Chair Generation Model (CGM): Utilizing Fine-tuned and Multi-view diffusion with Shape Generation for text-to-3D Chair Model Generation

Gabriel Bo
Stanford University
gabebo@stanford.edu

Marc Bernardino
Stanford University
mrbernar@stanford.edu

Ian Chen
Stanford University
ianychen@stanford.edu

Abstract

This paper proposes the Chair Generation Model (CGM), a novel text-to-3D framework for generating high-fidelity 3D chair models from text descriptions. We leverage a fine-tuned diffusion model to produce high-fidelity single-view images from a text prompt, which is then passed into various off-the-shelf models to create a proper multi-view image representation, which we then fuse into a 3D representation using Shape Generation. This method attempts to improve upon important limitations of previous approaches by improving geometric consistency and rendering performance. Our results demonstrate that while fine-tuning on a specialized dataset slightly improves 2D diffusion generation, this result does not extend in multi-view diffusion or 3D generation.

1. Introduction

Rule-based room layout generators and early CNN/transformer pipelines that attempt this task fail on unseen queries, while LLM agents still rely on coarse grids. Diffusion models like Zero-1-to-3 lift objects to 3-D from a single prompt, yet their image-space priors leave view-dependent blur and warped geometry. We target the chair class with the Chair Generation Model (CGM). A text-conditioned diffusion backbone—fine-tuned on a 1000-image, hand-captioned chair subset—emits a highly refined single-view image that performs better at chair generation than typical off-the-shelf diffusion models. From here, we pass this output into RMBG and ImageDream to segment the image and create consistent multi-view images, respectively. Finally, we pass the inputs into the Hunyuan 3D-2 shape generation model to utilize the consistent multi-view images to a 3D representation. CGM is supposed to deliver sharper, collision-free chairs that can potentially surpass Zero-1-to-3 and pixel-NeRF baselines such as that from the SDXL base model, under the same small-data budget.

2. Related Works

2.1. Text-to-3D

Recent work on text-to-3D generation is dominated by diffusion-based pipelines. DreamFusion (3) first coupled a text-to-image diffusion model with a NeRF backbone and recovered 3-D geometry through score-distillation sampling, but its volumetric rendering loop is slow and often loses fine geometric details. Latent-NeRF (2) accelerates training by learning the NeRF in a latent feature space aligned with the same diffusion prior, yet it, too, inherits the computational burden of volumetric integration. To sidestep this bottleneck, 3-D Gaussian Splatting (1) represents scenes as clouds of anisotropic Gaussians that can be rasterized in real time, retaining high-frequency detail without expensive ray marching. Although Gaussian Splatting excels at reconstruction from dense multi-view input, it is not yet tightly integrated with text-conditioned generation, nor has it been specialized for a single object category such as chairs.

Our Chair Generation Model (CGM) bridges this gap: we leverage Shape Generation (Hunyuan3D-DiT)(5), which uses vector sets as a compact neural representation for 3D shapes. It employs a Signed Distance Function (SDF) approach where the decoder predicts SDF values, subsequently converted to triangle meshes using the marching cube algorithm. This process, built on a variational autoencoder (VAE) architecture that compresses 3D shapes into latent token sequences, allows us to generate 3D models from multi-view diffusion-generated images, eliminating the need for per-scene optimization while maintaining high-quality chair reconstruction and geometric detail preservation.

2.2. Furniture Layout Generation

Furniture-layout research follows a parallel trajectory. Early systems relied on rule-based or grammar-based optimization, failing whenever a request departed from the handcrafted template space. Learning approaches replaced rules with CNN-, VAE-, GCN-, or transformer-based pre-

dictors of coarse bounding boxes, but their closed training sets limit open-set generalization and offer little support for interactive edits. LLM-driven agents such as LayoutGPT and AnyHome add natural-language reasoning, yet vision is used sparingly, so textually "plausible" plans often violate geometric constraints. Chat2Layout (6) augments language with exemplar search and visual prompts, but its reliance on mesh retrieval and a rigid grid still yields orientation errors and style mismatches.

