GROWFLOW: UNIFIED 3D STRUCTURE AND FEATURE GENERATION VIA EXPANDING STRUCTURAL FLOW MATCHING

Anonymous authors
Paper under double-blind review



Figure 1: **A room furnished with GrowFlow.** Every object here was generated from a single image by one network, without a separate structure stage and without retraining for a target resolution: thin frames, hollow helmets, open chairs and cluttered mechanical assemblies all come out of the same expanding flow. Materials were assigned afterwards for display only.

ABSTRACT

Structured latent representations have been widely adopted for efficient 3D shape modeling and generation in recent years, due to their favorable trade-off between geometric fidelity and computational efficiency. However, most existing generative approaches decompose the generation process into two stages: first predicting sparse voxel structure, and then generating latent features based on that structure. Although effective, this staged design introduces extra parameters to predict the voxel structure. While the voxel structure does not require many parameters compared to the detailed geometry, because of the need for generalization, it is necessary to build a huge structure prediction network. To address this, we propose GrowFlow, the first 3D generative framework to unify structure and feature generation across resolutions within a single flow-matching process. Specifically, we provide a principled mapping strategy that enables consistent flow matching across different resolutions and structural states, laying the foundation for multi-scale structural modeling. To avoid structure-feature decomposition, we integrate a dedicated structure generation module into the flow matching paradigm, which jointly models sparse voxel structure and fine-grained geometric features in an end-to-end manner. Using a Diffusion Transformer (DiT) as the backbone, our framework achieves generation quality comparable to the state-of-the-art while using fewer parameters than multi-stage pipelines and reducing the computation required. Extensive experiments on benchmark datasets show that GrowFlow in-

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.

fers 3D occupancy progressively from coarse to fine resolution, matching the geometric fidelity of state-of-the-art methods while improving generation efficiency. Notably, our model enables resolution-adaptive, progressive generation, letting users preview low-resolution results, allowing early termination or full-generation.

1 INTRODUCTION

3D generation benefits digital twin systems by enabling component-level synthesis (Yao et al., 2025), large-scale scene generation, and customized training pipelines (Wu et al., 2024b) while reducing manual modeling effort; we target reconstructing an object's 3D geometry from a single input image. Most 3D generators adopt flow matching (FM), learning time-dependent vector fields that map a Gaussian prior to the data distribution, and increasingly operate on latent structure representations (Xiang et al., 2025b; Li et al., 2025d; Chen et al., 2025b; Wu et al., 2025b; He et al., 2025; Ren et al., 2024) that concentrate capacity near the surface and improve geometric fidelity. These methods typically split generation into separate stages: a coarse structure is produced first, and fine-grained features are then generated conditioned on it.

While effective, this sequential design has clear limitations. Structure and feature generation are decoupled, so a wrong early structural decision cannot be corrected later and information cannot flow between stages; ensuring correct structure therefore demands large models, extra training and slower inference, and the train/inference distribution gap across stages compounds the error.

Some recent image and video generators instead adopt a piecewise flow across resolutions (Chen et al., 2025a; Jin et al., 2025), forming one process from noise to data by linking successive resolution intervals with upsampling. Because each upsampling shifts the global covariance and breaks cross-scale independence (the jump-point issue), these methods reschedule timesteps to keep the flow stable, which inflates the number of inference steps. Extending this idea to 3D is non-trivial: a dense full-resolution 3D grid is costly, so sparse structure is essential, yet the variable branching factor of a sparse hierarchy rules out the fixed-ratio re-noising used in 2D.

To address these challenges, we propose an expanding structural flow matching (Lipman et al., 2023) strategy that generates 3D structure and features jointly across resolutions. Analyzing the continuity of noise in time and the unit-variance requirement of FM, we derive explicit mappings between the inference equations at different resolutions and structures, and add a strategy that stabilizes structural decoding during inference. Prior multi-resolution image (Chen et al., 2025a) and video (Jin et al., 2025) approaches, which reschedule time and inject auxiliary noise, are special cases of our framework at a fixed branching factor. Unlike conventional 3D generators fixed to one output resolution, our model generates at arbitrary resolution, synthesizing simple objects cheaply at low resolution and intricate ones at high resolution. Finally, a resolution-aware classifier-free guidance (CFG) scheme assigns separate, level-dependent guidance to structure and features, emphasizing structure at coarse levels and features at fine levels within one trajectory.

Experiments show quality on par with state-of-the-art multi-stage approaches at fewer parameters and faster inference, with a single network read out at any octree level so resolution is chosen per object from a low-resolution preview.

Overall, our contributions include:

- an expanding structural flow matching formulation with a principled mapping that keeps flow matching consistent across resolutions and structural states,
- a 3D framework that jointly generates structure and features in one piecewise multi-resolution flow, avoiding separate stages,
- the ability to synthesize 3D content at arbitrary resolution and to terminate inference early, and
- experiments showing quality comparable to state-of-the-art methods with fewer parameters and faster inference.

2 Related Work

2.1 Diffusion as a genertive model

Generative diffusion-type modelsSong et al. (2021a); Ho et al. (2020) have emerged as a mainstream solution for high-fidelity generative tasks. As discrete instantiations of continuous-time score-basedSong et al. (2021b) generative modeling with stochastic differential equations, DDPMHo et al. (2020) proposes the classic denoising-diffusion pipeline, and DDIMSong et al. (2021a) leverages implicit deterministic sampling to reduce the inference cost of multi-step denoising. Instead of relying on score-matching denoising losses, flow matchingLipman et al. (2023) achieves simulation-free training of velocity fields for probability transport, and Rectified flowLiu et al. (2023) further optimizes transport paths to facilitate high-fidelity few-step sampling.

2.2 Multi-Resolution Flow Matching

Multi-resolution generation has been explored for 2D image and video synthesis. PixelFlow (Chen et al., 2025a) learns generative flows over a spatial feature hierarchy, Relay Diffusion (Teng et al., 2024) relays low-resolution representations to guide higher-resolution diffusion, and Pyramidal Flow Matching (Jin et al., 2025) adjusts the timestep schedule across resolutions during training. In sparse 3D spaces, however, upsampling assigns different timesteps to different voxels, which blocks a direct transfer of these schemes and motivates the multi-resolution formulation we develop for 3D.

