GeoLRM: Geometry-Aware Large Reconstruction Model for High-Quality 3D Gaussian Generation

Chubin Zhang[1,3] **Hongliang Song**[3] **Yi Wei**[2]
Yu Chen[3] **Jiwen Lu**[2] **Yansong Tang**[1,‡]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Department of Automation, Tsinghua University
[3]Alibaba Group
{zcb24, y-wei19}@mails.tsinghua.edu.cn,
{hongliang.shl, chenyu.cheny}@alibaba-inc.com,
lujiwen@tsinghua.edu.cn, tang.yansong@sz.tsinghua.edu.cn.
[‡] corresponding author

Abstract

In this work, we introduce the Geometry-Aware Large Reconstruction Model (GeoLRM), an approach which can predict high-quality assets with 512k Gaussians and 21 input images in only 11 GB GPU memory. Previous works neglect the inherent sparsity of 3D structure and do not utilize explicit geometric relationships between 3D and 2D images. This limits these methods to a low-resolution representation and makes it difficult to scale up to the dense views for better quality. GeoLRM tackles these issues by incorporating a novel 3D-aware transformer structure that directly processes 3D points and uses deformable cross-attention mechanisms to effectively integrate image features into 3D representations. We implement this solution through a two-stage pipeline: initially, a lightweight proposal network generates a sparse set of 3D anchor points from the posed image inputs; subsequently, a specialized reconstruction transformer refines the geometry and retrieves textural details. Extensive experimental results demonstrate that GeoLRM significantly outperforms existing models, especially for dense view inputs. We also demonstrate the practical applicability of our model with 3D generation tasks, showcasing its versatility and potential for broader adoption in real-world applications. The project page: https://linshan-bin.github.io/GeoLRM/.

1 Introduction

In fields ranging from robotics to virtual reality, the quality and diversity of 3D assets can dramatically influence both user experience and system efficiency. Historically, the creation of these assets has been a labour-intensive process, demanding the skills of expert artists and developers. While recent years have witnessed groundbreaking advancements in 2D image generation technologies, such as diffusion models [43, 44, 42] which iteratively refine images, their adaptation to 3D asset creation remains challenging. Directly applying diffusion models to 3D generation [20, 36] is less than satisfactory, primarily due to a dearth of large-scale and high-quality data. DreamFusion [40] innovatively optimize a 3D representation [2] by distilling the score of image distribution from pre-trained image diffusion models [43, 44]. However, this approach lacks a deep integration of 3D-specific knowledge, such as geometric consistency and spatial coherence, leading to significant issues such as the multi-head problem and the inconsistent 3D structure. Additionally, these methods require extensive per-scene optimizations, which severely limits their practical applications.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1772

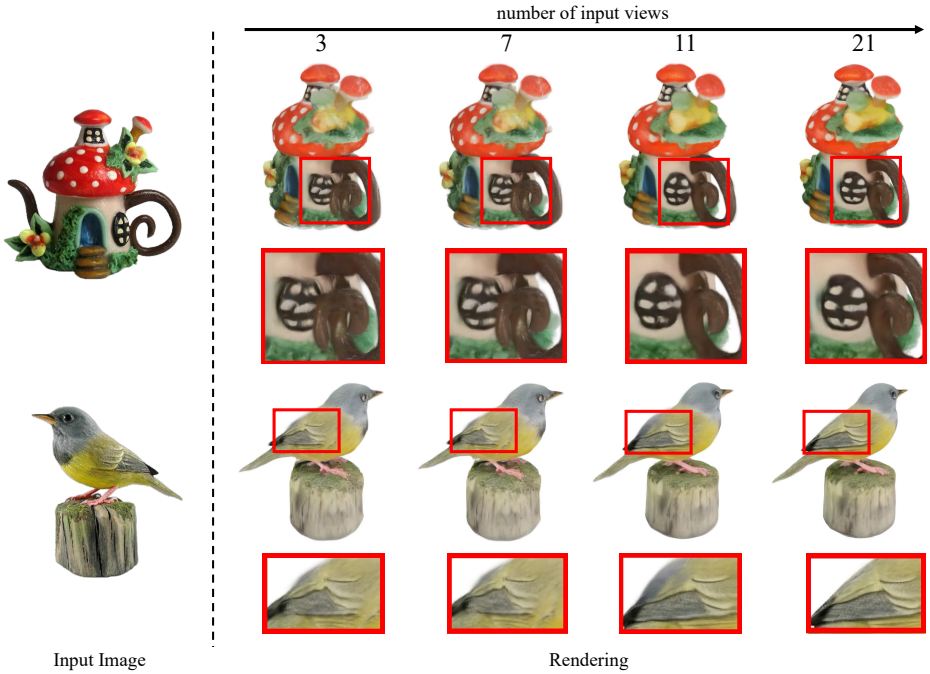

Figure 1: Image to 3D using GeoLRM. Initially, a 3D-aware diffusion model, specifically SV3D [60], transforms an input image into multiple views. Subsequently, these views are processed by our GeoLRM to generate detailed 3D assets. **Unlike other LRM-based approaches, GeoLRM notably improves as the number of input views increases.**

