HGA3D: HIGH-FIDELITY GEOMETRY-AWARE 3D GENERATION ACCELERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent VecSet-based 3D diffusion models, which represent 3D geometry as a compact set of latent vectors, achieve high generation quality but suffer from prohibitive inference latency due to multi-step denoising and expensive decoding. The central challenge is identifying reducible computational redundancy without corrupting the zero-level surface that determines the final mesh. To address this issue, we leverage **geometry preservation** as a guiding principle to accelerate 3D generation, implementing acceleration strategies across both the diffusion and decoding phases. In the diffusion stage, we introduce *geometry-aware token pruning*, which exploits decoder cross-attention between sparse dynamic surface anchors and latent tokens to retain surface-critical tokens while eliminating redundancy. In the decoding stage, we propose *curvature-adaptive volume decoding*, which facilitates efficient mesh decoding by adaptively concentrating high-resolution queries on geometrically complex regions while sparsifying redundant queries in smooth near-surface areas. Experiments on TripoSG and Hunyuan3D-2.0 demonstrate total inference speedups of $1.50\times$ and $1.48\times$, respectively, while preserving high-fidelity and structurally consistent 3D generation. Our code will be released.

1 INTRODUCTION

Recent advancements in native 3D generation methods Xiang et al. (2025b;a); He et al. (2025); Ye et al. (2025); Wu et al. (2024; 2025) have enabled the automated synthesis of high-fidelity 3D assets directly from text or visual prompts. As a key for content creation in virtual environments, these advancements attract increasing attention in robotics, gaming, and virtual reality.

Among native 3D generation approaches, VecSet-based 3D diffusion models Zhang et al. (2023); Zhao et al. (2023); Zhang et al. (2024); Chen et al. (2025); Li et al. (2025b;d); Lin et al. (2025); Li et al. (2025c); Zhao et al. (2025b); Lai et al. (2025a;b) have been widely studied due to their superior representational capacity. These methods employ a VAE Encoder to compress 3D shapes into a compact latent vector set (VecSet) and train a diffusion model to generate such representations. During inference, the generated VecSet is decoded by predicting point-wise SDF Park et al. (2019) or occupancy fields Mescheder et al. (2019) at sampled query coordinates via cross-attention, followed by surface extraction via Marching Cubes Lorensen & Cline (1987). While VecSet-based methods have demonstrated the ability to generate high-quality 3D shapes, their practicality is hindered by high inference latency from multi-step denoising and a computationally intensive decoding process, where the number of dense grid queries grows cubically with the grid resolution.

To accelerate the diffusion stage Ho et al. (2020); Song et al. (2021), caching-based acceleration methods have been extensively explored in 2D image and video synthesis Ma et al. (2024b); Wimbauer et al. (2024); Ma et al. (2024a); Zhao et al. (2025a); Shen et al. (2025); Ye et al. (2024); Liu et al. (2025a); Zhou et al. (2025); Xie et al. (2025); Liu et al. (2025b); Bolya et al. (2023); Bolya & Hoffman (2023); Fang et al. (2025); Zou et al. (2025); Zhang et al. (2025); Zou et al. (2024); Liu et al. (2025c); Jiang et al. (2025). By exploiting redundant computations along the denoising trajectory, these methods accelerate inference by caching intermediate features from previous timesteps and reusing or approximating them in subsequent ones.

Despite their success in 2D generation, directly transferring existing acceleration mechanisms to VecSet-based 3D diffusion models presents non-trivial challenges, primarily due to: **i) Inability to identify geometric importance in dense latent spaces.** Unlike 2D images, which contain

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.

vast redundant low-frequency backgrounds, each token in the VecSet is highly concentrated with information defining local surfaces or topology. Since existing 2D pruning methods are geometry-agnostic, they often fail to distinguish surface-critical tokens from less relevant ones. Consequently, applying these methods can result in erroneous removal of vital geometry information, leading to severe geometric distortions. **ii) Vulnerability to approximation errors.** 3D generation is far less tolerant of accumulated errors than its 2D counterpart, slight numerical fluctuations near the zero-level set can lead to SDF sign flipping. Therefore, large approximation errors from caching can disrupt the Marching Cubes reconstruction, resulting in holes and surface fractures in the final mesh.

