ControLRM: Fast and Controllable 3D Generation via Large Reconstruction Model

Hongbin Xu, Weitao Chen, Zhipeng Zhou, Feng Xiao, Baigui Sun, Liefeng Bo, Mike Zheng Shou, Wenxiong Kang,

Abstract—Despite recent advancements in 3D generation methods, achieving controllability still remains a challenging issue. Current approaches utilizing score-distillation sampling are hindered by laborious procedures that consume a significant amount of time. Furthermore, the process of first generating 2D representations and then mapping them to 3D lacks internal alignment between the two forms of representation. To address these challenges, we introduce ControLRM, an end-to-end feed-forward model designed for rapid and controllable 3D generation using a large reconstruction model (LRM). ControLRM comprises a 2D condition generator, a condition encoding transformer, and a triplane decoder transformer. Instead of training our model from scratch, we advocate for a joint training framework. In the condition training branch, we lock the triplane decoder and reuses the deep and robust encoding layers pretrained with millions of 3D data in LRM. In the image training branch, we unlock the triplane decoder to establish an implicit alignment between the 2D and 3D representations. To ensure unbiased evaluation, we curate evaluation samples from three distinct datasets (G-OBJ, GSO, ABO) rather than relying on cherry-picking manual generation. The comprehensive experiments conducted on quantitative and qualitative comparisons of 3D controllability and generation quality demonstrate the strong generalization capacity of our proposed approach. For access to our project page and code, please visit our project page.

Index Terms—Large Reconstruction Model, Controllable 3D Generation, Neural Radiance Fields.

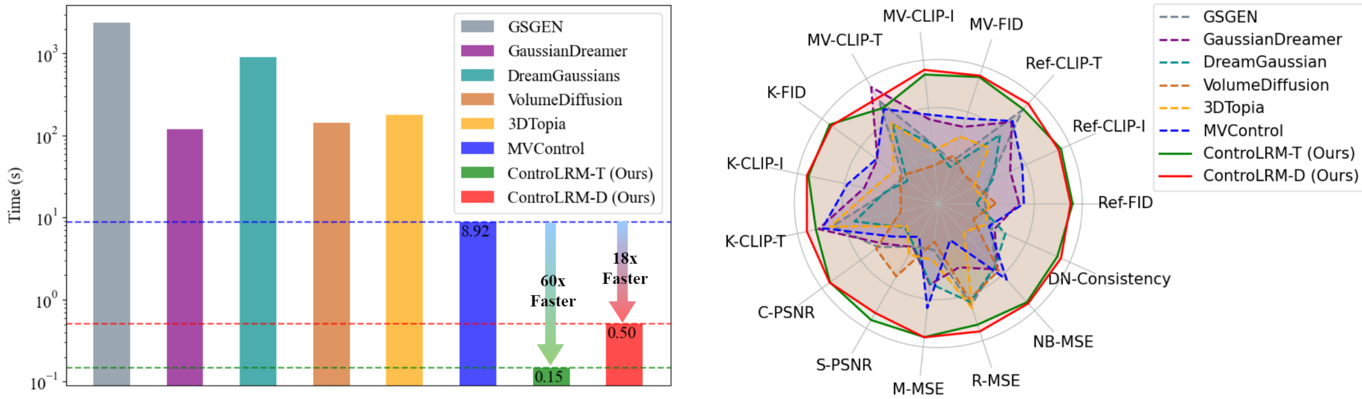
1 Introduction

THE potential of 3D content generation spans various sectors such as digital games, virtual reality/augmented reality (VR/AR), and filmmaking. Fundamental techniques in 3D content creation, such as text-to-3D and image-to-3D methods, offer substantial benefits by significantly reducing the need for laborious and costly manual work among professional 3D artists, thus enabling individuals without expertise to engage in the creation of 3D assets. Given the notable achievements in 2D content generation, exemplified by projects like DALL-E [1] and StableDiffusion [2], the community is increasingly focusing on advancements in 3D content generation. Recent progress in this field is credited to the advantageous characteristics of image diffusion models [2], [3], differentiable 3D representations [4], [5], and large reconstruction models [6], [7].

An appealing area of interest for 3D content creation is **text-to-3D** generation. Some groundbreaking advancements [8], [9] in text-to-3D synthesis have introduced methods to enhance a neural radiance field (NeRF) [4] through score distillation sampling (SDS) loss [8] for 3D asset generation. Building upon the influential work of DreamFusion [8], these SDS-based techniques aim to distill 3D information from pretrained large text-to-image generative models [1], [2]. Various strategies seek to elevate generation quality by expanding to multiple optimization phases [9], optimizing 3D representation and diffusion prior simultaneously [10], [11], and adjusting score distillation algorithms [12], [13].

Another crucial aspect of generating 3D content is the process of **image-to-3D** synthesis. The traditional approach to this challenge relies on 3D reconstruction methods such as Structure-from-Motion [15] and Multi-view Stereo [16], [17], [18], [19]. These techniques involve identifying 3D surface points by comparing similarities among point features extracted from source images, enabling the creation of highly precise surface and texture maps. Despite significant achievements in accurately reconstructing geometrical details, these methods still struggle to reproduce detailed view-dependent appearances. Consequently, recent advancements have focused on developing implicit 3D representations like neural radiance fields [4], [20] and neural implicit surfaces [21], [22]. These novel approaches explore volumetric representations that can be learned from dense multi-view datasets without explicit feature matching, offering more efficient and high-quality solutions [20], [23], [24]. Such efforts aim to move towards feed-forward models for radiance fields reconstruction, relaxing the need for dense views and per-scene optimization. Leveraging the capabilities and generalization power of large generative models like diffusion models, recent studies [25], [26], [27], [28], [29] have integrated pre-trained generative models with multi-view information to generate new views from sparse inputs. Additionally, the emergence of Large Reconstruction Models (LRM) [6], [30], [31] has emphasized learning internal perspective relationships through a triplane transformer [32] and cross-attention mechanisms with 2D visual features from single-view input images. Recent enhancements [7], [33] of LRM have focused on replacing triplane-based volume rendering with 3D Gaussian splatting [20] and extending single-view inputs to sparse multi-view configurations,

- H. Xu, F. Xiao and W. Kang are with South China University of Technology, Guangzhou, China.
- W. Chen, B. Sun, and L. Bo are with Alibaba Group, Hangzhou, China.
- Z. Zhou is with INF Group, Shanghai.
- Mike Z. Shou is with National University of Singapore, Singapre.
- W. Kang is the corresponding author. (E-mail: auwxkang@scut.edu.cn)



(a) Average time consumption of generating one Sample on a single V100-32G GPU

(b) Performance comparison among our ControLRM-T/D and other state-of-the-art methods

Fig. 1. Performance and efficiency comparison among different conditional 3D generation methods. Fig. (a) shows the average time consumption on a single V100-32G GPU of different methods. Our ControLRM-T and ControLRM-D can respectively achieve 60 and 18 times faster inference speed compared with the fastest baseline, MVControl [14]. Fig (b) shows the results of 15 evaluation metrics on the G-Objaverse test set, including 3D controllability metrics (introduced in Sec. 4.2.1) and controllable 3D generation metrics (introduced in Sec. 4.3.1).

facilitating comprehensive 3D object information.

To address the question of whether the current prompt-based or image-based 3D generation methods are adequate to fulfill our requirements, we can delve further into the necessities of 3D generation and categorize the issue into two distinct subproblems: **(1) Is 3D Generation Controllable?** In text-to-3D approaches, the prompt typically offers a basic description, requiring users to repeatedly input prompts to achieve the desired 3D output. Conversely, image-based methods necessitate acquiring the specific target image that meets the requirements before generating the desired 3D content. Therefore, integrating controllability into the 3D generation processes is crucial for ensuring user agency and customization. **(2) Is 3D Generation Efficient?** The optimization processes involved in text-to-3D and image-to-3D techniques are laborious and time-intensive, often demanding up to an hour to create a single 3D object based on input prompts or images. Such extensive computational requirements pose a significant barrier, rendering the production of 3D content unfeasible for many users. Consequently, addressing efficiency within the realm of 3D generation stands as a critical challenge to overcome.

To address the challenges identified, **this paper aims to develop an efficient and controllable 3D generation method**. An existing study named MVControl [14] endeavors to tackle this issue by extending ControlNet [34] to a multi-view diffusion model, MVDream [29]. The MVControl system produces four multi-view images, which are then fed into a multi-view Gaussian reconstruction model, LGM [7], to derive coarse 3D Gaussian representations. Subsequently, these coarse Gaussians undergo SDS optimization guided by a 2D diffusion model to refine the 3D Gaussian outputs. Despite demonstrating promising outcomes in 3D content generation, MVControl exhibits several limitations: **(1) Misalignment between 2D and 3D Representations**: In MVControl, the multi-view images generated by the 2D diffusion model are converted to 3D representations using the LGM reconstruction model. However, the direct integration of these distinct models may lead to discrepancies between 2D and 3D representations, as the reconstruction model might struggle to generalize across the generated images. **(2) Complex Multi-Stage Procedures Increase Time Consumption**: MVControl incorporates a two-stage approach: the initial stage involves the amalgamation of 2D diffusion and 3D reconstruction models, while the subsequent stage encompasses the SDS-based optimization process. These intricate multi-stage procedures contribute to a cumbersome and time-intensive generation process.

These identified challenges prompt the following solutions: (1) Resolving the misalignment between 2D and 3D through **an end-to-end aligned model**; (2) Streamlining complex procedures with **a fast feed-forward model**. This paper introduces **ControLRM**, a feed-forward model designed for controllable 3D generation founded on the Large Reconstruction Model (LRM). The architecture consists of: (1) A 2D condition generator with transformer or diffusion backbone that accept text and 2D visual conditions as input; (2) A 2D condition encoder that extract 2D latent features from the output feature of the 2D condition generator; (3) A triplane decoder transformer that interacts with the 2D features via cross-attention and generate a triplane-NeRF representation. Training directly with conditional inputs and ground truth multi-view images from scratch is computationally demanding and challenging. Therefore, we propose a joint training framework leveraging the strong priors of a pre-trained LRM model trained on extensive datasets. In the condition training stage, the condition 2D generator and the cross-attention layer are activated, while the parameters in the triplane decoder remain fixed. In the image training phase, both the image encoder and the triplane decoder are activated to ensure the alignment between 2D latents and 3D triplane transformer. Rather than utilizing the entire Objaverse [35] and MVImgNet [36] datasets like LRM [6], we opt for a smaller dataset, G-Objaverse [37], to train our ControLRM. To ensure unbiased evaluation, we curate evaluation samples from three distinct datasets (G-OBJ, GSO, ABO) rather relying on manual generation. The quantitative and qualitative results on 3D controllability evaluation and generation quality comparison demonstrate the superiority of our method.

In summary, our main contributions are as follows:

- We present ControLRM, a novel framework tailored for controllable 3D generation based on single-view 2D condition and text input. The model undergoes evaluation across four distinct condition types (edge, depth, normal, scribble), showcasing its robust generalization and diverse controllability features.

- We introduce an end-to-end feed-forward network architecture for controllable 3D generation. The end-to-end paradigm serves as a natural bridge between 2D latents and 3D triplanes, while the feed-forward network design guarantees rapid inference when compared to existing optimization-based approaches.
- We present an effective joint training scheme for training the controllable 3D generation model. This approach leverages the significant 3D reconstruction capabilities within pretrained LRM to enhance our controllable 3D generation task.
- Through comprehensive experiments conducted on G-OBJ, GSO, and ABO datasets, we demonstrate that our ControLRM significantly surpasses the performance of current state-of-the-art (SOTA) methods in 3D controllability, generation quality, and inference speed (as shown in Fig. 1).

2 Related Work

2.1 Optimization-based 3D Generation

Building on the accomplishments of text-to-image diffusion models [2], [3], optimization-based approaches present a practical alternative by circumventing the necessity for extensive text-3D datasets. DreamFusion [8] is a seminal work that introduced the SDS loss to optimize a neural field using diffusion priors for 3D asset generation. Additionally, Score Jacobian Chaining [38] is a study that elevates pretrained 2D diffusion models for 3D creation, utilizing the chain rule and the gradients learned from a diffusion model to backpropagate scores through the Jacobian of a differentiable renderer. However, these optimization-based techniques commonly encounter a shared challenge known as the Janus problem. MVDream [29] tackles this issue by refining a multi-view diffusion model, which replaces self-attention with multi-view attention in Unet to produce consistent multi-view images. Introducing the concept of 3D Gaussian splatting [20], DreamGaussian [39] optimizes 3D Gaussians using the SDS loss. Nonetheless, it grapples with the Janus problem stemming from the uncertainties of 2D SDS supervision and rapid convergence. Addressing this, GSGEN [40] and GaussianDreamer [41] incorporate a coarse 3D prior to generate more cohesive geometries. Furthermore, GSGEN proposes the use of the 3D SDS loss from Point-E [42] for joint optimization in the geometry phase. Despite SDS's benefits in terms of data requirements, it necessitates optimization for each new 3D object and demands hours to reach convergence.

2.2 Feed-forward 3D Generation

The extensive 3D datasets [35], [36] have unlocked new possibilities for training feed-forward models to generate 3D assets directly from text, single- or multi-view images. (1) **3D generation from single-view**: LRM [6] first scales up the triplane transformer on a large dataset to predict a triplane neural radiance field (NeRF) from single-view images, showing high generalization ability. TripoSR [30] integrates significant improvements in data processing, model design, and training techniques, enhancing the efficiency and effectiveness. (2) **3D generation from multi-view**: Methods based on multi-view are extensions designed to enhance the generation quality of single-view methods. Typically, multi-view images of an object are initially synthesized from a single image using a multi-view diffusion model [29]. Similar to single-view approaches, these methods can be broadly categorized as either diffusion-based or transformer-based architectures. Examples of diffusion-based architectures include SyncDreamer [27] and Wonder3D [28]. SyncDreamer necessitates dense views for 3D reconstruction, while Wonder3D employs a multiview cross-domain attention mechanism to process relatively sparse views. Transformer-based architectures like Instant3D [43] encodes multi-view images by a image encoder and concatenate the encoded results into a set of tokens for the image-to-triplane decoder. Additionally, LGM [7], GRM [44] and GS-LRM [33] enhance the generation quality using high-resolution features and increasing the number of surrounding views. (3) **3D generation from text**: Point-E [42] and Shap-E [45] utilize complex prompts to generate point clouds and neural radiance fields respectively. Representing 3D data as volumes, 3DTopia [46] and VolumeDiffusion [47] train diffusion models by fitting volumetric modules. ATT3D [48] employs a feed-forward transformer to generate the 3D contents and train the model with amortized training via pretrained diffusion model. Latte3D [49] extends the amortization architecture of ATT3D, significantly improving the efficiency and generation quality.

