3DIFY-ANYTHING: A UNIFIED MODEL FOR MULTIMODAL 3D GENERATION VIA CONTEXT PRETRAINING

Anonymous authors
Paper under double-blind review

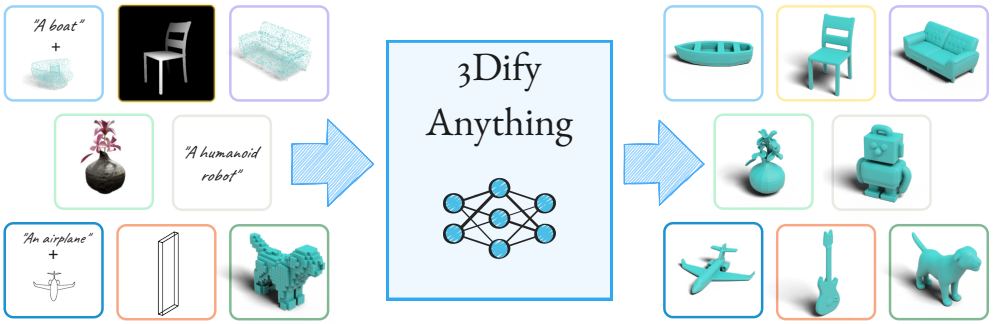

Figure 1: 3Dify-Anything generates 3D shapes in seconds from diverse inputs, such as text, images, sketches, depth maps, voxels, bounding boxes, and point clouds, or their combinations.

ABSTRACT

We introduce *3Dify-Anything*, a single model that turns diverse inputs into high-quality 3D assets with fast inference. Inputs can be text, images, sketches, depth maps, point clouds, voxels, bounding boxes, or any combination of these. The model rests on two ideas. The first is an *encoder–decoder design* that separates understanding the context from generating the shape. A large transformer context encoder reads the input once, and a lightweight rectified-flow decoder generates the 3D latent in a few ODE steps while attending to the encoder's representation. Because only the decoder runs at every step, capacity is concentrated where it is cheap, giving fast inference without sacrificing fidelity. The second idea is *context pretraining with cross-modal masking*, which makes the encoder strong enough for a small decoder to rely on. The encoder is pretrained with masked token prediction on single- and multi-modal contexts, including the target 3D shape tokens. For multi-modal contexts, one modality is kept visible while the other is heavily masked, so the encoder learns to predict one modality from another. Across a wide range of settings, *3Dify-Anything* is competitive with specialized single-task baselines while offering far greater flexibility in how inputs are composed. A compact variant enables near real-time multimodal 3D generation.

1 INTRODUCTION

Current 3D generative models have made tremendous progress in generating high-quality 3D shapes at scale (Xiang et al., 2025b; Sanghi et al., 2024; Li et al., 2025; Nichol et al., 2022b; Jun & Nichol, 2023; Zeng et al., 2022), particularly in image-to-3D (Liu et al., 2023b; Hong et al., 2023; Xu et al., 2024). However, the landscape remains predominantly single-modality rather than multi-modal: most models are trained and tuned for exactly one conditioning modality and do not enable flexible, condition-based generation (e.g., combining or switching among text, images, sketches, point clouds, or voxels). Supporting an alternative input typically requires training a separate model (Liu et al., 2023b; Poole et al., 2022) or substantial architectural modification (Zhang et al., 2023b; Ye et al., 2023). Text-to-3D often uses a two-stage pipeline (Poole et al., 2022; Lin et al., 2023; Wang et al., 2023) that first generates images and then lifts them to 3D, which increases latency and compounds errors across stages. Even single-modality inference is often slow due to iterative optimization or long sampling schedules.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Multi-modal training is inherently difficult, which is why most 3D generators remain single-modality. First, representation fusion is non-trivial: conditioning signals arrive from different encoders (e.g., DINOv2 for images (Oquab et al., 2023), CLIP for text (Radford et al., 2021)), yielding features with mismatched dimensionalities, spatial layouts/resolutions, and statistics. Naïve fusion (e.g., concatenation or averaging) leads to size mismatches, modality imbalance, and negative transfer across modalities. Moreover, some modalities lack strong pretrained encoders (e.g., low-resolution 16^3 voxels). Second, capacity and compute scale steeply with each added condition: ControlNet-style controllers (Zhang et al., 2023b), per-modality adapters or LoRA (Hu et al., 2022; Ye et al., 2023), and extra cross-attention blocks often replicate parts of the backbone, inflating parameters, memory, and wall-clock time.

We introduce 3Dify-Anything, which addresses these challenges with two complementary ideas: an *encoder–decoder design* separating context understanding from shape generation, and *context pretraining with cross-modal masking*, which makes the encoder strong enough for this separation.

Encoder–decoder generation. Diffusion and flow models spend almost all of their inference compute in an iterative loop, and conditioning is usually re-processed at every step of that loop. We instead use a large transformer *context encoder* that reads all conditioning inputs once and produces a rich multimodal representation. A lightweight rectified-flow *decoder* integrates a short ODE (5–18 Heun steps) in a compact 3D latent space and reads this representation through cross-attention at every step. Encoder capacity is paid for once per shape and decoder capacity at every step, so we make the encoder large and the decoder small, giving fast inference without sacrificing fidelity. In our experiments, a decoder-only model with the same total parameter count produces lower-quality shapes, and so does our encoder architecture trained from scratch, which isolates the benefit to pretraining. Since every modality enters through the same encoder, the generator needs no per-modality branches, adapters, or controller replicas. To make this possible, all inputs, including the target 3D shape, are tokenized into a shared vocabulary.

Context pretraining with cross-modal masking. A small decoder can only be as good as the representation it reads, so the encoder must be strong and must align the modalities with each other. We pretrain it with masked token prediction on single- and multi-modal contexts before training the decoder. Single-modality contexts use standard masking, which learns strong within-modality features. For multi-modal contexts, however, most randomly masked tokens can be recovered from neighbouring tokens of the same modality, so the encoder is never forced to relate the modalities. For modality pairs we therefore use *cross-modal masking*: one modality is kept fully visible while the other is heavily masked, up to entirely. The encoder must then predict one modality from the other, for example image or 3D shape tokens from text, or voxels from a point cloud. Because the target 3D shape tokens are among the pretraining modalities, this ties every conditioning signal to the geometry the decoder will generate. In zero-shot clustering on ModelNet40, cross-modal masking gives the best representations among the masking schemes we compare, with the largest gains on multi-modal inputs. Different modalities turn out to be best represented at different encoder depths, so the decoder conditions on a learned fusion of several layers rather than a single one.

In summary, we make the following contributions:

- We propose an *encoder–decoder architecture for multimodal 3D generation*. A large context encoder is run once, and a lightweight rectified-flow transformer decoder performs the iterative generation. Concentrating capacity in the single-pass encoder gives fast inference and lets one model accept any combination of inputs without per-modality modules.
- We propose *context pretraining with cross-modal masking*. It is a masked self-supervised objective over single- and multi-modal contexts, including the 3D shape tokens. Single-modality contexts use standard masking. In modality pairs, one side stays visible while the other is heavily masked, so the encoder learns to predict one modality from another. This yields better aligned, more semantic representations than standard masking, allowing the decoder to stay small.
- We present *3Dify-Anything*, a controllable, versatile, and fast multimodal 3D generator that accepts text, images, depth, sketches, masked images, voxels, point clouds, part point clouds, bounding boxes, and their combinations. It is competitive with specialized single-task baselines, outperforming them on point-cloud and voxel reconstruction, and a compact variant achieves near real-time generation.

2 BACKGROUND

3D shape generation. 3D generative models include autoregressive models, GANs, normalizing flows, and diffusion models (Nash et al., 2020; Gao et al., 2022b; Yang et al., 2019; Nichol et al., 2022a; Hui et al., 2024). Most recent methods train a generative model in the latent space of a learned autoencoder (Zhang et al., 2023a; Sanghi et al., 2024; Zhang et al., 2024), as we do. Conditional 3D generation either distills 2D diffusion models through score distillation (Poole et al., 2022; Wang et al., 2023) or trains conditional models on paired or zero-shot data (Nichol et al., 2022a; Xu et al., 2024; Sanghi et al., 2022; Siddiqui et al., 2024). Most of these models support a single conditioning modality, or train a separate model per condition. We discuss this literature in more detail in Appendix B.

Multimodal generation and 3D. In the 2D image–text setting, multimodal generation is typically achieved by adding control branches or adapters to pretrained text-to-image models (Zhang et al., 2023b; Ye et al., 2023), or by modeling interleaved text and image tokens with a single autoregressive backbone (Chameleon Team, 2024; Wu et al., 2024a). In 3D, CLAY (Zhang et al., 2024) adds separate adapters for conditions such as voxels, bounding boxes, and point clouds. Hunyuan3D-Omni (Team Hunyuan3D, 2025), a non-published technical report, extends an image-to-3D model with a control encoder for point clouds, voxels, bounding boxes, and skeletons. An image is always required, and the other signals act only as additional controls, so it cannot generate a shape from, for example, a point cloud or text alone. ShapeLLM-Omni (Ye et al., 2025) casts 3D generation and understanding as joint autoregressive modeling over text, image, and 3D tokens, which requires a large model to learn the full joint distribution. In contrast, our model treats every modality and pair of modalities uniformly through one pretrained context encoder. Text alone, a sketch, a part, or a bounding box with text are all equally valid inputs, and no per-condition modules are needed.

