[98] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. 1

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

[100] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Trans. Graph.*, 37(4), 2018. 2, 4, 5, 10, 22

[101] 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. 4, 10

[102] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 601–610, 2020. 10

[103] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 9

[104] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv preprint arXiv:2302.14859*, 2023. 9

[105] 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. 9

[106] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 9, 10

[107] 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, 3, 5, 6, 18, 21, 23, 27, 28

[108] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5, 18

[109] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 9

[110] Shilin Zhu, Zexiang Xu, Henrik Wann Jensen, Hao Su, and Ravi Ramamoorthi. Deep kernel density estimation for photon mapping. In *Computer Graphics Forum*, pages 35–45. Wiley Online Library, 2020. 10
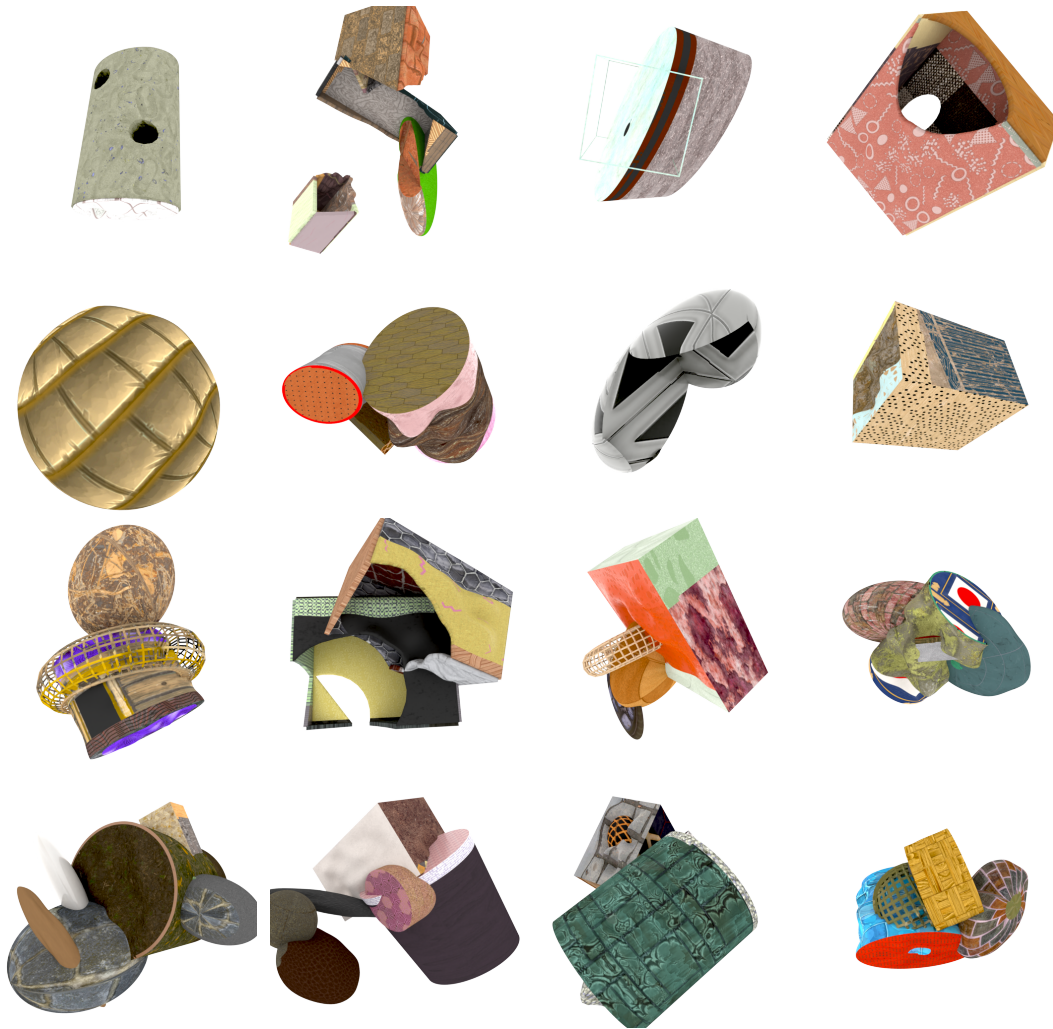
A Appendix summary

Figure 6: Uniformly sampled objects from *Zeroverse* to visualize its data distribution.