The introduction of the comprehensive 3D dataset Objaverse [12, 11] brings significant advancements for this field. Utilizing this dataset, researchers have fine-tuned 2D diffusion models to produce images consistent with 3D structures [28, 47, 48]. Moreover, recent innovations [74, 64, 54, 72, 65] have combined these 3D-aware models with large reconstruction models (LRMs) [18] to achieve rapid and accurate 3D image generation. These methods typically employ large transformers or UNet models that convert sparse-view images into 3D representations in a single forward step. While they excel in speed and maintaining 3D consistency, they confront two primary limitations. Firstly, previous works utilize triplanes [18, 72, 64] to represent the 3D models, wasting lots of features in regions devoid of actual content and involving dense computations during rendering. This *violates the sparse nature of 3D* as our analysis shows that the visible portions of the 3D models in the Objaverse dataset constitute only about 5% of the overall spatial volume. Though Gaussian-based methods [54, 74, 65] may use pixel-aligned Gaussians for better efficiency, this representation is incapable of recovering the unseen area and thus heavily relies on the input images. Secondly, previous works tend to *overlook the explicit geometric relationships between 3D and 2D images*, which results in ineffective processing. The tri-plane or pixel-aligned Gaussian tokens do not correspond to a specific space in 3D, thus being unable to utilize the projection relationship between 3D points and images. In other words, they conduct dense attention between the 3D queries and the image keys. This leads to the fact that these methods tend to reconstruct 3D with sparse view inputs but cannot achieve better performance with denser inputs.

To address these challenges, we introduce the geometry-aware large reconstruction model (GeoLRM) for 3D Gaussian generation. Our method centres on a 3D-aware reconstruction transformer that eschews conventional representations like triplanes or pixel-aligned Gaussians in favour of a direct interaction within the 3D space. However, directly generating 3D Gaussians in the whole 3D space requires huge memory costs. To this end, we first propose a specialized proposal network to predict an occupancy grid from input images. Only the occupied voxels will be further processed to generate 3D Gaussian features. The proposed transformer replaces the dense cross attention with deformable cross attention [86]. By projecting the input 3D tokens onto the corresponding image planes, these tokens only focus on the most relevant features, which greatly improves the effectiveness.

https://doi.org/10.52202/079017-1772

We trained our GeoLRM on the Objaverse dataset rendered by [41] and tested it on the Google Scanned Objects [13]. By integrating geometric principles, our model not only outperforms existing methods with the same number of inputs but also makes it possible to work with denser image inputs. Significantly, the model efficiently handles up to 21 images (even more if necessary), yielding superior 3D models in comparison to those generated from fewer images. Leveraging this capability, we integrated GeoLRM with SV3D [60] for high-quality 3D model generation.

In summary, our contributions are as follows:

- We introduce a two-stage pipeline that leverages the sparse nature of 3D data, resulting in a sparse 3DGS token representation suitable for extension to high resolution.
- We fully exploit the projection relationship between 3D points and 2D images, significantly reducing the space complexity of attention mechanisms in LRMs, thus enabling denser image input configurations.
- To the best of our knowledge, GeoLRM is the first to process dense inputs using LRM, potentially paving the way for integrating video generation models into 3D AIGC applications.

2 Related Work

2.1 Optimization-based 3D reconstruction

3D reconstruction from multi-view images has been extensively studied in computer vision for decades. While traditional methods like SfM [68, 58, 45] and MVS [46, 16] provide basic reconstruction and calibration, they lack robustness and expressiveness. Recent advancements leverage learning-based methods for better performance. Among these methods, NeRF [33] stands out for its capability of capturing high-frequency details. Following works [2, 83, 3, 34, 77, 8, 53, 4] further improve its performance and speed. Though NeRF has made a great improvement, the need to query tons of points during the rendering process makes it hard for real-time applications. 3D Gaussians [21] solves this problem by explicitly expressing a scene with 3D Gaussians and utilizing an efficient rasterization pipeline. These methods involve a per-scene optimization process and require dense multi-view images for a good reconstruction.

2.2 Large Reconstruction Model

Different from optimization-based 3D reconstruction methods, large reconstruction models [18, 22, 54, 74, 65, 82, 62, 64] are able to reconstruct 3D shapes in a feed-forward way. As the pioneer work of this area, the LRM [18] illustrates that the transformer backbone can effectively leverage the power of large-scale datasets and translate image tokens into implicit 3D triplanes under multi-view supervision. Beyond LRM, Instant3D [22] improves reconstruction quality with sparse-view inputs. It employs a two-stage paradigm, which first generates four views with the diffusion model and then regresses NeRF [33] from generated multi-view images. Instead of NeRF, InstantMesh [72] utilizes mesh representation to reconstruct 3D objects, which adopts a differentiable iso-surface extraction module. However, many works [54, 82, 74, 71] choose 3D Gaussians [21] as the outputs. GRM [74] proposes a transformer network to translate pixels to the set of pixel-aligned 3D Gaussians while LGM [54] uses an asymmetric UNet to predict and fuse 3D Gaussians. Compared with these methods, our GeoLRM projects multi-view features to the 3D space with cross-view attention mechanisms, which explicitly explores geometric knowledge.

2.3 3D generation

Early methods [6, 7, 15, 35, 51, 73, 37] in 3D generation area utilize 3D GANs to generate 3D-aware contents. Although some methods [32, 32, 85, 30, 10, 49, 80] replace 3D GANs with 3D diffusion models for high-quality generation, their generalization ability is bounded by the limited training data. Recently, proposed in DreamFusion [40], score distillation sampling (SDS) requires no 3D data and is able to leverage the great power of 2D text-to-image diffusion models [44, 43, 42]. Specifically, it optimizes a randomly-initialized 3D model and diffuses the render images with a pretrained diffusion model. As the follow-up works [63, 9, 26, 61, 55, 76, 27, 78, 25, 23, 41], many methods have been proposed to accelerate the optimization process or improve 3D generation quality. Different