Conditioning encoders for generation. Modern generators usually condition on a frozen pretrained encoder through cross-attention, such as CLIP or T5 text encoders in latent diffusion models (Rombach et al., 2022; Esser et al., 2024). Imagen (Saharia et al., 2022) showed that scaling the frozen text encoder improves generation more than scaling the diffusion model. These encoders are pretrained on a single modality with objectives unrelated to the generation target. Our encoder follows the same encoder–decoder pattern but is pretrained specifically for the generator. It covers all input modalities and their combinations, and its pretraining includes the target 3D shape tokens. REPA (Yu et al., 2025) likewise shows that generation quality depends strongly on representation quality, by aligning diffusion features with a frozen self-supervised vision encoder. In 3D, large-scale encoders such as Point-BERT (Yu et al., 2022) and Uni3D (Zhou et al., 2024) exist, but they operate on point clouds and are not designed to encode the diverse, mixed-modality contexts a multimodal generator is conditioned on.

Masked pretraining and flow matching. Masked token prediction is a standard self-supervised objective for text (Devlin et al., 2019) and images (He et al., 2022). MultiMAE (Bachmann et al., 2022) and 4M (Mizrahi et al., 2023) extend it to several modalities by masking tokens across all of them. Our cross-modal masking instead keeps one modality fully visible and masks the other heavily, up to entirely. This turns masked prediction into explicit cross-modal generation, with the 3D shape tokens as one of the modalities. For the decoder, we use flow matching and rectified flow (Lipman et al., 2023; Liu et al., 2023c). They learn a velocity field along (near-)straight paths from noise to data, and are now standard for large image generators (Esser et al., 2024) and 3D shape generation (Li et al., 2025).

3 METHOD

3.1 OVERVIEW

Figure 2 summarizes our pipeline. The model is an encoder–decoder that separates *understanding* the context from *generating* the shape. All conditioning inputs and the target 3D asset are tokenized into a shared vocabulary (Section 3.2). A large transformer context encoder p_θ is pre-trained on these tokens with cross-modal masking (Section 3.3) and then frozen. A lightweight rectified-flow decoder v_ϕ generates the 3D latent over a short ODE trajectory while attending to the encoder's representation (Section 3.4).

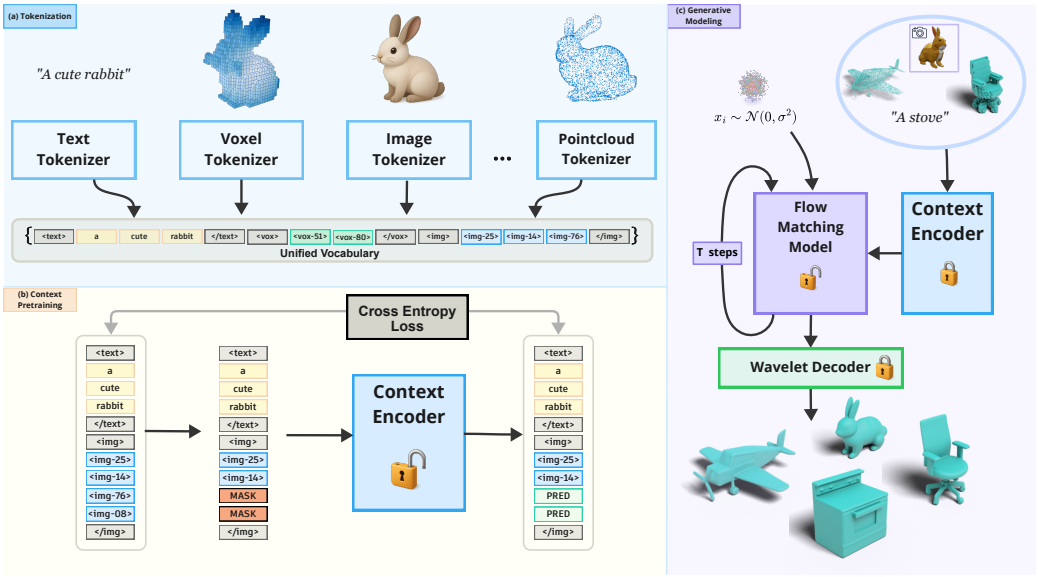

Figure 2: Overview of 3Dify-Anything. (a) Modality-specific tokenizers map every input into a unified vocabulary. (b) The context encoder is pre-trained with masked token prediction. In this cross-modal example, the text stays fully visible while image tokens are masked and predicted. (c) The frozen encoder reads the context once, and a rectified-flow model integrates T steps to produce the 3D latent, which the frozen wavelet decoder turns into a shape.

3.2 Tokenization

We represent every modality as a sequence of discrete tokens. This gives the model a single *uniform language*, supports single- and multi-modal contexts naturally, and avoids mixing features of different dimensionalities.

3D assets. Each asset is converted from a 256^3 TSDF into a wavelet tree (Hui et al., 2024), a 46^3 grid that is invertible back to the distance field. Following WaLa (Sanghi et al., 2024), we compress it with an autoencoder quantized by FSQ (Mentzer et al., 2023), giving a latent $Z \in \mathbb{R}^{8^3 \times 6}$ with a discrete index per position (64,000 codes). The discrete indices serve as *3D shape tokens* during context pre-training, and the continuous latent Z is the target of the decoder.

Context modalities. We tokenize each modality as follows (details in Appendix D):

1. *Images* (single-/multi-view, depth, sketch, masked): TA-TiTok (Kim et al., 2025), which represents each view with high fidelity using 128 tokens (vocabulary 8192). Masked images, with a random square region removed, enable image-based shape completion.
2. *Point clouds and part point clouds*: the Cube tokenizer (Bhat et al., 2025), 512 tokens (vocabulary 16,384). Parts are local chunks of the object, used for shape completion.
3. *Voxels* (16^3): our ResNet-based FSQ autoencoder, 64 tokens, vocabulary 15,360.
4. *Text*: GPT-2 BPE (Radford et al., 2019), vocabulary 50,257.
5. *Bounding box*: the axis-aligned box of the point cloud, serialized as numbers and tokenized as text.

Unified vocabulary. We merge all tokenizers into one vocabulary of about 154K entries: the 50,257 GPT-2 text tokens plus a disjoint index range per discrete tokenizer (image 8,192, point cloud 16,384, voxel 15,360, and 3D shape 64,000 codes). A code k from modality m maps to $o_m + k$, where o_m is the offset of its range, so image code 25 and voxel code 25 become distinct tokens (`<img-25>` vs. `<vox-25>`). Special tokens open and close each modality segment (e.g., `<text>...</text>`), and `[MASK]` and `[PAD]` are added. The encoder thus reads one token stream in which the modality of every token is explicit.

Context tasks. We define ten base contexts: image, depth, sketch, masked image, voxels, point cloud, part point cloud, text, bounding box, and 3D shape tokens. Pre-training uses every base context alone and every ordered pair, giving 100 context tasks. For example, the *text+image* task serializes a caption and a rendered view of the same object as <text> a cute rabbit </text> <img> $t_1 \dots t_{128}$ </img> (Figure 2a). The *voxel* task contains only the object's 64 voxel tokens between <vox> and </vox>. During pre-training, cross-modal masking might hide most of the image tokens in the first sequence, so that the encoder must predict them from the caption.

3.3 Context Pre-training with Cross-Modal Masking

We pre-train a bidirectional transformer encoder p_θ with masked token prediction (Devlin et al., 2019). For each sequence n in a batch of size B, a set of positions $\mathcal{M}^{(n)}$ is corrupted as in BERT, with random replacements drawn from the same modality's vocabulary. The loss is the per-sequence averaged cross-entropy:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{B} \sum_{n=1}^{B} \frac{1}{|\mathcal{M}^{(n)}|} \sum_{j \in \mathcal{M}^{(n)}} -\log p_\theta\big(t_j^{(n)} \mid \mathbf{s}^{(n)}\big), \tag{1}$$

where $t_j^{(n)}$ is the original token and $\mathbf{s}^{(n)}$ the corrupted sequence.

Cross-modal masking. With standard masking, most masked tokens can be recovered from unmasked neighbours of the *same* modality, so the encoder can reach a low loss while treating each modality independently. For multi-modal contexts we therefore change which tokens are hidden. Each token j is masked independently with probability π_j. Single-modality contexts use standard masking with one rate per sequence, $\pi_j = r_n \sim \mathcal{U}[0.2, 0.5]$. For a pair $\mathbf{c} = [\mathbf{c}_A; \mathbf{c}_B]$ we pick a target modality uniformly at random, say A, and set

$$\pi_j = \begin{cases} r_\times \sim \mathcal{U}[0.3, 1.0] & j \in A, \\ 0 & j \in B. \end{cases} \tag{2}$$

Modality B stays fully visible, so every prediction must use it. As $r_\times$ approaches 1, the task becomes *generating* one modality from the other, such as image tokens from a caption or 3D shape tokens from a sketch. Because the 3D shape tokens are one of the base contexts, this ties every conditioning signal to the geometry the decoder will generate. Section 4.2 shows that this objective gives better representations than standard masking, especially for multi-modal inputs. Encoder architecture and pre-training data are detailed in Appendix E.