In Appendix, we provide more visualization results, more analysis, and more implementation details of our paper. Also note that both *LRM-Zero* and GS-LRM (when involving the results) refer to the final model (i.e., with respect to their training data and training parameters) instead of just the model architecture. We also use GS-LRM to refer to the model architecture as well for simplicity.

Table 7: Quantitative results comparing *LRM-Zero* with GS-LRM [107] under the 4-input-view setting. We use GSO [28] and ABO [18] evaluation datasets and PSNR, SSIM, and LPIPS [108] metrics. *LRM-Zero* demonstrates competitive performance against GS-LRM.

			GSO			ABO		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
4 input views	Res-512	GS-LRM	**30.52**	**0.952**	**0.050**	**29.09**	**0.925**	**0.085**
		LRM-Zero	28.49	0.937	0.063	25.40	0.893	0.115
	Res-256	GS-LRM	**29.59**	**0.944**	**0.050**	**28.92**	**0.926**	**0.074**
		LRM-Zero	27.78	0.927	0.062	25.41	0.886	0.106

Figure 7: Uniformly sampled objects from Objavserse [23] to visualize its data distribution.

B   Uniformly-sampled data visualization

In Fig 1, we visualize a curated set of the two dataset *Zeroverse* and Objaverse [23]. We here present uniformly random samples of the two dataset for better understandings of the data distribution *Zeroverse* is in Fig. 6 and Objaverse is in Fig. 7. Our *Zeroverse* is a random procedural data thus the datasest is quite diverse but do not have any semantic. The Objaverse dataset is sourced from Sketchfab [1]. The Objaverse dataset contains more semantic meaning (e.g., the transformers in the bottom left, some humanoids, furniture, and structure of house). There are also some shapes without explicit semantic meaning (e.g., the objects at position row-1-column-1, row-1-column3, row-3-column-1). Comparing the two datasets visually, usually the *Zeroverse* data is more complex and has higher-frequency textures than the Objaverse dataset in average. However, there are some data in Objaverse have more fine-grained small structures (e.g., the lamp at row-3-column-2) and more overall details (e.g., the transformer at bottom left), which currently the *Zeroverse* can not achieve. We think that these data difference contributed to the gap of the metric-wise results, but might be mitigated with improved procedural process. The visual difference is not significant as shown in Fig. 1. More visualization of *LRM-Zero* can be found at our website https://desaixie.github.io/lrm-zero/.