To address these issues, we leverage geometry preservation as a guiding principle and propose a 3D diffusion co-acceleration framework that jointly accelerates denoising and decoding for efficient generation. In the denoising stage, we first propose a **geometry-aware token pruning** method. As demonstrated in Zhang et al. (2023), the VecSet-based decoder establishes a mapping between spatial query points and latent tokens through cross-attention. Intuitively, the latent tokens representing the points near the object surface (rather than those in the object) are critical for the geometric shape. Based on this insight, we propose to identify those critical latent tokens through the probe of the cross-attention score (of the 3D VAE Decoder) to a sparse set of surface anchors. That *sparse* set of surface anchors, rather than dense volumetric sampling of the full surface, facilitates efficient geometry scoring without heavy computational overhead.

While the aforementioned strategies significantly accelerate the denoising stage, decoding remains a computational bottleneck. Existing hierarchical decoding strategies Lai et al. (2025b) indiscriminately apply high-resolution refinement sampling across the entire near-surface region, neglecting local geometric variations. This leads to computational redundancy in smooth, low-frequency areas. We argue that the sampling objective should evolve across resolutions: while coarse levels handle global surface localization, the finest level should prioritize local detail refinement at geometrically complex regions such as *high-curvature areas*. To this end, we propose a **curvature-adaptive volume decoding** strategy that utilizes the discrete Laplacian to estimate local curvature, thereby allocating high-resolution queries to critical structures (e.g., sharp edges) and reducing query points.

We perform experiments on both TripoSG Li et al. (2025d) and Hunyuan3D-2.0 Zhao et al. (2025b), demonstrating that our framework achieves total inference speedups of 1.50× and 1.48×, respectively, with competitive performance on key geometric metrics such as Chamfer Distance and F-Score. The main contributions of this paper are as follows:

- We propose a geometry-aware token pruning strategy to reduce latent-token redundancy during denoising. By calculating the cross-attention scores between latent tokens and the set of sparse surface anchors, it efficiently identifies and retains tokens critical for surface reconstruction while pruning irrelevant information.
- We develop a curvature-adaptive volume decoding mechanism that allocates resources based on local geometric complexity. It concentrates high-resolution sampling on sharp details (high-curvature areas) while sparsifying queries in smooth regions.
- We integrate the proposed strategies into a unified 3D diffusion co-acceleration framework. Experiments on TripoSG and Hunyuan3D-2.0 demonstrate that we achieve total inference speedups of 1.50× and 1.48×, respectively, while preserving geometric fidelity.

2 Related Work

Native 3D Generation Models. Enabled by large-scale 3D datasets such as Objaverse Deitke et al. (2023), native 3D generation methods perform denoising directly in 3D space, often using advanced training objectives such as Flow Matching Lipman et al. (2023) or Rectified Flow Liu et al. (2023). These methods mainly adopt either spatially structured representations or implicit latent sets. Spatially structured methods Xiang et al. (2025b;a); Ye et al. (2025); He et al. (2025); Wu et al. (2025); Li et al. (2025a) typically organize features on voxel grids. For example, TRELLIS Xiang et al. (2025b) adopts a structured latent representation (SLAT), which associates local latent features with explicit active voxel coordinates. In contrast, implicit latent set approaches Zhang et al. (2023); Zhao et al. (2023); Zhang et al. (2024); Chen et al. (2025); Li et al. (2025b;d); Lin et al. (2025); Li et al. (2025c); Zhao et al. (2025b); Lai et al. (2025a) represent shapes using compact sets of latent vectors (e.g., VecSet) without voxel-grid correspondence. These vectors are decoded into SDF or occupancy fields,

avoiding dense voxel storage and aligning naturally with Transformers. However, reconstruction fidelity remains subject to a trade-off between latent set size and computational efficiency.