3.4 Generative Modelling with Rectified Flow

We train the decoder with rectified flow (Liu et al., 2023c; Lipman et al., 2023). Given a latent Z and noise $\epsilon \sim \mathcal{N}(0, I)$, we define $Z_t = (1-t)\, Z + t\, \epsilon$ and regress the velocity of this path:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{n,t,\epsilon}\Big[\big\| v_\phi\big(Z_t, t, h(\mathbf{c}_n)\big) - (\epsilon - Z)\big\|_2^2\Big], \tag{3}$$

where $h(\mathbf{c}_n)$ is the conditioning from the frozen encoder. Following Esser et al. (2024), t is drawn from a shifted logit-normal distribution. At inference we integrate the ODE with a Heun solver (Karras et al., 2022) in only 5–18 steps, using classifier-free guidance (Ho & Salimans, 2022) on the velocity. The number of steps and the guidance scale are set per input modality (Appendix E).

Conditioning. Different modalities are best represented at different encoder depths (Table 18). Rather than choose one layer, we tap nine layers $\{l_1, \dots, l_9\}$ spanning the encoder, normalize each, and fuse them with a learned linear map:

$$F = W\,\big[\mathrm{LN}_1(H^{(l_1)}) \,\|\, \cdots \,\|\, \mathrm{LN}_9(H^{(l_9)})\big] + b, \tag{4}$$

initialized to the average of the layers. The fused tokens are read through cross-attention in every decoder block, and a pooled summary modulates the blocks through adaLN.

Decoder. The decoder is a DiT-style transformer (Peebles & Xie, 2023) over the $8^3 = 512$ latent positions, with 3D rotary embeddings and shared adaLN (Chen et al., 2023). It is trained on 22 single- and paired-modality conditioning tasks, with rotations applied consistently to the target and geometric contexts. Architecture details and the task list are in Appendix E.

4 EXPERIMENTS

Models. All models share the same frozen context encoder (32 blocks, hidden size 1024, 407M parameters), pre-trained with cross-modal masking for 1M iterations. We train three rectified-flow decoders of increasing size on the same task mixture: **Ours-S** (width 448, 12 blocks, 65M parameters), **Ours-M** (width 1024, 32 blocks, 574M), and **Ours-L** (width 1536, 40 blocks, 1.56B). This lets us evaluate both the efficiency of a lightweight decoder and the effect of decoder capacity. For the benchmarks, we sample with a Heun solver, using per-modality steps and guidance scales (Appendix E). Training details and data are given in Appendix E.

Evaluation. We evaluate on three datasets: TOYS4K (Stojanov et al., 2021), GSO (Downs et al., 2022), and the MAS validation split (Hui et al., 2024). TOYS4K and GSO are out-of-distribution, evaluation-only datasets, allowing us to assess generalization beyond the training distribution. We compare against models that accept the same input as each benchmark: PointE (Nichol et al., 2022b), ShapE (Jun & Nichol, 2023), OpenLRM (Hong et al., 2023; He & Wang, 2023), TripoSR (Tochilkin et al., 2024), MeshAnything (Chen et al., 2024b), XCube (Ren et al., 2024), Make-A-Shape (Hui et al., 2024), WaLa (Sanghi et al., 2024), Trellis (Xiang et al., 2025b), and ShapeLLM-Omni (Ye et al., 2025). Hunyuan3D-Omni (Team Hunyuan3D, 2025), a non-published work, always requires an input image, so it does not apply to our point cloud, voxel, and text benchmarks. MAS validation aggregates data from 19 source datasets, covering diverse object categories and shapes. For geometric reconstruction, we report IoU and Chamfer distance for point cloud $\rightarrow$ 3D and voxel $\rightarrow$ 3D. Since alignment is challenging for image- and text-conditioned generation, we follow Trellis (Xiang et al., 2025b) and Direct3D (Wu et al., 2024b). We compute CLIP scores between the input image or text and eight rendered views of the predicted object. We also report ULIP2 (Xue et al., 2024) and UNI3D (Zhou et al., 2024) scores, following Wu et al. (2024b), for image-to-3D and text-to-3D. These measure cosine similarity between text/image and 3D shape embeddings.

4.1 QUANTITATIVE RESULTS

Image-to-3D. Table 1 compares our models with specialized image-to-3D baselines and the multimodal ShapeLLM-Omni (Ye et al., 2025) on TOYS4K; MAS results are in Appendix C. Our models support many input types in a single network. They also need only a single encoder pass followed by a short ODE integration with a small decoder, which keeps inference fast (0.56 s for Ours-S and 1.47 s for Ours-L, versus 31.2 s for ShapeLLM-Omni and 4.9 s for Trellis). The 407M-parameter encoder accounts for only 33 ms of this, because it runs once (Appendix E). Ours-M and Ours-L outperform every baseline except Trellis, and are far ahead of the multimodal ShapeLLM-Omni (ULIP2 0.4031 vs. 0.3379). Trellis, a specialized image-to-3D model, remains best on all three metrics, but is $3.3\times$ slower than Ours-L.

Text-to-3D. Table 2 reports text-to-3D results on TOYS4K. Ours-M and Ours-L outperform the multimodal ShapeLLM-Omni on all six metrics. They are on par with Trellis, a specialized text-to-3D model, on TOYS4K. Ours-L has the highest CLIP score (0.2418) and UNI3D score (0.2941), and its ULIP2 score is within 0.0001 of Trellis. On MAS (Appendix C), our models match Trellis on UNI3D but trail it on CLIP and ULIP2. Performance saturates from Ours-M to Ours-L.

Point-Cloud-to-3D Reconstruction. We report point-cloud-to-3D reconstruction results on MAS and GSO in Table 3. All three of our models achieve higher IoU than every specialized reconstruction baseline on both datasets, including WaLa (Sanghi et al., 2024). Ours-L raises IoU from 0.8625 to 0.9413 on MAS and from 0.9389 to 0.9650 on GSO. It also lowers Chamfer distance on MAS and matches WaLa on GSO.

Voxel-to-3D Reconstruction. Table 4 reports voxel-to-3D results from low-resolution 16^3 inputs. Ours-M and Ours-L achieve the highest IoU on both datasets. They outperform WaLa, which

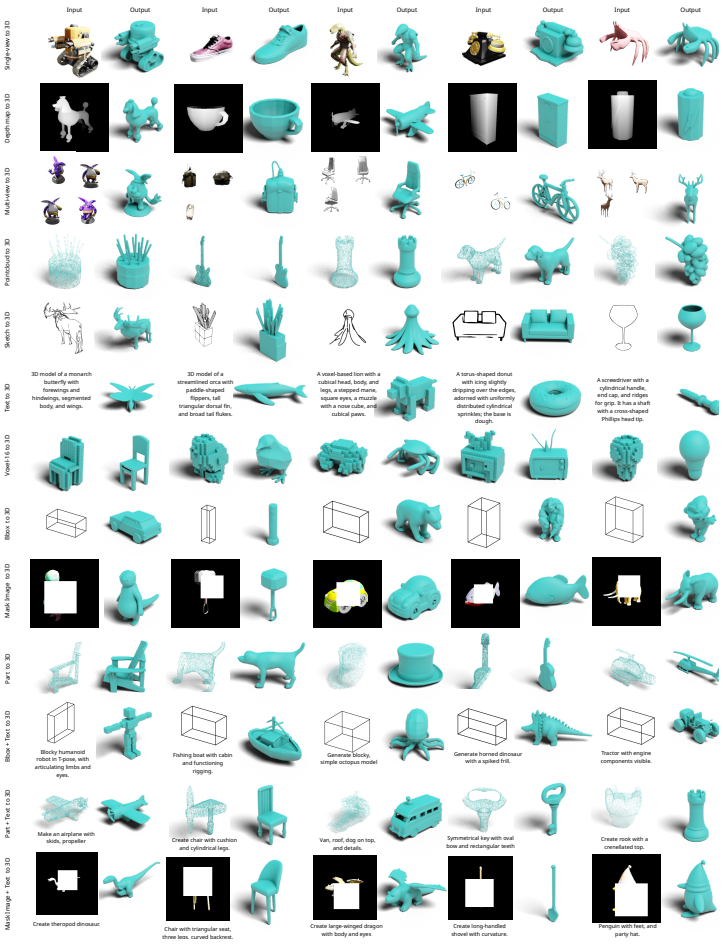

Figure 3: Generation from single- and multi-modal inputs with one model. Rows: single-view image, depth, multi-view images, point cloud, sketch, text, 16^3 voxels, bounding box, masked image, part point cloud, bbox+text, part+text, and masked image+text.

Table 1: Image-to-3D results on TOYS4K. Time is the mean over 10 samples on one H100, from input image to generated shape.

Method	Time↓	CLIP↑	ULIP2↑	UNI3D↑
Single-modal models				
PointE	9.82s	0.6630	0.2484	0.1950
ShapE	3.20s	0.6828	0.2780	0.2254
OpenLRM	0.07s	0.6757	0.2265	0.2020
Make-A-Shape	2.71s	0.7351	0.3768	0.3269
WaLa	0.83s	0.7584	0.3996	0.3467
TripoSR	0.16s	0.6912	0.2478	0.2169
Trellis	4.91s	0.7872	0.4205	0.3666
Multi-modal models				
ShapeLLM-Omni	31.17s	0.7286	0.3379	0.3117
Ours-S	0.56s	0.7497	0.3862	0.3353
Ours-M	1.16s	0.7678	0.4026	0.3508
Ours-L	1.47s	0.7684	0.4031	0.3520

Table 2: Text-to-3D results on TOYS4K.