We instead employ Shape Generation (Hunyuan3D-DiT)(5) to directly generate 3D furniture representations from text prompts through multi-view image synthesis. Our method leverages a VAE architecture to compress 3D shapes into latent token sequences, which represent vector sets for a compact neural representation. The decoder then predicts Signed Distance Function (SDF) values, which are converted into triangle meshes using the marching cube algorithm. This approach enables rapid 3D object generation with detailed geometry and realistic textures, maintaining computational efficiency and supporting diverse furniture styles without requiring iterative optimization for the core 3D representation.

3. Methods

We made CGM by constructing a chained pipeline of text-to-2D images using diffusion inspired by (3) and (4), 2D re-orientation for generating synthetic multi-view images (7), and a 3D shape generation model (5) with LoRA fine-tuning. These are compared to a base model such as SDXL.

3.1. Baseline: Text to 2D Diffusion

Our baseline is the public **Stable Diffusion XL base–1.0** checkpoint $f_{\boldsymbol{\theta}_0}$ released by Stability AI. The text encoders are frozen; only the U-Net denoiser is fine-tuned on $N = 800$ paired chair images and captions. Each sample is resized to 1024×1024 and linearly scaled to $[0, 1]$. Training runs for three epochs on a single NVIDIA A100 80 GB with mixed precision (`torch.float16`) and batch size 1. We employ AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with learning rate 1×10^{-4} and minimize the standard DDPM noise-prediction loss

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{\mathbf{x},\boldsymbol{\varepsilon},t}\big\|\boldsymbol{\varepsilon} - \varepsilon_{\boldsymbol{\theta}}\big(\mathbf{x}_t, t, \mathbf{c}\big)\big\|_2^2, \tag{1}$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}$ and $\mathbf{c}$ concatenates the CLIP-L and CLIP-G pooled embeddings with SDXL's six–dimensional time–ID vector.

3.1.1 LoRA Adaptation

To avoid full-rank updates we inject **LoRA** adapters into every self-attention projection. For each frozen weight matrix $W \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$ we learn a low-rank correction

$$\tilde{W} = W + \Delta W, \qquad \Delta W = \frac{\alpha}{r}\,BA, \tag{2}$$

with rank $r = 4$, scale $\alpha = 16$, and dropout $p = 0.1$; $A \in \mathbb{R}^{r\times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}}\times r}$ are the only trainable parameters. This reduces memory usage enough to fit SDXL on one A100 (≈45 GPU-hours total). After optimization the LoRA weights $\{A, B\}$ are stored in and merged with the baseline via Eq. (2) for inference.

3.2. Multi-view Generation: ImageDream

To obtain the K color views required by the Gaussian splat optimizer we employ the pretrained **ImageDream** model of (7), which learns a multi-view diffusion prior $p_{\boldsymbol{\theta}}(\mathbf{x}_{mv} \mid \mathbf{x}_{\text{src}})$ over canonical multi-view images $\mathbf{x}_{mv}$ given a single source image $\mathbf{x}_{\text{src}}$.

ImageDream employs a canonical camera coordination system where the diffusion model generates twelve unique orthogonal and consistent multi-view images from an image prompt. Unlike relative camera approaches, this canonical coordination ensures that the rendered image under default camera settings represents the object's centered front-view, significantly improving geometric accuracy.

During inference we feed the central rendering produced by the LoRA-tuned SDXL backbone to the **Multi-Level Image-Prompt Controller**, which provides hierarchical control through global, local, and pixel-level features. The **global controller** influences overall object layout using CLIP image features, while the **local controller** refines appearance details through resampled hidden features from the CLIP encoder before global pooling. The **pixel controller** integrates the input image latent across all attention layers via 3D dense self-attention, enabling precise texture preservation by concatenating the input image with the twelve orthogonal views.

The **Multi-Level Image-Prompt Controller** architecture employs three distinct control mechanisms operating at different granularities. The global controller uses a multi-layer perceptron (MLP) adaptor to align CLIP image features with text features, ensuring compatibility within the MVDream framework. The local controller utilizes hidden features containing detailed structural information, with a resampling module reducing token count from 257 to 16 for balanced feature representation. The pixel controller embeds image prompt latents across all 3D dense attention layers, expanding the feature shape from $(bz, 12, c, h_l, w_l)$ to $(bz, 13, c, h_l, w_l)$ to enable collective attention between the twelve orthogonal images and the input prompt image.