 https://doi.org/10.52202/079017-1772

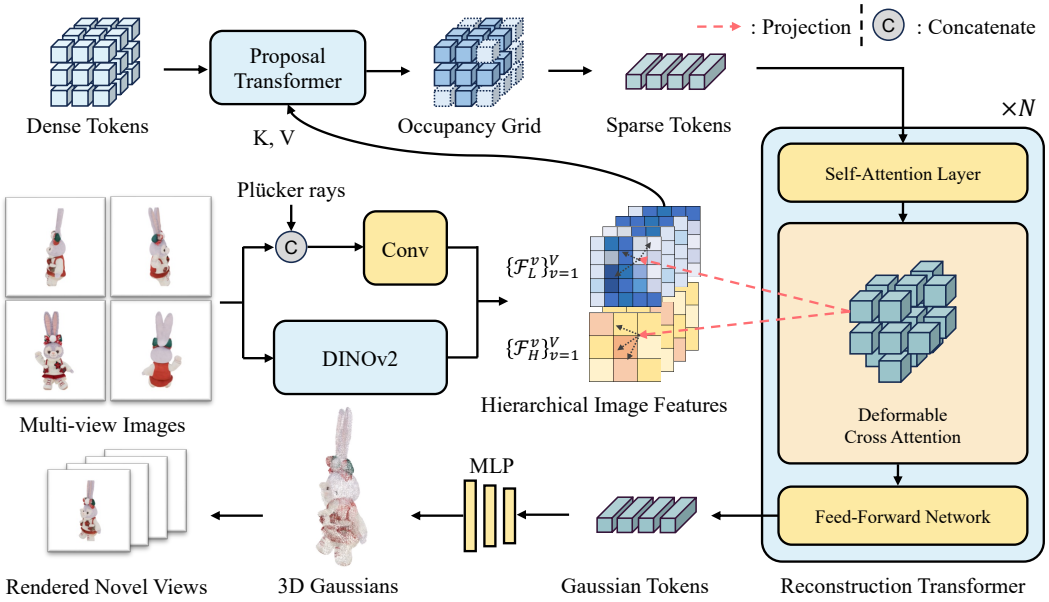

Figure 2: **Pipeline of the proposed GeoLRM**, a geometry-powered method for efficient image to 3D reconstruction. The process begins with the transformation of dense tokens into an occupancy grid via a Proposal Transformer, which captures spatial occupancy from hierarchical image features extracted using a combination of a convolutional layer and DINOv2 [38]. Sparse tokens representing occupied voxels are further processed through a Reconstruction Transformer that employs self-attention and deformable cross-attention mechanisms to refine geometry and retrieve texture details with 3D to 2D projection. Finally, the refined 3D tokens are converted into 3D Gaussians for real-time rendering.

with SDS-based methods, Zero-1-to-3 [28] fine-tunes the 2D diffusion models on a large-scale synthetic dataset to change the camera viewpoint of a given image. Similar to Zero-1-to-3, many other works [47, 60, 48, 75, 29, 67, 31, 69] aim to synthesize multi-view consistent images. Our method can reconstruct 3D contents based on these synthesis multi-view images.

3 Methodology

3.1 Overview

Figure 2 illustrates the pipeline of our proposed method. Our approach takes a set of images $\{I^i\}_{i=1}^N$ with their corresponding intrinsic $\{K^i\}_{i=1}^N$ and extrinsic $\{T^i\}_{i=1}^N$ as input. Initially, we encode input images into hierarchical image features and predict an occupancy grid with a proposal transformer. Each occupied voxel within this grid is considered a 3D anchor point. These 3D anchor points are then processed by a reconstruction transformer, refining their geometry and retrieving textural details. The proposal and reconstruction transformers share the same model architecture, which is further discussed in Section 3.2. The outputs of the reconstruction transformer are decoded into Gaussian features with a shallow MLP for rendering. Loss functions are described in Section 3.3.

3.2 Model Architecture

Our model architecture features a hierarchical image encoder for extracting high and low-level image feature maps along with a geometry-aware transformer for lifting 2D features into 3D representations.

Hierarchical Image Encoder Our method integrates both high and low-level features to enhance model performance. For high-level features, we utilize DINOv2 [38], which excels in single-image 3D tasks [1]. To capture low-level features, we combine Plücker ray embeddings and RGB values. The Plücker ray parameterizes each ray corresponding to a pixel by $\mathbf{r} = (\mathbf{d}, \mathbf{o} \times \mathbf{d})$, with $\mathbf{d}$ representing the ray's direction and $\mathbf{o}$ its origin [50, 75]. These embeddings, denoted as R^v for each image I^v, are concatenated with the RGB values of the image. This combined data is then integrated through a

https://doi.org/10.52202/079017-1772

convolution layer. The encoding processes are succinctly described by the equations:

$$\mathcal{F}_H^v = \text{DINOv2}(I^v), \tag{1}$$

$$\mathcal{F}_L^v = \text{Conv}(\text{Concat}(I^v, R^v)), \tag{2}$$

where $\mathcal{F}_H^v$ and $\mathcal{F}_L^v$ represent the high and low-level feature maps of image I^v, respectively.

Geometry-aware Transformer The geometry-aware transformer aims to efficiently lift image features to 3D. The proposal transformer and reconstruction transformer are both instances of this architecture. Previous methods [18, 54, 74, 65, 82] use tri-planes or pixel-aligned Gaussians to represent 3D contents. However, these data structures make it hard to utilize the projection relationships, causing dense computations. Instead, we use 3D anchor points, which serve as proxies for their surrounding points, significantly reducing the number of points we need to process. As detailed in Figure 2, each transformer block contains a self-attention layer, a deformable cross-attention layer and a feed-forward network (FFN). The model takes N anchor point features $\mathcal{F}_A = \{\boldsymbol{f}_i\}_{i=1}^N$ as input tokens. Each token $\boldsymbol{f}_i$ comprises the coordinate of the corresponding point and a shared learnable feature.