Method	CLIP↑	ULIP2↑	UNI3D↑
Single-modal models			
PointE	0.1870	0.1858	0.1734
ShapE	0.1921	0.2148	0.1941
XCube	0.2054	0.2427	0.2198
Trellis	0.2412	0.3156	0.2884
Multi-modal models			
ShapeLLM-Omni	0.2234	0.2899	0.2679
Ours-S	0.2220	0.2823	0.2663
Ours-M	0.2411	0.3155	0.2927
Ours-L	0.2418	0.3155	0.2941

uses the same input resolution, on both IoU and Chamfer distance. They also exceed the IoU of ShapeLLM-Omni (Ye et al., 2025) (MAS: 0.8080 vs. 0.7004), even though it receives $64\times$ more voxels (64^3), while ShapeLLM-Omni has a lower Chamfer distance.

Table 3: Point-cloud-to-3D results on MAS and GSO. CD is reported $\times 10^3$.

	MAS		GSO	
Method	IoU↑	CD↓	IoU↑	CD↓
Single-modal models				
Point-E SDF model	0.4899	15.8	0.6006	3.7
MeshAnything	0.3378	9.1	0.3731	6.4
Make-A-Shape	0.7595	3.6	0.7769	1.9
WaLa	0.8625	1.4	0.9389	1.1
Multi-modal models				
Ours-S	0.9095	1.156	0.9425	1.169
Ours-M	0.9379	1.095	0.9631	1.085
Ours-L	0.9413	1.087	0.9650	1.082

Table 4: Voxel-to-3D results on MAS and GSO. CD is reported $\times 10^3$.

		MAS		GSO	
Method	Input Res.	IoU↑	CD↓	IoU↑	CD↓
Single-modal models					
NN-Interpolation	16^3	0.2500	30.6	0.3790	23.9
Trilinear Interpolation	16^3	0.2596	31.9	0.3899	25.0
Make-A-Shape	16^3	0.6459	3.8	0.7602	3.1
WaLa	16^3	0.6929	2.9	0.8048	2.3
Multi-modal models					
ShapeLLM-Omni	64^3	0.7004	2.2	0.8341	1.9
Ours-S	16^3	0.7361	3.134	0.8014	2.883
Ours-M	16^3	0.8040	2.277	0.8410	2.224
Ours-L	16^3	0.8080	2.312	0.8484	2.174

Table 5: Multimodal voxel-to-3D results on TOYS4K with Ours-L. CD is reported $\times 10^3$.

Context	IoU↑	CD↓
voxel 16^3	0.8215	2.000
voxel 16^3 + text	0.8255	1.907
voxel 16^3 + image	**0.8518**	**1.674**

Multimodal results. Figure 3 shows qualitative results for all 13 single- and multi-modal input types handled by one model. Table 5 tests whether combining context signals helps, using 16^3 voxel inputs on TOYS4K with Ours-L. Adding a text prompt improves Chamfer distance ($2.0 \rightarrow 1.907$) and slightly improves IoU. Adding an image improves both substantially (IoU $0.8215 \rightarrow 0.8518$, CD $2.0 \rightarrow 1.674$), as the image resolves details that coarse voxels cannot. Cross-modal pre-training is designed for this setting: each two-modality pre-training sequence asks the encoder to explain one modality in terms of the other.

4.2 ABLATION STUDIES

Probing protocol. To compare pre-training objectives without training a generator for each, we probe the frozen encoder on ModelNet40 (Wu et al., 2015). We run K-means ($K = 40$) on pooled features of single- and two-modality contexts, and report adjusted mutual information (AMI) (Vinh et al., 2010) and Hungarian-matched accuracy. All encoders share the same architecture and data and are compared at 300k iterations. Details are in Appendix E.

Which pre-training objective? Table 7 compares four masking schemes. *Fixed* masks every token at rate 0.3, and *Uniform* samples one rate per sequence, $r \sim \mathcal{U}[0.2, 0.5]$. The two cross-modal variants (Eq. 2) keep one modality of a pair fully visible and mask the other. *Cross-modal, low* masks the target at $r_\times \sim \mathcal{U}[0.2, 0.5]$, the same range as Uniform, and *Cross-modal, high* (ours) masks it at $r_\times \sim \mathcal{U}[0.3, 1.0]$.

Sampling the rate already helps: Uniform improves over Fixed on five of the seven inputs, including text and all voxel and point-cloud inputs. Changing *which* tokens are masked helps much more. With the same rate range as Uniform, Cross-modal, low improves every input that contains an image, point cloud, or voxel, and the largest gains are on multi-modal inputs. For example, image+text AMI rises from 0.355 to 0.492 and voxel+text from 0.587 to 0.665; only text alone drops at this rate. Masking more aggressively, so that the encoder must sometimes generate the entire target modality from the other, gives the best AMI on *every* input, including text. Compared with Uniform, Cross-modal, high raises image+text AMI from 0.355 to 0.592 ($+67\%$ relative) and voxel+text from 0.587 to 0.728. For each multi-modal context, the pair is also clustered better than either of its single-modality components, which shows that the encoder combines the two modalities rather than relying on one of them. The same ranking holds for generation (Table 6, bottom). With an identical decoder, the cross-modal encoder outperforms the fixed and uniform encoders on image, voxel, text, and depth inputs, and is within 0.004 IoU of the best on point clouds. We therefore use cross-modal masking with $r_\times \sim \mathcal{U}[0.3, 1.0]$ for all generative models.

Table 6: Context training regime (top) and pre-training objective and conditioning layers (bottom) on 250 Toys4K objects at 100k iterations. Decoder parameters include all non-text conditioning modules; single-layer variants have no layer-fusion parameters; encoder parameters exclude token embeddings. Time is for 512 tokens. CD is reported $\times 10^3$.

				Image→3D			Voxel→3D		Text→3D			Point cloud→3D		Depth→3D		
Model type	Enc. params	Dec. params	Time	ULIP2↑	Uni3D↑	CLIP↑	IoU↑	CD↓	ULIP2↑	Uni3D↑	CLIP↑	IoU↑	CD↓	ULIP2↑	Uni3D↑	CLIP↑
Decoder only	–	66.4M	875 ms	0.244	0.213	0.675	0.713	3.57	0.185	0.192	0.185	0.829	1.59	0.265	0.237	0.683
Decoder only (big)	–	472.2M	1197 ms	0.267	0.238	0.688	0.754	2.73	0.201	0.207	**0.195**	0.882	1.21	0.288	0.254	0.697
Dec. + ctx encoder	407.3M	46.0M	672 ms	0.250	0.223	0.677	0.729	3.18	0.163	0.174	0.176	0.843	1.47	0.274	0.243	0.685
Pre-train: fixed	407.3M	64.9M	660 ms	0.334	0.290	0.711	0.739	2.99	0.186	0.199	0.185	0.904	1.36	0.349	0.302	0.717
Pre-train: uniform	407.3M	64.9M	652 ms	0.340	0.295	0.715	0.740	2.88	0.187	0.197	0.182	**0.914**	**1.03**	0.352	0.304	0.719
Pre-train: cross, layer 15	407.3M	46.0M	660 ms	0.348	0.302	0.718	0.746	2.72	0.201	0.202	0.188	0.894	1.10	0.361	0.316	0.726
Pre-train: cross, concat (ours)	407.3M	64.9M	654 ms	**0.364**	**0.315**	**0.730**	**0.755**	**2.65**	**0.209**	**0.212**	0.190	0.910	1.04	**0.377**	**0.325**	**0.735**

Table 7: Context pre-training objective. Zero-shot K-means clustering of the frozen encoder on ModelNet40 (best layer and pooling per input). Each entry reports AMI / accuracy; best value of each metric in **bold**. Masking variants are defined in Table 12; all encoders at 300k iterations.

Masking	**Text**	**Image**	**Image+Text**	**PC**	**PC+Text**	**Voxel**	**Voxel+Text**
Fixed	0.511 / 0.429	0.367 / 0.304	0.367 / 0.310	0.625 / 0.489	0.621 / 0.485	0.585 / 0.468	0.570 / 0.459
Uniform	0.523 / 0.420	0.355 / 0.291	0.355 / 0.299	0.628 / 0.499	0.630 / 0.510	0.586 / 0.464	0.587 / 0.462
Cross-modal, low	0.421 / 0.352	0.400 / 0.336	0.492 / 0.395	0.666 / **0.577**	0.675 / 0.570	0.593 / 0.462	0.665 / 0.578
Cross-modal, high	**0.531 / 0.431**	**0.437 / 0.353**	**0.592 / 0.510**	**0.695** / 0.569	**0.705 / 0.620**	**0.631 / 0.542**	**0.728 / 0.625**

Which encoder layer to condition on? Repeating the probe layer by layer (Table 18, Appendix) shows that no single layer is best for every modality. In AMI, image and voxel features peak at layer 15 and point clouds at layer 19, while text is best at the final layer, where image and voxel AMI have dropped substantially ($0.437 \rightarrow 0.314$ and $0.631 \rightarrow 0.457$). Conditioning on any one layer would therefore favour some modalities over others, which motivates fusing all tapped layers (Eq. 4). This fusion outperforms conditioning on layer 15, the best single layer in the probe, on every generation metric (Table 6).

Do we need context pre-training? Table 6 compares four setups trained with the same flow objective, data, task mixture, and (because of compute constraints) budget of 100k iterations. *Decoder only* embeds the context tokens with a learned table and processes them directly in the flow decoder (width 448, 18 blocks). *Decoder only (big)* scales this to width 1024 and 27 blocks, matching the parameter count of our encoder plus decoder. *Dec. + ctx encoder* trains our encoder architecture from scratch jointly with the decoder. *Pre-train: cross, concat* is ours (Ours-S), with the frozen encoder pre-trained with cross-modal masking.