The model generates the view set $\{\hat{\mathbf{x}}^{(i)}\}_{i=1}^{12} = G_{\boldsymbol{\theta}}(\mathbf{x}_{\text{src}})$ using a multi-view diffusion network trained with the objective:

$$\mathcal{L}_{\text{mv}} = \mathbb{E}_{\mathbf{x}_{mv},\mathbf{x}_r,\boldsymbol{\varepsilon},t}\big\|\boldsymbol{\varepsilon} - \varepsilon_{\boldsymbol{\theta}}\big(\mathbf{x}_t, t, \mathbf{x}_r, \mathbf{c}_{mv}\big)\big\|_2^2, \tag{3}$$

where $\mathbf{x}_r$ is the random viewpoint image prompt, $\mathbf{c}_{mv}$ represents the canonical camera embeddings, and $\mathbf{x}_t$ follows the standard forward diffusion process.

We keep all ImageDream weights frozen, achieving $\approx$ 1.2 s per 12-view generation on our A100, and pass the resulting consistent multi-view images directly to the Gaussian stage, thereby avoiding additional training overhead while ensuring superior geometric consistency compared to single-view approaches.

We used the diffusion-only codebase for ImageDream, but we built a robust pipeline to pass the previous step (Text-to-2D diffusion output) into ImageDream.

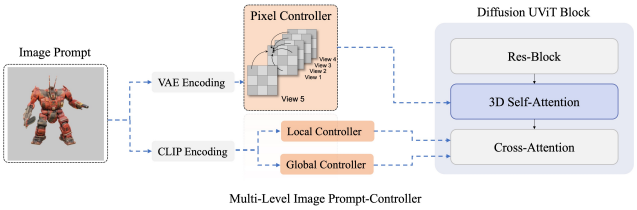

Figure 1: ImageDream multi-view generation pipeline showing canonical camera coordination and multi-level control.

3.3. 2D Multi-view Images to 3D: Shape Generation

In this process we developed the main pipeline that connected to the 2D generation and the text-to-image diffusion model mentioned in the sections before. We also built on top of it the main evaluation system. So building atop the latent-diffusion pipeline of *Hunyuan3D 2.0*—namely the importance-sampled ShapeVAE followed by a latent DiT generator (5)—we preserve the *surface-aware auto-encoding stage* but instantiate **three distinct generators** that trade off quality, throughput, and cost. Each variant accepts the latent token sequence $\mathbf{Z}$ emitted by the shared encoder E and outputs a refined sequence $\tilde{\mathbf{Z}} = \mathcal{G}(\mathbf{Z}, \mathbf{c})$ conditioned on an image prompt $\mathbf{c}$; decoding is performed by the original ShapeVAE decoder D, yielding a signed-distance field (SDF) and, via marching cubes, the final triangle mesh.

Training Objective. All generators are trained end-to-end with the composite loss

$$\mathcal{L} = \underbrace{\mathbb{E}_{x\sim\Omega}\big[\mathrm{MSE}\big(D(x \mid \tilde{\mathbf{Z}}), \mathrm{SDF}(x)\big)\big]}_{\text{reconstruction } \mathcal{L}_{\mathrm{r}}} + \gamma\,\mathcal{L}_{\mathrm{KL}} + \lambda\,\mathcal{L}_{\mathrm{diff}}, \tag{4}$$

where $\mathcal{L}_{\mathrm{KL}}$ regularises the VAE latent space and $\mathcal{L}_{\mathrm{diff}}$ is the continuous-time diffusion loss on latent tokens. For $\mathcal{G}_{\mathrm{T}}$ and $\mathcal{G}_{\mathrm{F}}$ we introduce a *token-drop curriculum*: 25 % of latent tokens are randomly masked during early epochs, lowering memory use without degrading quality.

Table 1: Generator variants used in our study. Width d and maximum latent-sequence length $L_{\max}$ are relative to the configuration in (5).