2.3 Controllable 3D Generation

Despite the rapid advancements in 3D generation techniques discussed earlier, achieving controllability in 3D generation remains a significant challenge. The current state-of-the-art controllable 3D generation method is MVControl [14]. This method incorporates a trainable control network that interacts with the base multi-view diffusion model to facilitate controllable multi-view image generation. In the coarse stage, the MVControl model produces four-view images, which are subsequently input into the 3D reconstruction model LGM [7]. The generated coarse Gaussians are then utilized to initialize the SDS-based training in the refinement stage. However, there are still some limitations in MVControl: (1) The direct integration of distinct models may lead to discrepancies between 2D and 3D representations, as the reconstruction model may not generalize well on the generated multi-view images. (2) The complex procedures for generating a single 3D content may increase the time consumption. In response to these limitations, we propose ControLRM, an end-to-end feed-forward controllable 3D generation model which also has fast inference speed.

3 Method

In this section, we present the ControLRM framework as depicted in Fig. 2. We commence by outlining the fundamen-

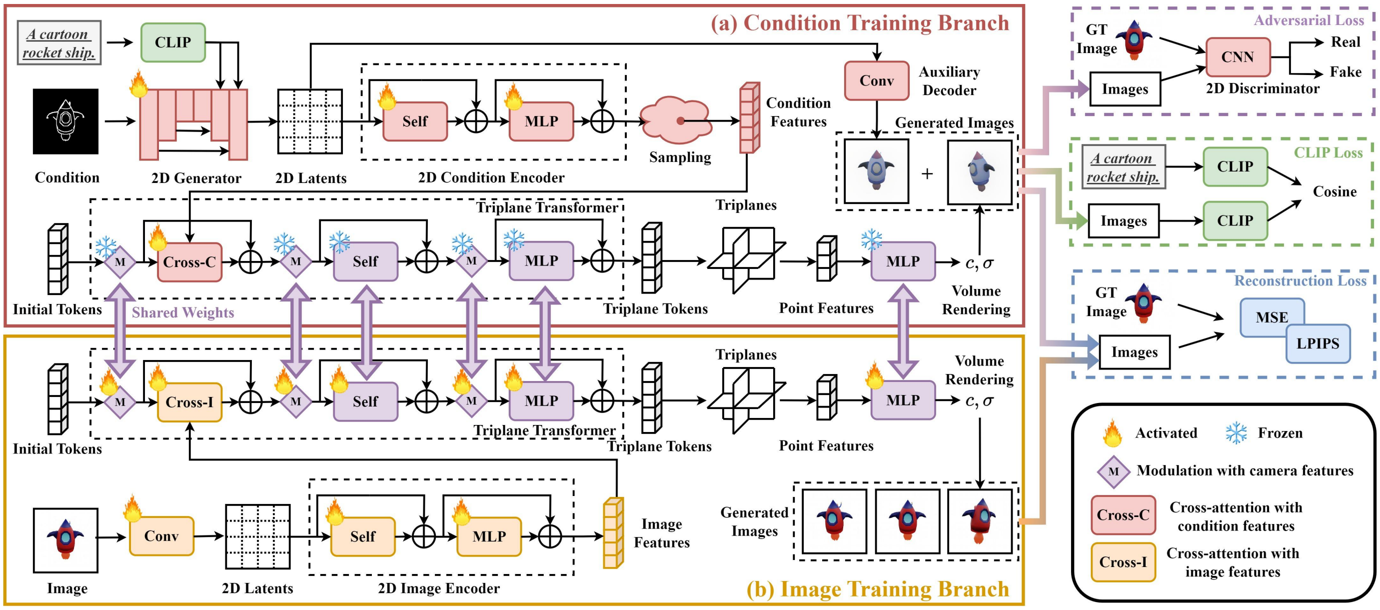


Fig. 2. The overall framework of **ControLRM**, a feed-forward controllable 3D generation model.

tals of LRM in Sec. 3.1. Next, we delve into a comprehensive examination of the LRM framework from the perspective of the Variational Auto-encoder (VAE) in Sec. 3.2. Building on the insights from Sec. 3.2, we elucidate the process of enhancing the LRM to our proposed ControLRM in Sec. 3.3. Subsequently, we elaborate on the components of each module within ControLRM and expound on the training objectives in Sec. 3.4.

3.1 Preliminary of LRM

Large Reconstruction Model (LRM) is an advanced method that efficiently generates a 3D object from a single 2D image input. The LRM primarily consists of the following components:

Image Encoder: Given an RGB image as input, we utilize a pre-trained visual transformer (ViT) [50] to encode the image into patch-wise feature tokens denoted by $\{h_i|h_i \in \mathbb{R}^{D_e}\}_i^{N_p}$, where i represents the index of the image patch, N_p is the total number of image patches, and D_e signifies the dimension of the feature tokens. Specifically, the pre-trained self-supervised model DINO (Caron et al., 2021) is used. The ViT incorporates a predefined `[CLS]` token $h_{\text{cls}} \in \mathbb{R}^{D_e}$, which is then concatenated with the feature sequence $\{h_i\}_{i=1}^{N_p}$ to form the output.

Camera Features: The camera feature $c \in \mathbb{R}^{20}$ is comprised of the flattened vectors of camera extrinsic and intrinsic parameters. The 4-by-4 extrinsic matrix E is flattened to a 16-dimensional vector $E_{1\times16}$. The intrinsic parameters, including the camera focal length and principal points, are combined as a 4-dimensional vector: $[\text{foc}_x, \text{foc}_y, \text{pp}_x, \text{pp}_y]$. To embed the camera feature, a multi-layer perceptron (MLP) is employed to transform the camera feature c into a 1024-dimensional camera embedding $\tilde{c}$.

$$\tilde{c} = \text{MLP}_{\text{cam}}(c) = \text{MLP}_{\text{cam}}([E_{1\times16}, \text{foc}_x, \text{foc}_y, \text{pp}_x, \text{pp}_y]) \quad (1)$$

Modulation with Camera Features: The camera modulation incorporates an adaptive layer normalization (adaLN) [51] to adjust image features using denoising iterations and class designations. When provided with the camera feature $\tilde{c}$ as input, a multi-layer perceptron (MLP) predicts the scaling factor γ and the shifting factor β:

$$\gamma, \beta = \text{MLP}_{\text{mod}}(\tilde{c}) \quad (2)$$

Subsequently, the modulation function will process the sequence of vectors in the transformer $\{f_j\}$ as follows:

$$\text{ModLN}(f_j) = \text{LN}(f_j) \cdot (1 + \gamma) + \beta \quad (3)$$

where LN is the layer Normalization [52].

Transformer Layers: Each transformer layer consists of a cross-attention sub-layer, a self-attention sub-layer, and a multi-layer perceptron sub-layer (MLP), where the input tokens for each sub-layer are modulated by the camera features. The feature sequence f^{in}, serving as the input to the transformer layers, can also be viewed as triplane hidden features. As illustrated in Fig. 2 (b), the cross-attention module uses the feature sequence f_{in} as the query and the image features $\{h_{\text{cls}}, h_i\}_{i=1}^{N_p}$ as the key/value pairs.

$$f_j^{\text{cross-i}} = \text{Cross-I}(\text{ModLN}(f_j^{\text{in}}); \{h_{\text{cls}}, h_i\}_{i=1}^{N_p}) + f_j^{\text{in}} \quad (4)$$

where Cross-I represents the cross-attention between the image features and the triplane features.

Subsequent to the original transformer [53], the self-attention sub-layer denoted as Self(·) and the multi-layer perceptron sub-layer labeled as MLP(·) handle the input feature sequence in the ensuing manner:

$$f_j^{\text{self}} = \text{Self}(\text{ModLN}(f_j^{\text{cross-i}}); \text{ModLN}(f_{j'}^{\text{cross-i}})) + f_j^{\text{cross-i}} \quad (5)$$

$$f_j^{\text{out}} = \text{MLP}(\text{ModLN}(f_j^{\text{self}})) + f_j^{\text{self}} \quad (6)$$

where f_j^{out} represents the triplane feature output. This final output undergoes upsampling via a trainable de-convolution layer and is subsequently reshaped into the final triplane representation $TP \in \mathbb{R}^{3\times64\times64\times D_t}$, where D_t signifies the dimension of the triplane.

Triplane NeRF: The triplane TP comprises three axis-aligned feature planes: $\text{TP}_{xy}/\text{TP}_{yz}/\text{TP}_{xz} \in \mathbb{R}^{64\times64\times D_t}$. Given any 3D point $p = [p_x, p_y, p_z]^T$ within the NeRF object bounding box $[-1, 1]^3$, the point's feature can be extracted from the triplane TP using bilinear sampling.

$$\text{TP}_p = \text{Concat}(\text{TP}_{xy}[p_x, p_y], \text{TP}_{yz}[p_y, p_z], \text{TP}_{xz}[p_x, p_z]) \quad (7)$$

where $\mathrm{Concat}(\cdot)$ represents the concatenation function, and $\mathrm{TP}_p \in \mathbb{R}^{3 \cdot D_t}$ denotes the sampled feature corresponding to point p.

Training Objectives: During training, V views are randomly selected from the dataset. One view is chosen as the reference view and passed to the LRM, while the other $V-1$ views serve as auxiliary training views. Let the rendered views of the LRM be denoted as $\hat{x}$, and the ground truth views as x^{GT}. Particularly, for each input image x, we aim to minimize:

$$L_{\mathrm{recon}}(x) = \frac{1}{V}\sum_{v=0}^{V}(L_{\mathrm{MSE}}(\hat{x}_v, x_v^{\mathrm{GT}}) + \lambda L_{\mathrm{LPIPS}}(\hat{x}_v, x_v^{\mathrm{GT}})) \tag{8}$$

where L_{MSE} represents the normalized pixelwise L2 loss, L_{LPIPS} denotes the perceptual image similarity loss [54], and λ is a customizable weight used to balance these losses.

3.2 Understanding LRM in a Perspective of VAE

From the perspective of Variational Autoencoder (VAE) [55], the LRM can be viewed as an intricate architecture that encompasses certain fundamental principles akin to VAEs.

Similar to the encoder in a VAE, the image encoder of LRM processes an input image, transforming it into a series of feature tokens. These tokens serve as the encoded latent representation of the input image, mirroring the latent space in a VAE. The decoding component of LRM functions analogously to the decoder in a VAE by reconstructing images from the latent space. Specifically, LRM maps the latent trilinear representation to a 3D object within NeRF and subsequently generates images with new perspectives, akin to the generation or decoding process within a VAE framework. LRM employs a reconstruction loss to reduce the dissimilarity between the input image and the rendered images altered based on camera parameters. In the subsequent section, we will offer a theoretical overview of LRM, including a form of Evidence Lower Bound (ELBO).

Given the 3D representation $\mathbf{x}_{3d}$, a set of projected 2D images $\{x_i\}_{i=1}^{N_V}$ with corresponding camera parameters $\{T_i\}_{i=1}^{N_V}$, where N_V denotes the number of viewpoints. It is assumed that the ground-truth distribution of the 3D representation is represented by the density $p(\mathbf{x}_{3d})$. In LRM, this 3D representation is characterized by a triplane Neural Radiance Field (NeRF). Under this assumption, one can write:

$$p(\mathbf{x}_{3d}) = \int_z p(\mathbf{x}_{3d}, z)dz = \int_z p(\mathbf{x}_{3d}|z)p(z)dz \tag{9}$$

z represents the latent variable associated with $\mathbf{x}_{3d}$, following a simple distribution $p(z)$ referred to as the prior distribution. The primary objective of the VAE is to acquire a robust approximation of $p(\mathbf{x}_{3d}|z)$ based on the provided data. This approximated distribution is denoted by $p_\theta(\mathbf{x}_{3d}|z)$, where θ symbolizes the learnable parameters. Subsequently, we can compute the log likelihood $\log p_\theta(\mathbf{x}_{3d})$ in the following manner:

$$\begin{aligned}
&\log p_\theta(\mathbf{x}_{3d}) \\
&= \log \int_T p_\theta(\mathbf{x}_{3d}|T)p(T)dT \geq \int_T \log p_\theta(\mathbf{x}_{3d}|T)p(T)dT \\
&\approx \frac{1}{N_V}\sum_{i=1}^{N_V} \log p_\theta(\mathbf{x}_{3d}|T_i) = \frac{1}{N_V}\sum_{i=1}^{N_V} \log \int_z p_\theta(\mathbf{x}_{3d}, z|T_i)dz \\
&= \frac{1}{N_V}\sum_{i=1}^{N_V} \log \int_z \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)q_\varphi(z|x_i, T_i)}{q_\varphi(z|x_i, T_i)}dz \\
&\geq \frac{1}{N_V}\sum_{i=1}^{N_V} \mathbb{E}_{q_\varphi} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_\varphi(z|x_i, T_i)}
\end{aligned} \tag{10}$$

where $p_\theta(\mathbf{x}_{3d}|T_i)$ indicates that the 3D representation $\mathbf{x}_{3d}$ is conditioned on the camera parameters T_i corresponding to viewpoint i. Given that our $\mathbf{x}_{3d}$ embodies a triplane NeRF, when conditioned on T_i, it serves as a representation of the rendered image from viewpoint i. The final row in Eq. 10 denotes the Evidence Lower Bound (ELBO). By isolating the inner term of ELBO at viewpoint i, we obtain:

$$\begin{aligned}
&\mathbb{E}_{q_\varphi} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_\varphi(z|x_i, T_i)} \\
=&\mathbb{E}_{q_\varphi} \log \frac{p_\theta(\mathbf{x}_{3d}|z, T_i)p_\theta(z)}{q_\varphi(z|x_i, T_i)} \\
=&\mathbb{E}_{q_\varphi} \log p_\theta(\mathbf{x}_{3d}|z, T_i) - \mathrm{KL}(q_\varphi(z|x_i, T_i)||p_\theta(z)) \\
=&\mathbb{E}_{q_\varphi} \log \int_T p_\theta(\mathbf{x}_{3d}|z, T_i, T)p(T)dT - \mathrm{KL}(q_\varphi(z|x_i, T_i)||p_\theta(z))] \\
\geq&\mathbb{E}_{q_\varphi} \int_T \log p_\theta(\mathbf{x}_{3d}|z, T_i, T)p(T)dT - \mathrm{KL}(q_\varphi(z|x_i, T_i)||p_\theta(z)) \\
\approx&\frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{q_\varphi} \log p_\theta(\mathbf{x}_{3d}|z, T_i, T_j) - \mathrm{KL}(q_\varphi(z|x_i, T_i)||p_\theta(z))
\end{aligned} \tag{11}$$