2.3 3D Generative Models

Advances in 3D datasets (Deitke et al., 2023b;a; Stojanov et al., 2021; Chang et al., 2015), reconstruction (Mildenhall et al., 2022; Kerbl et al., 2023) and surface extraction (Lorensen & Cline, 1987; Shen et al., 2023) underpin modern 3D generative modeling.

2D diffusion-based methods either optimize a 3D model through a pre-trained 2D diffusion prior (Poole et al., 2023; Wang et al., 2023; Lin et al., 2023) or synthesize multi-view images for feed-forward reconstruction (Hong et al., 2024; Tang et al., 2024; Zhang et al., 2024a; Xu et al., 2024), but inherit the weak multi-view consistency of 2D diffusion and thus produce artifacts at slow speeds.

VecSet-based methods encode a shape as an unordered set of latent vectors (Zhang et al., 2023) and have been adopted (Li et al., 2025a; Zhao et al., 2023; Li et al., 2025c; Zhao et al., 2025; Chen et al., 2025b; Zhang et al., 2024b; Li et al., 2025b), with Direct3D (Wu et al., 2024a) adding a triplane variant; as a global code decoded by cross-attention against query points, this family tends to over-smooth fine geometry.

SLAT-based methods encode a shape into sparse voxels with per-voxel features (Xiang et al., 2025b), improving fine geometry and inspiring many follow-ups (He et al., 2025; Li et al., 2025d; Ye et al., 2025; Wu et al., 2025b). Most adopt a two-stage pipeline that predicts structure first and features second, and this decoupling limits information exchange and causes structure–geometry misalignment; recent single-stage variants (Wu et al., 2025a; Anonymous, 2026) fold VecSet decoding into SLAT but their field-like transforms still limit structural fidelity.

3 Enhancing Structural Flow Matching

Flow matching (FM) (Lipman et al., 2023) casts generation as learning a time-dependent velocity field that transports a Gaussian prior to the data distribution. For a noise sample $x_0 \sim \mathcal{N}(0, 1)$ and a data sample $x_1 \sim p_{data}$, the interpolation path

$$x_t = (1 - t)x_0 + tx_1, \qquad t \in [0, 1] \tag{1}$$

induces the ODE $dx_t/dt = v(x_t, t)$ whose closed-form velocity is regressed by a network $v_\theta(x_t, t, c)$ by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,x_1,t}[||v_\theta(x_t, t, c) - v(x_t, t)||_2^2], \tag{2}$$

after which inference integrates the learned ODE from $t = 0$ to $t = 1$. Piecewise flow matching (Chen et al., 2025a; Jin et al., 2025) splits $[0, 1]$ into R overlapping intervals $[s_r, e_r]$, each refining one resolution level into the next; because early intervals carry a coarser grid with far fewer tokens, the network is evaluated far fewer times than in a single-resolution flow of equal length. Its per-interval boundaries are

$$\text{Start:} \quad x_s^r = (1 - s_r)x_0 + s_r \,\mathrm{Up}(\mathrm{Down}(x_1, r + 1)) \tag{3}$$

$$\text{End:} \quad x_e^r = (1 - e_r)x_0 + e_r \,\mathrm{Down}(x_1, r) \tag{4}$$

where $\mathrm{Down}(\cdot)$ and $\mathrm{Up}(\cdot)$ are the down- and upsampling operators, and the state inside the r-th window follows a rescaled local time $t' = (t - s_r)/(e_r - s_r)$:

$$x_t^r = (1 - t')x_s^r + t' x_e^r \tag{5}$$

(superscripts index the window, not powers). Continuity at the jump point between windows is normally restored by rescheduling the next interval's start time (Jin et al., 2025), e.g. $s_{r+1} = 2e_r/(1 + e_r)$ in 2D. We generalize this piecewise paradigm to 3D, jointly generating geometric structure and latent features in a single stage over sparse voxels. Unlike the dense 2D grid, the sparsity and spatial irregularity of 3D geometry demand a formulation in which structure and features are modeled together, and in which the non-fixed branching factor of the sparse hierarchy rules out the fixed-ratio re-noising used in 2D.

3.1 Multi-Scale Generation using an Octree

Dense grids are inefficient for high-fidelity 3D representation, so we adopt an octree to organize sparse voxels hierarchically across resolutions. As in prior piecewise flow matching approaches (Chen et al., 2025a; Jin et al., 2025), we partition continuous time into R intervals, each corresponding to a distinct octree resolution level, ranging from level L_0 to L_{R-1}. Here, L_{R-1} corresponds to the highest resolution, where only voxels located on the underlying geometry are retained, while L_0 represents the initial low-resolution level, organized as a dense grid. At intermediate levels, the octree is adaptively pruned according to voxel occupancy, yielding a progressively sparser representation. In this hierarchical structure, downsampling $\mathrm{Down}(\cdot)$ and upsampling $\mathrm{Up}(\cdot)$ differ fundamentally from those used in the (non-sparse) image domain. $\mathrm{Down}(\cdot)$ averages the features of all children for each parent node, whereas $\mathrm{Up}(\cdot)$ broadcasts the feature of a parent node to its m children. The number of child nodes depends on whether the corresponding sub-voxels intersect the underlying geometry, resulting in a structure with a variable branching factor.

Using the above design, GrowFlow progressively generates structured latent representations starting from a coarse dense noise grid, through iterative upsampling, structural pruning, feature re-noising, and denoising at various resolution levels. As a result, GrowFlow unifies 3D structure and feature generation within a single flow matching process. Based on Equation 5, the starting states x_s^r and ending states x_e^r are defined as:

$$x_s^r = \begin{cases} \mathrm{Up}_\epsilon((1 - s_r)\,\mathrm{Up}_\epsilon(x_0, r - 1) + s_r \,\mathrm{Down}(x_1, R - r)), & r > 0 \\ x_0, & r = 0 \end{cases} \tag{6}$$

$$x_e^r = (1 - e_r)\,\mathrm{Up}_\epsilon(x_0, r) + e_r \,\mathrm{Down}(x_1, R - r - 1) \tag{7}$$

where $\mathrm{Up}_\epsilon(\cdot)$ denotes upsampling with re-noising, which will be explained in detail in Sec. 3.2.