---

[1] https://sketchfab.com/
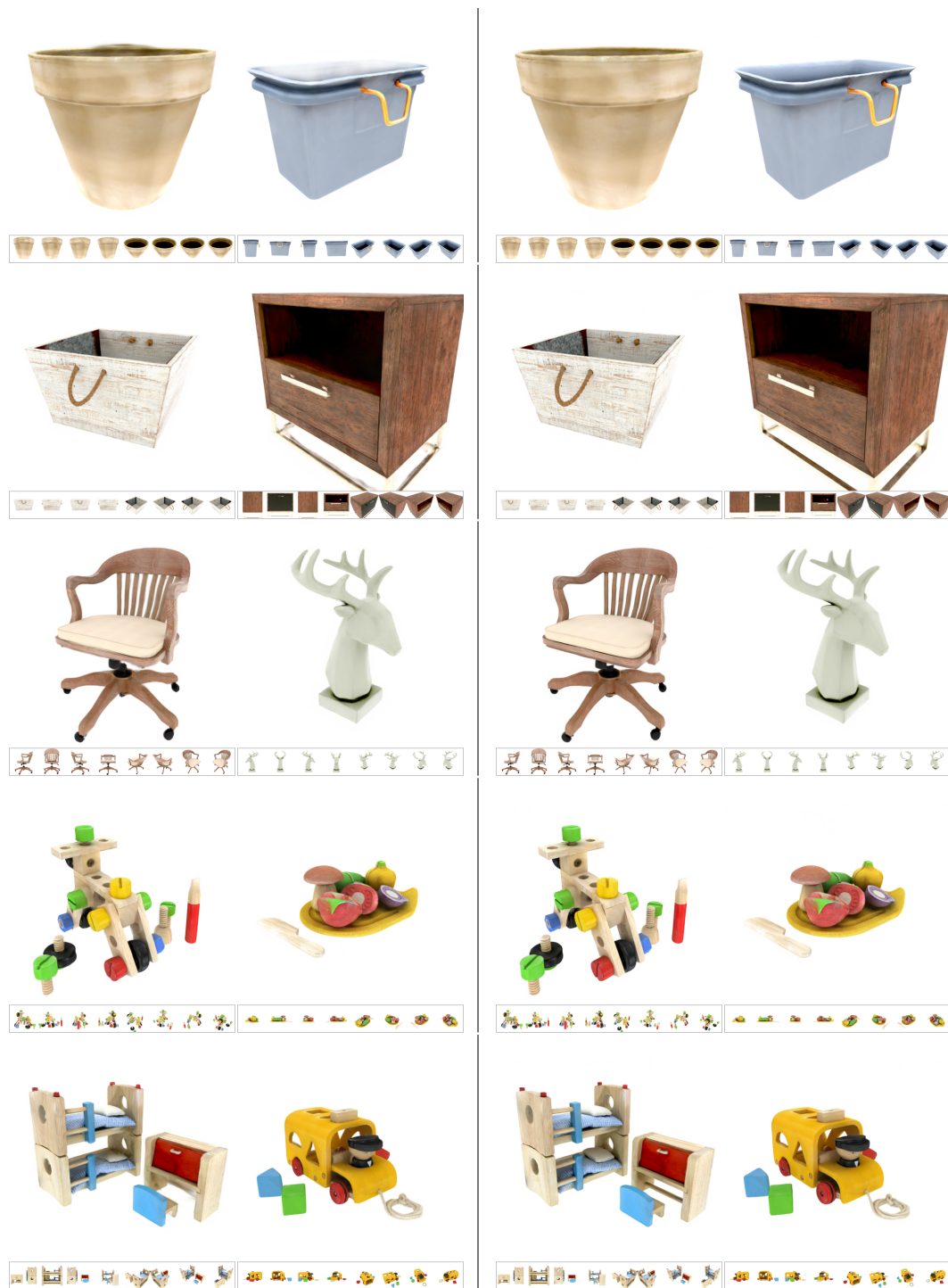
Figure 8: Qualitative comparison of *LRM-Zero* (left two columns) and GS-LRM (right two columns). When there is invisible region in the input views (first row), *LRM-Zero* produces poor reconstruction results. When the input views have good coverage (second row to fifth row), *LRM-Zero* performs similarly well as GS-LRM.

As shown in Tab. 1, *LRM-Zero* performs worse than GS-LRM on average. In Fig. 8, we show some qualitative comparisons of the two model's reconstruction results. Although we provide 8 input views, for the first row in Fig. 8, there are still invisible region in the input view. This leads to *LRM-Zero*'s

poor reconstruction results on them, as *LRM-Zero* does not learn sufficient semantics of real-world objects like GS-LRM. When there is sufficient coverage in the input views, as in the second row in Fig. 8, we can see that *LRM-Zero* and GS-LRM produces similar reconstruction results. In the third row to the fifth row, both *LRM-Zero* and GS-LRM performs similarly well on objects with complex shape.