Note that the extrinsic matrix of the input reference view is normalized to an identity matrix, while the extrinsic matrices of the other views are adjusted to the relative transformation matrix with respect to the normalized reference view. The intrinsic parameters remain constant across all views. Consequently, the input camera parameter T_i is consistent and fixed within the LRM, thereby allowing for its exclusion from the formulas:

$$\begin{aligned}
&\mathbb{E}_{q_\varphi} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_\varphi(z|x_i, T_i)} \\
\geq&\frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{q_\varphi} \log p_\theta(\mathbf{x}_{3d}|z, T_j) - \mathrm{KL}(q_\varphi(z|x_i)||p_\theta(z))
\end{aligned} \tag{12}$$

where $p_\theta(x_{3d}|z, T_j)$ represents the triplane decoder (depicted as purple modules in Fig. 2), while $q_\phi(z|x_i)$ denotes the image encoder (illustrated as orange modules in Fig. 2).

3.3 Upgrading LRM to ControLRM

Eq. 10, 11, and 12 elaborate on the extension of LRM, interpreting it as a specialized variant of the Variational Autoencoder (VAE). By analogy, these expressions can be further expanded to cater to the objective of controllable 3D generation. Consider e_i as indicative of the input 2D visual condition on view i and the associated textual prompt concerning the 3D object, the ELBO can be formulated as:

$$\begin{aligned}
&\log p_\theta(\mathbf{x}_{3d}) \\
=&\log \int_T p_\theta(\mathbf{x}_{3d}|T)p(T)dT \geq \int_T \log p_\theta(\mathbf{x}_{3d}|T)p(T)dT \\
\approx&\frac{1}{N_V}\sum_{i=1}^{N_V} \log p_\theta(\mathbf{x}_{3d}|T_i) = \frac{1}{N_V}\sum_{i=1}^{N_V} \log \int_z p_\theta(\mathbf{x}_{3d}, z|T_i)dz \\
=&\frac{1}{N_V}\sum_{i=1}^{N_V} \log \int_z \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)q_{\varphi'}(z|e_i, T_i)}{q_{\varphi'}(z|e_i, T_i)}dz \\
=&\frac{1}{N_V}\sum_{i=1}^{N_V} \log \mathbb{E}_{q_{\varphi'}}[\frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_{\varphi'}(z|e_i, T_i)}] \geq \frac{1}{N_V}\sum_{i=1}^{N_V} \mathbb{E}_{q_{\varphi'}} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_{\varphi'}(z|e_i, T_i)}
\end{aligned} \tag{13}$$

By isolating the inner term of ELBO at viewpoint i, we can get:

$$\begin{aligned}
&\mathbb{E}_{q_{\varphi'}(z|e_i,T_i)} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_{\varphi'}(z|e_i,T_i)} \\
=&\mathbb{E}_{q_{\varphi'}(z|c_i,T_i)} \log \frac{p_\theta(\mathbf{x}_{3d}|z,T_i)p_\theta(z)}{q_{\varphi'}(z|e_i,T_i)} \\
\geq&\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}_{q_{\varphi'}(z|e_i,T_i)} \log p_\theta(\mathbf{x}_{3d}|z,T_i,T_j) - \mathrm{KL}(q_{\varphi'}(z|e_i,T_i)||p_\theta(z))
\end{aligned} \tag{14}$$

Due to the normalization operation towards reference viewe in the extrinsic matrix, T_i is a fixed identity matrix, which can be further simplified in Eq. 14.

$$\begin{aligned}
&\mathbb{E}_{q_{\varphi'}(z|e_i,T_i)} \log \frac{p_\theta(\mathbf{x}_{3d}, z|T_i)}{q_{\varphi'}(z|e_i,T_i)} \\
\geq&\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}_{q_{\varphi'}(z|e_i)} \log p_\theta(\mathbf{x}_{3d}|z,T_j) - \mathrm{KL}(q_{\varphi'}(z|e_i)||p_\theta(z))
\end{aligned} \tag{15}$$

where $p_\theta(x_{3d}|z,T_j)$ represents the same triplane decoder as Eq. 12 (depicted as purple modules in Fig. 2), while $q_{\varphi'}(z|e_i)$ denotes the condition encoder part (illustrated as red modules in Fig. 2).

Eq. 15 represents the ELBO of our ControLRM. However, the optimization of Eq. 15 might be much more difficult than the optimization of Eq. 12 in LRM, given the relaxation of input from detailed images to coarse conditions (visual condition maps and text descriptions). Typically, achieving convergence of ControLRM necessitates an even larger scale of data compared to what was utilized in training LRM (Objaverse [35] and MVImgNet [36]). Consequently, direct optimization of Eq. (15) is not the optimal solution, considering the computational cost and convergence issues encountered during training.

To address this issue, we have to explore an alternative training approach for our ControLRM model. Remarkably, it is observed that the triplane decoder denoted by $p_\theta(x_{3d}|z,T_j)$ is common to both Eq. 12 and Eq. 15. This implies that leveraging the convergence of the triplane decoder $p_\theta(x_{3d}|z,T_j)$ and the image encoder $q_\phi(z|x_i)$ under the guidance of Eq. (12) can enhance the training process in Eq. 15. If $p_\theta(x_{3d}|z,T_j)$ is kept constant in Eq. 15, the focus shifts to maximizing the remaining term $-\mathrm{KL}(q_{\varphi'}(z|e_i)||p_\theta(z))$, aligning the condition encoder $q_{\varphi'}(z|e_i)$ with the latent space z. Consequently, the need for a vast amount of paired data (input condition and 3D object) can be significantly reduced, and the convergence can also be enhanced by leveraging the strong prior knowledge embedded in pre-trained LRM models.

Following these discussions, we propose a joint training paradigm which comprises two branches: the Image Training Branch and the Condition Training Branch. The former encompasses a 2D image encoder ($q_\phi(z|x_i)$ in Eq. 12) and a 3D triplane decoder ($p_\theta(x_{3d}|z,T_j)$ in Eq. 12). The latter comprises a 2D condition encoder ($q_{\varphi'}(z|e_i)$ in Eq. 15) and utilizes the same 3D triplane decoder ($p_\theta(x_{3d}|z,T_j)$ in Eq. 15. It is noteworthy that the cross-attention layers interacting with $q_\phi(z|x_i)$ and $q_{\varphi'}(z|e_i)$ are denoted as Cross-I and Cross-C, respectively. Illustrated in Figure 2, the Image Training Branch optimizes the ELBO in Eq. 12, aiming to refine the triplane decoder $p_\theta(x_{3d}|z,T_j)$ and 2D image encoder $q_\phi(z|x_i)$ for optimal performance. On the other hand, the Condition Training Branch retains the fixed parameters of the triplane decoder $p_\theta(x_{3d}|z,T_j)$ and focuses on optimizing the ELBO in Eq. 15. This process naturally aligns the distributions of the latent spaces in Eq. 12 and Eq. 15 using the shared 3D Triplane Transformer.

3.4 ControLRM

In this section, we delve into the specific modules of ControLRM. The design of the conditional generator was detailed in Fig. 2 in Section 3.4.1. Depending on the chosen backbone for the conditional generator, ControLRM manifests in two variants: 1) ControLRM-T featuring a transformer-based conditional generator (Section 3.4.2); 2) ControLRM-D integrating a diffusion-based conditional generator (Section 3.4.3). Subsequently, we present the condition-to-triplane transformer decoder in Section 3.4.6. The training objectives encompassing adversarial loss, clip loss, and rendering loss are expounded upon in Section 3.4.7.

3.4.1 Design of Conditional Generator

As depicted in Fig. 2, the conditional generator utilizes the 2D condition and the text embedding of CLIP [56] as input to produce the 2D latents required for subsequent procedures. A naive design of this generator is a transformer-based backbone with cross-attention mechanism between the feature sequence extracted from condition image and the text feature. However, this design with only the cross-attention mechanism fails to generate a regular results but yielding meaningless results in the experiments. A similar issue was observed in [57], indicating that the main reason for this optimization failure stems from the notable disparity between the 2D renderings and the ground truth images. As noted by [58], the optimization gradient becomes unreliable when the generated distribution and the target distribution are disjoint. In contrast, the backward gradients to the 2D latents in our model must traverse a series of modules, including the condition encoder, triplane transformer, and NeRF modules. This complexity of pathways may significantly impede the optimization process, consequently resulting in unexpected failures. A straightforward remedy proposed in [57] involves the incorporation of randomness (e.g., Gaussian noise) into the network architecture. By increasing the overlap between the rendered distribution and the target distribution, the gradients during training become more meaningful, promoting convergence. In summary, the key considerations for designing the condition generator in ControLRM are: 1) Incorporation of randomness for improved training outcomes. 2) Emphasis on the efficiency of the generator for fast inference speed.

3.4.2 Transformer-based Conditional Generator

For **ControLRM-T** model, we have devised a lightweight transformer-based generator, illustrated in Figure 3 (a). Building upon the preceding discussion, we introduce randomness through a style injection module. Drawing inspiration from the original style injection concept in StyleGAN [59], where style features and random noise are integrated into the generator via Adaptive Instance Normalization

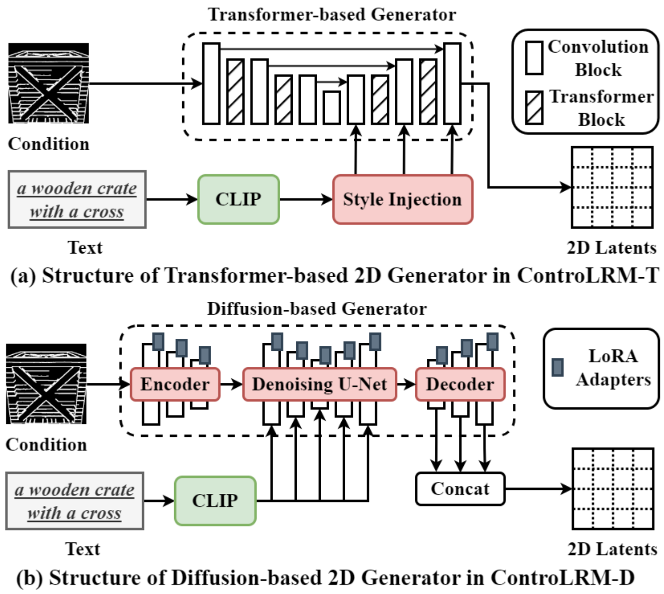


Fig. 3. The architecture of the 2D conditional generator in ControLRM. (a) shows the transformer-based generator in **ControLRM-T**, and (b) shows the diffusion-based generator in **ControLRM-D**.

(AdaIN), we adapt this approach by treating the text embedding as the style feature. This text embedding is concatenated with random Gaussian noise and passed through a 3-layer MLP within our style injection module. The resulting feature vector is then combined with the output of each convolution layer to incorporate the text feature. In Figure 3 (a), the convolution blocks and transformer blocks are stacked together, with residual connections applied to the convolution blocks in a U-Net configuration.

3.4.3 Diffusion-based Conditional Generator

For the **ControLRM-D** model, we have intricately integrated LoRA adapters [60] into the original latent diffusion model, incorporating small trainable weights. Leveraging the inherent randomness within the diffusion model, and aided by the pre-trained weights obtained from large-scale datasets, we aim to address the discrepancy issue highlighted in Section 3.4.4. In addressing efficiency concerns, we opt for the fast one-step diffusion model [61] as the foundational framework. Specifically, we initialize the Diffusion-based generator with the pre-trained weights of SD-Turbo [62]. To form the 2D latents for subsequent procedures, we concatenate the outputs of the last three layers of the decoder depicted in Figure 3 (b).

3.4.4 Condition Encoder

In Figure 2 (a), the 2D latents are firstly interpolated to match the resolution of the input condition image, and then divided into the feature sequence $\{g_i | g_i \in \mathbb{R}^{D_e}\}_i^{N_p}$. Similar to the feature sequence $\{h_i\}_i^{N_p}$ extracted from the input image discussed in Sec. **??**, D_e denotes the feature dimension, while N_p corresponds to the number of patches. Within the condition encoder, the feature sequence $\{g_i\}_i^{N_p}$ is passed through a sequence of transformer layers, each comprising a self-attention sub-layer and an MLP sub-layer.

$$g_i^{\text{self}} = \text{Self}(g_i; g_i) + g_i \tag{16}$$

$$g_i^{\text{out}} = \text{MLP}(g_i^{\text{self}}) + g_i^{\text{self}} \tag{17}$$

where g_i^{out} is the output feature.