3.2 Re-noising at the Jump Point

Existing re-noising strategies suit the regular grids of image and video but fail on octrees, because the rescaled start time s_{r+1} after upsampling depends on each node's branching factor. In an image grid every pixel maps to a fixed number of children (e.g. 4), so one global time rescheduling applies everywhere. On an octree a voxel expands to a variable number of children (0 to 8) according to local occupancy, so no globally consistent rescheduling exists and existing re-noising becomes unsuitable for sparse latent generation.

At a jump point the latent must be upsampled while keeping the unit variance and inter-voxel independence that flow matching assumes. Mean downsampling of m children shrinks the noise variance to $1/m$, and naively broadcasting a parent value yields perfectly correlated siblings; both effects

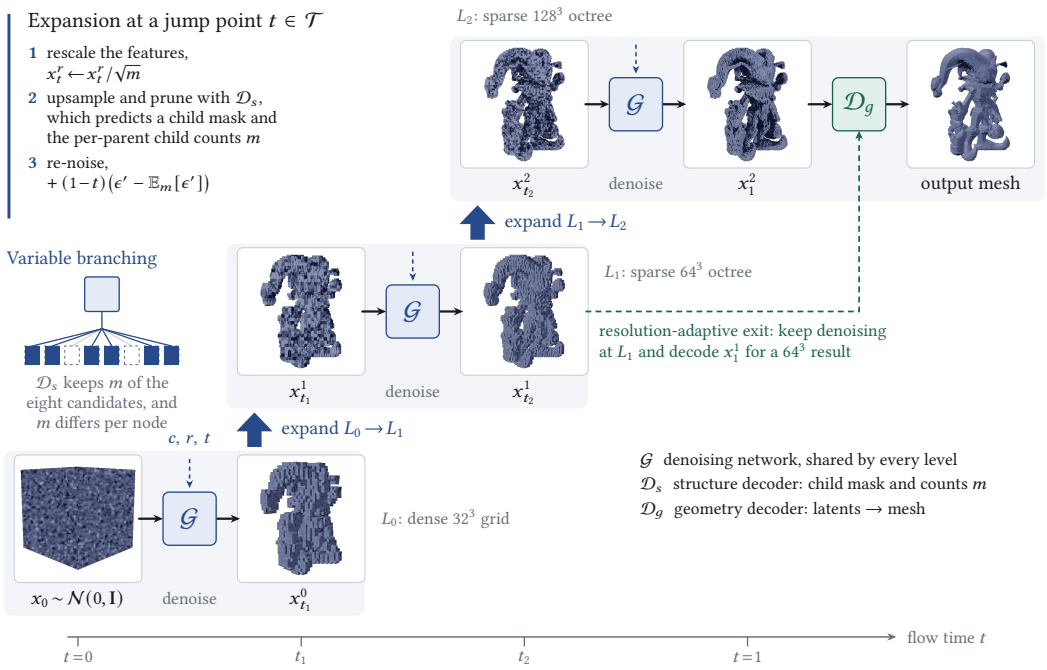

Figure 2: **Inference with expanding structural flow matching.** Each band holds the states on one octree level, drawn as their occupied voxels; flow time runs to the right and the octree level grows upward. From dense Gaussian noise on the L_0 grid, the shared denoiser $\mathcal{G}$ integrates the flow, and at each jump point $t \in \mathcal{T}$ the latent is rescaled, upsampled and pruned by the structure decoder $\mathcal{D}_s$, then re-noised, so structure and features grow within a single flow matching process. Because the re-noising is exact, unlike prior strategies (Chen et al., 2025a; Jin et al., 2025) the timestep schedule is never rescheduled. Stopping the same trajectory at an intermediate level (green) gives a resolution-adaptive family of outputs from one model; the expansion itself is detailed in Fig. 3.

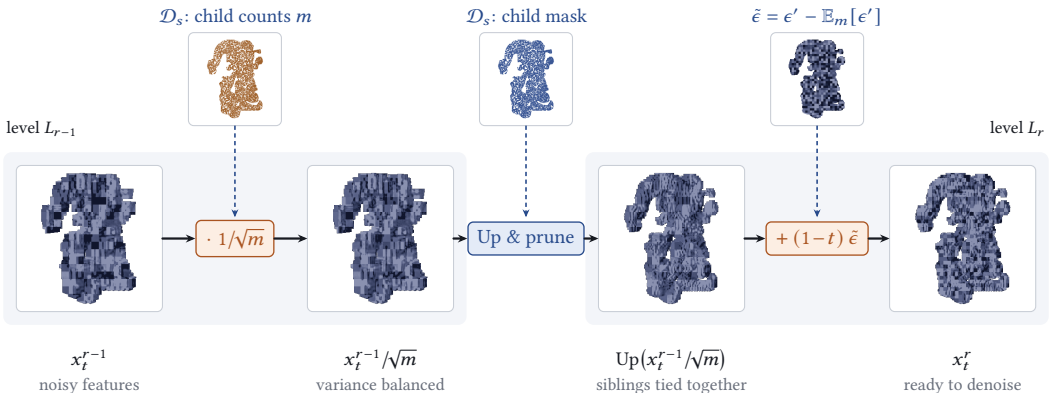

Figure 3: **The expansion operator Up_ϵ applied at a jump point** (Eq. 9). The octree level changes only here, so the two lanes mark the level before and after the jump. From left to right: the features are divided by $\sqrt{m}$ to undo the variance that mean downsampling removes; Up then hands each parent feature to the m children that the structure decoder $\mathcal{D}_s$ keeps, which leaves siblings perfectly correlated; a zero-mean residual $\tilde{\epsilon} = \epsilon' - \mathbb{E}_m(\epsilon')$ finally restores independence, all without moving the flow time t.

must be undone. We keep the resulting operator here and defer the full variance and independence derivation to Appendix A.3.