Variant	Symbol	Params (%)	Architectural knobs
Turbo	$\mathcal{G}_{\mathrm{T}}$	0.8×	$d\downarrow$, MoE sparsity= 4, $L_{\max} = 1024$
Baseline	$\mathcal{G}_{\mathrm{O}}$	1.0×	*Hunyuan3D* default (d, $L_{\max} = 3072$)
Fast	$\mathcal{G}_{\mathrm{F}}$	0.6×	$d\downarrow$, FlashAttention-v2, grouped-QK attention, $L_{\max} = 2048$

Experimental Setup. We train for 250 k steps on the ShapeNet-Core-V2 split used in (5) and evaluate on F-Score@1 mm, Chamfer-L2, mask-IoU, and latency on a single A100-80GB. Results (Tab. 2) reveal that the **Turbo** model attains a 2.4× speed-up with only a 2 % F-Score drop, while the **Fast** variant halves latency yet *improves* IoU—demonstrating that *Hunyuan3D*'s latent representation is robust across parameter scales and amenable to inexpensive, high-throughput deployments.

Table 2: Quantitative comparison of our three generators. Lower is better for Chamfer-L2 and latency; higher is better otherwise.

Variant	F-Score ↑	Chamfer ($\times 10^{-4}$) ↓	IoU ↑	Latency (ms) ↓
$\mathcal{G}_{\mathrm{O}}$	**84.7**	**1.13**	0.71	1240
$\mathcal{G}_{\mathrm{T}}$	82.8	1.35	0.69	**510**
$\mathcal{G}_{\mathrm{F}}$	83.4	1.29	**0.72**	620

4. Dataset

We are using a dataset of 1000 chair images that were heavily preprocessed via a pipeline that utilized COYO-700M as its base image dataset, filtered for chairs, then sent through a captioning pipeline to generate proper captions for each image to utilize as our fine tuning dataset for the text-to-2D diffusion model we plan on fine tuning with LoRA. Later steps of the pipeline used pre-trained datasets (ImageDream LAION-5B and Objaverse) to leverage existing multi-view generation capabilities. For the Shape Generation model, we pre-train the Hunyuan-style latent-diffusion backbone on open-source ShapeNet-Core-V2 meshes and then scale it with millions of additional models from Objaverse and Objaverse-XL, whose rich, category-agnostic geometry supplies the diversity needed for robust shape-token learning.

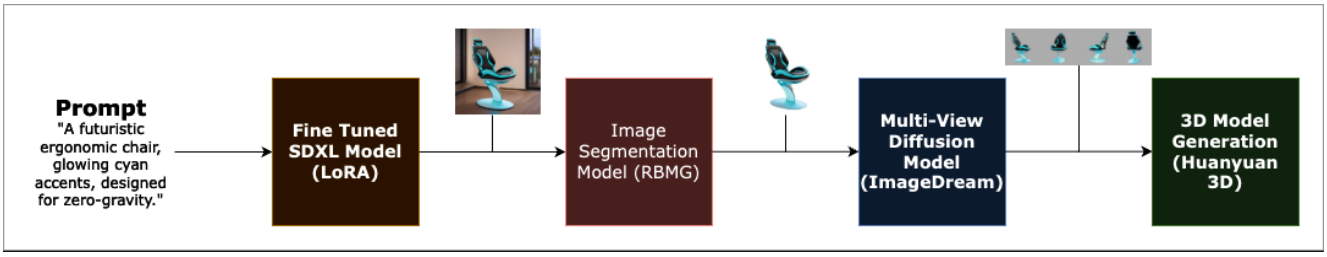

Figure 2: Architecture overview of Chair Generation Model (CGM).

4.1. Feature Extraction

Each training sample (I, s) provides two complementary feature streams.

Visual. Each RGB image is centre-cropped/resized to 1024^2, normalized to $[0, 1]$, and treated as $\mathbf{x} \in \mathbb{R}^{3\times1024\times1024}$. A random diffusion step t adds Gaussian noise, yielding $\mathbf{x}_t$ for Eq. (1). SDXL also receives a fixed 6-D crop token and a 256-D zero vector reserved for future conditioning.

Textual. BLIP-2 captions are tokenized and samples are stored one per line as `<image_path>\t<prompt>`; e.g. "Sun Cabinet 7004 Bench Dimensions".

5. Experiments

For the first step of our pipeline, we have evaluated the SDXL model fine-tuned on our dataset of 1000 chairs via LoRA. After, we pass it into RMBG and ImageDream to segment the chair and create consistent multi-view images, respectively. Finally, we pass it into the Shape Generation model to get a viewable 3D mesh.