To integrate the random sampling process, the output g_i^{out} of the final transformer layer is fed to another MLP to regress the mean and variance results:

$$\mu_{g_i}, \sigma_{g_i} = \text{MLP}(g_i^{\text{out}}) \tag{18}$$

where μ_g is the mean feature and σ_g represents the variance. Throughout training, the output feature sequence $\{\tilde{g}_i\}_i^{N_p}$ is is stochastically sampled from a Gaussian distribution, where $\tilde{g}_i \sim \mathcal{N}(\mu_{g_i}, \sigma_{g_i}^2)$.

3.4.5 Auxiliary Decoder

To boost the performance, we further introduce an auxiliary decoder for the 2D latents to enhance the training process. The generated 2D latents from the conditional generator (refer to Sections 3.4.2 and 3.4.3) are passed through a lightweight three-layer convolutional neural network. The resulting image x_{aux} is combined with the 2D renderings to compute the loss function for the generated images. The inclusion of the auxiliary decoder offers direct guidance to the 2D generator, aiding in overall network convergence.

3.4.6 Triplane Transformer Decoder

The condition-to-triplane decoder receives the condition feature sequence $\{\tilde{g}_i\}_i^{N_p}$ and the triplane feature sequence f^{in}. Analogous to the image-to-triplane decoder discussed in Sec. 3.1, each transformer layer consists of a cross-attention sub-layer, a self-attention sub-layer, and an MLP layer. The input tokens for each sub-layer are influenced by the camera features $\tilde{c}$. The operation of each transformer layer can be described as follows:

$$f_j^{\text{cross-c}} = \text{Cross-C}(\text{ModLN}(f_j^{\text{in}}); \{\tilde{g}_i\}_i^{N_p}) + f_j^{\text{in}} \tag{19}$$

$$f_j^{\text{self}} = \text{Self}(\text{ModLN}(f_j^{\text{cross-c}}); \text{ModLN}(f_j^{\text{cross-c}})) + f_j^{\text{cross-c}} \tag{20}$$

$$f_j^{\text{out}} = \text{MLP}(\text{ModLN}(f_j^{\text{self}})) + f_j^{\text{self}} \tag{21}$$

3.4.7 Training Objectives

In Fig. 2, the training objectives consist of three components: adversarial loss, CLIP loss, and rendering loss. For each sample, we designate one reference view and randomly select $V-1$ side views. Denoting the rendered images of ControLRM as $\hat{x}$ and the ground truth images as x^{GT}, the index of the reference view is designated as 0. The resultant image from the auxiliary decoder (refer to Sec. 3.4.5) is denoted as x_{aux}. The calculation of the loss can be expressed as follows:

Adversarial Loss: To incentivize the alignment of the generated images with the corresponding ground truth domains, we apply an adversarial loss [63]. In line with the approach advocated by Vision-Aided GAN [64], the discriminator utilizes the CLIP model as its foundation. The adversarial loss is defined as follows:

$$\begin{aligned} L_{\text{adv}} = & \frac{1}{V+1}\{\sum_{v=0}^{V} \mathbb{E}[\log \mathcal{D}(x_v^{\text{GT}})] + \sum_{v=0}^{V} \mathbb{E}[\log(1 - \mathcal{D}(\hat{x}_v))] + \\ & \mathbb{E}[\log \mathcal{D}(x_0^{\text{GT}})] + \mathbb{E}[\log(1 - \mathcal{D}(x_{\text{aux}}))\} \end{aligned} \tag{22}$$

CLIP Loss: To improve the consistency between the generated images and the text prompt y_{text}, a CLIP loss [56] is employed for text-image alignment.

$$\begin{aligned}L_{\text{clip}} =& \frac{1}{V+1}[\sum_{v=0}^{V}(1-\cos(\text{CLIP-I}(\hat{x}_v), \text{CLIP-T}(y_{\text{text}})))+ \\ & (1-\cos(\text{CLIP-I}(x_{\text{aux}}), \text{CLIP-T}(y_{\text{text}})))]\end{aligned} \tag{23}$$

where CLIP-I is the CLIP image encoder, and CLIP-T is the CLIP text encoder.

Reconstruction Loss: The generated images are compared to the ground truth images to ensure consistency through a reconstruction loss. For each input condition image and text prompt, we aim to minimize:

$$\begin{aligned}L_{\text{recon}} =& \frac{1}{V+1}[\sum_{v=0}^{V}(L_{\text{MSE}}(\hat{x}_v, x_v^{\text{GT}}) + \lambda\sum_{v=0}^{V} L_{\text{LPIPS}}(\hat{x}_v, x_v^{\text{GT}}))+ \\ & L_{\text{MSE}}(x_{\text{aux}}, x_0^{\text{GT}}) + \lambda L_{\text{LPIPS}}(x_{\text{aux}}, x_0^{\text{GT}})]\end{aligned} \tag{24}$$

where L_{MSE} is the normalized pixel-wise L2 loss, L_{LPIPS} is the perceptual image patch similarity [54]. λ is a customized weight to balance the losses. In default, $\lambda = 1.0$.

Overall Loss: The overall loss is a weighted sum of the aforementioned losses:

$$L_{\text{overall}} = L_{\text{recon}} + \lambda_{\text{adv}} L_{\text{adv}} + \lambda_{\text{clip}} L_{\text{clip}} \tag{25}$$

where $\lambda_{\text{adv}} = 0.5$, $\lambda_{\text{clip}} = 5.0$ in default.

Efficient Training: By default, we configure the rendered image resolution to 256×256. However, performing direct computations on the entire 256×256 renderings using L_{overall} is likely to lead to GPU memory overflow during training, mainly due to NeRF's significant memory requirements. To address this issue, we opt for a straightforward yet efficient approach that trades space for time. Firstly, we partition the original 256×256 images into smaller local patches with a resolution of 128×128. These local patches are randomly chosen based on weighted sampling of foreground pixels using the ground truth image mask. Secondly, we downsample the original 256×256 images to smaller global images with a resolution of 128×128. Similar to the approach in LRM [6], we utilize the deferred back-propagation technique [65] to conserve GPU memory. In essence, the adjusted loss function is as delineated below:

$$L_{\text{overall}} = L_{\text{recon}}^{\text{local}} + L_{\text{recon}}^{\text{global}} + \lambda_{\text{adv}} L_{\text{adv}}^{\text{global}} + \lambda_{\text{clip}} L_{\text{clip}}^{\text{global}} \tag{26}$$

where $^{\text{global}}$ means the loss is computed on the global images. $^{\text{local}}$ means the loss is computed on the sampled local patches.

4 EXPERIMENT

4.1 Experiment Details

4.1.1 Training Details

Our training dataset comprises the training split of the G-Objaverse dataset [37], which is a subset of Objaverse [35]. We have randomly selected 260k samples from the original G-Objaverse for training, while the remaining samples are allocated for validation and evaluation purposes. The text prompts for each sample are sourced from Cap3D [66]. Additionally, the visual condition maps are derived from the multi-view images in the dataset, encompassing edge, sketch, depth, and normal annotations. Edge annotations are generated using the Canny edge detector [67], sketch annotations are produced with the sketch generation model from ControlNet [34]. Depth and normal annotations are provided by G-Objaverse, and further normalized to match the format of MVControl (Li et al., 2024).

We initialize our network using the weights from the pre-trained OpenLRM-base [68]. The image-conditioned transformer from OpenLRM is removed, and our proposed conditional backbone, incorporating text and visual conditions (such as sketch, edge, depth, and normal), is appended as input. During training, the cross-attention layers in the triplane transformer of OpenLRM are activated, while the remaining layers are kept frozen. We utilize the AdamW optimizer with a conservative learning rate of 4e-4 for training ControLRM on 16 Nvidia V100-32G GPUs. Each batch comprises 96 text-condition-image pairs. The training duration is estimated to be approximately 4-6 days for ControLRM-T and 5-6 days for ControLRM-D. The input resolution of the condition image is set to 336, while the rendered image resolution is set to 256

4.1.2 Evaluation Dataset

For evaluation, we collect test samples from real world datasets rather than manually generated samples [14] to ensure unbiased generation. Following the selection principle of MVControl [14] and TripoSR [30], test data is gathered from three distinct datasets for comparative analysis in the subsequent experiments.

(1) **G-OBJ**: We collect 118 samples with highest clip scores between the text annotation and multi-view images from the test split of G-Objaverse dataset [37], ensuring they are absent from the training data. The text annotation is obtained from Cap3D [66]. We manually select one reference view from all provided 40 views in the dataset, and extract the edge/sketch/depth/normal condition maps on that reference view. The remaining views are used as ground truth multi-view images for benchmark evaluation.

(2) **GSO**: We also collect 80 samples from the Google Scanned Objects dataset [69] for zero-shot evaluation. This dataset features more than one thousand 3D-scanned household items, serving as a valuable resource for assessing the zero-shot generalization capabilities of the proposed method. In analogy with **G-OBJ**, we manually select a single reference view from the 32 available views in the dataset. Subsequently, edge/sketch/depth/normal condition maps are generated for this chosen reference view. Text annotations are obtained using BLIP2 [70]. The input data contains the prepared 2D condition map and the corresponding text prompt. The remaining views are utilized as the ground truth for evaluation benchmark.

(3) **ABO**: We also select 80 samples from the Amazon Berkeley Objects dataset [71] for zero-shot evaluation. The Amazon Berkeley Objects dataset is a comprehensive 3D dataset comprising product catalog images, metadata, and artist-designed 3D models featuring intricate geometries and materials based on real household objects. Text annotations are generated using BLIP2 caption model [70]. We manually select one reference view from the 72 available views provided in the dataset and extract the four condition maps (edge/sketch/depth/normal). The remaining views are emplyed as ground truth for benchmark evaluation.

TABLE 1
Quantitative results of controllability under **Edge (Canny)** condition in comparison with other SOTA 3D generation methods on **G-OBJ**. ↑ denotes higher result is better, while ↓ means lower is better. We report the metrics of **C-PSNR**, **C-SSIM**, and **C-MSE** in the table. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Methods	Edge (Canny)		
	C-PSNR ↑	C-SSIM ↑	C-MSE ↓
GSGEN [40]	11.54	0.768	0.0807
GaussianDreamer [41]	11.08	0.755	0.0866
DreamGaussians [39]	8.98	0.667	0.1341
VolumeDiffusion [47]	11.75	0.803	0.0773
3DTopia [46]	8.78	0.692	0.1430
MVControl [14]	10.14	0.738	0.1052
ControLRM-T (Ours)	16.14	0.891	0.0349
ControLRM-D (Ours)	16.17	0.886	0.0314

TABLE 2
Quantitative results of Controllability under **Sketch** condition in comparison with other SOTA 3D generation methods on **G-OBJ**. ↑ denotes higher result is better, while ↓ means lower is better. We report the metrics of **S-PSNR**, **S-SSIM**, and **S-MSE** in the table. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Methods	Sketch		
	S-PSNR ↑	S-SSIM ↑	S-MSE ↓
GSGEN [40]	13.19	0.7629	0.0499
GaussianDreamer [41]	13.23	0.7934	0.0503
DreamGaussians [39]	13.14	0.7740	0.0516
VolumeDiffusion [47]	15.16	0.8247	0.0326
3DTopia [46]	13.73	0.7902	0.0443
MVControl [14]	12.56	0.7406	0.0603
ControLRM-T (Ours)	18.02	0.9084	0.0189
ControLRM-D (Ours)	17.56	0.9000	0.0208

4.1.3 Baselines

We compare our proposed ControLRM with other state-of-the-art baselines in the 3D generation task, including: (1) **Score-Distillation-Sampling (SDS) methods**: GSGEN, GaussianDreamer, and DreamGaussians [8]; (2) **3D-based Diffusion models**: VolumeDiffusion and 3DTopia [46], [47]; (3) **Controllable 3D Diffusion models**: MVControl [14]. It is important to note that MVControl is the most relevant state-of-the-art controllable 3D generation method. For comparison purposes, we utilize the official implementations of the aforementioned methods in the subsequent experiments.

4.2 Experiment Results of 3D Controllability

4.2.1 Evluation Metrics

To assess the controllability of various 3D generation methods, we have developed metrics tailored to gauge the consistency of input 2D conditions following ControlNet++ [72]. Four distinct conditions are taken into account: edge (canny), sketch, depth, and normal. Specific metrics have been intricately designed for each condition to quantify the extent to which the condition is maintained throughout the generation process:

(1) **Edge Condition**: Given the 2D edge map on the reference view, we use the generated 3D content to render a new image at the same view. Subsequently, a Canny detector [67] is employed to extract the edge image from the rendered image, allowing for a comparison between the edge image and the original condition image. The associated hyperparameters for Canny detector is the same as ControlNet [34]. To evaluate the resemblance of the edge maps, performance metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Mean Squared Error (MSE) are computed following [6]. These metrics are further noted as **C-PSNR**, **C-SSIM**, and **C-MSE**.

(2) **Sketch Condition**: Given the 2D sketch image on the reference view, we use the generated 3D content to render a new image at the same view. Subsequently, the sketch extraction network provided by ControlNet [34] is employed to derive the sketch map from the rendered image. We emply PSNR, SSIM, MSE assess the similarity between the generated sketch map and the original sketch map. These metrics are referred to as **S-PSNR**, **S-SSIM**, and **S-MSE** in this study.

(3) **Depth Condition**: Given the 2D depth image on the reference view, we use the generated 3D content to render the image and the depth at the same viewpoint.

On the one hand, we can evaluate the depth consistency with foundation models in monocular depth estimation (i.e. Midas [73], ZoeDepth [74]) following ControlNet++ [72]. These foundation models are utilized to produce a depth map based on the input rendered image. Analogously, they are capable of estimating the depth map given a ground truth image as input on the reference view. By leveraging the depth prior obtained from these foundation models, the Mean Squared Error (MSE) distance between the estimated depth maps of the ground truth image and the rendered image can indicate controllability under various depth conditions. When using Midas as the foundation model, the metric is denoted as **M-MSE**; whereas, if ZoeDepth is employed, the metric is referred to as **Z-MSE**.