To address this issue, we further inject an independent noise component after upsampling to restore the correct noise covariance structure. Specifically, we compensate for the correlation by rescaling the latent features by $\sqrt{m}$, ensuring that each sub-voxel has unit variance and is statistically independent. After simplification, the final upsampling formulation can be written as:

$$x_t^r = \mathrm{Up}_\epsilon(x_t^{r-1}) \tag{8}$$

$$= \mathrm{Up}(\frac{x_t^{r-1}}{\sqrt{m}}) + (1-t)(\epsilon' - \mathbb{E}_m(\epsilon')) \tag{9}$$

where $\epsilon' \sim \mathcal{N}(0,1)$ is the newly sampled noise and $\mathbb{E}_m(\cdot)$ indicates the average of m sub-voxels. This formulation ensures temporal continuity, variance consistency across scales, and the independence required for flow matching. It enables stable training and inferencing, allowing the model to progressively upsample noise $\epsilon \sim \mathcal{N}(0,1)$ from low to high resolution for high-quality generation (see Fig. 3 for the three steps of a single jump, and Fig. 2 for the whole trajectory).

Leveraging the denoising strategy in Eqs. 6, 7 and the core multi-resolution upsampling and downsampling relationships in Eq. 9, the training and inference pipeline can be clearly interpreted via Algorithm 1 in Appendix A.2 and Fig. 2. Specifically, we initialize the generative process with Gaussian noise x_0 at the lowest resolution and perform denoising using the network $\mathcal{G}$ over the first time interval. We then apply feature rescaling to balance local variance. Subsequently, the latent representation is upsampled, and the noise level is adjusted to accommodate structural changes. A pruning step, guided by the structure decoder $\mathcal{D}_s$, updates the 3D occupancy. This pipeline alternates between denoising and upsampling at successive stages until the target resolution is reached. No further upsampling is performed after the final denoising stage. Finally, the latent features are decoded into the 3D object using a decoder.

4 Architecture for 3D Generation

With a complete generation strategy, we then consider the specific design of the neural network and the representation of its features. The neural networks serve two roles: the structure decoder $\mathcal{D}_s$ performs feature rescaling and structural pruning, while the denoising network $\mathcal{G}$(details are given in Appendix A.1) learns the flow direction for denoising (Algorithm 1 in Appendix A.2). Although $\mathcal{D}_s$ is a separate module, it prunes each level in place within the single flow rather than fixing the whole structure in a first stage, so the pipeline stays single-stage.

4.1 Octree Structured Latent Feature

We adopt the SLAT representation (Xiang et al., 2025b) as our latent feature; it consists of a set of voxel coordinates $p \in \mathcal{V}$ at resolution R and their corresponding geometric latent features $f_g \in \mathcal{F}_g$. Specifically, we use a VAE pretrained on SparseFlex (He et al., 2025), which takes point clouds as input and outputs meshes. Based on the sparse voxels $\mathcal{V}$, we construct an R-level octree $\mathcal{O} = \{\mathcal{O}_0, \ldots, \mathcal{O}_{R-1}\}$, spanning resolutions from $K - R + 1$ to K, which serves as the underlying structure for the expanding structure flow matching process. The latent features at each octree level are obtained using the downsampling operator $\mathrm{Down}(\cdot)$ introduced in Section 3.1.

Because mean downsampling makes a coarse feature consistent with several high-resolution structures, we add an octree structure embedding: the 8-bit child-occupancy mask is projected by a linear layer to the structure embedding f_s and normalized for scale stability. Finally, at each level r and timestep t, the structure embedding $f_s^{(r,t)}$ is concatenated with the geometric feature $f_g^{(r,t)}$ to form the joint latent representation x_t^r, which serves as the latent space of the generative model:

$$x_t^r = \mathrm{concat}(f_g^{(r,t)}, f_s^{(r,t)}), r \in [0, \ldots, R-1] \tag{10}$$

4.2 Denoiser

Following Trellis (Xiang et al., 2025b), we employ a Diffusion Transformer (DiT) (Peebles & Xie, 2023) to progressively denoise the joint latent representation across octree levels, from the initial

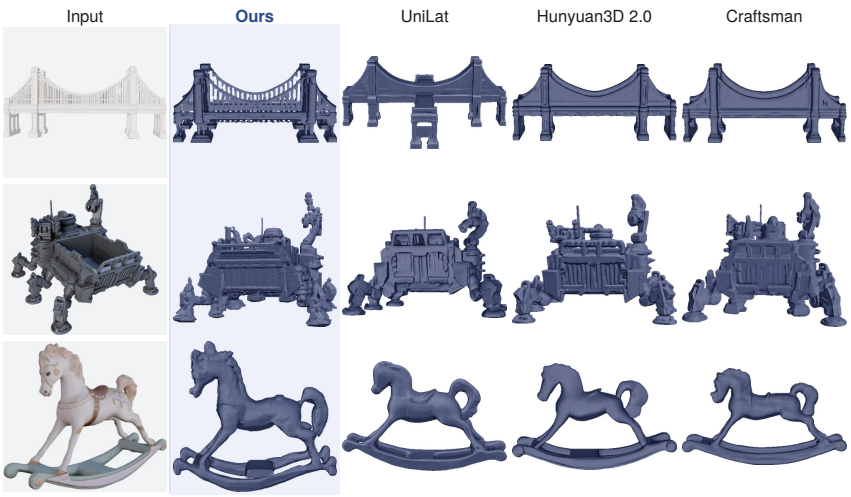

Figure 4: **Comparison with single-stage generators.** Every result is rendered with the same material and camera, so only the geometry differs. These baselines predict the shape in one pass at a fixed resolution and tend to smooth thin parts away, whereas GrowFlow grows the octree progressively and recovers structures such as the suspension cables of the bridge, the mechanical clutter on the vehicle and the open frame of the rocking horse.

Table 1: Comparison to other methods. Latent resolution is given in parentheses; a dash marks the methods whose latent is a fixed-length token set and therefore has no spatial resolution. Inference time is measured end to end on a single NVIDIA 4090, for geometry generation only—texture synthesis is excluded where a baseline provides it—using each method's released default sampler settings.