On the other side, there is an extension of Objaverse named ObjaverseXL [24], which contains 10M 3D data from the Internet. We did not show it here and did not use it in our exploration for now. The quality of ObjavereXL is worse than Objaverse, and the data format (untextured shapes, point clouds) does not meet our data requirements. ObjaverseXL contains four subsets, 1. Sketchfab, which is the same to Objaverse, 2. Smithsonian [2], only about 2K data in this subset. 3. Thingiverse [3], with untextured mesh data. 4. Github, which we have not got license clearance on downloading all of the data. For these reasons, we did not use ObjaverseXL in our initial exploration.

C   Early Exploration on Scalability of *LRM-Zero*

We get mixed results from our scalability experiments on training steps, model size, and data size. We hypothesize that training convergence is the key to *LRM-Zero*'s performance. For many experiments in Tab. 8, the model underperforms its counterpart due to undertraining and lack of convergence. We suspect that the complexity of *Zeroverse* and our modified, lowered perceptual loss scale (discussed in Sec. 5.2), and the lowered learning rate for 2x and 3x model sizes contribute to the limited training convergence of *LRM-Zero*.

Table 8: *LRM-Zero*'s scaling experiment results.

	scaling			GSO			ABO		
id	Training Steps def. 1x, 80K	Model Size def. 1x, 300M	Data Size def. 1x, 400K	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
1	1x	1x	1x	30.78	0.957	0.036	28.82	0.934	0.065
2	2x	1x	1x	31.47	0.962	0.032	29.33	0.938	0.061
3	3x	1x	1x	31.11	0.960	0.035	29.18	0.937	0.063
4	1x	2x	1x	30.19	0.952	0.041	28.43	0.927	0.071
5	1x	3x	1x	30.34	0.954	0.040	28.58	0.929	0.069
6	2x	2x	1x	30.00	0.949	0.042	28.25	0.925	0.073
7	2x	2x	2x	30.56	0.955	0.038	28.84	0.931	0.068
8	2x	3x	10x	30.10	0.951	0.042	28.28	0.926	0.073
9	2x	1x	4x	31.15	0.960	0.034	29.02	0.935	0.064
10	2x	1x	20x	31.08	0.960	0.034	28.95	0.936	0.063
11	3x	1x	20x	**31.51**	**0.963**	**0.031**	**29.41**	**0.940**	**0.060**

Training steps   We discover that both GS-LRM [107] (experiment 2 in Tab. 9) and *LRM-Zero* (experiment 2 in Tab. 8) benefit from training on 2x (160K) steps, which was not explored in [107]. The gap between *LRM-Zero* and GS-LRM at 2x training steps is larger than the gap at 1x training steps. This shows that GS-LRM models, especially *LRM-Zero*, are undertrained at 1x (80K) training steps and aligns with our hypothesis that training convergence is the key.

At 1x (400K) training data size, Experiment 3 in Tab. 8, with 3x training steps, performs worse than experiment 2 in Tab. 8, indicating that the optimal training step for 1x model size and 1x *Zeroverse* data is somewhere between 2x and 3x. At 20x (8M) data size, 3x training steps (experiment 11) outperforms 2x training steps (experiment 10), indicating that more training steps is needed to

---

[2] https://3d.si.edu/
[3] https://www.thingiverse.com/

Table 9: GS-LRM's scaling experiment results.

	scaling			GSO			ABO		
id	Training Steps def. 1x, 80K	Model Size def. 1x, 300M	Data Size def. 2x, 800K	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1	1x	1x	2x	31.90	0.966	0.030	30.66	0.949	0.055
2	2x	1x	2x	**33.12**	**0.973**	**0.024**	**31.75**	**0.957**	**0.047**

converge well on more training data. This again aligns with our hypothesis about the importance of training convergence.

Model size   We experiment with 2x (700M parameters) and 3x (1B parameters) model sizes with a lowered learning rate of 3-e4 instead of the default 4e-4. At both 1x (experiment 4, 5) and 2x training steps (experiment 6, 7, 8), the larger models underperform compared their 1x model size counterparts (experiment 2, 3). We suspect that this is due to the fact that they are trained with a reduced learning rate of 3e-4 instead of the default 4e-4 for training stability purpose, which limits the models' training convergence. Also, we might need to tune more hyperparameters for the 2x and 3x model sizes to see their benefit. Due to the limited computation resources, we leave the exploration of larger model sizes as future work.