For the self-attention layer, a crucial problem is how to inject positional information into the sparse 3D tokens. We extend the Rotary Positional Embedding (RoPE) [52] to 3D conditions for relative positional embedding. For a query $\boldsymbol{q_m}$ and a key $\boldsymbol{k_n}$ at absolute position $\boldsymbol{m}$ and $\boldsymbol{n}$, we ensure that the inner product of embedded values reflects only the relative position information $\boldsymbol{m} - \boldsymbol{n}$. A direct yet promising way is splitting the features into three parts and applying RoPE [52] on each part with x, y, and z positions respectively.

As we can locate each anchor point in the 3D space, a possible way to lift 2D features to 3D is to project them to the feature maps with known poses and average the corresponding features. However, this method assumes an accurate anchor position, an equal contribution of all images and a good 3D correspondence of input images, which is often impractical, especially in 3D generation tasks. To tackle these issues, we employ deformable attention [86, 24, 66] for a robust fusion of image features. Given a 3D anchor point feature $\boldsymbol{f}_i$, its spatial coordinate $\boldsymbol{x}_i$ and multiple feature maps $\{\mathcal{F}^v\}_{v=1}^V$, the deformable attention mechanism is formulated as:

$$\text{DeformAttn}(\boldsymbol{f}_i, \boldsymbol{x}_i, \{\mathcal{F}^v\}_{v=1}^V) = \sum_{v=1}^{V} w_v [\sum_{k=1}^{K} A_k \mathcal{F}^v \langle \boldsymbol{p}_{iv} + \Delta \boldsymbol{p}_{ivk} \rangle], \tag{3}$$

where k indexes the sampled keys and K is the total sampled key numbers. $\boldsymbol{p}_{iv}$ is the projected 2D coordinate on feature map $\mathcal{F}^v$ and $\Delta \boldsymbol{p}_{ivk}$ is the sampled offset. $\langle \cdot \rangle$ indicates the interpolation operation. A_k is the attention weight predicted from $\boldsymbol{f}_i$. w_v is a per-view weight derived from the feature it weights. Notably, the prediction of $\Delta \boldsymbol{p}_{ivk}$ allows the network to correct the geometry error of anchor points and the inconsistency of input images; The w_v enables different importance levels for each image. To further enhance the representation ability of the model, this mechanism is extended to multi-head and multi-scale conditions.

Given input tokens $\mathcal{F}_A^{in}$, the transformer block enhances these tokens through a series of sophisticated transformations described as follows:

$$\mathcal{F}_A^{self} = \mathcal{F}_A^{in} + \text{SelfAttn}(\text{RMSNorm}(\mathcal{F}_A^{in})), \tag{4}$$

$$\mathcal{F}_A^{cross} = \mathcal{F}_A^{self} + \text{DeformCrossAttn}(\text{RMSNorm}(\mathcal{F}_A^{self}), \{(\mathcal{F}_H^v, \mathcal{F}_L^v)\}_{v=1}^V), \tag{5}$$

$$\mathcal{F}_A^{out} = \mathcal{F}_A^{cross} + \text{FFN}(\text{RMSNorm}(\mathcal{F}_A^{cross})). \tag{6}$$

This design introduces several improvements over the original transformer architecture [59]. By incorporating RMSNorm [79] for normalization and SiLU [14] for activation, we achieve more stable training dynamics and better performance.

Post-processing The proposal network takes a low-resolution dense grid (16^3) as anchor points. The output is upsampled to a high-resolution grid (128^3) with a linear layer. This grid is formulated to represent the occupancy probability of the corresponding area ($[-0.5, 0.5]^3$). The reconstruction transformer takes occupied voxels as anchor points. Each output token $\boldsymbol{f}_i$ is decoded into multiple 3D Gaussians $\{\boldsymbol{G}_{ij}\}_{j=1}^M$ with a multilayer perceptron. The 3D Gaussian $\boldsymbol{G}_{ij}$ is parameterized by the offset $\boldsymbol{o}_{ij}$ regarding the anchor points, 3-channel RGB $\boldsymbol{c}_{ij}$, 3-channel scale $\boldsymbol{s}_{ij}$, 4-channel rotation quaternion $\boldsymbol{\sigma}_{ij}$, and 1-channel opacity α_{ij}. We employ activation functions to limit the range of the

https://doi.org/10.52202/079017-1772

offset, scale and opacity for better training stability similar to [54]:

$$\boldsymbol{o}_{ij} = \text{Sigmoid}(\boldsymbol{o}'_{ij}) \cdot o_{\max}, \tag{7}$$

$$\boldsymbol{s}_{ij} = \text{Sigmoid}(\boldsymbol{s}'_{ij}) \cdot s_{\max}, \tag{8}$$

$$\alpha_{ij} = \text{Sigmoid}(\alpha'_{ij}), \tag{9}$$

where $o_{\max}, s_{\max}$ are predefined maximum values of offsets and scales. Given target camera views $\{\boldsymbol{c}_t\}_{t=1}^{T}$, the 3D Gaussians can be further rendered into images $\{\hat{I}_t\}_{t=1}^{T}$, alpha masks $\{\hat{M}_t\}_{t=1}^{T}$ and depth maps $\{\hat{D}_t\}_{t=1}^{T}$ through Gaussian splatting [21].