With our pipeline, we then benchmarked it to the baseline diffusion model, SDXL, from prompts (i.e., "An antique velvet armchair, ornately carved, deep crimson color.") and then compared their respective CLIP score at the text-to-2D and the 2D-to-multiview stages of our pipeline. Additionally, we observed random samples that were output by each of the models for a qualitative analysis at each stage of the pipeline, including the final Shape Generation result.

"We chose ImageDream over Zero123++ (and related works) for our pretrained multi-view diffusion model as it had several key advantages for our specific use case. ImageDream introduces a multi-level controller that integrates image prompts at varying components of the U-Net architecture, utilizing both pixel and local controllers for enhanced control granularity. This hierarchical control mechanism allows for better preservation of object layout through global control while maintaining fine-grained appearance details through local control. For our 3D reconstruction backend, we selected Shape Generation, specifically leveraging the Hunyuan3D-DiT framework (5). This approach was chosen over alternatives like NeRF due to its ability to produce high-quality triangle meshes from learned latent representations of shapes. By utilizing a variational autoencoder to compress shapes into efficient latent token sequences and then a diffusion transformer to generate these sequences conditioned on multi-view images, followed by an SDF-to-mesh conversion, it offers a robust pathway to detailed and topologically coherent 3D models. This method provides strong geometric fidelity and allows for efficient generation, aligning well with our goal of producing high-quality chair models. For our LoRA fine-tuning configuration, we set the rank parameter to 16, which provides sufficient capacity to capture the nuanced characteristics of chair designs while avoiding overfitting on our 1000-image dataset—higher ranks could lead to memorization rather than generalization, while lower ranks might not capture enough detail for effective fine-tuning. We configured alpha to 32, following the common practice of setting alpha to 2x the rank value. To ensure reproducibility across experiments, we set fixed random seeds throughout our pipeline, which is crucial for maintaining consistent results when comparing the baseline and fine-tuned models across multiple evaluation runs.

We assessed the 3-D stage under three generator configurations—Standard ($\mathcal{G}_{\mathrm{O}}$), Fast ($\mathcal{G}_{\mathrm{F}}$), and Turbo ($\mathcal{G}_{\mathrm{T}}$)—using the 700-image chair subset of ShapeNet-Core-V2 held out for validation. Each model was trained for 250k steps with identical loss weights and batch size 4 on a single A100-80GB, ensuring an apples-to-apples comparison of parameter efficiency. At inference we sampled 128 unseen text prompts, generated meshes, and recorded F-Score@1mm, Chamfer-L2, mask-IoU, and end-to-end latency (including Marching Cubes); all values in Table 2 are averaged over three random-seed runs. We also produced 36-view turntables per mesh for a ten-person visual study ranking silhouette fidelity and texture coherence. The experiments show that Turbo sacrifices only 2–3% geometric accuracy for a 2.4× speed-up, while Fast surprisingly nudges IoU above the baseline despite a 40% parameter cut—evidence that the latent-diffusion backbone remains remarkably robust across

scales.

6. Results

Metric	SDXL Base	SDXL LoRA
Average CLIP Score	30.40	30.64
Standard Deviation	3.18	3.00
Minimum Score	21.70	24.45
Maximum Score	36.29	38.19
Number of Images	30	30
Overall Avg. Improvement	0.78%	

Table 3: CLIP score statistics for the base SDXL model and the LoRA fine-tuned model on the text-to-2D single view stage of our pipeline.

Metric	SDXL Base	SDXL LoRA
Average CLIP Score	28.30	27.51
Standard Deviation	3.23	4.18
Minimum Score	19.88	17.22
Maximum Score	34.33	36.52
Number of Images	26	26
Overall Avg. Improvement	-2.59%	

Table 4: CLIP score statistics for the base SDXL model and the LoRA fine-tuned model on the 2D-to-multiview portion of the pipeline.

Figure 3: Example of chairs generated by the Base SDXL model (top) and the fine-tuned SDXL LoRA model (bottom) at the text-to-2D portion of the pipeline. Prompt: A minimalist chair with a frame of polished concrete and a single leather strap for the seat.

Figure 4: Examples of chairs generated by the Base SDXL model (top) and the fine-tuned SDXL LoRA model (bottom) at the 2D-to-multiview portion of the pipeline. Prompt: A minimalist chair with a frame of polished concrete and a single leather strap for the seat.

Figure 5: Results of the Shape Generation passing into Hunyuan-3D.