Type	Method (latent resolution)	Structure	ULIP score ↑	Uni3D score ↑	inferencing time↓	parameters↓
multi-stage	Trellis (64)	SLAT	0.1530	0.2825	∼ 10s	1.64B
	Direct3D-S2 (128)	SLAT	0.1557	0.2874	∼ 215s	2.72B
	Hi3DGen (128)	SLAT	0.1535	0.2799	∼ 32s	2.63B
	Trellis2 (128)	SLAT	0.1567	**0.2925**	∼ 47s	3.11B
single-stage	Craftsman (–)	VecSet	0.1520	0.2588	∼ **8s**	**1.35B**
	Hunyuan3D 2.0 (–)	VecSet	0.1548	0.2879	∼ 32s	2.60B
	UniLat (64)	VecSet	0.1506	0.2704	∼ 13s	2.50B
ours	Ours (64)	Growable SLAT	0.1537	0.2830	∼ **8s**	1.53B
	Ours (128)	Growable SLAT	**0.1583**	0.2898	∼ 27s	1.53B

noise toward the target state given by Eqs. 6 and 7. The image condition c is extracted by DINOv2 (Oquab et al., 2024), and the octree level r serves as an additional conditioning signal. Fig. 6 details a single block: the timestep and level are embedded, summed and mapped by one MLP to the scale, shift and gate parameters of the two modulated norms in the AdaLN-Zero manner of (Peebles & Xie, 2023), so both signals act on every block at negligible cost; image tokens enter through a cross attention, and rotary embeddings of the voxel coordinates place tokens within each attention window. During training, for a randomly sampled timestamp $t \in [0, 1]$, we first determine the corresponding time interval according to predefined time windows, which evenly partitions the entire timeline in our experiments. The start and end points for the current interval are then computed according to Eqs. 6 and 7, which allows us to calculate the ground-truth velocity of the flow trajectory for supervising the Denoiser $\mathcal{G}$: $\mathbb{E}_{x_t^r,t,r}[||\mathcal{G}(x_t^r, t, c, r) - v_t||_2^2]$.

4.3 Structure Decoder

During the upsampling transition between successive intervals, the structure of the next octree level must be inferred from the current state. To do so, we train an additional structure decoder $\mathcal{D}_s$ that predicts the child-node structure for each octree node at the current level, conditioned on the current noisy latent feature x_t^r, the resolution level r, and the timestep t, as follows: $o = \mathcal{D}_s(x_t^r, r, t)$, where

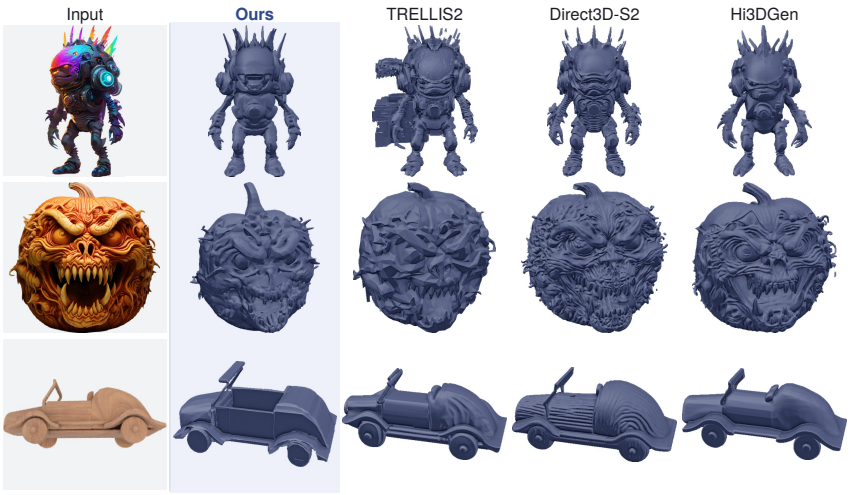

Figure 5: **Comparison with multi-stage generators.** Multi-stage baselines show displaced limbs and interiors that are closed over. GrowFlow keeps structure and features on one flow, so the occupancy is revisited at every jump point: the spikes of the first row stay separate from the body, and the car keeps an open cockpit with its windscreen raised.

o is the 8-bit occupancy mask for each node, indicating which child nodes should be retained. We train the structure decoder using the ground-truth structure $\hat{o}$ with a binary cross-entropy loss.

4.4 Adaptive Classifier Free Guidance

Different from existing methods adopting a fixed CFG weight, we assign separate guidance scales to structural and feature latent components and adaptively adjust them across flow intervals. At low-resolution early stages, we use stronger CFG for structure to ensure geometric correctness and weaker CFG for features. As the resolution grows, we gradually reduce structural guidance while increasing feature guidance to recover fine-grained details. This design naturally generalizes conventional two-stage paradigms: it prioritizes structure at coarse levels and refines geometry at fine levels within a single unified flow-matching process.

5 Experiments

Implementation details, datasets and evaluation metrics are given in Appendix B.

5.1 Comparison

We compare against single-pass methods (single-stage in Table 1) and multi-stage methods; ours is single-stage yet progressive across octree levels, so it forms its own group. All baselines use released checkpoints trained on different corpora, so the comparison is system-level.

Comparison to Single-Stage Models UniLat (Wu et al., 2025a) equips a VecSet (Zhang et al., 2023) architecture with a voxel-aware decoder. Comparative results are shown in Fig. 4 and Table 1. Our results recover thin structures these baselines (Li et al., 2025a; Zhao et al., 2025; Wu et al., 2025a) smooth away (e.g. the suspension cables of the bridge in Fig. 4); on Table 1 we match the strongest of them at latent resolution 64 and lead all three at 128. Unlike UniLat (Wu et al., 2025a), which predicts structure in a single pass and cannot easily correct errors, our approach determines structure progressively, allowing refinement and yielding more stable results.

Comparison to Multi-Stage Models Comparative results for these baselines are shown in Figs. 5, 9 and 10, and Table 1(the Figs. 9 and 10 are given in Appendix D). Our results exhibit more stable 3D structures than other methods (Xiang et al., 2025a;b; Wu et al., 2024a; Ye et al.,

Table 2: Ablations, scored with ULIP and Uni3D. *Top*: one model read out at four octree resolutions in a single inference run (with end-to-end time). *Bottom*: components removed at resolution 64; the full model there is the "64" row above. Because our design decodes the noisy latent directly, the noise-subtraction variant adds that step and is therefore listed as "w/" rather than "w/o".