3.3 Training Objectives

We employ a two-stage training mechanism for our model. In the first stage, we train the proposal transformer using 3D occupancy ground truth. This stage presents a challenge as it involves a highly unbalanced binary classification task; only about 5% of the voxels are occupied. To address this imbalance, we employ a combination of binary cross-entropy loss and the scene-class affinity loss, as proposed in [5], to supervise the training process. For the generation of ground truth data, see A.1.

For the second stage, we supervise the rendered T images, alpha masks and depth maps with corresponding ground truth:

$$\mathcal{L} = \sum_{t=1}^{T} \left(\mathcal{L}_{\text{img}}(\hat{I}_t, I_t) + \mathcal{L}_{\text{mask}}(\hat{M}_t, M_t) + 0.2\mathcal{L}_{\text{depth}}(\hat{D}_t, D_t, I_t) \right), \tag{10}$$

$$\mathcal{L}_{\text{img}}(\hat{I}_t, I_t) = ||\hat{I}_t - I_t||_2 + 2\mathcal{L}_{\text{LPIPS}}(\hat{I}_t, I_t), \tag{11}$$

$$\mathcal{L}_{\text{mask}}(\hat{M}_t, M_t) = ||\hat{M}_t - M_t||_2, \tag{12}$$

$$\mathcal{L}_{\text{depth}}(\hat{D}_t, D_t, I_t) = \frac{1}{|\hat{D}_t|} \left|\left| \exp(-\Delta I_t) \odot \log(1 + |\hat{D}_t - D_t|) \right|\right|_1, \tag{13}$$

where $\mathcal{L}_{\text{LPIPS}}$ is the perceptual image patch similarity loss [84], $|\hat{D}_t|$ is the total number of pixels in $|\hat{D}_t|$, ΔI_t is the gradient of the current RGB image and $\odot$ is the element-wise multiplication operation. As demonstrated in [57], applying a logarithmic penalty and weighting the per-pixel depth errors with the image gradients result in a smoother geometric representation.

4 Experiments

4.1 Datasets

G-buffer Objaverse (GObjaverse) [41]: Used for training. Derived from the original Objaverse [12] dataset, GObjaverse includes high-quality renderings of albedo, RGB, depth, and normal images. These images are generated through a hybrid technique combining rasterization and path tracing. The dataset comprises approximately 280,000 normalized 3D models scaled to fit within a cubic space of $[-0.5, 0.5]^3$. GObjaverse employs a diverse camera setup involving:

- Two orbital paths yielding 36 views per model. This includes 24 views at elevations between 5° and 30° (incremented by 15° rotations) and 12 views at near-horizontal elevations from -5° to 5° (with 30° rotation steps).
- Additional top and bottom views for comprehensive spatial coverage.

Google Scanned Objects (GSO) [13]: Used for evaluation, this dataset is rendered similarly to GObjaverse. A random subset of 100 objects is selected to streamline the evaluation process.

OmniObject3D [70]: Also used for evaluation, this dataset is consistently rendered like GObjaverse. A random subset of 100 objects is chosen for efficient evaluation.

4.2 Implementation details

Our model features 330 million parameters distributed across two distinct image encoders and two transformers. The first encoder processes geometry with the 6-layer proposal transformer, while the

Table 1: Quantitative results on Google Scanned Objects (GSO) [13], where we used six views for inputs and four for evaluation. Inference time and memory usage account only for the reconstruction process. **Bold** and underline denote the highest and second-highest scores, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	FS ↑	Inf. Time (s)	Memory (GB)
LGM	20.76	0.832	0.227	0.295	0.703	**0.07**	7.23
CRM	22.78	0.843	0.190	0.213	0.831	0.30	5.93
InstantMesh	23.19	0.856	**0.166**	0.186	0.854	0.78	23.12
Ours	**23.57**	**0.872**	0.167	**0.167**	**0.892**	0.67	**4.92**

Table 2: Quantitative results on OmniObject3D [70]. **Bold** and underline denote the highest and second-highest scores, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	CD ↓	FS ↑
LGM	21.94	0.824	0.203	0.256	0.787
CRM	23.12	0.855	0.175	0.204	0.810
InstantMesh	23.86	0.860	0.139	0.178	0.834
Ours	**24.74**	**0.883**	**0.134**	**0.156**	**0.863**

second focuses more on textures crucial with the 16-layer reconstruction transformer. During training, we maintain a maximum number of transformer input tokens of 4k and randomly select 8 views from a possible 38 for supervision. From these 8 views, we randomly select 1 to 7 views as inputs to predict the remaining views. This flexibility in view selection not only tests the robustness of our method but also mimics real-world scenarios where complete data may not always be available. Both input and rendering resolutions are maintained at 448x448 pixels. At the testing and inference stages, we use a resolution of 512x512 to align with existing methods. Besides, the number of input tokens is extended to 16k during testing, showcasing its scalability without the need for fine-tuning. Detailed information on our model's architecture and training procedures can be found in Section A.3.

4.3 Quantitative Results

We evaluated the quality of reconstructed assets from sparse view inputs by analyzing both 2D visual and 3D geometric aspects on the GSO and OmniObject3D dataset [13]. Visual quality was assessed by comparing rendered views to ground truth images using metrics such as PSNR, SSIM, and LPIPS. Geometric accuracy was evaluated by aligning our models to the ground truth coordinate systems and measuring discrepancies using Chamfer Distance and F-Score at a threshold of 0.2, with point samples totalling 16,000 from the ground truth surfaces. Our method was quantitatively compared against established baselines, including LGM [54], CRM [64], and InstantMesh [72]. We avoided comparisons with proprietary methods due to the unavailability of their test splits. Similarly, we excluded comparisons with OpenLRM [17] and TripoSR [56] as these methods are tailored for single image inputs, which would be unfair to compare with.