On the other hand, an alternative method to assess depth

TABLE 3
Quantitative results of Controllability under **Depth** condition in comparison with other SOTA 3D generation methods on **G-OBJ**. ↓ denotes lower result is better. We report the metrics of **M-MSE**, **Z-MSE**, and **R-MSE** in the table. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Methods	Depth		
	M-MSE ↓	Z-MSE ↓	R-MSE ↓
GSGEN [40]	0.1504	0.1381	0.0425
GaussianDreamer [41]	0.1019	0.1271	0.0558
DreamGaussians [39]	0.1035	0.1284	0.0435
VolumeDiffusion [47]	0.1615	0.1156	0.0444
3DTopia [46]	0.1364	0.1374	0.0412
MVControl [14]	0.0692	0.0695	0.0655
ControLRM-T (Ours)	0.0287	0.0198	0.0355
ControLRM-D (Ours)	0.0285	0.0174	0.0331

TABLE 4
Quantitative results of Controllability under **Normal** condition in comparison with other SOTA 3D generation methods on **G-OBJ**. ↓ denotes lower result is better. We report the metrics of **NB-MSE**, and **DN-Consistency** in the table. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Methods	Normal	
	NB-MSE ↓	DN-Consistency ↓
GSGEN [40]	0.0140	0.0412
GaussianDreamer [41]	0.0133	0.0404
DreamGaussians [39]	0.0141	0.0372
VolumeDiffusion [47]	0.0129	0.0468
3DTopia [46]	0.0240	0.0431
MVControl [14]	0.0103	0.0421
ControLRM-T (Ours)	0.0038	0.0216
ControLRM-D (Ours)	0.0034	0.0205

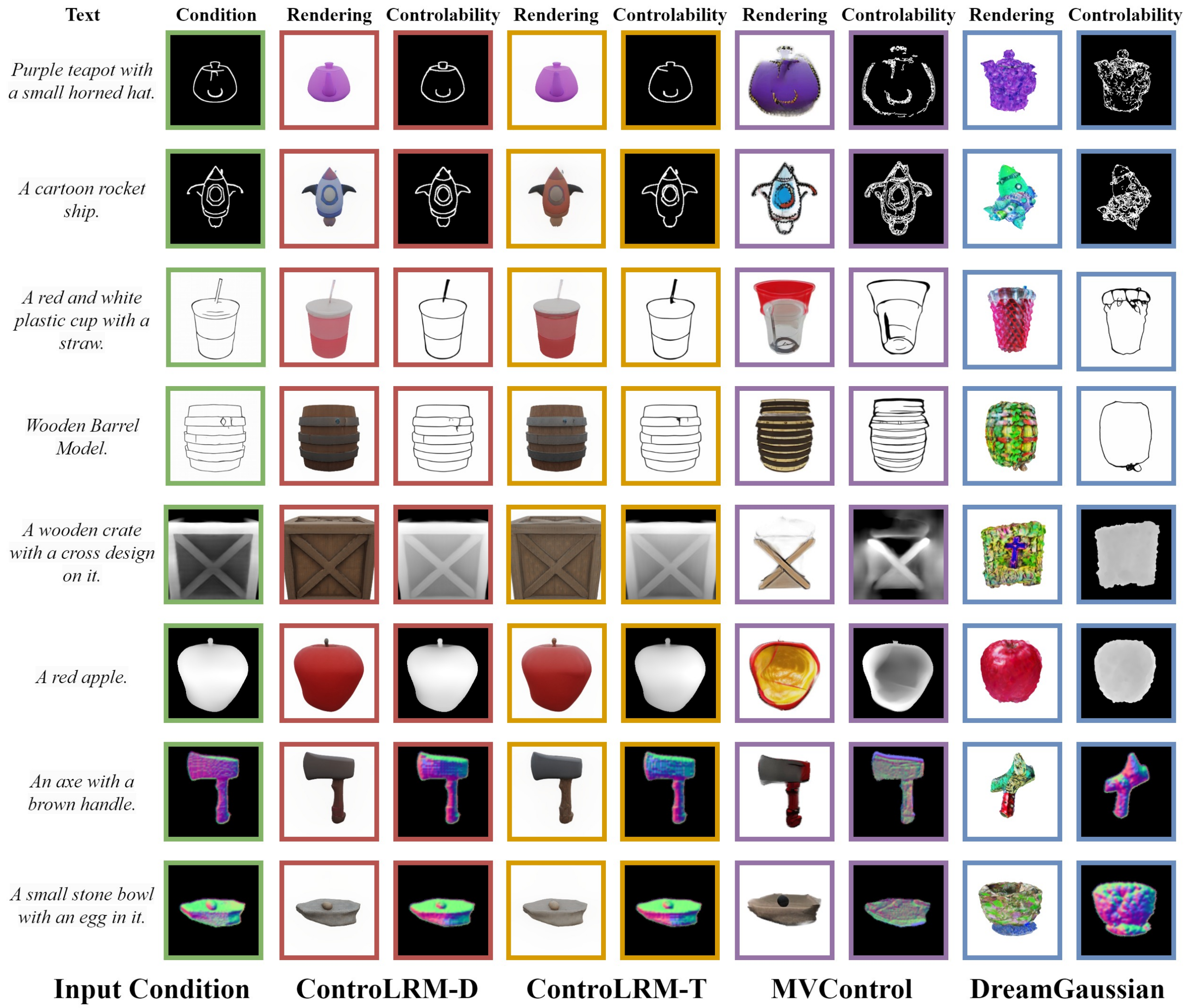


Fig. 4. Visualization comparison of controllability under different conditional controls (Edge/Depth/Normal/Sketch).

consistency in 3D space involves comparing the disparity between the rendered depth map and the input conditional depth map. The disparity, measured by MSE distance, between the rendered depth map and the input conditional depth map can also reflect the model's controllability performance. However, discrepancies in scale between the estimated relative depth map and the input conditional depth map may adversely affect the accuracy of the MSE metric. Thus, it becomes essential to address the scale discrepancy before evaluating the similarity between these depth maps. Following the approach outlined in [73], we compute an ordinary least squares solution to adjust for the scale and shift between these depth maps. Subsequently, the scale and shift transformation is applied to the relative depth map, and the MSE is then calculated between it and the input conditional depth map. This enables the calculation of a scale-agnostic MSE metric to evaluate the similarity between the depth maps, providing an effective way to evaluate the 3D consistency of the rendered depth map, denoted as **R-MSE**.

(4) **Normal Condition**: Given the 2D normal map on the reference view, we use the generated 3D results to render the image and depth at the same viewpoint.

Firstly, we can assess the normal consistency with pre-trained models in surface normal estimation, such as Normal-BAE [75] following ControlNet++ [72]. The model for surface normal estimation facilitates the extraction of normal maps from rendered images. Similarly, the ground truth image can be input into the model to derive estimated normal maps. As the pre-trained model can grasp the surface normal priors from the input images, the Mean Squared Error (MSE) distance between these normal maps can indicate the controllability performance of the generation model. This evaluation metric, based on Normal-BAE, is referred to as **NB-MSE**.

Secondly, the evaluation of normal consistency in 3D space involves comparing the resemblance between the rendered depth maps and the input conditional normal maps. The rendered depth map on the reference view is normalized to 0 to 1 first, and then used to calculate the normal map. The MSE distance between this converted normal map and the input conditional normal map can demonstrate the normal consistency throughout the generation process. This metric, influenced by the depth-normal consistency in 3D space, is labeled as **DN-consistency**.

4.2.2 Quantitative Results

Results with Canny Condition: The comparison of the controllability of 3D generation methods under the Edge

TABLE 5
Quantitative comparison with SOTA 3d generation methods on G-Objaverse (**G-OBJ**) test set. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**. We also provide the time consumption of each method on a single V100-32G GPU to compare the efficiency.

Methods \ Metrics	Time ↓	Reference View			All Views			Front-K Views		
		FID ↓	CLIP-I ↑	CLIP-T ↑	FID ↓	CLIP-I↑	CLIP-T ↑	FID ↓	CLIP-I ↑	CLIP-T ↑
GSGEN [40]	≈ 40 min	235.09	0.756	0.308	340.39	0.762	0.298	357.13	0.781	0.310
GaussianDreamer [41]	≈ 2 min	182.70	0.802	0.295	268.95	0.803	0.309	282.59	0.823	0.321
DreamGaussians [39]	≈ 15 min	247.48	0.763	0.281	351.87	0.761	0.279	368.76	0.783	0.293
VolumeDiffusion [47]	142.55 sec	218.09	0.728	0.237	327.76	0.725	0.241	348.39	0.752	0.257
3DTopia [46]	177.89 sec	228.79	0.719	0.267	289.02	0.749	0.280	329.29	0.808	0.312
MVControl [14]	8.92 sec	175.43	0.829	0.296	251.71	0.811	0.291	280.40	0.856	0.318
ControLRM-T (Ours)	0.148 sec	100.58	0.915	0.309	166.03	0.879	0.292	144.02	0.932	0.323
ControLRM-D (Ours)	0.503 sec	104.08	0.911	0.315	163.25	0.887	0.300	148.76	0.935	0.330

TABLE 6
Quantitative comparison with SOTA controllable 3D text-to-3d method (MVControl [14]) on G-Objaverse (**G-OBJ**) [37] test set. 4 kinds of different visual condition types are utilized for comparison here, including **Edge**, **Depth**, **Normal**, and **Sketch**. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Metrics	Methods	Reference View				All Views				Front-K Views			
		Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch
FID ↓	**MVControl** [14]	226.01	158.38	144.59	172.73	300.10	229.45	215.27	262.03	328.99	257.62	244.51	290.49
	ControLRM-T	99.51	102.88	97.49	102.43	165.21	165.49	163.11	170.33	141.54	147.18	140.73	146.63
	ControLRM-D	98.45	109.20	103.09	105.57	158.73	166.36	161.62	166.28	139.02	156.91	148.17	150.95
CLIP-I ↑	**MVControl** [14]	0.771	0.854	0.866	0.825	0.768	0.831	0.840	0.806	0.816	0.875	0.883	0.851
	ControLRM-T	0.915	0.914	0.919	0.912	0.879	0.881	0.881	0.876	0.933	0.932	0.932	0.930
	ControLRM-D	0.920	0.902	0.912	0.911	0.889	0.885	0.888	0.886	0.939	0.931	0.935	0.935
CLIP-T ↑	**MVControl** [14]	0.262	0.311	0.312	0.300	0.264	0.302	0.304	0.291	0.295	0.326	0.330	0.318
	ControLRM-T	0.309	0.308	0.310	0.309	0.291	0.292	0.293	0.290	0.322	0.323	0.324	0.322
	ControLRM-D	0.318	0.311	0.316	0.315	0.301	0.299	0.300	0.299	0.332	0.327	0.330	0.329

TABLE 7
Quantitative comparison with SOTA 3d generation methods on Google Scanned Objects (**GSO**) test set. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**. We also provide the time consumption of each method on a single V100-32G GPU to compare the efficiency.

Methods \ Metrics	Time ↓	Reference View			All Views			Front-K Views		
		FID ↓	CLIP-I ↑	CLIP-T ↑	FID ↓	CLIP-I ↑	CLIP-T ↑	FID ↓	CLIP-I ↑	CLIP-T ↑
GSGEN [40]	≈ 40 min	273.54	0.734	0.286	344.61	0.740	0.289	360.57	0.759	0.300
GaussianDreamer [41]	≈ 2 min	189.41	0.815	0.305	278.70	0.810	0.300	287.20	0.829	0.311
DreamGaussians [39]	≈ 15 min	271.80	0.761	0.281	359.65	0.760	0.279	373.53	0.784	0.290
VolumeDiffusion [47]	142.55 sec	236.01	0.719	0.261	299.61	0.715	0.259	316.35	0.742	0.273
3DTopia [46]	177.89 sec	274.99	0.698	0.274	331.39	0.727	0.283	369.27	0.799	0.311
MVControl [14]	8.92 sec	194.97	0.848	0.298	278.08	0.816	0.288	301.31	0.870	0.312
ControLRM-T (Ours)	0.148 sec	165.44	0.899	0.309	260.75	0.846	0.289	251.57	0.912	0.316
ControLRM-D (Ours)	0.503 sec	162.28	0.896	0.313	171.13	0.838	0.302	247.06	0.908	0.322

(Canny) condition is presented in Table 1. The aforementioned metrics of **C-PSNR**, **C-SSIM**, and **C-MSE** in Sec. 4.2.1 are utilized for evaluating controllability. Our approach establishes a new state-of-the-art benchmark, surpassing other methods by a significant margin. Specifically, our ControLRM-D and ControLRM-T achieve C-PSNR scores of 16.17 and 16.14 respectively, exhibiting an improvement of approximately 6 compared to the baseline performance. Similar improvements can also be witnesses in C-SSIM and C-MSE.

Results with Sketch Condition Tab. 2 presents the state-of-the-art comparison for the controllability of 3D generation results on Sketch condition. The evaluation includes the results of three metrics introduced in Sec. 4.2.1: **S-PSNR**, **S-SSIM**, and **S-MSE**. These metrics can reflect how much sketch control information is preserved in the generated 3D results. The results reveals that our models, ControLRM-D and ControLRM-T, outperform other methods significantly across all three metrics. In comparison to the baseline method, MVControl, our approach showcases a significant enhancement, boasting around 6 points in S-PSNR, 0.25 in S-SSIM, and 0.04 in S-MSE.

Results with Depth Condition Tab. 3 shows the state-of-the-art comparison for the controllability of 3D generation methods on Depth condition. We report the scores of **M-MSE**, **Z-MSE**, and **R-MSE** introduced in Sec. 4.2.1. From the results in the table, our proposed methods, ControLRM-D and ControLRM-T, outperform other baselines across all three metrics of depth controllability. Specifically, our proposed method demonstrates an improvement of approximately 0.04 in the M-MSE, 0.05 in Z-MSE, 0.03 in R-MSE, compared to MVControl.