Cache-Based 2D Diffusion Acceleration. Cache-based methods exploit spatiotemporal redundancy to reuse intermediate computations. For instance, block-wise methods Ma et al. (2024b); Wimbauer et al. (2024); Ma et al. (2024a); Zhao et al. (2025a); Shen et al. (2025) bypass redundant layers based on feature temporal coherence, while step-wise strategies Ye et al. (2024); Liu et al. (2025a); Zhou et al. (2025); Xie et al. (2025); Liu et al. (2025b) exploit temporal redundancy across adjacent timesteps to approximate intermediate results. Another line of research focuses on token-wise acceleration, which aims to reduce spatial redundancy during the diffusion stage Bolya et al. (2023); Bolya & Hoffman (2023); Fang et al. (2025); Zou et al. (2025); Zhang et al. (2025); Zou et al. (2024); Liu et al. (2025c); Jiang et al. (2025), thereby alleviating the quadratic complexity of the attention mechanism. While efficient for 2D images, these strategies do not explicitly measure the relevance of VecSet tokens to the reconstructed surface and may degrade geometric fidelity. In this work, we focus on token-wise acceleration and introduce a geometry-aware caching mechanism tailored to overcome this critical limitation in 3D diffusion while preserving high-fidelity 3D generation.

3D Diffusion Acceleration. Existing 3D diffusion acceleration methods broadly fall into training-free and training-based approaches. Training-free methods, such as Fast3Dcache Yang et al. (2026), accelerate SLAT-based structure generation through caching. Training-based methods, including FlashVDM Lai et al. (2025b) and MDT-dist Zhou et al. (2026), instead distill pretrained generators for few-step sampling. In contrast, our method targets pretrained VecSet backbones without additional training. Appendix A provides a detailed comparison of these methods in terms of representations, acceleration mechanisms, and training requirements.

Efficient Implicit-Field Decoding. VecSet-based 3D generators recover meshes by predicting an implicit field over spatial query points, resulting in decoding complexity that grows cubically with target resolution. Hierarchical surface extraction strategies reduce this cost by progressively refining candidate voxels near the zero-level set Lai et al. (2025b). However, near-surface localization alone does not distinguish smooth regions from high-frequency structures. Our curvature-adaptive decoder concentrates the finest-resolution computation on geometrically complex regions.

3 PRELIMINARIES

3.1 UNIFIED FLOW MATCHING PARADIGM

Recent 3D generative models Li et al. (2025d); Zhao et al. (2025b) employ Rectified Flow Liu et al. (2023) or Flow Matching Lipman et al. (2023) to train diffusion models. These methods share a unified ODE-based formulation with straight flows. To align with standard diffusion conventions, let $x_0 \sim p_{\text{data}}$ be the clean latent representation and $x_1 \sim \mathcal{N}(0, I)$ be the standard Gaussian noise. Both frameworks define a conditional linear interpolant:

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

The generative process simulates an Ordinary Differential Equation (ODE) from $t = 1$ to $t = 0$:

$$dx_t = v_\theta(x_t, t \mid c)dt, \tag{2}$$

where v_θ denotes the velocity field parameterized by a neural network (e.g., DiT models), conditioned on input c. The network is trained to regress the target velocity $u_t(x_t \mid x_0, x_1) = x_1 - x_0$, which represents the direction from data to noise. During inference, an ODE solver integrates Eq. (2) backwards in time to reconstruct x_0.

3.2 VECSET REPRESENTATION FOR 3D GEOMETRY

Recent approaches Zhang et al. (2023); Li et al. (2025d) utilize the VecSet representation to compress geometry $\mathcal{G}$ into a compact, fixed-size latent set $x_0 = \{z_i\}_{i=1}^N \in \mathbb{R}^{N \times D}$, where N and D denote the number of tokens and channel dimension, respectively. As shown in Figure 1, the inference pipeline consists of two stages: i) a DiT denoises Gaussian noise into the clean latent set x_0 via the ODE process described above, and ii) a VAE decoder interprets x_0 as an implicit field.

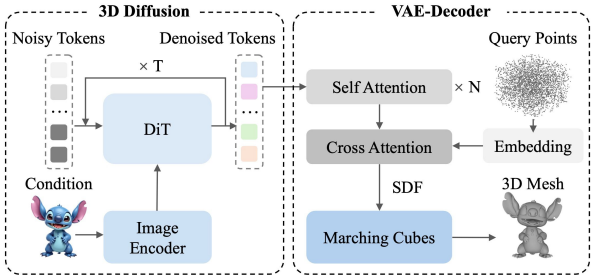

Figure 1: **VecSet-based generation pipeline.** i) a 3D Diffusion stage, where a DiT denoises image-conditioned latent tokens, and ii) a VAE-Decoder stage, reconstructing the mesh via cross-attention between tokens and query points.