Our approach achieved state-of-the-art performance in four out of the five metrics studied. Although InstantMesh showed slightly higher LPIPS on the GSO dataset, attributed to its mesh-based smoothing capabilities, our method demonstrated superior geometric accuracy, benefiting from explicit modelling of the 3D-to-2D relationship.

In another experiment, outlined in Table 3, we observed a notable trend: our model's performance consistently improves with more input views while maintaining low computational costs. This indicates robust scalability, a critical feature for practical applications. In contrast, the performance of InstantMesh [72], does not follow this pattern. Specifically, InstantMesh shows a decline in performance when the input views increase to 12. This degradation could be due to two primary factors. First, the low-resolution tri-planes may reach their maximum capacity to represent details. Second, the model tends to oversmooth details when handling a large volume of image tokens. Our approach strategically addresses these issues. We employ an extendable sequence of 3D tokens that can be dynamically adjusted to fit the resolution requirements. Additionally, our model features deformable attention mechanisms that intelligently focus on the most pertinent information, preventing the loss of critical details.

https://doi.org/10.52202/079017-1772

Table 3: Quantitative results on Google Scanned Objects (GSO) with different numbers of input views. We keep the same four views for testing while changing the number of input views. **Bold** denotes the highest score.

Num Input	PSNR		SSIM		Inf. Time (s)		Memory (GB)	
	InstantMesh	Ours	InstantMesh	Ours	InstantMesh	Ours	InstantMesh	Ours
4	**22.87**	22.84	0.832	**0.851**	0.68	**0.51**	22.09	**4.30**
8	23.22	**23.82**	0.861	**0.883**	0.87	**0.84**	24.35	**5.50**
12	23.05	**24.43**	0.843	**0.892**	**1.07**	1.16	24.62	**6.96**
16	23.15	**24.79**	0.861	**0.903**	**1.30**	1.51	26.69	**8.23**
20	23.25	**25.13**	0.895	**0.905**	**1.62**	1.84	28.73	**9.43**

Figure 3: Qualitative comparisons of different image-3D methods. **Better viewed when zoomed in.**

4.4 Qualitative Results

We conducted a qualitative analysis comparing our method with several LRM-based baselines, including TripoSR [17], LGM [54], CRM [64], and InstantMesh [72], maintaining their original settings to ensure optimal performance. In our approach, we utilized the SV3D [60] technology to generate 21 multi-view images, significantly enhancing the resolution and textural details of the 3D Gaussians produced, as illustrated in Figure 3. Furthermore, as shown in Figure 4, employing InstantMesh to reconstruct these images did not yield satisfactory outcomes, corroborating our quantitative findings. This demonstrates the superior capability of our method in handling more complex 3D reconstructions.

https://doi.org/10.52202/079017-1772

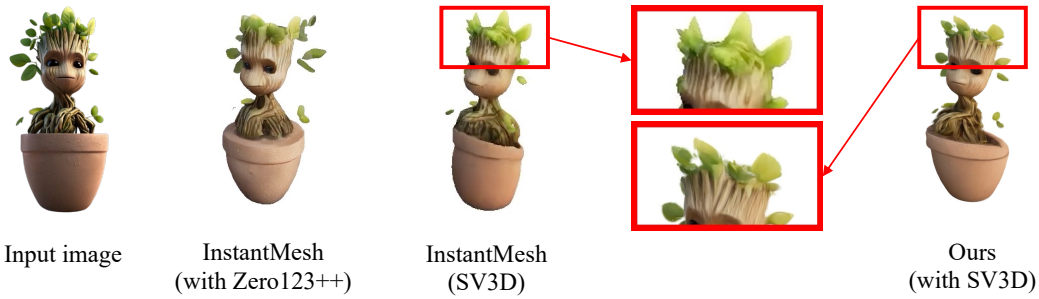

Figure 4: Qualitative comparison concerning scalability in input views.

4.5 Ablation Study

In this part, We provide ablation studies for the key designs of our method as shown in Table 4. Due to the limited computational sources, the ablation is done using a smaller reconstruction model (12 layers) and lower resolution (224x224).

Table 4: Ablation study of some key designs. Models are tested on the GSO dataset [13]. Upper: 6 input views and 4 testing views. Lower: different input views. **Bold** and underline denote the highest and second-highest scores, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
W/o Plücker rays	20.64	0.826	0.244
W/o low-level features	20.29	0.817	0.246
W/o high-level features	15.85	0.798	0.289
W/o 3D RoPE	20.52	0.827	0.224
Fixed # input views	**20.97**	**0.839**	0.220
Full model	20.73	0.831	**0.216**

Method	4 Inputs		8 Inputs		12 Inputs	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Fixed # input views	19.72	0.822	20.85	0.833	21.43	0.838
Full model	**19.94**	**0.835**	**21.16**	**0.840**	**22.04**	**0.853**

Figure 5: Effects of excluding high-level and low-level features in the image encoder.

Hierarchical Image Encoder Our ablation study underscores the critical role of hierarchical image features in reconstruction tasks, which necessitate both high-level semantic information (e.g., object identity and arrangement) and low-level texture information (e.g., surface patterns and colors). As illustrated in Figure 5, the absence of high-level features leads to model instability, while omitting low-level features results in a loss of textural detail. This dual requirement emphasizes the model's reliance on a comprehensive feature set for accurate image reconstruction. We also performed an ablation study regarding the Plücker ray embeddings in the low-level encoder. These coordinates assist the model in learning camera directions, contributing to an improved performance.

https://doi.org/10.52202/079017-1772

3D RoPE In transformer-based architectures, the role of positional embeddings is critical for accurately interpreting sequence data positions. A key question arises: With the reconstruction transformer employing deformable cross-attention to elevate 2D features to 3D, is positional embedding still necessary? Our ablation studies confirm its necessity. Notably, 3D RoPE significantly enhances the model's ability to handle longer sequences. For instance, increasing the sequence length from 4k to 16k elements, models equipped with 3D RoPE exhibited a PSNR improvement of 0.4, compared to a 0.2 improvement in models lacking 3D RoPE. This observation aligns with the 1D RoPE [52].