Results with Normal Condition Tab. 4 shows the state-of-the-art comparison for the controllability of 3D generation methods on Normal condition. The evaluation metrics include **NB-MSE** and **DN-Consistency** introduced in Sec. 4.2.1. From the comparison results, our proposed ControLRM-D/ControLRM-T models outperforms other baselines in both NB-MSE and DN-Consistency metrics. Specifically, ControLRM-D and ControLRM-T achieve NB-MSE scores of 0.0034 and 0.0038, respectively, representing a notable improvement compared to MVControl (0.0103

TABLE 8
Quantitative comparison with SOTA controllable 3D text-to-3d method (MVControl [14]) on Google Scanned Objects (**GSO**) [69] test set. 4 kinds of different visual condition types are utilized for comparison here, including **Edge**, **Depth**, **Normal**, and **Sketch**. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Metrics	Methods	Reference View				All Views				Front-K Views			
		Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch
FID ↓	**MVControl** [14]	219.89	187.77	179.74	192.48	311.91	256.08	255.68	288.64	342.59	272.83	270.02	319.81
	ControLRM-T	156.60	167.11	173.60	164.44	253.59	262.95	269.62	256.84	245.12	253.76	257.62	249.79
	ControLRM-D	151.48	165.90	171.33	160.42	165.38	174.29	174.98	169.88	234.42	253.02	256.24	244.55
CLIP-I ↑	**MVControl** [14]	0.782	0.877	0.890	0.843	0.762	0.841	0.851	0.811	0.815	0.895	0.907	0.864
	ControLRM-T	0.915	0.896	0.879	0.904	0.855	0.842	0.835	0.852	0.923	0.910	0.894	0.919
	ControLRM-D	0.916	0.892	0.870	0.906	0.854	0.826	0.820	0.850	0.928	0.901	0.885	0.919
CLIP-T ↑	**MVControl** [14]	0.265	0.312	0.312	0.301	0.263	0.298	0.299	0.291	0.290	0.321	0.324	0.314
	ControLRM-T	0.311	0.306	0.301	0.318	0.293	0.288	0.284	0.291	0.320	0.317	0.311	0.317
	ControLRM-D	0.316	0.312	0.308	0.314	0.304	0.301	0.300	0.303	0.326	0.317	0.316	0.323

TABLE 9
Quantitative comparison with SOTA 3d generation methods on Amazon Berkeley Objects (**ABO**) test set. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**. We also provide the time consumption of each method on a single V100-32G GPU to compare the efficiency.

Methods \ Metrics	Time ↓	Reference View			All Views			Front-K Views		
		FID ↓	CLIP-I ↑	CLIP-T ↑	FID ↓	CLIP-I ↑	CLIP-T ↑	FID ↓	CLIP-I ↑	CLIP-T ↑
GSGEN [40]	≈ 40 min	304.49	0.664	0.257	366.47	0.669	0.259	376.56	0.691	0.272
GaussianDreamer [41]	≈ 2 min	148.70	0.820	0.297	225.38	0.787	0.277	226.56	0.831	0.306
DreamGaussians [39]	≈ 15 min	340.64	0.729	0.248	392.95	0.723	0.247	406.35	0.750	0.290
VolumeDiffusion [47]	142.55 sec	288.28	0.698	0.247	350.46	0.679	0.242	372.49	0.715	0.262
3DTopia [46]	177.89 sec	247.89	0.692	0.273	231.55	0.751	0.272	259.88	0.844	0.312
MVControl [14]	8.92 sec	149.61	0.857	0.305	217.97	0.802	0.291	236.81	0.868	0.316
ControLRM-T	0.148 sec	85.08	0.913	0.311	202.14	0.827	0.282	160.12	0.915	0.320
ControLRM-D	0.503 sec	80.12	0.914	0.320	181.84	0.836	0.292	152.37	0.918	0.324

TABLE 10
Quantitative comparison with SOTA controllable 3D text-to-3d method (MVControl [14]) on Amazon Berekely Objects (**ABO**) [71] test set. 4 kinds of different visual condition types are utilized for comparison here, including **Edge**, **Depth**, **Normal**, and **Sketch**. We provide the zero-shot evaluation results of **FID** ↓, **CLIP-I** ↑ and **CLIP-T** ↑ on the test samples. The best results are highlighted with **underline**, and the second best ones are highlighted with **wavy-line**.

Metrics	Methods	Reference View				All Views				Front-K Views			
		Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch	Edge	Depth	Normal	Sketch
FID ↓	**MVControl**	204.89	127.76	101.67	164.11	254.40	197.09	183.39	236.99	268.67	221.58	201.11	255.85
	ControLRM-T	88.22	93.60	84.00	74.49	207.31	220.16	196.77	184.30	162.46	171.38	157.38	149.24
	ControLRM-D	74.89	86.32	86.82	72.45	173.57	189.60	192.93	171.27	149.25	156.44	159.65	144.15
CLIP-I ↑	**MVControl**	0.818	0.884	0.895	0.833	0.784	0.812	0.815	0.798	0.839	0.886	0.897	0.848
	ControLRM-T	0.909	0.909	0.907	0.925	0.828	0.800	0.830	0.848	0.913	0.911	0.908	0.926
	ControLRM-D	0.919	0.909	0.898	0.931	0.850	0.814	0.821	0.859	0.922	0.913	0.903	0.934
CLIP-T ↑	**MVControl**	0.292	0.312	0.316	0.299	0.282	0.295	0.300	0.287	0.310	0.319	0.323	0.311
	ControLRM-T	0.302	0.313	0.313	0.317	0.276	0.279	0.284	0.290	0.323	0.317	0.317	0.321
	ControLRM-D	0.319	0.320	0.316	0.326	0.294	0.289	0.285	0.300	0.323	0.324	0.319	0.330

NB-MSE). Significant improvment of our models in DN-Consistency score can also be found in the table.

4.2.3 Qualitative Results

Controllable 3D generation requires the persistence of input conditions as a crucial ability. The generated 3D contents should retain the control information of the input conditions. For qualitative comparison of 3D controllability, we visualize the generated results and the extracted condition maps in Fig. 4. The first two columns display the visualization of text and 2D visual conditions. Subsequent columns exhibit the visualization results of the rendered images and the extracted visual condition map from them. The comparison encompasses several methods: our ControLRM-D (columns 3-4), ControLRM-T (columns 5-6), MVControl (columns 7-8) [14], and DreamGaussian (columns 9-10) [39]. Each row of the figure corresponds to a specific control condition: Rows 1-2 (Edge), Rows 3-4 (Sketch), Rows 5-6 (Depth), and Rows 7-8 (Normal). As shown in the figure, ControLRM-D and ControLRM-T can effectively preserve the control information in the generated 3D content. For instance, in the first and second rows, the controllability results of MVControl and DreamGaussian under the Canny condition appear noticeably fuzzier compared to those of ControLRM-D/T. It demonstrates our proposed method can effectively maintain the controllability during 3D generation, providing better scalability compared with existing methods.

4.3 Experimental Results of Controllable 3D Generation

4.3.1 Evaluation Metrics

For evaluation, we quantitatively compare our proposed method with baselines by measuring the quality of generated 3D contents with **FID**, the consistency to the reference ground truth image with **CLIP-I**, and the consistency to the reference text description with **CLIP-T**.

Render FID: Following LATTE3D [49], we compute the Fréchet Inception Distance (FID) [76] between the renderings of the generated 3D contents and the collected ground

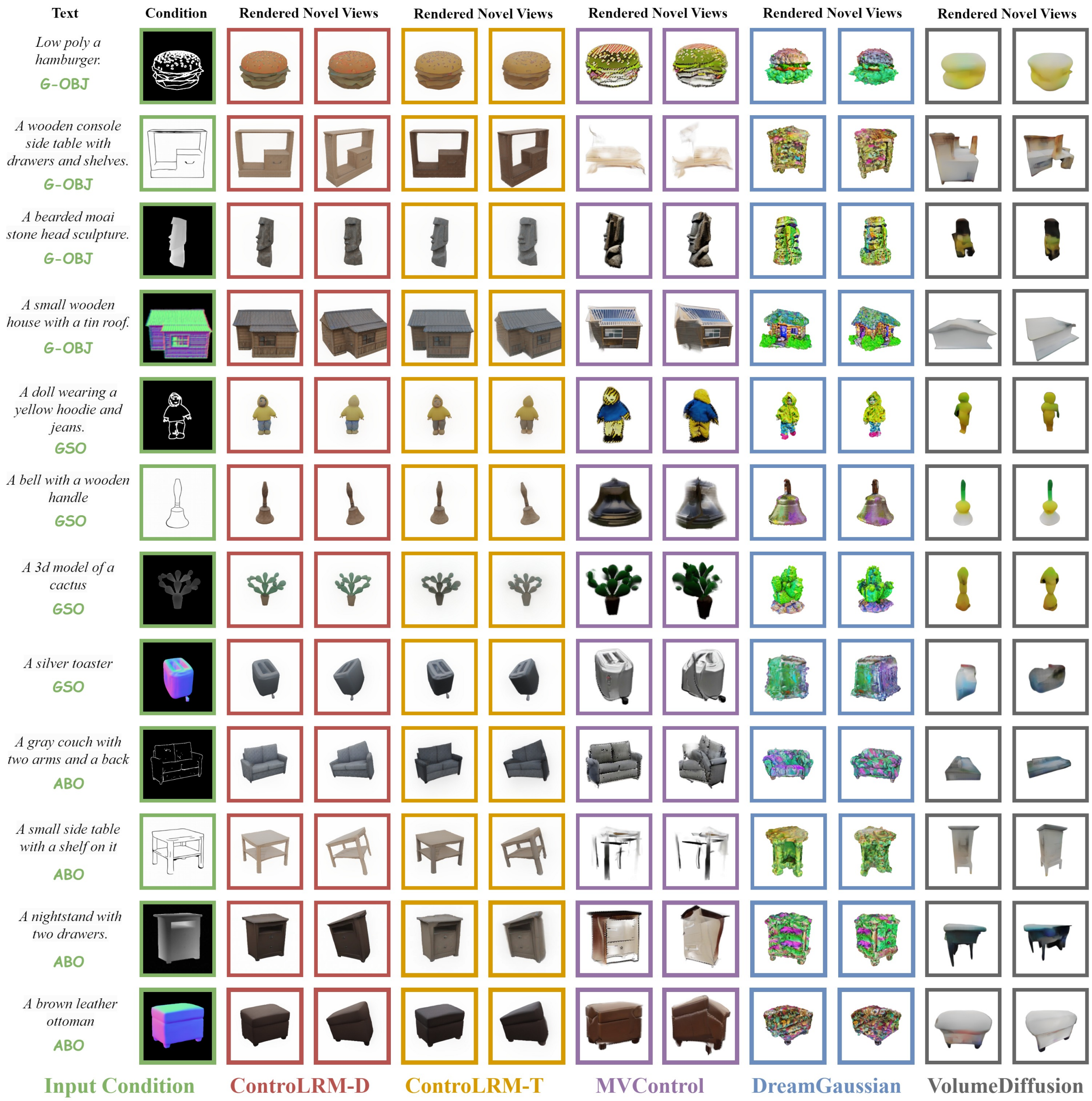


Fig. 5. Qualitative comparison with SOTA 3D generation methods, including MVControl [14], DreamGaussian [39], and VolumeDiffusion [47]. To avoid cherry-picking, the input conditions are extracted from **G-OBJ**, **GSO**, and **ABO** datasets. None of the images are observed by our model during training. Please zoom in for clearer visualization.

truth multi-view images. This metric can measure how well the generated shapes align with those from the 2D prior in visual quality.

CLIP-I: Following MVControl [14], we measure the CLIP scores of image features extracted from the renderings of the generated 3D contents and the collected ground truth images on different views. This metric aims to reveal the similarity between the rendering results of generated 3D contents and the ground truth images.

CLIP-T: Following MVControl [14], we also measure the CLIP scores of the image features extracted from the renderings and the given text prompt. This metric can measure the similarity between the generated 3D contents and the given text descriptions.