Specifically, the decoder reconstructs the 3D mesh by querying point-wise attributes (e.g., SDF) through **cross-attention** between spatial query points and the latent tokens. To improve efficiency, hierarchical sampling Lai et al. (2025b) is often employed to refine queries near the implicit surface progressively. Finally, the implicit fields are discretized into meshes using Marching Cubes.

4 METHODS

4.1 OVERVIEW

As illustrated in Figure 2, we aim to preserve geometric details while accelerating both the diffusion and decoding stages. For the diffusion stage, we introduce ***geometry-aware token pruning*** (Section 4.2) to reduce token computation through geometry-guided selection. For mesh decoding, we propose ***curvature-adaptive volume decoding*** (Section 4.3), which allocates computational resources to high-curvature regions for efficient surface extraction. Both modules prioritize surface reconstruction, using token relevance and local geometric variation, respectively.

To align with the discrete sampling strategy, we slightly abuse the timestep notation and set discrete $t \in \{T, T-1, \ldots, 0\}$ in the following sections to denote the timestep index instead of continuous time. The inference starts from pure noise at $t = T$ and progresses to clean data at $t = 0$.

4.2 GEOMETRY-AWARE TOKEN PRUNING

Motivation. In 3D generation, we aim to reconstruct the surface points, rather than the inner or outer ones, as surface points directly contribute to the 3D shape. We thus identify the tokens associated with the surface points and prune the remaining tokens for acceleration, which leads to a novel geometry-aware token pruning framework. In order to do that, we measure the importance (i.e., surface-ness) of a latent token by its cross-attention score within the VAE decoder to a set of surface query points. We further maintain a **sparse** and dynamic set of surface query points (called anchors) for efficient calculation of the cross-attention scores.

Geometry-Aware Score Formulation. Specifically, we define a set of query points $\mathcal{V}$ located near the zero-level set to compute these attention weights. Since the decoder operates on clean data x_0, we first approximate x_0 from x_t. By rearranging the interpolant defined in Eq. (1) and substituting the learned velocity field $v_\theta \approx x_1 - x_0$, we derive the clean data estimation $\hat{x}_0^{(t)}$ at timestep t[1]:

$$\hat{x}_0^{(t)} = x_t - \frac{t}{T} v_\theta(x_t, \frac{t}{T} \mid c). \tag{3}$$

Subsequently, we feed the estimated denoised latent $\hat{x}_0^{(t)}$ into the VAE decoder to compute the cross-attention map $\mathbf{C} \in \mathbb{R}^{|\mathcal{V}| \times N}$ between query points $\mathcal{V}$ and latent tokens. This map serves as a geometric proxy to identify and prune redundant tokens. However, obtaining such zero-level set points through dense volume sampling is computationally expensive. To bypass this bottleneck, we introduce a

[1]Note that we require only a coarse and rapid approximation of $\hat{x}_0^{(t)}$ using Eq. (3) to establish the anchors. While initially imprecise, they are dynamically refined across subsequent timesteps.