Setting	ULIP ↑	Uni3D ↑	time ↓
Resolution (single model, one run)			
128	**0.1583**	**0.2898**	∼27s
64	0.1537	0.2830	∼8s
32	0.1529	0.2533	∼6s
16	0.1499	0.2499	∼**5s**
Component ablation (resolution 64)			
w/o structure normalization	0.1511	0.2635	
w/o noisy prediction	0.1534	0.2799	
w/ noise subtraction	0.1280	0.2588	
w/o progressive CFG	0.0741	0.1268	

2025), though on the aggregate scores of Table 1 we only match the strongest of them—Trellis2 keeps the best Uni3D score—while running about $1.7\times$ faster with under half the parameters.

5.2 Ablation Studies

5.2.1 Generation at different resolutions

Reading one trajectory out at successive octree levels yields several resolutions from a single run (Fig. 7 in Appendix C, Table 2): Since silhouette integrity can be reliably evaluated at low resolution, poor-quality previews are discarded prior to executing computationally expensive higher-resolution levels, and an acceptable 64^3 output can also be adopted as the final result but not the 128^3 — enabling a flexible quality-efficiency trade-off.

5.2.2 Component ablation

We ablate each component at resolution 64 (Fig. 8 in Appendix C, Table 2). Omitting normalization for the structure feature $f_s^(r,t)$ (normalize-structure) creates non-linear flow-matching trajectories and degrades results. Predicting only the clean structure feature $f_s^(r,t)$ in every upsampling step (noisy-prediction) causes timestep misalignment against latent feature $f_g^(r,t)$, diminishing fine details. One-step pure-noise prediction with subsequent subtraction from the noisy structure feature (noise-subtraction) introduces extra artifacts. Additionally, a fixed CFG weight inadequately constrains structure in early stages and features in later stages, lowering final generation quality.

6 Conclusions and Future Works

We present an expanding structural flow-matching framework that generates 3D structure and features jointly across resolutions in one stage, running faster than two-stage pipelines at matched resolution, free of cross-stage information loss, and preserving thin detail that implicit methods smooth away, with multi-resolution readout and early termination. More results are given in the gallery of Appendix D. Current results are constrained by a limited maximum resolution, no post-generation editing, and an evaluation based on image–shape embedding scores rather than surface metrics, which we leave to future work.

AI Use Statement

Large language models were used only to polish author-written prose. The research itself was carried out by the authors.

REFERENCES

Anonymous. Voxset: Sparse voxel set tokenizer for 3d shape generation. In *The Fourteenth International Conference on Learning Representations (ICLR)*, pp. 14, Virtual Event, May 2026. OpenReview. URL https://openreview.net/forum?id=7cLvFw1ZGu. Under review.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. doi: 10.48550/ARXIV.1512.03012. URL https://arxiv.org/abs/1512.03012.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *CoRR*, abs/2504.07963, 2025a. doi: 10.48550/ARXIV.2504.07963. URL https://doi.org/10.48550/arXiv.2504.07963.

Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *CoRR*, abs/2507.17745, 2025b. doi: 10.48550/ARXIV.2507.17745. URL https://doi.org/10.48550/arXiv.2507.17745.

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 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, New York, NY, USA, 2023a. Curran Associates. URL http://papers.nips.cc/paper_files/paper/2023/hash/70364304877b5e767de4e9a2a511be0c-Abstract-Datasets_and_Benchmarks.html.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 13142–13153, Los Alamitos, CA, USA, 2023b. IEEE. doi: 10.1109/CVPR52729.2023.01263. URL https://doi.org/10.1109/CVPR52729.2023.01263.

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *CoRR*, abs/2503.21732, 2025. doi: 10.48550/ARXIV.2503.21732. URL https://doi.org/10.48550/arXiv.2503.21732.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=sllU8vvsFF.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=66NzcRQuOq.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433. URL https://doi.org/10.1145/3592433.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 5307–5317. Computer Vision Foundation / IEEE, 2025a. doi: 10.1109/CVPR52734.2025.00500. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_CraftsMan3D_High-fidelity_Mesh_Generation_with_3D_Native_Diffusion_and_Interactive_CVPR_2025_paper.html.

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, Xiao Chen, Feipeng Tian, Jianxiong Pan, Zeming Li, Gang Yu, Xiangyu Zhang, Daxin Jiang, and Ping Tan. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *CoRR*, abs/2505.07747, 2025b. doi: 10.48550/ARXIV.2505.07747. URL https://doi.org/10.48550/arXiv.2505.07747.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *CoRR*, abs/2502.06608, 2025c. doi: 10.48550/ARXIV.2502.06608. URL https://doi.org/10.48550/arXiv.2502.06608.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *CoRR*, abs/2505.14521, 2025d. doi: 10.48550/ARXIV.2505.14521. URL https://doi.org/10.48550/arXiv.2505.14521.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 300–309. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00037. URL https://doi.org/10.1109/CVPR52729.2023.00037.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In Maureen C. Stone (ed.), *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, pp. 163–169. ACM, 1987. doi: 10.1145/37401.37422. URL https://doi.org/10.1145/37401.37422.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65 (1):99–106, 2022. doi: 10.1145/3503250. URL https://doi.org/10.1145/3503250.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=FjNys5c7VyY.

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 4209–4219. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00403. URL https://doi.org/10.1109/CVPR52733.2024.00403.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 42(4):37:1–37:16, 2023. doi: 10.1145/3592430. URL https://doi.org/10.1145/3592430.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 1798–1808. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00184. URL https://openaccess.thecvf.com/content/CVPR2021/html/Stojanov_Using_Shape_To_Categorize_Low-Shot_Learning_With_an_Explicit_Shape_CVPR_2021_paper.html.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: large multi-view gaussian model for high-resolution 3d content creation. *CoRR*, abs/2402.05054, 2024. doi: 10.48550/ARXIV.2402.05054. URL https://doi.org/10.48550/arXiv.2402.05054.

Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=qTlcbLSm4p.

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. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1a87980b9853e84dfb295855b425c262-Abstract-Conference.html.