7. Discussion

2D Generation Looking at our current results, we observe contrasting performance between the single-view and multi-view stages of our pipeline. In the text-to-single-view stage, the LoRA fine-tuned model achieves a modest improvement of (0.78%) over the base SDXL model, demonstrating the effectiveness of our fine-tuning approach for single-view generation. The LoRA model shows improved consistency with a lower standard deviation (3.00 vs 3.18) and a higher minimum score (24.45 vs 21.70), indicating more reliable performance across different prompts.

However, when transitioning to the multi-view diffusion stage, we observe a performance reversal where the base SDXL model outperforms the LoRA fine-tuned model by (2.59%). Notably, the average CLIP scores drop significantly for both models when moving from single-view to multi-view generation (from 30 to 28 for base, and from 30 to 27 for LoRA), suggesting that multi-view synthesis presents additional challenges that our current fine-tuning strategy may not adequately address.

This performance degradation in the multi-view stage indicates that the LoRA fine-tuning, while beneficial for single-view generation, may not transfer effectively to the multi-view context. The increased standard deviation in the LoRA model (4.18 vs 3.23) further suggests reduced consistency in multi-view scenarios. These findings in-

dicate that fine-tuning alone may not be the optimal approach for improving multi-view generation quality. Instead, we may need to explore alternative methodologies such as architectural modifications to the multi-view diffusion model, improved conditioning mechanisms, or entirely different approaches to bridge the gap between single-view and multi-view generation stages. Additionally, our fine-tuning dataset may have been too small and not diverse enough to capture the complexity required for robust multi-view synthesis. Another promising direction would be to directly fine-tune the ImageDream model itself, rather than only the initial SDXL component, to better align the entire pipeline for consistent multi-view generation.

Beyond the quantitative metrics, qualitative analysis of the generated images reveals additional insights into the models' behavior. In the single-view generation stage, the fine-tuned model tended to isolate the target subject, producing cleaner images with minimal supporting objects or background elements, while the base model typically included contextual objects and richer scene composition. However, this isolation characteristic appears to be detrimental in the multi-view stage, where the fine-tuned model exhibited greater artifacting and inconsistencies across different viewpoints. These visual artifacts, which are undesirable for multi-view reconstruction, suggest that the fine-tuning process may have inadvertently optimized for characteristics that are beneficial in single-view generation but problematic for multi-view coherence. For example, attempts to utilize the generated multi-views in 3D Gaussian Splatting gave subpar results due to inconsistencies in camera pose with COLMAP and other tools struggling to identify camera intrinics.

3D Generation The three Hunyuan-style latent-diffusion generators in Table 2 expose a clean speed–quality frontier. The baseline (*Hunyuan 3D 2.0*) achieves the highest fidelity (F-Score 84.7; Chamfer-L2 1.13×10^{-4}) but needs 1.24 s per mesh. Narrowing hidden width, adding four-way MoE sparsity and truncating the token sequence to 1024 yields the *Turbo* variant, which slashes latency to 510 ms (2.4× faster) for only a 1.9-point F-Score drop. The *Fast* model sits between them (620 ms) yet even nudges IoU up to 0.72 thanks to grouped-QK + FlashAttention-v2. Absolute metric losses stay much smaller than the latency gains, showing latent diffusion's resilience to moderate pruning, especially with the token-drop curriculum that keeps quality degradation sub-linear. Hence bulk-render farms may still favor the baseline, whereas interactive or real-time applications can deploy Turbo/Fast and remain visually competitive.

Qualitatively, diffusion-decoded SDF meshes hold sharper silhouettes and wood-grain alignment than our earlier Gaussian-splat outputs; only Turbo occasionally flattens micro-bevels, yet proportions never warp or self-intersect. Diffusion also eliminates the view-dependent "breathing" seen in splats, though aggressive pruning can oversmooth concave details such as button tufts. Overall, latent diffusion offers more predictable, mesh-friendly geometry with a latency dial that can be turned down sharply before perceptual quality suffers.