In the top block, context pre-training gives the best results on almost every task, with large margins over the decoder-only baseline. For example, image→3D ULIP2 rises from 0.244 to 0.364 ($+49\%$), depth→3D from 0.265 to 0.377, and point cloud→3D IoU from 0.829 to 0.910, with a 35% lower Chamfer distance. Capacity alone does not explain these gains. The parameter-matched decoder-only model remains well behind on image, depth, and point cloud inputs. It comes close only on voxel IoU and is better only on text CLIP score, while ours uses $7\times$ fewer decoder parameters and is $1.8\times$ faster. Adding the same encoder without pre-training helps little over decoder only, and it is worse on text. The benefit therefore comes from context pre-training rather than from the encoder's architecture or size.

5 Conclusion

We present *3Dify-Anything*, a single model that generates 3D assets from diverse inputs and their combinations. A large context encoder runs once and a lightweight rectified-flow decoder runs at every step, and context pretraining with cross-modal masking teaches the encoder to predict one modality from another. The result is a flexible multimodal generator that is competitive with specialized models. Our ablations show that the gains come from context pretraining rather than from encoder size, and that a small decoder suffices for near real-time generation. Future work includes extending context pretraining to further modalities, such as video, and to appearance generation.

AI USE STATEMENT

Large language models (LLMs) were used in the following ways in preparing this paper. **Drafting sections:** LLMs drafted parts of the text, including the abstract, introduction, method, experimental analysis, related work, and appendix, from the authors' notes, code, and experimental results; the authors revised all drafted text. **Aiding and polishing writing:** LLMs edited and condensed the text, improved clarity, and adapted the paper to the ICLR 2027 template and page limit. **Retrieval and discovery:** LLMs helped identify and check related work; the authors read the cited works and verified every citation. LLMs also helped transfer numbers from the authors' experiment logs into tables and compute derived quantities such as parameter counts and relative improvements. All research ideas, experiments, and results are the authors' own. The authors verified all content, citations, and numbers and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

Implementation and training details, tokenizer specifications, datasets, and additional evaluation details are provided in the main paper and appendix.

REFERENCES

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision (ECCV)*, 2022.

Raphael Bensadoun, Tom Monnier, Yanir Kleiman, Filippos Kokkinos, Yawar Siddiqui, Mahendra Kariya, Omri Harosh, Roman Shapovalov, Benjamin Graham, Emilien Garreau, et al. Meta 3d gen. *arXiv preprint arXiv:2407.02599*, 2024.

Kiran Bhat, Nishchaie Khanna, Karun Channa, Tinghui Zhou, Yiheng Zhu, Xiaoxia Sun, Charles Shang, Anirudh Sudarshan, Maurice Chu, Daiqing Li, et al. Cube: A roblox view of 3d intelligence. *arXiv e-prints*, pp. arXiv–2503, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision (ECCV)*, 2024a.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024b.

An-Chieh Cheng, Xueting Li, Sifei Liu, Min Sun, and Ming-Hsuan Yang. Autoregressive 3d shape generation via canonical mapping. In *European Conference on Computer Vision*, pp. 89–104. Springer, 2022.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4456–4465, 2023.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21126–21136, 2022.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023a.

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

Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20637–20647, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, pp. 2553–2560, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129: 3313–3337, 2021.

Chenjian Gao, Qian Yu, Lu Sheng, Yi-Zhe Song, and Dong Xu. Sketchsampler: Sketch-based 3d reconstruction via view-dependent depth sampling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pp. 464–479. Springer, 2022a.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022b.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3D shape generation. In *ACM SIGGRAPH Asia*, pp. 1–9, 2022.

Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. In *ICML*, 2024.

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 867–876, 2022.

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

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. *arXiv preprint arXiv:2501.07730*, 2025.

Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *European Conference on Computer Vision*, pp. 694–710. Springer, 2020.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9601–9611, 2019.

Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12642–12651, 2023.

Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12706–12716, 2021.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023a.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309, 2023b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023c.

Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021.

Zhengzhe Liu, Peng Dai, Ruihui Li, Xiaojuan Qi, and Chi-Wing Fu. Iss: Image as stetting stone for text-guided 3d shape generation. *arXiv preprint arXiv:2209.04145*, 2022.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pp. 67–77. IEEE, 2017.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8446–8455, 2023.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13492–13502, 2022.

Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 306–315, 2022.

David Mizrahi, Roman Bachmann, Oğuzhan Fatih Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020.

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

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pp. 725–741, 2018.

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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

Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. CLIP-Forge: Towards zero-shot text-to-shape generation. In *CVPR*, pp. 18603–18613, 2022.

Aditya Sanghi, Rao Fu, Vivian Liu, Karl DD Willis, Hooman Shayani, Amir H Khasahmadi, Srinath Sridhar, and Daniel Ritchie. Clip-sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18339–18348, 2023.

Aditya Sanghi, Aliasghar Khani, Pradyumna Reddy, Arianna Rampini, Derek Cheung, Kamal Rahimi Malekshan, Kanika Madan, and Hooman Shayani. Wavelet latent diffusion (wala): Billion-parameter 3d generative model with compact wavelet encodings. *arXiv preprint arXiv:2411.08017*, 2024.

Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10397–10407, 2021.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials. *arXiv preprint arXiv:2407.02445*, 2024.

Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1798–1808, 2021.

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

Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 61–70, 2020.

Team Hunyuan3D. Hunyuan3d-omni: A unified framework for controllable generation of 3d assets. *arXiv preprint arXiv:2509.21245*, 2025.

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

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023.

Karl DD Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4): 1–24, 2021.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024a.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems*, 37:121859–121881, 2024b.

Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. *arXiv preprint arXiv:2512.14692*, 2025a.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21469–21480, 2025b.

Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360 views. *arXiv e-prints*, pp. arXiv–2211, 2022.

Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20908–20918, 2023.

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

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6239–6249, 2022.

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X. Chang. An object is worth 64x64 pixels: Generating 3d object via image diffusion, 2024. URL https://arxiv.org/abs/2408.03178.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4541–4550, 2019.

Lior Yariv, Omri Puny, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4630–4639, 2024.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Junliang Ye, Zhengyi Wang, Ruowen Zhao, Shenghao Xie, and Jun Zhu. Shapellm-omni: A native multimodal llm for 3d generation and understanding. *arXiv preprint arXiv:2506.01853*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. *Advances in Neural Information Processing Systems*, 35:21871–21885, 2022.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *arXiv preprint arXiv:2301.11445*, 2023a.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023b.

Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *arXiv preprint arXiv:2305.04461*, 2023.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

A Additional Experimental Results

A.1 Qualitative Results

Figure 5 shows qualitative results across diverse single- and multi-modal inputs. The model faithfully reconstructs geometry from strong cues such as point clouds and voxels, while generating plausible, semantically aligned shapes from weaker signals including text, sketches, depth maps, and bounding boxes. Multi-modal conditioning improves disambiguation by combining coarse geometry with semantics, sparse outlines with text, or masked images with text to complete missing regions while preserving object identity. Figure 6 further demonstrates controllability: the same text prompt can be steered by visual or geometric context, such as part point clouds, bounding boxes, or reference renderings, to adjust shape, scale, and proportions across categories. Overall, these results demonstrate robust cross-modal composition from partial visual, semantic, and geometric context.

B Extended related work

3D shape generation. There is a long line of work on 3D generative models, including autoregressive models (Nash et al., 2020; Cheng et al., 2022; Mittal et al., 2022; Zhang et al., 2022; Sun et al., 2020; Yan et al., 2022), GANs (Wu et al., 2016; Gao et al., 2022b), and likelihood-based approaches such as normalizing flows (Yang et al., 2019; Klokov et al., 2020) and diffusion models (Nichol et al., 2022a; Hui et al., 2024; Zhang et al., 2023a). Many current methods follow a two-stage pipeline, in which they first train a VQ-VAE and then train a generative model on the resulting latent space (Zhang et al., 2023a; Cheng et al., 2023; Sanghi et al., 2024; Zhang et al., 2024; Li et al., 2023), which helps scale training to high-dimensional 3D signals. In this work, we also train our generative model in the latent space; however, our framework is largely representation-agnostic and can, in principle, also be applied directly to the 3D representation (Nichol et al., 2022a; Zheng et al., 2023; Ren et al., 2024; Hui et al., 2024; 2022).

Conditional 3D shape generation. Most modern conditional 3D generative models can be broadly divided into two categories. The first line of work explores using the SDS (score distillation sampling) loss (Poole et al., 2022), which essentially backpropagates gradients from a frozen 2D text-to-image diffusion model (Rombach et al., 2022; Saharia et al., 2022) into a 3D representation so that its rendered views become consistent with the input text. This strategy has been adopted to optimize or generate 3D shapes directly (Jain et al., 2022; Wang et al., 2023; Deitke et al., 2023b; Deng et al., 2023; Melas-Kyriazi et al., 2023; Xu et al., 2022; Liu et al., 2023b; Qian et al., 2023; Shi et al., 2023; Michel et al., 2022; Liu et al., 2023a). The second category comprises conditional generative models, which are either trained with paired data (Lun et al., 2017; Gao et al., 2022a; Chen et al., 2024a; Zhang et al., 2022; Xu et al., 2024; Siddiqui et al., 2024; Bensadoun et al., 2024; Nichol et al., 2022a; Yariv et al., 2024) or in a zero-shot fashion (Yan et al., 2024; Liu et al., 2022; Xu et al., 2023; Sanghi et al., 2022; 2023). Many prior works are single-modality conditional generative models, or they train a separate model per conditioning signal; as a result, they can't handle multiple conditions within one model, limiting practical use. By contrast, this work studies a unified generative framework that maps diverse inputs—text, single-/multi-view images, sketches, depth maps, point clouds, voxels, and their combinations—into high-quality 3D assets by discretizing all modalities into a shared vocabulary and using context pretraining to learn aligned cross-modal representations.