Multi-view Settings: The evaluation protocol of MVControl [14] only calculate the CLIP score between the generated multi-view images and real ground truth images on the reference view. However, merely evaluating the performance with ground truth on only one reference view is not comprehensive for comparing 3D generated contents. Because a single view can only capture a portion of the 3D object, often omitting unseen parts. Consequently, utilizing multi-view ground truth is essential to enhance the evaluation protocol. As discussed in Sec. 4.1.2, we collect samples with multi-view ground truth from **G-OBJ**, **GSO**, and **ABO**. By incorporating these multi-view samples, we enhance the original benchmark used in MVControl [14] to be more comprehensive in the following manner: (1) **Reference View**: The rendered image and ground truth image on the reference view are utilized to compute metrics including FID, CLIP-I, and CLIP-T; (2) **All Views**: All views are taken into account when calculating the three metrics

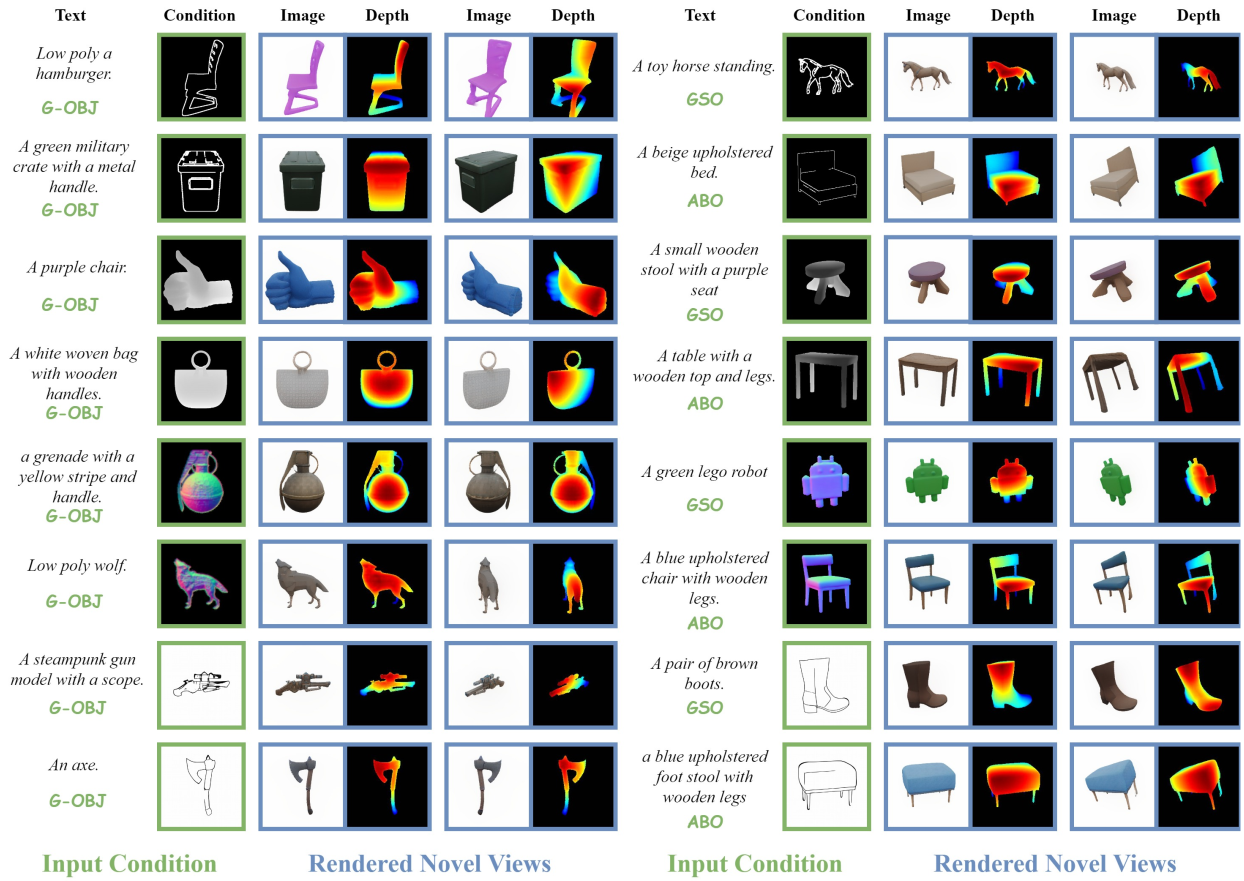


Fig. 6. Visualization of rendered novel views (RGB and depth) generated by our ControLRM-D. The samples are extracted from **G-OBJ**, **GSO**, and **ABO** datasets. None of the images are observed by our model during training. Please zoom in for clearer visualization.

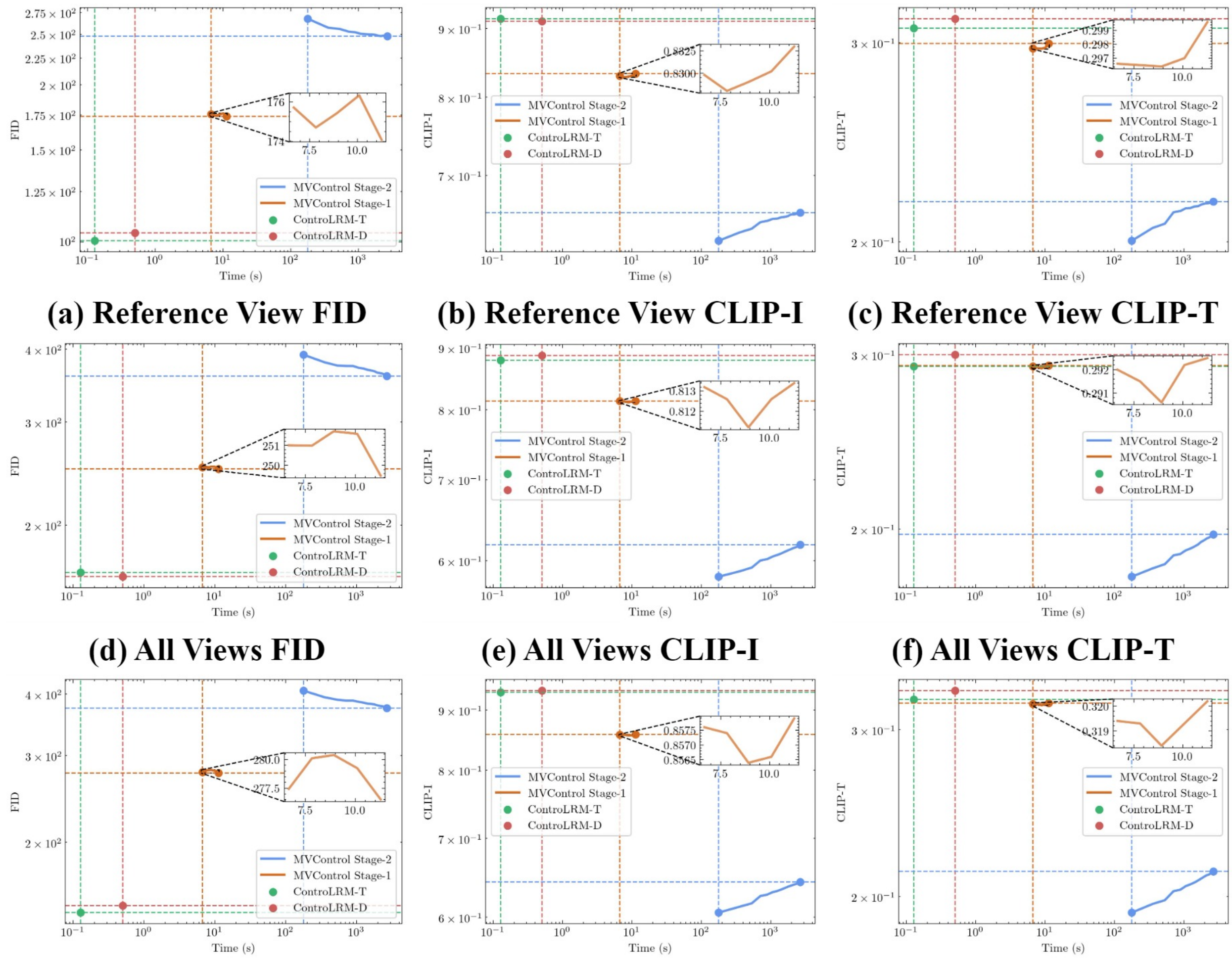
between the rendered and ground truth images; (3) **Front-K Views**: Given the provision of only one reference view, the views on the back side may lack crucial cues for precise prediction, potentially leading to unreliable results in multi-view scenarios. Therefore, incorporating an additional evaluation of the views in front of the reference view is necessary. Consequently, we select the K views closest to the given reference view for further metric computation, with the default value of K set to 4.

4.3.2 Quantitative Comparison on G-OBJ

To demonstrate the effectiveness of the proposed method in controllable 3D generation, we present the quantitative results on the **G-OBJ** benchmark in Tab. 5 and 6. Tab. 5 shows the comparison of **FID**, **CLIP-I**, **CLIP-T** with other baselines. We report the mean score of these metrics under four different conditions (edge/depth/normal/sketch). The time efficiency of each method on a single V100-32G GPU is reported as well. In the tables, we adopt three different multi-view settings during evaluation as discussed in Sec. 4.3.1. As shown in Tab. 5, ControLRM-T achieves an inference speed of 0.148 seconds per sample, while ControLRM-D achieves 0.503 seconds per sample. Our ControLRM models significantly enhance the inference speed by an order of magnitude when compared to alternative methods. In addition to the siginificant improvement in time efficiency, the benchmark results on nine metrics also show that our ControLRM can achieve significantly better performance than other baselines. For example, ControLRM-D/ControLRM-T achieves 104.08/101.06 Reference FID score, 0.911/0916 Reference CLIP-I score, and 0.315/0.309 Reference CLIP-T score. The baselines achieve over 175 FID score, which is significantly higher than ControLRM. It demonstrates the superior ability and efficiency of the proposed method in controllable 3D generation. Tab. 6 shows the direct comparison with SOTA method (MVControl [14]) on four different visual conditions . Similar to Tab. 5, the metrics of **FID**, **CLIP-I** and **CLIP-T** under three different multi-view settings are used to reveal the quality of the generated 3D contents. On most of the evaluation metrics, our ControLRM can achieve competetive and even better performance than MVControl, and the inference speed is significantly faster Specifically, the inference speeds of ControLRM-D (0.503 sec/sample) and ControLRM-T (0.148 sec/sample) are much faster than MVControl (8.92 sec/sample). It demonstrates the superior ability and efficiency of the proposed method in controllable 3D generation.

4.3.3 Quantitative Comparison on GSO

To demonstrate the generalization ability of the proposed method on the task of controllable 3D generation, we provide the experimental results on **GSO** benchmark and compare our model with other state-of-the-art methods introduced in Sec. 4.1.3. Similar to Sec. 4.3.2, we also use the evaluation metrics of **FID**, **CLIP-I**, and **CLIP-T** to measure the performance on controllable 3D generation. These metrics are also calculated under 3 different multi-view settings as introduced in Sec. 4.3.1. In Tab. 7, we



(a) Reference View FID (b) Reference View CLIP-I (c) Reference View CLIP-T

(d) All Views FID (e) All Views CLIP-I (f) All Views CLIP-T

(g) Front-K Views FID (h) Front-K Views CLIP-I (i) Front-K Views CLIP-T

Fig. 7. Visualization of the evaluation results (**FID**/**CLIP-I**/**CLIP-T**) at different amounts of optimization time on a single V100-32G GPU. In comparison with the state-of-the-art controllable 3D generation method, MVControl [14], our ControLRM can achieve over faster speed and better performance.

present the quantitative comparison among our proposed ControLRM and the baselines. In most of the reported metrics, our ControLRM can achieve competetive and even better performance compared with the baselines. As the table shows, our ControLRM-D and ControLRM-T outperform other baselines on the metrics of FID and CLIP-I in all view settings. For example, under the reference view setting, our ControLRM-D/T can achieve 169.73/169.69 FID, significantly lower than the best one of the baselines, MVControl (194.97 FID score). The zero-shot experiments on **GSO** can demonstrate the great generalization ability of the proposed method on unseen test cases. We also provide the quantitative comparison between our ControLRM and MVControl [14] in Tab. 8 under 4 different input conditions. The table shows that our proposed ControLRM is still competitive compared with MVControl. For Edge and Sketch condition, both of ControLRM-D and ControLRM-T achieves better performance than MVControl in terms of FID, CLIP-I, and CLIP-T. For the Depth and Normal conditions, ControLRM-D competes effectively with MVControl, although ControLRM-T shows slightly inferior performance. An important reason is the preciseness of the given depth or normal map in controllable 3D generation. Our ControLRM is trained using the ground truth depth or normal map of the dataset, which provides absolutely precise geometric prior as conditional input. Whereas in the **GSO** benchmark, we extract the depth and normal maps using the annotator provided by MVControl. The estimated depth and normal maps generated by the models provided by MVControl lack precision, leading to significant deviations in the predicted results. This inaccuracy can be misleading for ControLRM, which relies on precise geometric conditions.

4.3.4 *Quantitative Comparison on ABO*

To evaluate the zero-shot generalization performance on controllable 3D generation, we further conduct experiments on **ABO** benchmark. The quantitative comparison with other state-of-the-art methods in 3D generation introduced in Sec. 4.1.3 is presented in Tab. 9. The table employs the metrics of **FID**, **CLIP-I**, and **CLIP-T** to evaluate the performance of controllable 3D generation. These metrics are computed under three distinct multi-view settings discussed in Section 4.3.1. From the table, we can find that ControLRM-D outperforms other baselines on all metrics. ControLRM-T achieves the second best performance in most of these metrics. In Tab. 10, we compare our ControLRM with MVControl quantitatively under 4 different input conditions. In

TABLE 11
Ablation analysis of each component in the training losses.

Models	Canny					
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	CLIP-I ↑	CLIP-T ↑
Basic Training (L_{recon})	18.810	0.8198	0.1723	105.752	0.9061	0.3031
+Adv Loss (L_{adv})	19.445	0.8303	0.1581	100.163	0.9145	0.3085
+CLIP Loss (L_{clip})	19.452	0.8306	0.1579	99.867	0.9147	0.3087
+2D Auxiliary (x_{aux})	19.454	0.8306	0.1579	99.512	0.9150	0.3091
Models	Depth					
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	CLIP-I ↑	CLIP-T ↑
Basic Training (L_{recon})	19.476	0.8314	0.1578	106.049	0.9079	0.3036
+Adv Loss (L_{adv})	20.051	0.8414	0.1469	103.625	0.9127	0.3068
+CLIP Loss (L_{clip})	20.066	0.8416	0.1465	103.220	0.9131	0.3075
+2D Auxiliary (x_{aux})	20.070	0.8417	0.1464	102.875	0.9135	0.3078
Models	Normal					
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	CLIP-I ↑	CLIP-T ↑
Basic Training (L_{recon})	19.425	0.8312	0.1618	102.247	0.9133	0.3033
+Adv Loss (L_{adv})	19.903	0.8371	0.1518	98.694	0.9168	0.3063
+CLIP Loss (L_{clip})	19.905	0.8374	0.1517	97.724	0.9180	0.3103
+2D Auxiliary (x_{aux})	19.909	0.8375	0.1516	97.489	0.9189	0.3103
Models	Sketch					
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	CLIP-I ↑	CLIP-T ↑
Basic Training (L_{recon})	18.910	0.8205	0.1703	109.158	0.9023	0.3048
+Adv Loss (L_{adv})	19.546	0.8315	0.1588	103.164	0.9113	0.3085
+CLIP Loss (L_{clip})	19.552	0.8318	0.1585	102.710	0.9118	0.3087
+2D Auxiliary (x_{aux})	19.554	0.832	0.1583	102.426	0.9121	0.309

the **ABO** benchmark, ControLRM-D exhibits competitive or superior performance in terms of **FID**, **CLIP-I**, and **CLIP-T** when compared to MVControl across all four conditions. Conversely, our lightweight model, ControLRM-T, performs slightly less effectively than MVControl under depth and normal conditions but excels in canny and sketch conditions. As outlined in Sec. 4.3.3, the extraction of depth and normal maps relies on pre-trained models supplied by MVControl. Notably, ControLRM is trained using ground truth depth and normal maps, which differ from the estimated maps provided by the pre-trained models. This distribution discrepancy between the ground truth and estimated maps adversely impacts the performance of ControLRM.