Data size   At 2x training steps with 1x model size, increasing data size in experiments 9 and 10 does not help model's performance compared to experiment 2. At 3x training steps with 1x model size, experiment 11 with 20x training data outperforms experiment 3, which overfits on 1x training data. By training on 8M *Zeroverse* objects with sufficient training steps, Experiment 11 is also our overall best model. We also observe that experiment 11 performs similarly as experiment 2, which uses only 400K training data. We hypothesize this is because the benefit of 20x data is limited by the capacity of the 1x model and that training a larger model on 20x data is promising. Due to the limited computing resources, we leave this as future work.

D   Zeroverse creation details

D.1   Sampling and Compositions of Primitives

We randomly sampled the number of the primitives from 1-to-9, with an unnormalized probability density of [5, 5, 5, 5, 5, 4, 3, 2, 1] accordingly. This sampling strategy samples more shapes with lower number of primitives that can smooth the dataset and increase the LRM training stability. Also, the later augmentations (especially the wireframe conversion and the boolean difference) can possibly introduce more fractions and parts.

For sampled primitives, we will scales their size randomly through each axis of the shape. We first put the shape in a normalized frames (e.g., align each edge of the cube with one of the axis) and randomly sample the scaling factors of each axis independently.

For compositing the scaled primitives, we randomly sample the centers of each primitive in the bounding box $[-1, 1]^3$, then we randomly sample a rotation of the shape (with three degrees of freedom). The compsition is done by simply putting all primitives together to a single shape. We do not consider the connectivity of the synthesized objects. See Fig. 6 for an example of disjoint shape (e.g., the top right one).

D.2   Augmentations

For the height-field augmentations, we constrain the maximal value of the height map to be proportional to the size of the face to avoid over-displacement. For more details, please refer to Xu et al. [100] and their released codebase.

For the boolean difference augmentation and wireframe conversion augmentation, we use the function from Blender [4]. In details, we use Blender's boolean modifier and solidify modifier to augment the initial shape. We first add a random primitive shape with random size and at a random vertex on the initial shape. Then, we apply the boolean modifier, where the new primitive shape, treated as a cutter, is subtracted from the initial shape. We do not use the torus for the subtraction as it might reduce training stability. Finally, we apply the solidify modifier to the cut shape to add thickness to it. The inside faces of the resulting cut shape will have the same texture as the outside faces.

We use Blender's wireframe modifier and subdivision modifier to create the wireframe of a primitive shape. We also add randomness to the thickness of the wireframe. The wireframes make up the thin structures that are missing in the initial shapes (Sec. 3.1) and add diversity to *Zeroverse*.

The augmentation is applied randomly with a given probability. We do not apply 'boolean difference' and 'wireframe' augmentations at the same time. This will breaks the independence assumption of different augmentations, but would avoid ultra-complex shapes, which improve training stability of the reconstruction model. In our implementation, we by default take 0.4 probability of the 'boolean difference' augmentation, 0.2 of the 'wireframe', and 0.4 of not applying both (as we want disjoint distribution for 'boolean difference' and 'wireframe' augmentations). The 'height-map' augmentation is applied independently to the above two shape augmentations, and always set at 0.5 independently for each surface. The results of other configuration can be found in the stability section (Sec. 5.2).

E Data synthesis distributed implementation details

In order to synthesize the massive amount of data, we tackle the synthesis and rendering of 8M shapes in *Zeroverse* with job parallelization. We run the independent shape synthesis and rendering jobs in parallel on 400 CPU nodes with a total of 38,400 CPU cores. The whole process takes 88 hours, where 5% of the time is spent on shape synthesis and 95% of the time is spent on rendering. Despite the relatively large number of CPU cores, the cost is negligible comparing with the training experiments cost on GPU. The training experiments in the main paper are mostly carried on a subset of 400K of the data. The early exploration of scalability uses the full dataset.