C IMAGE- AND TEXT-TO-3D RESULTS ON MAS

Tables 8 and 9 report image-to-3D and text-to-3D results on the MAS validation set, complementing the TOYS4K results in the main paper. The trends match TOYS4K. For image-to-3D, Ours-M and Ours-L outperform every baseline except Trellis and are well ahead of the multimodal ShapeLLM-Omni. For text-to-3D, our models match Trellis on UNI3D but trail it on CLIP and ULIP2.

Table 8: Image-to-3D results on MAS.

Method	CLIP↑	ULIP2↑	UNI3D↑
Single-modal models			
PointE	0.6955	0.2389	0.1916
ShapE	0.7148	0.2643	0.2166
OpenLRM	0.7064	0.2118	0.1827
Make-A-Shape	0.7648	0.3526	0.2962
WaLa	0.7694	0.3604	0.3074
TripoSR	0.7187	0.2274	0.1920
Trellis	0.7932	0.3845	0.3295
Multi-modal models			
ShapeLLM-Omni	0.7433	0.2787	0.2392
Ours-S	0.7682	0.3595	0.3080
Ours-M	0.7786	0.3668	0.3185
Ours-L	0.7800	0.3680	0.3185

Table 9: Text-to-3D results on MAS.

Method	CLIP↑	ULIP2↑	UNI3D↑
Single-modal models			
PointE	0.1944	0.1768	0.1601
ShapE	0.1973	0.1868	0.1672
XCube	0.2152	0.2488	0.2287
Trellis	0.2450	0.2829	0.2572
Multi-modal models			
ShapeLLM-Omni	0.2340	0.2506	0.2372
Ours-S	0.2316	0.2495	0.2430
Ours-M	0.2425	0.2671	0.2574
Ours-L	0.2421	0.2664	0.2571

D TOKENIZERS

3D Asset Tokenizer. We construct our 3D tokenizer by first encoding 256^3 signed distance fields into a wavelet-tree representation (Hui et al., 2024), arranged as a 46^3 grid. We then train an autoencoder on this representation, following the architecture and balanced fine-tuning procedure of WaLa (Sanghi et al., 2024). WaLa is trained on Toys4K, a dataset widely used for evaluation, so we retrain our tokenizer from scratch while explicitly *excluding* Toys4K. We also increase capacity by using 6 instead of 4 ResBlocks in each downsampling and upsampling block. Finally, we replace the vector-quantization codebook with finite scalar quantization (FSQ) (Mentzer et al., 2023) for stability. The encoder downsamples to an 8^3 latent grid with 6 channels, quantized with levels $(8, 8, 8, 5, 5, 5)$. This gives 512 tokens per shape with an implicit codebook of $8^3 \cdot 5^3 = 64{,}000$ entries. The integer indices are used as *3D shape tokens* during context pre-training. The continuous post-quantization latent ($8^3 \times 6$) is the target of the rectified-flow decoder, and generated latents are decoded by the frozen decoder followed by the inverse wavelet transform. We validate on the MAS validation set, which includes Toys4K, following the protocol of WaLa (Sanghi et al., 2024). We report IoU and MSE, as well as D-IoU and D-MSE, which weight each MAS subset equally to mitigate data imbalance (Table 10). WaLa compresses each shape to a $12^3 \times 4$ latent, whereas ours uses only $8^3 \times 6$. That is $3.4\times$ fewer spatial positions (512 vs. 1728) and $2.25\times$ fewer latent values (3072 vs. 6912). The generative decoder therefore operates on a much shorter sequence. Despite this stronger compression, our tokenizer outperforms WaLa on IoU, MSE (15% lower), and D-IoU, and is comparable on D-MSE (0.00152 vs. 0.00148).

Table 10: **3D asset tokenizer comparison.** Reconstruction metrics on SDFs (IoU, MSE, D-IoU, D-MSE). Our latent grid is 8^3, versus 12^3 for WaLa.

Method	Latent	IoU ↑	MSE ↓	D-IoU ↑	D-MSE ↓
WaLa	$12^3 \times 4$	0.95979	0.00078	0.94820	**0.00148**
Ours (FSQ)	$\mathbf{8^3 \times 6}$	**0.96333**	**0.00066**	**0.94920**	0.00152

Voxel Tokenizer. Pretrained low-resolution voxel autoencoders trained on large-scale data are not available, so we train a ResNet-based autoencoder from scratch. It downsamples a 16^3 occupancy

grid to a 4^3 latent (64 tokens) and quantizes it with FSQ (Mentzer et al., 2023), using a vocabulary of 15,360. We use the same datasets as described below.

Image Tokenizer. We use the pretrained TA-TiTok (Kim et al., 2025) tokenizer (TiTok-BL-128), which quantizes each image into 128 tokens with a vocabulary of 8192. We use the same tokenizer for images, sketches, depth maps, masked images, and image crops. Image-like contexts contain one to four views, sampled with probabilities $0.7/0.1/0.1/0.1$, and each view is wrapped in its own block.

Point Cloud Tokenizer. We use the pretrained Roblox Cube shape tokenizer (Bhat et al., 2025) (v0.1), which tokenizes 8,192 oriented points into 512 tokens with a vocabulary of 16,384.

Part Point Clouds. Part point clouds let the model complete a shape from a local piece of it. For each object we take its dense oriented point cloud and generate 8 local chunks independently. Each chunk is grown around a random anchor point by selecting the nearest points under a randomly chosen distance: Euclidean, Chebyshev, cosine (angle around the centroid), or distance along a single random axis. The chunk size is a fraction of the object drawn from $\mathrm{Beta}(0.6, 0.6)$, which produces both small fragments and large portions of the shape. Chunks are sampled from the full object, so they may overlap. Each chunk is re-normalized to the unit cube, resampled to 8,192 points, and tokenized with the Cube tokenizer into 512 tokens.

Bounding Box. For coarse global control we encode the axis-aligned bounding box of the object. Its coordinates are rounded to three decimal places, serialized as a short string of numbers, and consumed through the GPT-2 text tokenizer. The box is rotated consistently with the other geometric modalities.

E Training details and datasets

Data. We train on roughly 4M 3D assets collected from the following sources (in alphabetical order): 3D-FUTURE (Fu et al., 2021), ABC (Koch et al., 2019), ABO (Collins et al., 2022), BuildingNet (Selvaraju et al., 2021), COMA (Ranjan et al., 2018), DeformingThings4D (Li et al., 2021), FG3D (Liu et al., 2021), Fusion 360 (Willis et al., 2021), House3D (Wu et al., 2018), Objaverse (Deitke et al., 2023b), two subsets of Objaverse-XL (Thingiverse and GitHub) (Deitke et al., 2023a), ShapeNet (Chang et al., 2015), SMAL (Zuffi et al., 2017), SMPL (Loper et al., 2015), and Thingi10K (Zhou & Jacobson, 2016). Toys4K and GSO are held out for evaluation. Geometric modalities (voxels, point clouds, bounding boxes, and 3D shape tokens) are precomputed under a set of rotations. When a context contains several geometric modalities, they share the same rotation.

Context encoder pre-training. The encoder is a bidirectional transformer with 32 blocks, hidden size 1024, 16 heads, and one register token, for 407M parameters excluding token embeddings. It uses RMSNorm (Zhang & Sennrich, 2019), SwiGLU (Shazeer, 2020), QK-normalization (Henry et al., 2020), rotary position embeddings (Su et al., 2024), and depth-scaled initialization. Contexts are truncated or padded to 1050 tokens. The data mixture is 80% 3D assets, 10% ImageNet-21K (Ridnik et al., 2021), and 10% Wikipedia. The ImageNet-21K samples are divided equally between image-only, text-only, and image+text tasks, where the text is a caption of the image. For 3D assets, we sample single-modality tasks with probability 0.4 and ordered pairs with probability 0.6 from the 100 tasks built from the 10 base contexts. Within a task size, each semantic modality group is equally likely. Table 11 lists the hyper-parameters, and Table 12 the masking variants of Table 7. All variants share every other setting.

Generative model training. All three decoders are trained with the frozen encoder, the rectified-flow objective of Eq. 3, and the same task mixture (Table 14). They share the following settings:

- *Conditioning*: nine tapped layers $\{0, 3, 7, 11, 15, 19, 23, 27, 31\}$ combined with concatenation fusion, packed context tokens, and shared adaLN.
- *Flow objective*: logit-normal time sampling with shift $s = 2$, and condition dropout 0.1 for classifier-free guidance.

Table 11: Context encoder pre-training hyper-parameters.