4.3.5 Qualitative Results

In Fig. 5, we compare our ControLRM-D/T with state-of-the-art 3D generation methods: MVControl [14], DreamGaussian [39], and VolumeDiffusion [47]. The figure displays rendered novel views under four different condition controls (Edge/Depth/Normal/Sketch). Our model demonstrates superior performance compared to other baselines, exhibiting higher quality and consistency in the generated 3D contents. To ensure unbiased evaluation, we adopt input samples collected from **G-OBJ**, **GSO**, and **ABO** which are unseen in the training dataset following LRM [6]. The figure illustrates the capability of our ControLRM-D/T to infer semantically plausible 3D content from a single-view input visual condition. Additionally, we showcase more examples of generated 3D content from input conditions generated by **G-OBJ**, **GSO**, and **ABO** in Figure 6, produced by our ControLRM-D. The rendered images and depth maps in novel views are jointly visualized. Our model adeptly captures the intricate geometry of diverse input conditions (such as hands, guns, axes, etc.), and maintains consistent texture generation across the outputs. The fidelity to the input visual conditions in the generated results underscores the exceptional performance and generalization capabilities of our model.

4.4 Extra Experiments

Efficiency Comparison: To provide a direct comparison of efficicency, we compare our ControLRM-D/T with the SOTA controllable 3D generation model MVControl [14] in Fig. 7. MVControl consists of two stages: the first stage generates a coarse 3D content, and the second stage attempts to optimize the 3D content with test-time optimization using SDS loss [8]. The quality of the generated 3D content improves over prolonged test-time optimization. Both of these stages are compared in the figure. We present visualizations of three evaluation metrics (FID, CLIP-I, CLIP-T) across three different multi-view settings (Reference View, All Views, Front-K Views) alongside the corresponding time consumption. The average time consumed for generating a single 3D content per sample on a V100-32G GPU is reported. We find that the refinement stage of MVControl tends to return worse performance than the coarse stage on the real-world data rather than the manually generated data used in their paper.

Ablation Study: We conduct additional experiments to comprehensively analyze the contributions of the key components in our ControLRM framework. The ablation results under four different conditions are provided in Tab. 11. By default, we utilized ControLRM-T in the ablation experiments. For evaluation, we reported the metrics of **PSNR**, **SSIM**, **LPIPS**, **FID**, **CLIP-I**, and **CLIP-T** in the table following MVControl [14]. In the table, ”Basic Training” indicates that the model was solely trained with the reconstruction loss L_{recon}. ”+Adv Loss” signifies the addition of adversarial loss L_{adv} to the reconstruction loss L_{recon}. Similarly, ”+CLIP Loss” indicates the incorporation of clip loss L_{clip}. ”+2D Auxiliary” refers to the adoption of auxiliary supervision on x_{aux}. The results demonstrate that the basic training scheme could achieve relatively good performance and meaningful generation with the support of large-scale pre-training weights from LRM [6], achieving a PSNR of approximately 18-19 in each of the four different conditions. The inclusion of adversarial loss L_{adv} can led to an improvement of 3 to 5 in FID. Furthermore, the addition of clip loss L_{clip} and 2D auxiliary supervision x_{aux} can slightly enhance the FID by about 0.5. Overall, the results in the table highlight the effectiveness of each component in our ControLRM framework in enhancing the performance of controllable 3D generation.

5 LIMITATION

In this study, the quantitative and qualitative analysis prove the superiority of our proposed method, but we also realize that this work is still insufficient and discuss the following limitations: **(1) Condition Expansion:** While significant advancements have been made under four control conditions, it is crucial to extend this framework to encompass additional control conditions such as segmentations, pose, and others. **(2) Generalization Bottleneck:** The bottleneck of the proposed method is attributed to the utilization of the pre-trained Large Reconstruction Model (LRM). Although the proposed approach effectively aligns the controllable 2D generator with the pre-trained triplane decoder, failures in the pre-trained LRM could result in the failure of our ControLRM. Therefore, enhancing the performance by employing a more robust backbone can address this issue.

6 Conclusion

This paper introduces ControLRM, a novel controllable 3D generation framework characterized by high speed and superior generation quality. Our model offers support for four different types of controls: Edge (Canny), Depth, Normal, and Sketch. The architecture comprises an end-to-end feed-forward network that includes a 2D condition encoder based on transformer or diffusion models and a 3D triplane decoder leveraging a pre-trained LRM, where only the cross-attention layers are active during training. Additionally, we introduce an joint training pipeline encompassing adversarial loss, clip loss, and reconstruction loss. To ensure fair evaluation, we collect unseen evaluation samples from three different datasets: G-OBJ, GSO, and ABO. The comprehensive quantitative and qualitative evaluation findings demonstrate that our model surpasses existing state-of-the-art methods and achieves generation speeds significantly faster by an order of magnitude.

References

[1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in International conference on machine learning. Pmlr, 2021, pp. 8821–8831.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10 684–10 695.

[3] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 9298–9309.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," Communications of the ACM, vol. 65, no. 1, pp. 99–106, 2021.

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." ACM Trans. Graph., vol. 42, no. 4, pp. 139–1, 2023.

[6] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, "Lrm: Large reconstruction model for single image to 3d," arXiv preprint arXiv:2311.04400, 2023.

[7] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," arXiv preprint arXiv:2402.05054, 2024.

[8] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," arXiv preprint arXiv:2209.14988, 2022.

[9] R. Chen, Y. Chen, N. Jiao, and K. Jia, "Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation," in Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 22 246–22 256.

[10] J. Sun, B. Zhang, R. Shao, L. Wang, W. Liu, Z. Xie, and Y. Liu, "Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior," arXiv preprint arXiv:2310.16818, 2023.

[11] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," Advances in Neural Information Processing Systems, vol. 36, 2024.

[12] O. Katzir, O. Patashnik, D. Cohen-Or, and D. Lischinski, "Noise-free score distillation," arXiv preprint arXiv:2310.17590, 2023.

[13] X. Yu, Y.-C. Guo, Y. Li, D. Liang, S.-H. Zhang, and X. Qi, "Text-to-3d with classifier score distillation," arXiv preprint arXiv:2310.19415, 2023.

[14] Z. Li, Y. Chen, L. Zhao, and P. Liu, "Controllable text-to-3d generation via surface-aligned gaussian splatting," arXiv preprint arXiv:2403.09981, 2024.

[15] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113.

[16] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in 2007 IEEE 11th International Conference on Computer Vision. IEEE, 2007, pp. 1–8.

[17] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 767–783.

[18] H. Xu, Z. Zhou, Y. Qiao, W. Kang, and Q. Wu, "Self-supervised multi-view stereo via effective co-segmentation and data-augmentation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 4, 2021, pp. 3030–3038.

[19] H. Xu, Z. Zhou, Y. Wang, W. Kang, B. Sun, H. Li, and Y. Qiao, "Digging into uncertainty in self-supervised multi-view stereo," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6078–6087.

[20] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," ACM Transactions on Graphics, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[21] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 3504–3515.

[22] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," Advances in Neural Information Processing Systems, vol. 33, pp. 2492–2502, 2020.

[23] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 5855–5864.

[24] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in European conference on computer vision. Springer, 2022, pp. 333–350.

[25] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," 2023.

[26] W. Li, R. Chen, X. Chen, and P. Tan, "Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d," arXiv preprint arXiv:2310.02596, 2023.

[27] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang, "Syncdreamer: Generating multiview-consistent images from a single-view image," arXiv preprint arXiv:2309.03453, 2023.

[28] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt et al., "Wonder3d: Single image to 3d using cross-domain diffusion," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 9970–9980.

[29] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, "Mvdream: Multi-view diffusion for 3d generation," arXiv preprint arXiv:2308.16512, 2023.

[30] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, "Triposr: Fast 3d object reconstruction from a single image," arXiv preprint arXiv:2403.02151, 2024.

[31] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang, "Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 10 324–10 335.

[32] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis et al., "Efficient geometry-aware 3d generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16 123–16 133.

[33] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu, "Gs-lrm: Large reconstruction model for 3d gaussian splatting," arXiv preprint arXiv:2404.19702, 2024.

[34] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3836–3847.

[35] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13 142–13 153.

[36] X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, C. Zhu, Z. Xiong, T. 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, 2023, pp. 9150–9161.

[37] L. Qiu, G. Chen, X. Gu, Q. zuo, M. Xu, Y. Wu, W. Yuan, Z. Dong, L. Bo, and X. Han, "Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d," arXiv preprint arXiv:2311.16918, 2023.

[38] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich, "Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation," 2022. [Online]. Available: https://arxiv.org/abs/2212.00774

[39] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," arXiv preprint arXiv:2309.16653, 2023.

[40] Z. Chen, F. Wang, Y. Wang, and H. Liu, "Text-to-3d using gaussian splatting," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 21 401–21 412.

[41] T. Yi, J. Fang, J. Wang, G. Wu, L. Xie, X. Zhang, W. Liu, Q. Tian, and X. Wang, "Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 6796–6807.

[42] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," 2022. [Online]. Available: https://arxiv.org/abs/2212.08751

[43] J. Li, H. Tan, K. Zhang, Z. Xu, F. Luan, Y. Xu, Y. Hong, K. Sunkavalli, G. Shakhnarovich, and S. Bi, "Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model," 2023. [Online]. Available: https://arxiv.org/abs/2311.06214

[44] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein, "Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation," 2024. [Online]. Available: https://arxiv.org/abs/2403.14621

[45] H. Jun and A. Nichol, "Shap-e: Generating conditional 3d implicit functions," 2023. [Online]. Available: https://arxiv.org/abs/2305.02463

[46] F. Hong, J. Tang, Z. Cao, M. Shi, T. Wu, Z. Chen, T. Wang, L. Pan, D. Lin, and Z. Liu, "3dtopia: Large text-to-3d generation model with hybrid diffusion priors," arXiv preprint arXiv:2403.02234, 2024.

[47] Z. Tang, S. Gu, C. Wang, T. Zhang, J. Bao, D. Chen, and B. Guo, "Volumediffusion: Flexible text-to-3d generation with efficient volumetric encoder," arXiv preprint arXiv:2312.11459, 2023.

[48] J. Lorraine, K. Xie, X. Zeng, C.-H. Lin, T. Takikawa, N. Sharp, T.-Y. Lin, M.-Y. Liu, S. Fidler, and J. Lucas, "Att3d: Amortized text-to-3d object synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2306.07349

[49] K. Xie, J. Lorraine, T. Cao, J. Gao, J. Lucas, A. Torralba, S. Fidler, and X. Zeng, "Latte3d: Large-scale amortized text-to-enhanced3d synthesis," arXiv preprint arXiv:2403.15385, 2024.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[51] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4195–4205.

[52] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," ArXiv e-prints, pp. arXiv–1607, 2016.

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

[54] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

[55] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[56] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning. PMLR, 2021, pp. 8748–8763.

[57] M. Li, P. Zhou, J.-W. Liu, J. Keppo, M. Lin, S. Yan, and X. Xu, "Instant3d: Instant text-to-3d generation," International Journal of Computer Vision, pp. 1–17, 2024.

[58] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," arXiv preprint arXiv:1701.04862, 2017.

[59] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4401–4410.

[60] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021.

[61] G. Parmar, T. Park, S. Narasimhan, and J.-Y. Zhu, "One-step image translation with text-to-image models," arXiv preprint arXiv:2403.12036, 2024.

[62] A. Sauer, F. Boesel, T. Dockhorn, A. Blattmann, P. Esser, and R. Rombach, "Fast high-resolution image synthesis with latent adversarial diffusion distillation," arXiv preprint arXiv:2403.12015, 2024.

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, 2014.

[64] N. Kumari, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Ensembling off-the-shelf models for gan training," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10 651–10 662.

[65] K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely, "Arf: Artistic radiance fields," in European Conference on Computer Vision. Springer, 2022, pp. 717–733.

[66] T. Luo, C. Rockwell, H. Lee, and J. Johnson, "Scalable 3d captioning with pretrained models," Advances in Neural Information Processing Systems, vol. 36, 2024.

[67] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence, no. 6, pp. 679–698, 1986.

[68] Z. He and T. Wang, "Openlrm: Open-source large reconstruction models," https://github.com/3DTopia/OpenLRM, 2023.

[69] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 2553–2560.

[70] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in International conference on machine learning. PMLR, 2023, pp. 19 730–19 742.

[71] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Yago Vicente, T. Dideriksen, H. Arora, M. Guillaumin, and J. Malik, "Abo: Dataset and benchmarks for real-world 3d object understanding," CVPR, 2022.

[72] M. Li, T. Yang, H. Kuang, J. Wu, Z. Wang, X. Xiao, and C. Chen, "Controlnet++: Improving conditional controls with efficient consistency feedback," arXiv preprint arXiv:2404.07987, 2024.

[73] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 3, pp. 1623–1637, 2020.

[74] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," arXiv preprint arXiv:2302.12288, 2023.

[75] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13 137–13 146.

[76] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," Advances in neural information processing systems, vol. 30, 2017.