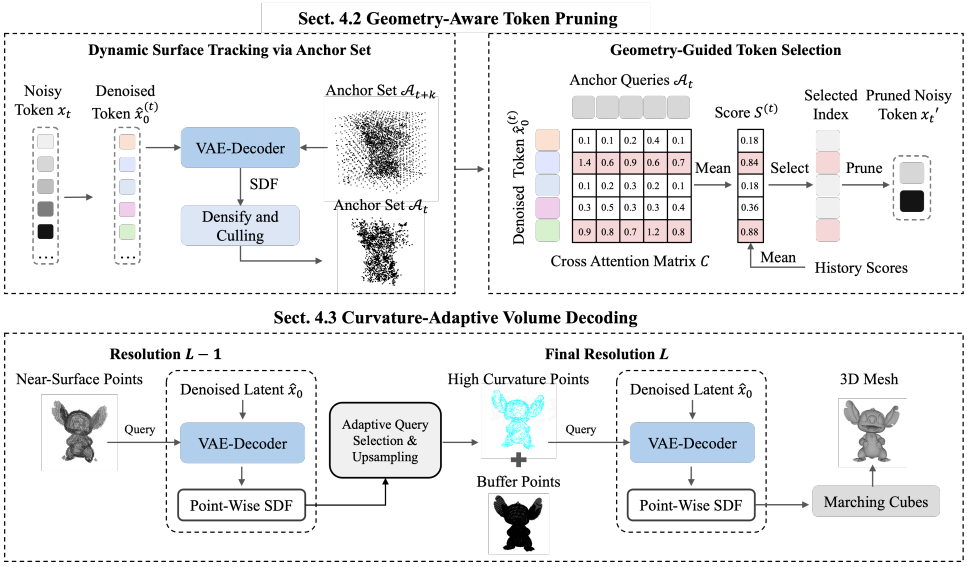

Figure 2: **Overview of our proposed acceleration methods.** We introduce two key strategies to accelerate generation while preserving fidelity: i) Geometry-Aware Token Pruning (Section 4.2) removes geometrically redundant tokens by calculating the cross-attention scores between latent tokens and the dynamic surface, and ii) Curvature-Adaptive Volume Decoding (Section 4.3) accelerates mesh decoding by concentrating high-resolution queries on complex, high-curvature regions.

dynamic surface tracking mechanism, which maintains a sparse set of anchors that iteratively track the evolving surface, enabling geometry-aware token scoring without heavy overhead.

Dynamic Surface Tracking via Anchor Set. To localize the implicit surface without prohibitive computational overhead, we maintain a sparse set of anchor points $\mathcal{A}_t = \{a_i \in \mathbb{R}^3\}_{i=1}^{K_t}$ that approximates the evolving geometry. Initialized as a coarse grid with resolution r_c (e.g., $r_c = 16$), this anchor set is updated via a coarse-to-fine strategy, comprising two key phases:

1) Surface Localization. Given the sparse anchor candidates, we first decode their SDF values. An anchor a_i is identified as a valid surface point if it satisfies either of the following criteria: i) its absolute SDF value falls below a predefined threshold ϵ, or ii) a sign inversion is detected within its local spatial neighborhood $\mathcal{N}(a_i)$:

$$|\text{SDF}(a_i)| < \epsilon, \text{ or } (\exists a_j \in \mathcal{N}(a_i), \text{sgn}(\text{SDF}(a_i)) \neq \text{sgn}(\text{SDF}(a_j))), \tag{4}$$

where $\text{sgn}(\cdot)$ is the signum function that indicates whether a point is inside or outside the surface. Anchors satisfying these conditions are marked as valid active points.

2) Dynamic Densification and Culling. To enhance the fidelity of surface approximation, we implement a dynamic growth mechanism. Anchors that consistently deviate from the surface over consecutive timesteps are identified as redundant and subsequently removed. Conversely, for confirmed surface points, we sample candidate sub-anchors within their local neighborhoods at a higher resolution. Specifically, the number of newly generated points matches the number of discarded ones. This ensures the anchor set $\mathcal{A}_t$ maintains a sparse set with a fixed candidate budget, while the number of valid surface anchors may vary for efficient estimation.

Geometry-Guided Token Selection. Leveraging the geometric correspondence established by the active anchors $\mathcal{A}_t$, we quantify the contribution of each latent token to the surface reconstruction. Let $\mathbf{C}^{(t)} \in \mathbb{R}^{K_t \times N}$ denote the cross-attention map between K_t active anchors and N latent tokens that form the estimated clean set $\hat{x}_0^{(t)} = \{z_j\}_{j=1}^{N}$. We formulate the instantaneous geometry-aware importance score $\tilde{S}_j^{(t)}$ for token z_j by aggregating attention weights exclusively from valid surface anchors:

$$\tilde{S}_j^{(t)} = \sum_{i=1}^{K_t} \mathbf{C}_{i,j}^{(t)}. \tag{5}$$

Since the estimated clean latent $\hat{x}_0^{(t)}$ is imperfect and may contain noise during the denoising process, we need to mitigate the impact of fluctuations in SDF or attention values calculated at individual timesteps. Therefore, we adopt a sliding window smoothing strategy. Instead of relying solely on the current step, we compute the final importance score $S_j^{(t)}$ by averaging the instantaneous scores over the preceding steps within a window of size W:

$$S_j^{(t)} = \frac{1}{L} \sum_{k=0}^{L-1} \tilde{S}_j^{(t+k)}, \text{ where } L = \max(1, \min(T - t + 1, W)). \tag{6}$$