To avoid duplication in job parallelism, we first assign unique uuids and corresponding seeds for each shape to synthesize. Then, given the uuid, the seed, and the seed-induced fixed set of parameters, the synthesis and rendering jobs of each shape is independent from each other. In order to avoid exceeding local disk storage, we regularly upload the synthesized shapes and the rendered images to remote storage (e.g., AWS s3 in our experiment) and free the memory of their local copies. On each CPU node, we run multiple shape synthesis and rendering jobs in parallel to maximize the CPU utilization.

F More training stability results

As discussed in Sec. 5.1, adding augmentation substantially boosts *LRM-Zero*'s performance. However, it also makes *Zeroverse* much more complex and thus *LRM-Zero*'s training unstable, as shown in experiment 2 and 5 in Tab. 10. We explore various techniques to help stabilize the training. First, we adjust the perceptual loss weight. Compared to GS-LRM [107] trained on Objaverse [23] and *LRM-Zero* trained on no-augmentation *Zeroverse*, boolean augmented *Zeroverse* objects have high perceptual loss magnitudes that causes the excessive gradient norm. In experiment 2, we observe unusual, excessive gradient norm values in the range of 2-5. By reducing perceptual loss weight from 0.5 to 0.2 in experiment 3, the gradient norm values drop to the reasonable range of 0-1. However, experiment 3 still failed due to gradient norm explosion later.

Suspecting that boolean-augmented objects added too much dataset complexity, we reduced their ratio while increased the ratio no-augmentation objects in experiment 5 but still ended up with gradient norm explosion. Then, while keeping the same training dataset, we experimented with two techniques to further stabilize the training, i.e. reducing the Gaussian's scale clipping and the view angle threshold between the sampled views, in experiment 6, 7, and 8. Both techniques turn out to allow model to train stably for the full training steps. However, we notice that they also reduce the model's training set convergence and testing set performance.

---

[4] https://docs.blender.org/manual/en/latest/copyright.html

Table 10: Illustrating the training stability issues when constructing the procedural *Zeroverse* dataset. The instability can be resolved either with training stabilizing techniques (e.g., reducing perceptual loss weight, Gaussian scale clipping, and view angle threshold), or with reducing the complexity of *Zeroverse*. 'failed' experiments are usually due to model divergence.

	dataset			training			result
id	hf-only	boolean	wireframe	perceptual loss weight (default 0.5)	Gaussian scale clipping (default -1.2)	view angle threshold (default 60)	GSO PSNR, if finished
1	100%	0%	0%	default	default	default	29.54
2	20%	80%	0%	default	default	default	failed
3				0.2	default	default	failed
4				0	default	default	failed
5	40%	60%	0%	0.2	default	default	failed
6				0.2	-1.6	40	30.32
7				0.2	-1.6	default	30.42
8				0.2	default	40	30.85
9	92%	8%	0%	0.2	-1.6	40	30.14
10				0.2	-1.6	default	30.46
11				0.2	default	default	30.86
12	40%	40%	20%	0.2	default	default	30.78
13	85%	10%	5%	default	default	default	30.62

We shift the data distribution of *Zeroverse* to include more easy data and less complex boolean-augmented shapes. In experiment 9, 10, and 11, we find that training stabilizing techniques constraints the model's training set convergence and testing set performance. Experiment 11 further shows that by reducing the ratio of boolean-augmented complex shapes, we can stably train *LRM-Zero*. In our final version of *Zeroverse*, we aim to reduce the dataset complexity in order to achieve stable training from the data distribution side to avoid the convergence constraints that Gaussian's scale clipping or view angle threshold entails. In our empirical observation of the training data, we notice that adding boolean augmentation to objects with many basic shapes can be over complex. In experiment 12, which has the same data distribution as our final version of *Zeroverse*, we adopt a more even, smooth distribution between no-aug, boolean, and wireframe augmented objects and an easier basic-shape-number distribution to favor less basic shapes per object. It performs similarly as experiment 8 and 11. Additionally, in experiment 13, when reducing the ratio of data with boolean difference augmentation, we do not have training instability issues with the default training hyperparameters from GS-LRM, including the 0.5 perceptual loss weight.