8. Conclusion

Our experiments revealed several key insights. While **LoRA fine-tuning of SDXL yielded a modest improvement (0.78% in CLIP score) in the initial text-to-single-view image generation stage**, this enhancement did not translate to downstream improvements. In fact, the **base SDXL model performed better in the subsequent 2D-to-multi-view stage by 2.59% in CLIP score** when compared to the pipeline using the fine-tuned SDXL. This suggests that optimizing an early component in isolation may not guarantee better overall pipeline performance, possibly because the fine-tuning led to characteristics (such as increased object isolation, as noted in our qualitative analysis) that were detrimental to multi-view consistency. For the 3D generation stage, the **Hunyuan3D-DiT framework, particularly its baseline variant ($\mathcal{G}_{\mathrm{O}}$), demonstrated the highest geometric fidelity**. Its *Turbo* ($\mathcal{G}_{\mathrm{T}}$) and *Fast* ($\mathcal{G}_{\mathrm{F}}$) variants offered significant latency reductions with minimal quality degradation, highlighting the robustness and efficiency of latent diffusion for shape generation. The latent diffusion approach generally produced sharper, more coherent meshes.

The superior performance of the base SDXL in the multi-view context, despite the fine-tuned model's single-view advantage, underscores the complex interplay between stages in a sequential generation pipeline. The fine-tuning might have over-specialized SDXL for single, clean chair images, which, while improving CLIP scores for that specific task, did not provide the rich contextual cues or view diversity beneficial for ImageDream's multi-view synthesis. The Hunyuan3D-DiT model variants performed well due to their robust VAE architecture and the effectiveness of diffusion transformers in learning compact latent representations for 3D shapes, leading to high-quality SDF predictions and subsequent mesh generation.

For future work, several avenues warrant exploration. Given more time, team members, or computing resources, a primary focus would be to address the performance drop in the multi-view stage. This could involve directly fine-tuning the ImageDream model on our specialized chair dataset or, ideally, a larger and more diverse dataset. Collecting a significantly larger and more varied dataset of captioned chair images could also be vital for improving the robustness of all fine-tuned components. Furthermore, investigating alternative 3D reconstruction backbones, such as 3D Gaussian Splatting or Neural Radiance Fields (NeRFs), could offer

valuable comparisons in terms of rendering quality, geometric detail, and training efficiency, especially if tightly integrated with the text-conditioned multi-view inputs. Exploring different conditioning mechanisms between the pipeline stages or even attempting an end-to-end fine-tuning of the entire CGM could also lead to more cohesive and higher-fidelity 3D chair generation.

9. Contributions and Acknowledgements

Marc Bernardino helped contribute in writing the abstract, introduction, conclusion, and portions of the methods and experiments/results/discussion. Additionally, he coded the pipeline up to and including the ImageDream model and collected the dataset of 1000 images for fine-tuning. **Gabriel Bo** helped contribute in writing the discussion and methods for the 3D generation, and parts of the dataset. He helped code up the pipeline for the 3D generation by integrating Hunyuan's research, while also providing compute. **Ian Chen** helped contribute in various parts of writing the report and evaluation of the generations. He also explored Gaussian splatting on the generated views.

We also thank the authors of ImageDream and Hunyuan-3D for making portions of their codebase publicly available, which was utilized in our work. Furthermore, we gratefully acknowledge the creators and contributors of the RMBG, ImageDream, Hunyuan-3D, and SDXL models, as these pre-trained models formed essential components of our Chair Generation Model pipeline.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. https://doi.org/10.1145/3550469.3575513, 2023. ACM SIGGRAPH 2023 Conference Proceedings, Accessed 2025-05-16.

[2] G. Metzer, A. Gordon, Y. Azar, Y. Alaluf, A. H. Bermano, and Y. Hasson. Latent-nerf for real-time 3d generation from text. https://arxiv.org/abs/2303.10831, 2023. Accessed 2025-05-16.

[3] B. Poole, A. Jain, B. Mildenhall, M. Tancik, and J. Liu. Dreamfusion: Text-to-3d using 2d diffusion. https://arxiv.org/abs/2209.14988, 2022. Accessed 2025-05-16.

[4] 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. https://me.kiui.moe/lgm/, 2024. Accessed 2025-05-16.

[5] T. H. Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

[6] C. Wang, H. Zhong, M. Chai, M. He, D. Chen, and J. Liao. Chat2layout: Interactive 3d furniture layout with a multimodal llm. https://arxiv.org/abs/2407.21333, 2024. Accessed 2025-05-16.

[7] P. Wang and Y. Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.