Based on the geometric score S, we adopt a cumulative importance thresholding strategy. Tokens are first ranked in descending order according to their importance scores. Instead of retaining a fixed number of tokens, we dynamically select a subset of top-ranked tokens such that their cumulative score reaches a predefined fraction ρ (e.g., $\rho = 0.8$) of the total score across all tokens. This approach allows the pruning strength to be adjusted adaptively while retaining the most informative tokens.

Following this partition, we maintain a token-wise velocity cache during denoising. Retained tokens participate in the current forward pass and write their newly predicted velocities to the cache; the remaining tokens reuse their most recently cached velocities. The assembled full-length velocity then updates all latent tokens, reducing computation while preserving the complete latent representation. As a result, the model substantially reduces computational overhead while maintaining the geometric structure of the generated 3D object.

We summarize the Geometry-Aware Token Scoring in Algorithm 1 of Appendix B. In Appendix C, we study a decoder-side token-pruning model for a given latent representation. The analysis bounds the expected SDF perturbation in terms of the discarded attention mass and examines how approximating dense-query importance with anchor-based scores affects this bound. These results provide a theoretical motivation for our geometry-aware scoring and cumulative token selection strategy.

4.3 Curvature-Adaptive Volume Decoding

Motivation. To accelerate the decoding process with cubic computational complexity w.r.t. the sampling density of query points (as discussed in Section 3), we employ a **curvature-adaptive** coarse-to-fine sampling strategy. This approach follows the multi-resolution $\{R_1, R_2, \ldots, R_L\}$ framework in Lai et al. (2025b) but further accelerates the sampling at the highest (and slowest) resolution. This is based on the observation that lower resolutions $(R_1, \ldots, R_{L-1})$ facilitate *global surface reconstruction*, whereas the highest resolution (R_L) prioritizes *local detail refinement at high curvature regions*. Consequently, we propose a curvature-adaptive sampling method that dynamically concentrates the computational budget on geometrically complex (i.e., high curvature) regions during the final stage, significantly reducing redundancy while preserving high-fidelity geometric details.

Curvature Estimation via Discrete Laplacian. At coarse resolutions $\{R_1, \ldots, R_{L-1}\}$, we follow the standard protocol Lai et al. (2025b) to identify near-surface voxels $\mathcal{V}_s^{(l)}$ for reliable surface localization, subject to the criteria in Eq. (4). Query points $\mathcal{V}^{(l+1)}$ for the next level R_{l+1} are upsampled from $\mathcal{V}_s^{(l)}$. At the target resolution R_L, we focus on high-frequency regions by selectively sampling points with high curvature Lin et al. (2020).

Specifically, for a given voxel $v \in \mathcal{V}_s^{(l)}$ at the resolution level l, we approximate the local mean curvature $\kappa(v)$ of the underlying implicit surface. To minimize computational overhead, we utilize the discrete Laplacian of the predicted SDF values $\text{SDF}(v)$ as a robust proxy for curvature:

$$\kappa(v) \approx |\Delta \text{SDF}(v)| = |\sum_{v_n \in \mathcal{N}(v)} (\text{SDF}(v_n) - \text{SDF}(v))|, \tag{7}$$

where $\mathcal{N}(v)$ denotes the 6-neighborhood system within the voxel grid, and Δ represents the discrete Laplacian operator. This approximation effectively highlights regions characterized by rapid changes in surface normals.

Adaptive Query Selection. Based on this metric, we define a subset of curvature points $\mathcal{V}_c^{(l)}$, comprising voxels where the geometric variation exceeds a statistical threshold:

$$\mathcal{V}_c^{(l)} = \left\{ v \in \mathcal{V}_s^{(l)} \text{ and } |\kappa(v)| > \tau_{curve} \right\}, \tag{8}$$

where τ_{curve} is the 90th percentile of curvature magnitudes within the current hierarchy level.