Architecture	32 blocks, width 1024, 16 heads, 1 register, RoPE
Tokenizer / context length	extended GPT-2 vocabulary, 1050 tokens
Optimizer	AdamW (Loshchilov & Hutter, 2019), lr 2×10^{-4}, weight decay 0.05
Batch size	1024 (32 per GPU $\times$ 32 H100 GPUs)
Iterations	1M
EMA	yes
Masking	cross-modal; $r \sim \mathcal{U}[0.2, 0.5]$ (single), $r_\times \sim \mathcal{U}[0.3, 1.0]$ (pairs)
BERT corruption	80% `[MASK]`, 10% same-modality random, 10% unchanged

Table 12: Masking variants compared in Table 7. r is the rate for single-modality sequences (and for both modalities of a pair when cross-modal masking is off). $r_\times$ is the rate of the target modality of a pair; the other modality is left fully visible.

Variant	Rate r	Cross-modal	Target rate $r_\times$
Fixed	0.3	no	–
Uniform	$\mathcal{U}[0.2, 0.5]$	no	–
Cross-modal, low	$\mathcal{U}[0.2, 0.5]$	yes	$\mathcal{U}[0.2, 0.5]$
Cross-modal, high (ours)	$\mathcal{U}[0.2, 0.5]$	yes	$\mathcal{U}[0.3, 1.0]$

- *Optimization*: AdamW (Loshchilov & Hutter, 2019) with $\beta_2 = 0.95$ and weight decay 0.01, 5k warm-up steps, gradient clipping at norm 1.0, and an EMA of the weights with decay 0.9999.
- *Precision*: bf16.

Latents are used without normalization. Table 13 lists the per-model settings. The global batch size is 256 for all three models, and we scale the learning rate down as the decoder grows.

Table 13: Decoder configurations. All models use the same frozen context encoder (407M parameters) and global batch size 256. Decoder parameters include the conditioning (projection and layer-fusion) modules, about 24–29M for each model. Ours-S and Ours-M are trained on 8 H100 GPUs and Ours-L on 8 B200 GPUs.

Model	Width	Blocks	Heads	Batch (per GPU $\times$ GPUs)	Learning rate	Decoder params
Ours-S	448	12	7	32×8	2.3×10^{-4}	64.9M
Ours-M	1024	32	16	32×8	1.0×10^{-4}	574.1M
Ours-L	1536	40	24	32×8	6.7×10^{-5}	1.56B

Unified vocabulary. We extend the GPT-2 tokenizer with a dedicated index range for each modality tokenizer (image, point cloud, voxel, and 3D shape), so the same local index from two tokenizers maps to distinct vocabulary IDs. We add modality-specific start/end tokens (e.g., `<text>`, `<pointcloud_dis>`, `<crop_image>`), together with `[PAD]` and `[MASK]`. Sequences are right-padded to 1050 tokens, and an attention mask excludes padding. Masking follows BERT: 80% of selected tokens become `[MASK]`, 10% a random token from the same modality, and 10% are unchanged. Structural tokens are never masked, and at least one token of the target modality is always masked.

Decoder architecture. The decoder operates directly on the $8^3 = 512$ latent positions (patch size 1), encoded with 3D axial rotary embeddings. Each block applies adaLN-modulated self-attention, cross-attention to the context tokens, and a SwiGLU MLP (Shazeer, 2020). It uses RMSNorm (Zhang & Sennrich, 2019) and QK-normalization (Henry et al., 2020), and the cross-attention output projection is zero-initialized. We use shared adaLN ("adaLN-single") (Chen et al., 2023): one projection computes the modulation parameters from the timestep and a global vector, and each block adds only a learned $6 \times d$ table. The encoder layers tapped for conditioning are $\{0, 3, 7, 11, 15, 19, 23, 27, 31\}$. Two independent concatenation fusions (Eq. 4) produce the con-

Table 14: Conditioning tasks used to train the generative decoders. "Rot." indicates whether the item (context and target) may be randomly rotated. Tasks whose only pose information comes from a pose-locked modality (image, sketch, depth, masked image, part) are kept in canonical pose. When a geometric modality is present, it fixes the pose and the pose-locked modality acts as a canonical reference.

Type	Context	Rot.
Single	text; image (1–4 views); part point cloud; sketch; depth; masked image	no
Single	voxel 16^3; point cloud; bounding box	yes
Pair	voxel+text; point cloud+text; bbox+text	yes
Pair	image+text; part+text; sketch+text; depth+text; masked image+text	no
Pair	image+voxel; image+point cloud; image+bbox; part+point cloud	yes
Pair	image+part	no

ditioning. The first is a token sequence, projected to the decoder width and read through cross-attention. The second is a global vector, obtained by attentive pooling and a zero-initialized MLP and added to the timestep embedding. Context tokens are packed without padding and processed with variable-length attention. For classifier-free guidance, the conditioning is replaced by learned null embeddings with probability 0.1. The time shift is $s = 2$, $t = s\sigma/(1 + (s - 1)\sigma)$ with $\sigma = \mathrm{sigmoid}(u)$ and $u \sim \mathcal{N}(0, 1)$, and the number of Heun steps and the guidance scale w are set per modality (Table 15).

Sampling settings. For the benchmark results, we integrate the probability-flow ODE with Heun's second-order method (Karras et al., 2022) on the same shifted time schedule used in training. Because the conditioning signals differ greatly in how much they constrain the shape, a single setting is not optimal for all inputs. We therefore select the number of steps and the guidance scale for each input modality by grid search (Table 15). Strongly constraining inputs need few steps and little guidance. Voxel-to-3D uses only 5 steps, and point-cloud-to-3D uses guidance 2.5. Weakly constraining inputs benefit from more steps and stronger guidance, and text-to-3D uses 18 steps with guidance 7.

Table 15: Per-modality sampling settings (Heun solver), selected by grid search.

Task	Steps	Guidance scale w
Text→3D	18	7.0
Image→3D	15	5.0
Point cloud→3D	10	2.5
Voxel→3D	5	4.5

Rotation handling. Geometric contexts (voxels, point clouds, and boxes) share the exact rotation of the target, and text is rotation-invariant. Pose-locked contexts (images, sketches, depth, and parts) exist only in the canonical pose. An item with a pose-locked context but no geometric context therefore uses the canonical pose for both context and target.

Decoder scaling. Table 16 scales the decoder from Ours-S to Ours-L with the encoder, conditioning, and task mixture fixed. Going from Ours-S to Ours-M improves every task substantially, but the larger Ours-L decoder brings only small further gains, and voxel Chamfer distance even worsens slightly. This is consistent with our design. The pre-trained encoder does the heavy lifting of interpreting the context, so beyond a moderate size, additional decoder capacity yields diminishing returns while increasing inference time.

Inference time breakdown. Table 17 breaks down image-to-3D inference time for our three models. Timings are the mean over 10 samples on a single H100 GPU, measured from the input image to the generated shape after 10 warm-up runs. We use bf16 and `torch.compile`, and the settings of Table 15: 15 Heun steps, i.e., 29 decoder evaluations, with guidance scale 5. The keys and values of the context tokens for cross-attention are computed once and cached across steps. The encoder runs once per shape and costs about 33 ms for every model, only 2–6% of the total, even though it has 407M parameters. The flow decoder dominates and scales with decoder size, from 13.7 ms

Table 16: Scaling study of decoder capacity with a fixed pre-trained context encoder. Image- and text-to-3D are evaluated on TOYS4K, point-cloud- and voxel-to-3D on MAS. Time is image-to-3D inference time on one H100 (Appendix E). CD is reported $\times 10^3$.

Model	Decoder (width × depth)	Decoder params	Time	Image→3D			Text→3D			Point cloud→3D		Voxel→3D	
				CLIP↑	ULIP2↑	UNI3D↑	CLIP↑	ULIP2↑	UNI3D↑	IoU↑	CD↓	IoU↑	CD↓
Ours-S	448 × 12	64.9M	0.56s	0.7497	0.3862	0.3353	0.2220	0.2823	0.2663	0.9095	1.156	0.7361	3.134
Ours-M	1024 × 32	574.1M	1.16s	0.7678	0.4026	0.3508	0.2411	**0.3155**	0.2927	0.9379	1.095	0.8040	**2.277**
Ours-L	1536 × 40	1.56B	1.47s	**0.7684**	**0.4031**	**0.3520**	**0.2418**	**0.3155**	**0.2941**	**0.9413**	**1.087**	**0.8080**	2.312

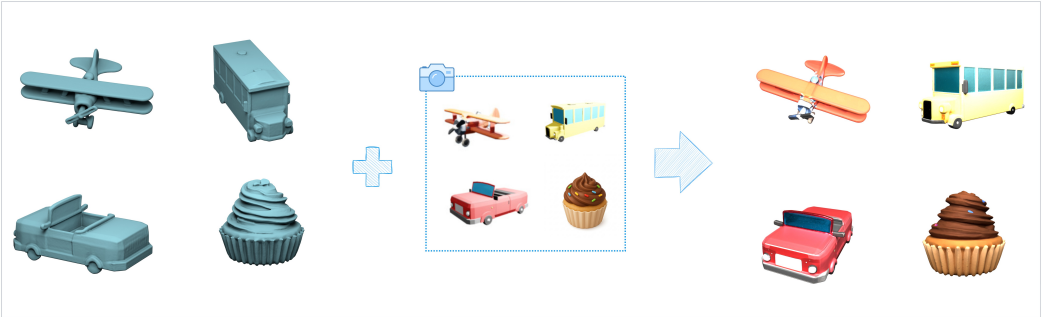

Figure 4: Using off-the-shelf texture generation pipelines (e.g. Trellis texture generation pipeline) conditioned on input images (middle) to get texturized assets (right)

per evaluation for Ours-S to 44.8 ms for Ours-L. Decoding the latent into an SDF takes a constant ≈120 ms. Concentrating capacity in the single-pass encoder is therefore almost free at inference, while every parameter added to the decoder is paid for at each of the 29 evaluations.