G Additional Experiments

We conduct additional experiments to seek more explanations on the performance gap between *LRM-Zero* and GS-LRM. We perform an experiment on GS-LRM, training it on a randomly sampled subset of 200K Objaverse objects. As shown in Tab. 11, GS-LRM's performance only drops by 0.1 PSNR on GSO. We conduct joint training on both Objaverse and *Zeroverse* and compare it to training only on one of the two datasets in Tab. 12. Our result shows that training on both Objaverse and Zeroverse performs better than Zeroverse only, but worse than Objaverse only. Both of these experiments likely indicate that for the single-object reconstruction task, the results start to saturate with about 200K realistic data. Since the size of Objaverse is adequate for the single object reconstruction task, and the advantages of Zeroverse are not exploited in this task. However, we believe that the advantages

Table 11: Scaling down GS-LRM's training data size. When training on only 200K instead of 800K Objaverse data, GS-LRM's performance drops by only 0.1 PSNR on GSO.

	scaling			GSO			ABO		
id	Training Steps def. 1x, 80K	Model Size def. 1x, 300M	Data Size def. 2x, 800K	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
1	1x	1x	2x	**29.59**	**0.944**	**0.050**	**28.92**	**0.926**	**0.074**
2	1x	1x	0.5x	29.42	0.942	0.052	28.75	0.924	0.075

Table 12: *LRM-Zero* vs. *LRM-Zero-Obja* vs. GS-LRM at the 8-input-view, 256 resolution setting. Z means Zeroverse and O means Objaverse. The *LRM-Zero* (first row) and GS-LRM (second row) results are from experiment 9 in Tab. 8 and experiment 2 in Tab. 9. The *LRM-Zero-Obja* result (third row) is obtained by training on 800K *Zeroverse* data and 800K Objaverse data. While *LRM-Zero-Obja* outperforms *LRM-Zero*, it underperforms GS-LRM.

	scaling			GSO			ABO		
data	Training Steps def. 1x, 80K	Model Size def. 1x, 300M	Data Size def. 1x, 400K	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Z	2x	1x	4x	31.15	0.960	0.034	29.02	0.935	0.064
O	2x	1x	2x	**33.12**	**0.973**	**0.024**	**31.75**	**0.957**	**0.047**
Z&O	2x	1x	4x	32.11	0.968	0.027	30.70	0.950	0.052

of *Zeroverse*, i.e. the data size, texture quality, controllability are more valuable when extended to other tasks, such as scene reconstruction and relighting, where data is scarce.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main focus of this paper is to explore the possibility to train feed-forward reconstruction model synthetic data. This goal aligns with the title, abstract, and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see the Limitation Section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper proposes a feed-forward reconstruction model trained on synthetic data. For the model training, the paper follows the experimental setup (e.g., hyperparameters) in Zhang et al. [107]. We illustrate some key implementations but would refer the readers to the original paper for clarity. For the synthetic data, the paper tries the best to cover all the implementation details. Also, to fully support reproducing, the authors will release the data generation script for a concrete specification of the method.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper will release the synthetic data generation script, which is the main blocker for reproducing.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test follows Zhang et al. [107]. We describe most of the details in this paper and would refer readers to Zhang et al. [107] for clarity.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: As each experiment takes significant amount of compute resource, we don't have enough computing resources to complete the significance test.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes, provide both data creation and model training.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The paper is written on our own and follows the honest codes.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Please refer to the main paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited all previous papers that lead to our current paper's presentation.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release assets in the traditional meaning (e.g., labeled data). For the synthetic data creation, we have documented them well and will release the code to specify the process.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

53316