Dynamic Input The ablation study demonstrates a decrease in performance when employing our dynamic input view strategy compared to the fixed 6-input view setting when the training and testing phases were consistent. Despite this, the dynamic input strategy enhances the model's ability to generalize across different input configurations. This adaptability is critical for handling more complex scenarios, aligning with our primary objectives.

Deformable attention As shown in Table 5, the ablation results indicate that increasing the number of sampling points in the deformable attention generally improves performance. Given the trade-off between computational cost and performance gain, we find that using 8 sampling points strikes the best balance.

Table 5: Ablation study of deformable attention. '0 sampling points' means directly using the projected points without any deformation. **Bold** and underline denote the highest and second-highest scores, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
0 sampling points	19.52	0.802	0.265
4 sampling points	20.21	0.819	0.238
8 sampling points	<u>20.73</u>	<u>0.839</u>	<u>0.220</u>
16 sampling points	**20.80**	**0.846**	**0.219**

5 Conclusion

In this paper, we present GeoLRM, a geometry-aware large reconstruction model designed to improve the efficiency and quality of 3D generation. Our approach distinguishes itself from previous methods by effectively utilizing the inherent sparsity of 3D structures and explicitly integrating geometric relationships between 3D and 2D images. The GeoLRM framework employs a 3D-aware transformer architecture that predicts 3D Gaussians through a sophisticated coarse-to-fine methodology. Initially, a proposal network estimates coarse occupancy grids, which serve as foundational 3D anchor points for subsequent refinement. The second stage leverages deformable cross-attention to enhance the 3D structure, integrating detailed textural information. Extensive experiments validate that GeoLRM can process higher resolutions and accommodate denser image inputs, outperforming existing models in terms of detail and accuracy. This innovation demonstrates significant potential for real-world applications, particularly in domains where dense view inputs can enhance output quality and user experience. GeoLRM's ability to handle up to 21 images efficiently underscores its scalability and adaptability, paving the way for integration with advanced video generation technologies.

6 Limitation

While GeoLRM achieves impressive reconstruction quality, it does so through a two-stage process, which is not inherently end-to-end. This segmentation can lead to the accumulation of errors. The reliance on a proposal network is currently indispensable due to the computational intensity of processing Gaussian points across the entire 3D space. This necessity introduces potential inefficiencies and constraints that could hinder real-time applications. Future research will focus on developing an end-to-end solution that integrates these stages seamlessly, reducing error propagation and optimizing processing time. By addressing these limitations, we aim to enhance the model's robustness and applicability across a broader range of 3D generation tasks.

Acknowledgement

This work was supported in part by the Beijing Natural Science Foundation under Grant No. L247009 and in part by Young Elite Scientists Sponsorship Program by CAST (No. 2024QNRC003).

References

[1] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. *arXiv preprint arXiv:2404.08636*, 2024.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, pages 19697–19705, 2023.

[5] Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022.

[6] 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 *CVPR*, pages 16123–16133, 2022.

[7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021.

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, pages 333–350. Springer, 2022.

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, pages 22246–22256, 2023.

[10] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *ICCV*, pages 2262–2272, 2023.

[11] 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. *NeurIPS*, 36, 2024.

[12] 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 *CVPR*, pages 13142–13153, 2023.

[13] 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 *ICRA*, pages 2553–2560. IEEE, 2022.

[14] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[15] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *NeurIPS*, 35:31841–31854, 2022.

[16] Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *CVPR*, volume 2, pages 2402–2409. IEEE, 2006.

[17] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM`, 2023.

[18] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2023.

 https://doi.org/10.52202/079017-1772

[19] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. *arXiv preprint arXiv:2311.12754*, 2023.

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

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

[22] 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. In *ICLR*, 2023.

[23] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. In *CVPR*, 2024.

[24] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, pages 1–18. Springer, 2022.

[25] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *CVPR*, 2024.

[26] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, pages 300–309, 2023.

[27] Fangfu Liu, Diankun Wu, Yi Wei, Yongming Rao, and Yueqi Duan. Sherpa3d: Boosting high-fidelity text-to-3d generation via coarse 3d prior. In *CVPR*, 2024.

[28] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9298–9309, 2023.

[29] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *ICLR*, 2024.

[30] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *ICLR*, 2023.

[31] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

[32] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *CVPR*, pages 12923–12932, 2023.

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):102:1–102:15, July 2022.

[35] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, pages 7588–7597, 2019.

[36] 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.

[37] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021.

[38] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[39] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*, 2023.

[40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[41] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv preprint arXiv:2311.16918*, 2023.

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

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022.

[45] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016.

[46] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–528. IEEE, 2006.

[47] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[48] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2023.

[49] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *CVPR*, pages 20887–20897, 2023.

[50] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 34:19313–19325, 2021.

[51] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *NeurIPS*, 35:24487–24501, 2022.

[52] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[53] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. 2022 ieee. In *CVPR*, pages 5449–5459, 2021.

[54] 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.

[55] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024.

[56] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

[57] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024.

[58] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

 https://doi.org/10.52202/079017-1772

[60] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024.

[61] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pages 12619–12629, 2023.

[62] 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. In *ICLR*, 2023.