To maintain topological continuity and prevent structural holes, we identify a set of buffer points $\mathcal{V}_b^{(l)}$:

$$\mathcal{V}_b^{(l)} = \{v \in \mathcal{V}_s^{(l)} \text{ and } |\text{SDF}(v)| < \epsilon_b\}, \tag{9}$$

where ϵ_b is a tight SDF threshold. The final query set $\mathcal{V}^{(L)}$ is then constructed as:

$$\mathcal{V}^{(L)} = \text{Upsample}\left(\mathcal{V}_c^{(L-1)} \cup \mathcal{V}_b^{(L-1)}\right). \tag{10}$$

This yields a representation significantly sparser than global surface upsampling $\mathcal{V}_s^{(L-1)}$, effectively eliminating computational redundancy in smooth regions while adaptively allocating capacity to capture fine-grained details. We provide the pseudocode of the Curvature-Adaptive Volume Decoding in Algorithm 2 of Appendix B.

5 Experiments

We validate the efficacy of our proposed methods by integrating them into two state-of-the-art VecSet-based 3D generation methods: TripoSG Li et al. (2025d), and Hunyuan3D-2.0 Zhao et al. (2025b). We begin by detailing the experimental setup in Section 5.1, followed by a comprehensive analysis of the quantitative results in Section 5.2. In Section 5.3, we present ablation studies to illustrate the trade-offs between generation efficiency and geometric fidelity.

5.1 Experimental Setup

Backbone Models. To demonstrate the applicability and effectiveness of our approach, we adopt TripoSG Li et al. (2025d) and Hunyuan3D-2.0 Zhao et al. (2025b) as our backbone models. TripoSG extends the DiT framework Peebles & Xie (2023) and is trained via Rectified Flow with a sequence length of 2048 latent tokens. In contrast, Hunyuan3D-2.0 Zhao et al. (2025b) adopts a dual- and single-stream network structure inspired by FLUX Labs et al. (2025) and utilizes Flow Matching for training with 3072 latent tokens. We employ these two distinct architectures to validate the broad applicability of our method across different model structures. Implementation details can be found in Appendix D.

State-of-the-art Methods for Comparison. We benchmark our method against recent token-wise diffusion acceleration methods, including RAS Liu et al. (2025c) and SADA Jiang et al. (2025), adapted to the evaluated VecSet backbones. Since SADA Jiang et al. (2025) simultaneously performs step-wise and token-wise acceleration, we isolate our comparison to its token-pruning variant for evaluation. These comparisons allow us to assess the efficiency–fidelity trade-off of our complete framework against existing token-wise accelerators.

Datasets and Evaluation Setup. For quantitative evaluation, we follow the protocol established by TRELLIS Xiang et al. (2025b), using all 3,229 objects in the Toys4K evaluation set Stojanov et al. (2021). For each object, we render a condition image from the ground-truth mesh under a fixed frontal viewpoint.

We evaluate geometric fidelity using Chamfer Distance (CD), F-Score at a threshold of 0.1, Intersection over Union (IoU), Normal Consistency (NC), and Normal-FID (N-FID). Following TripoSG Li et al. (2025d), Normal-FID compares normal-map renderings of generated and ground-truth meshes across the evaluation set; lower values indicate better agreement and may reflect visible defects not fully captured by the other metrics. To evaluate efficiency, we report denoising, decoding, and total inference time ($\text{T}_{denoise}$ / T_{decode} / T_{total}) in seconds. Inference latency is measured on a single NVIDIA GeForce RTX 4090 GPU with 50 inference steps.

5.2 Main Results

Experimental Results on TripoSG. As shown in Table 1, our method achieves a $1.50\times$ end-to-end speedup on TripoSG, reducing total inference time from 8.13s to 5.42s. It is the fastest of the evaluated acceleration methods and improves all five geometric metrics over RAS. Compared with SADA, which achieves a $1.28\times$ speedup, our method achieves higher IoU and NC and, most notably, reduces

Table 1: **Quantitative comparison on the TripoSG Li et al. (2025d) and Hunyuan3D-2.0 Zhao et al. (2025b) benchmark.** The **best** results are bolded and underlined, and the second-best results are underlined (excluding Vanilla).