Table 17: Image-to-3D inference time breakdown (ms, mean over 10 samples on one H100). Tokenize covers image preprocessing, TA-TiTok tokenization, and context assembly. Decode covers the tokenizer decoder and inverse wavelet transform.

Model	Decoder params	Tokenize	Encoder	Flow (29 NFE)	ms / NFE	Decode	Total
Ours-S	64.9M	11.9	33.1	397.9	13.7	121.5	564
Ours-M	574.1M	12.0	33.4	988.6	34.1	124.2	1158
Ours-L	1.56B	12.3	34.4	1299.3	44.8	120.5	1466

Evaluation of the pre-training objective. For the clustering probes of Tables 7 and 18, we encode the ModelNet40 test set in canonical pose. We extract features from layers $\{11, 15, 19, 23, 27\}$ and the final layer, with register, mean, and max pooling. The features are clustered with K-means ($K = 40$). We report AMI and accuracy under the optimal Hungarian assignment between clusters and categories. For each input we report the best layer and pooling.

F TEXTURIZING

While the primary focus of our work is high-fidelity geometric generation, the resulting assets are inherently compatible with modern appearance synthesis workflows. To demonstrate this, we apply texture generation pipeline of TRELLIS.2 Xiang et al. (2025a) to our generated geometries. As shown in Figure 4, the structural consistency and clean topology of our 3D outputs provide a reliable foundation for automated texturing, allowing for the rapid creation of complete, render-ready 3D assets without additional manual refinement.

Table 18: Layer-wise zero-shot K-means clustering on ModelNet40 for the encoder pre-trained with cross-modal masking ($r_\times \sim \mathcal{U}[0.3, 1.0]$, 300k iterations). Each cell reports AMI / accuracy; the best layer for each metric is in **bold**. The best layer differs across modalities, which motivates conditioning on a fusion of all layers.

Layer	Text	Image	Point Cloud	Voxel
11	0.365 / 0.326	0.342 / 0.283	0.628 / 0.510	0.586 / 0.453
15	0.371 / 0.326	**0.437** / 0.353	0.692 / **0.594**	**0.631** / **0.542**
19	0.338 / 0.305	0.416 / 0.329	**0.695** / 0.569	0.605 / 0.481
23	0.344 / 0.297	0.422 / **0.359**	0.678 / **0.594**	0.567 / 0.456
27	0.374 / 0.317	0.380 / 0.329	0.657 / 0.545	0.511 / 0.402
Final	**0.404** / **0.334**	0.314 / 0.283	0.647 / 0.555	0.457 / 0.394

G LIMITATIONS AND FUTURE WORK

Our model inherits the quality ceiling of its underlying 3D tokenizer; future improvements in 3D representations could directly benefit our framework. Additionally, while we support a broad set of modalities, extending to other inputs such as audio or video descriptions remains unexplored. Finally, scaling context pretraining to even larger datasets and model sizes may further improve generation quality and generalization.

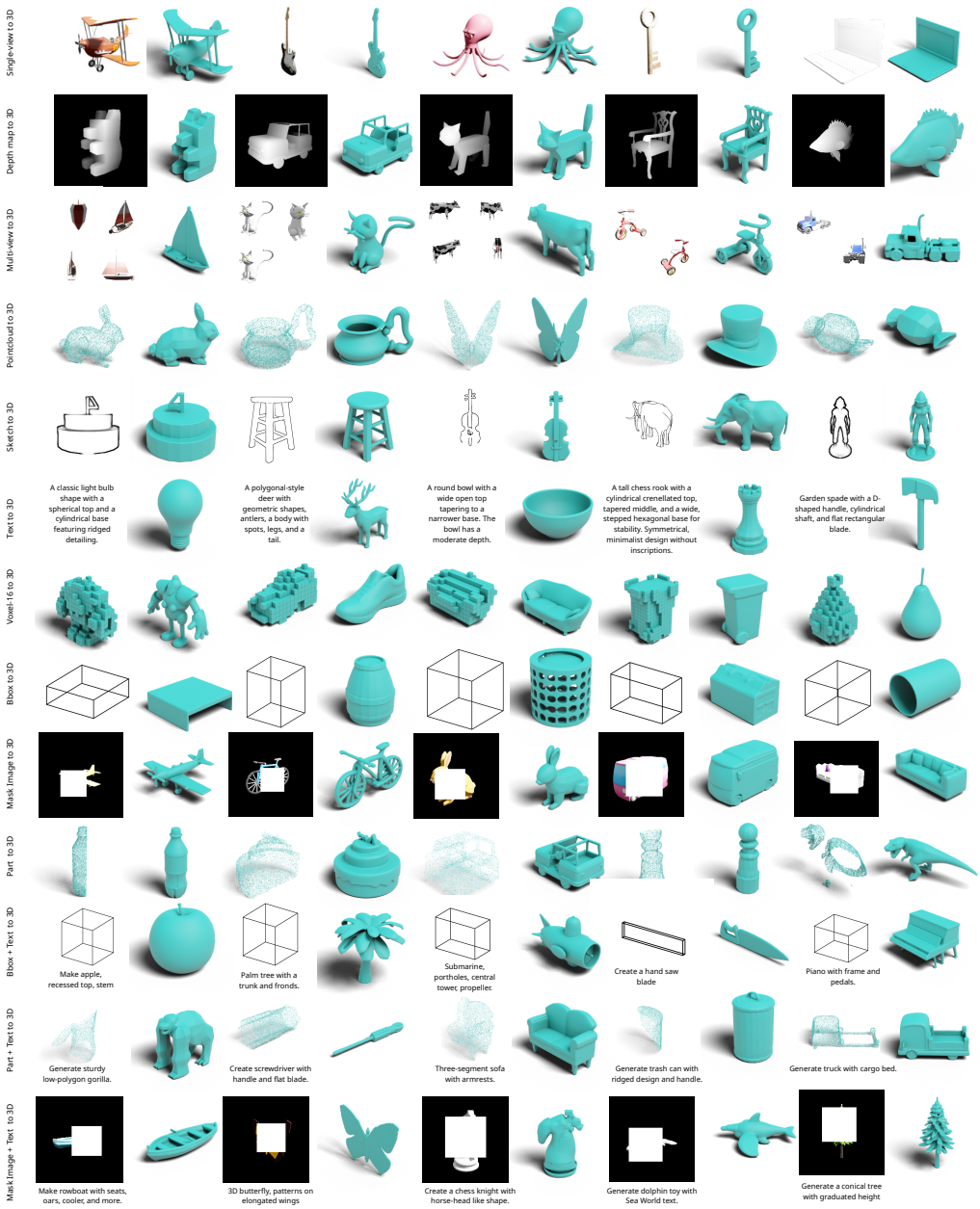

Figure 5: Examples of 3D generation from single- and multi-modal context inputs with a single model. Rows, top to bottom: single-view image, depth map, multi-view images, point cloud, sketch, text, 16^3 voxels, bounding box, masked image, part point cloud, bounding box + text, part point cloud + text, and masked image + text. Each input is followed by the generated shape.

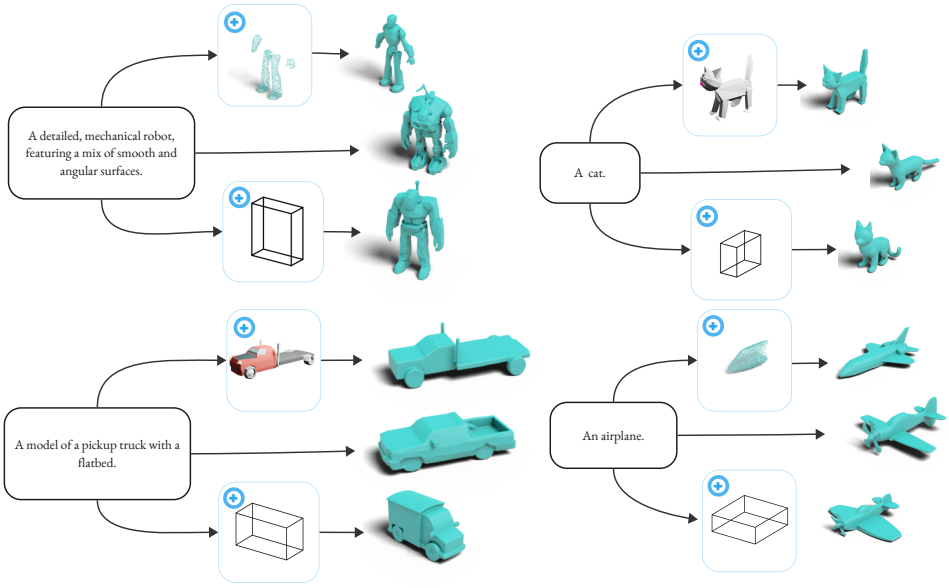

Figure 6: Examples of multi-modal generation illustrating how a single text prompt can be combined with various types of conditioning inputs to guide the creation of 3D models. For each object category (robot, cat, pickup truck, and airplane), the central path shows generation from text alone. The top and bottom paths show generation conditioned on both the text prompt and an additional modality, such as a part point cloud, a 3D bounding box, or a reference rendering, demonstrating enhanced control over the shape, scale, and proportions of the final 3D output.