Guanjun Wu, Jiemin Fang, Chen Yang, Sikuang Li, Taoran Yi, Jia Lu, Zanwei Zhou, Jiazhong Cen, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Xinggang Wang, and Qi Tian. Unilat3d: Geometry-appearance unified latents for single-stage 3d generation. *CoRR*, abs/2509.25079, 2025a. doi: 10.48550/ARXIV.2509.25079. URL https://doi.org/10.48550/arXiv.2509.25079.

Shuang Wu, Youtian Lin, Yifei Zeng, Feihu Zhang, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/dc970c91c0a82c6e4cb3c4af7bff5388-Abstract-Conference.html.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *CoRR*, abs/2505.17412, 2025b. doi: 10.48550/ARXIV.2505.17412. URL https://doi.org/10.48550/arXiv.2505.17412.

Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, and Pan Ji. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *ACM Trans. Graph.*, 43(4):43:1–43:17, 2024b. doi: 10.1145/3658188. URL https://doi.org/10.1145/3658188.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *CoRR*, abs/2512.14692, 2025a. doi: 10.48550/ARXIV.2512.14692. URL https://doi.org/10.48550/arXiv.2512.14692.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 21469–21480. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.02000. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiang_Structured_3D_Latents_for_Scalable_and_Versatile_3D_Generation_CVPR_2025_paper.html.

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. *CoRR*, abs/2404.07191, 2024. doi: 10.48550/ARXIV.2404.07191. URL https://doi.org/10.48550/arXiv.2404.07191.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: learning a unified representation of language, images, and point clouds for 3d understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1179–1189. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00120. URL https://doi.org/10.1109/CVPR52729.2023.00120.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. CAST: component-aligned 3d scene reconstruction from an RGB image. *ACM Trans. Graph.*, 44(4):86:1–86:19, 2025. doi: 10.1145/3730841. URL https://doi.org/10.1145/3730841.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *CoRR*, abs/2503.22236, 2025. doi: 10.48550/ARXIV.2503.22236. URL https://doi.org/10.48550/arXiv.2503.22236.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4):92:1–92:16, 2023. doi: 10.1145/3592442. URL https://doi.org/10.1145/3592442.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: large reconstruction model for 3d gaussian splatting. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4,*

2024, Proceedings, Part XXII, volume 15080 of *Lecture Notes in Computer Science*, pp. 1–19. Springer, 2024a. doi: 10.1007/978-3-031-72670-5_1. URL https://doi.org/10.1007/978-3-031-72670-5_1.

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. *ACM Trans. Graph.*, 43(4):120:1–120:20, 2024b. doi: 10.1145/3658146. URL https://doi.org/10.1145/3658146.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ea1a7f7bc0fc14142106a84c94c826d0-Abstract-Conference.html.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo, Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu, Changrong Hu, Tianyu Huang, Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He, Tian Liu, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *CoRR*, abs/2501.12202, 2025. doi: 10.48550/ARXIV.2501.12202. URL https://doi.org/10.48550/arXiv.2501.12202.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=wcaE4Dfgt8.

A Additional Method Details

A.1 Denoiser Block Architecture

Fig. 6 expands the condensed denoiser description in the main text, detailing one block of the Diffusion Transformer (Peebles & Xie, 2023) that realizes the shared denoiser $\mathcal{G}$.

A.2 Inference Algorithm

Algorithm 1 states the full inference procedure summarized in the main text: starting from noise at the initial resolution, the shared denoiser $\mathcal{G}$ integrates the flow while, at each jump point $t \in \mathcal{T}$, the structure decoder $\mathcal{D}_s$ rescales, upsamples, prunes and re-noises the latent before continuing.

A.3 Derivation of the Re-noising Operator

This appendix derives the expansion operator Up_ϵ of Eq. 9, which restores the unit variance and inter-voxel independence that flow matching assumes when a latent is upsampled across a jump point.

To assess the continuity of flow matching at the jump point between intervals, it is helpful to reverse our perspective and first examine the changes in the corresponding down-sampling operation. Suppose that before downsampling, the noise follows a standard normal distribution $\mathcal{N}(0, 1)$. After

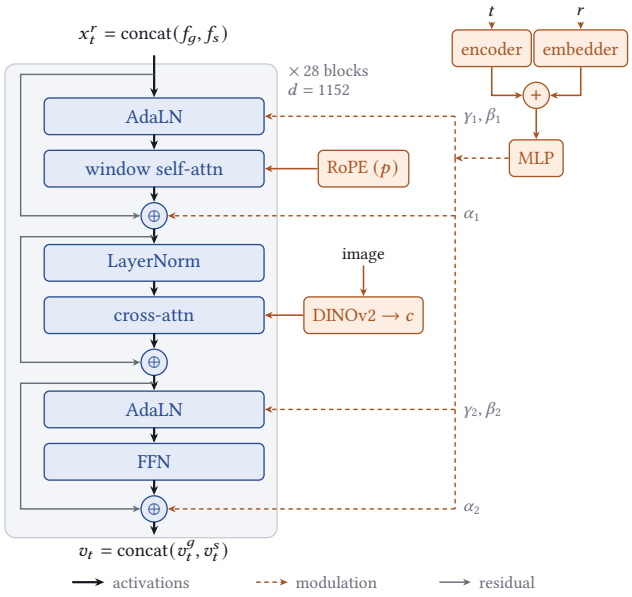

Figure 6: **One denoiser block.** A window self attention, a cross attention over the DINOv2 tokens, and a feed-forward layer, each preceded by a norm. Timestep t and octree level r are embedded, added, and mapped by one MLP to the two scale-shift pairs (γ_i, β_i) and the two gates α_i; a gated branch is scaled by its α_i before it rejoins the residual stream at $\oplus$, and the middle sum is a plain residual add.