[63] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *NeurIPS*, 36, 2024.

[64] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024.

[65] 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.

[66] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, pages 21729–21740, 2023.

[67] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. In *ICLR*, 2024.

[68] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

[69] Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. In *CVPR*, 2024.

[70] 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 *CVPR*, pages 803–814, 2023.

[71] Dejia Xu, Ye Yuan, Morteza Mardani, Sifei Liu, Jiaming Song, Zhangyang Wang, and Arash Vahdat. Agg: Amortized generative 3d gaussians for single image to 3d. *arXiv preprint arXiv:2401.04099*, 2024.

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

[73] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, pages 18430–18439, 2022.

[74] 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.

[75] 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. In *ICLR*, 2023.

[76] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. In *CVPR*, 2024.

[77] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2(3):6, 2021.

[78] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. In *ICLR*, 2024.

[79] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *NeurIPS*, 32, 2019.

[80] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *TOG*, 42(4):1–16, 2023.

[81] Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *arXiv preprint arXiv:2312.09243*, 2023.

[82] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024.

[83] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

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

[85] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, pages 5826–5835, 2021.

[86] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020.

 https://doi.org/10.52202/079017-1772

A Appendix

A.1 Occupancy Ground Truth

Previous studies [39, 81, 19] have investigated the task of vision-centric occupancy prediction. However, these approaches often exhibit significant performance discrepancies when compared to 3D methods. To bridge this gap, we leverage depth maps from the GObjaverse dataset to generate accurate 3D occupancy ground truths. This process begins by transforming each pixel in the depth map, represented as $\mathbf{p^i} = [u, v, 1]^T$, into a point in world coordinates. This transformation uses both the intrinsic matrix K and the extrinsic parameters T, consisting of a rotation matrix R and a translation vector $\mathbf{t}$, as shown in the equation:

$$\mathbf{p^w} = R(d \cdot K^{-1}\mathbf{p^i}) + \mathbf{t}, \tag{14}$$

where d denotes the depth at pixel $\mathbf{p^i}$. Subsequently, these world coordinates are voxelized to pinpoint occupied voxel centres:

$$V = \left\{ \left\lfloor \frac{P}{\epsilon} \right\rfloor \right\} \cdot \epsilon, \tag{15}$$

where P includes all points in three-dimensional space, V represents the voxel centers, and the voxel size ϵ is set at $1/128$. The voxelization helps in reducing redundancy by removing duplicate entries.

Figure A: Image-to-3D generation with mesh extraction results.

A.2 Mesh Extraction from 3D Gaussians

We adopt the mesh extraction pipeline from [54] to derive high-quality mesh representations from 3D Gaussians. Figure A illustrates the mesh generation results of our method, while Figure B compares our generated mesh with other techniques. The results demonstrate the effectiveness of our approach, despite some loss of detail during conversion.

Figure B: Comparison of the generated meshes.

A.3 More Implementation Details

We illustrate the details of network architecture and training procedure in Table A. We train both the proposal transformer and the reconstruction transformer for 12 epochs on GObjaverse [41], which takes 0.5 and 2 days respectively on 32 A100 40G. For the proposal transformer, we use a batch size of 2 per GPU and apply mixed-precision training with BF16 data type. For the reconstruction transformer, we use a batch size of 1 per GPU and keep the full precision. We note that the second stage is particularly sensitive to the data type and would fail if using mixed-precision.

https://doi.org/10.52202/079017-1772

Table A: Implementation details.

Proposal Transformer	Image encoder	DINOv2 (ViT-B/14) + Conv
	# layers	6
	# attention head	16
	# deformed points	8
	Image feature dimension	384
	3D feature dimension	384
	Max sequence length	4096
Reconstruction Transformer	Image encoder	DINOv2 (ViT-B/14) + Conv
	# layers	16
	# attention head	16
	# deformed points	8
	Image feature dimension	384
	3D feature dimension	768
	Max sequence length	4096
	# Gaussians per token	32
Training details	Epoch	12
	Learning rate	1e-4
	Learning rate scheduler	Cosine
	Optimizer	AdamW
	(Beta1, Beta2)	(0.9, 0.95)
	Weight decay	0.05
	Warm-up	1500
	Gradient accumulation	8
	Gradient clip	4
	# GPU	32

A.4 Social Impact

3D AIGC is transforming sectors by automating realistic 3D model creation. In entertainment, it streamlines film and game production, reducing costs and enhancing experiences. Education benefits from immersive VR simulations for deeper learning. Architecture sees rapid design visualization and urban planning improvements. Challenges include job displacement and ethical concerns over content authenticity. Addressing these requires legal and policy measures, such as clear copyright laws and standards to protect intellectual property. Developing advanced content moderation tools can detect false content, and enhancing AI security can prevent misuse. By focusing on these solutions, we can mitigate negative impacts and maximize the positive contributions of 3D AIGC to society.

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 claims made in the abstract and introduction do reflect the paper's contributions and scope.

   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: Limitations are discussed.

   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]

https://doi.org/10.52202/079017-1772

Justification: The paper does not include theoretical results.

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: We provide detailed information about our methods and the results are reproducible.

   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: We have released the code.

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: We have included necessary details.

   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: The computationally intensive nature of the training procedure made it impractical to report error bars.

   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).

https://doi.org/10.52202/079017-1772

- 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: We reported sufficient information on the computer resources.

   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: Our research adheres fully to the NeurIPS Code of Ethics.

   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: We have discussed both the positive and negative social impacts of our method in the appendix.

    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: The dataset we use has no risks.

    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 properly credited used assets.

    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.

https://doi.org/10.52202/079017-1772

- 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: The paper does not release new assets.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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.