Algorithm 1 Inferencing

Require: Noise x_0 at initial resolution r
Require: Jump points $\mathcal{T}$
Require: Stepsize δt
Require: Structure decoder network $\mathcal{D}_f$ for pruning
Require: Denoising network $\mathcal{G}$
Initialize geometric features $x_0^r \leftarrow x_0$
for $t = 0$ **to** 1 **step** δt **do**
  if $t \in \mathcal{T}$ **then**
    Rescale x_t^r with $\mathcal{D}_s$
    Upsample x_t^r and prune with $\mathcal{D}_s$
    Renoise x_t^r, Equation (9)
    $r \leftarrow r + 1$
  end if
  Denoise x_r^t with $\mathcal{G}$
end for
return x_1^r

averaging over m independent noise samples during downsampling, the resulting noise distribution becomes $\mathcal{N}(0, \frac{1}{m})$. This reduction in variance breaks the consistency of the noise schedule required by flow matching. Therefore, when performing downsampling, it is necessary to compensate for the missing noise energy by rescaling the full latent features by $\sqrt{m}$. During upsampling, we must first remove the corresponding compensated noise introduced in the downsampling step. However, directly repeating the same voxel value m times leads to the sub-voxels having correlated noise, since all replicated values share the same noise realization. This violates the independent noise assumption underlying flow matching: each spatial location is expected to be perturbed by independent noise.

The compensation this analysis implies is exactly the expansion operator Up_ϵ of Eq. 9.



Figure 7: **One trajectory, four usable resolutions.** All four models come from a single run of the same network, read out at successive octree levels; no retraining or re-sampling is involved.

B Implementation Details

Model Configuration, Training Settings, and Training Set Our neural network uses 28-layer Transformer blocks, each with 1152 dimension. Our flow matching model has a feature dimensionality of 32 and a structural dimensionality of 8. The denoiser DiT (Peebles & Xie, 2023) has 850M parameters; together with the structure decoder and the pretrained VAE decoder, the full generation stack reported in Table 1 totals 1.53B. Our model can adapt to various resolutions, without needing retraining, and at the same time, it can meet the needs of users with fixed computing resources. The total training time is about 3 weeks on 32 NVIDIA A800 GPUs. Our model was trained on a 500,000 subset of the ObjaverseXL (Deitke et al., 2023a) dataset.

Test Set All tests run on NVIDIA 4090 GPUs rather than on the training hardware, because their smaller memory budget is closer to the environment in which such models are actually deployed. Quantitative results are reported on the public Toys4k (Stojanov et al., 2021) benchmark, a standard evaluation set of 4,000 clean single-object scans over 105 categories that shares no assets with our ObjaverseXL training subset; each object is rendered from one randomly chosen view, so the evaluation measures single-image reconstruction rather than best-of-n selection. We additionally report qualitative comparisons on the in-the-wild images released with prior work (Xiang et al., 2025b; Li et al., 2025c).

Evaluation Metrics Because single-image 3D generation is one-to-many—many plausible shapes are consistent with a single view—comparing a generation against one specific ground-truth scan penalizes valid outputs, so we follow common practice and instead measure how well the generated shape agrees with the conditioning image in a joint image–shape embedding space. ULIP (Xue et al., 2023) and Uni3D (Zhou et al., 2024) each provide such an embedding, trained on different data with different point encoders; we report the cosine similarity between the input image and the generated point cloud under both, and treat agreement between the two as the signal of interest. Alongside these we report end-to-end inference time and parameter count, since a central claim of this work is that folding structure and features into one flow removes a stage rather than adding one. Both embeddings score how well a shape agrees with its conditioning image rather than how accurate its surface is, so the geometric claims below rest on the qualitative comparisons; a surface-level evaluation against ground-truth scans is left for future work. Each entry is a single evaluation pass with one view per object and no averaging over seeds, so gaps of a few thousandths should be read as ties rather than as a ranking.

C Qualitative Ablation Examples

Qualitative resolution ablation results incorporating our quality-efficiency trade-off are provided in Fig. 7, and component ablation results evaluating various flow-matching design variants are given in Fig. 8.

D Additional Results

This appendix collects the material that did not fit in the main paper: two further comparison sheets, each against both multi-stage and single-stage baselines (Figs. 9 and 10) and a gallery of inputs and outputs produced by a single model (Fig. 11). Every render in the paper, including these, uses one material and one virtual light, so shape differences cannot be mistaken for shading differences.

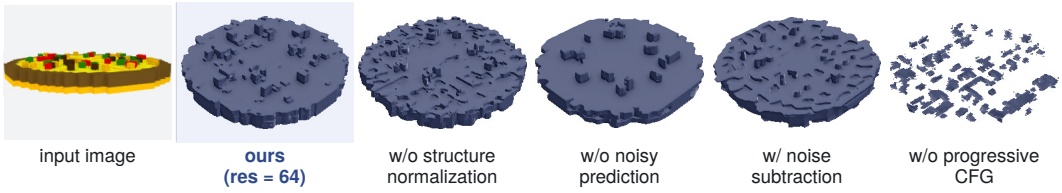

Figure 8: **Component ablation** at a fixed resolution of 64, on a block 'pizza' whose flat top and many small toppings make each failure mode easy to read.



Figure 9: **Additional comparisons, part one.** Left to right the columns are the input image, GrowFlow, Trellis2, Direct3D-S2, Hi3DGen and Hunyuan3D 2.0, so both multi-stage and single-stage baselines are represented. Hollow interiors and thin members, which is where fixing the structure in a first stage costs the most: a voxel ruled empty there can never be filled again, and one ruled solid can never be opened. The cup is the clearest case. Its stripes are painted rather than modelled, yet several baselines cut them into the wall or thicken it, while GrowFlow keeps the wall a single clean surface because the occupancy of the rim is re-decided at every jump point. The two humanoid rows and the mushroom stalk show the same effect on slender geometry.



Figure 10: **Additional comparisons, part two.** Left to right the columns are the input image, GrowFlow, Trellis2, Direct3D-S2, Hi3DGen and Hunyuan3D 2.0, again spanning both families. Two rows carry most of the evidence. On the warning sign, several baselines emboss the printed lettering and the print pattern into the geometry, whereas GrowFlow keeps the board a flat plate and the frame a thin open loop. On the shoe, GrowFlow is the only method that leaves the interior cavity open; the others close it over, because the occupancy of the opening is settled in a first stage that is never revisited.



Figure 11: **Gallery of results**, from in-the-wild images and from Toys4k (Stojanov et al., 2021). All eighteen objects come from the same checkpoint at a latent resolution of 128, with no per-object tuning; the inputs span photographs, renders and stylised illustrations.