Methods	CD↓	F-Score↑	IoU↑	NC↑	N-FID↓	$T_{denoise}$↓	T_{decode}↓	T_{total} (Speedup)↓
Framework: TripoSG								
Vanilla	0.0665	0.9147	0.5410	0.7381	11.32	6.34s	1.79s	8.13s (1.00×)
RAS	0.0729	0.8973	0.5127	0.7175	15.39	**4.42s**	1.77s	6.19s (1.31×)
SADA (Token Prune)	**0.0668**	**0.9143**	0.5359	0.7315	14.93	4.60s	1.76s	6.36s (1.28×)
Ours	0.0671	0.9125	**0.5369**	**0.7343**	**13.07**	4.67s	**0.75s**	**5.42s** (1.50×)
Framework: Hunyuan3D-2.0								
Vanilla	0.0671	0.9111	0.5491	0.7387	11.41	9.33s	0.68s	10.01s (1.00×)
RAS	0.0685	0.9088	0.5305	0.7276	15.53	6.90s	0.65s	7.55s (1.33×)
SADA (Token Prune)	0.0677	0.9116	0.5423	0.7316	16.61	7.53s	0.80s	8.33s (1.20×)
Ours	**0.0670**	**0.9146**	**0.5491**	**0.7381**	**12.45**	**6.42s**	**0.36s**	**6.78s** (1.48×)

N-FID by 12.5% (from 14.93 to 13.07), while maintaining comparable CD and F-Score. Together, these results show that joint denoising and decoding acceleration delivers substantial end-to-end gains with strong geometric fidelity among competing accelerators. Qualitative visual comparisons are provided in Appendix E.

Experimental Results on Hunyuan3D-2.0. To assess whether the same framework transfers to another VecSet generator, we further evaluate it on Hunyuan3D-2.0. As shown in Table 1, our method reduces total inference time from 10.01s to 6.78s, corresponding to a $1.48\times$ speedup. Among the evaluated acceleration methods, our method achieves both the lowest overall latency and the best performance across all five geometric metrics. Together with the TripoSG results, this performance indicates that the same framework can deliver strong end-to-end acceleration and competitive geometric fidelity on VecSet backbones with different transformer designs and latent-token counts.

5.3 Ablation Study

Analysis of Curvature-Aware Sampling. To evaluate our curvature-adaptive sampling strategy, we perform an ablation study on TripoSG, using the hierarchical sampling of Lai et al. (2025b) as the baseline (the "w/o" setting in Table 2). Relative to this baseline, our curvature-adaptive sampling reduces decoding latency from 1.77s to 0.75s, yielding a $2.36\times$ decoding speedup, and lowers total inference time from 6.45s to 5.42s. Our method with curvature-adaptive sampling achieves a CD of 0.0671 and an F-Score of 0.9125, slightly improving upon the "w/o" setting (0.0676 and 0.9120, respectively). These results suggest that curvature-adaptive sampling substantially accelerates decoding with geometric quality comparable to that of the "w/o" setting.

Table 2: **Ablation study on the Curvature-Aware sampling strategy.**

Setting	Curve	CD↓	F-Score↑	IoU↑	NC↑	N-FID↓	$T_{denoise}$↓	T_{decode}↓	T_{total}↓
Vanilla	N/A	0.0665	0.9147	0.5410	0.7381	11.32	6.34s	1.79s	8.13s
Ours	w/o	0.0676	0.9120	**0.5375**	**0.7356**	**12.85**	4.68s	1.77s	6.45s
	w/	**0.0671**	**0.9125**	0.5369	0.7343	13.07	**4.67s**	**0.75s**	**5.42s**

Effects of the cumulative Threshold (ρ). The cumulative threshold ρ determines the number of retained tokens, thereby directly trading off generation quality and computational efficiency. As shown in Table 3, increasing ρ generally improves geometric fidelity at the cost of higher denoising latency. At lower thresholds (e.g., $\rho = 0.6, 0.7$), the model achieves faster inference speed but suffers from geometric degradation, as indicated by the higher Normal-FID. As ρ increases to 0.9, the generation quality achieves the best values for F-score, IoU, NC, and Normal-FID, yet the denoising latency increases to 4.93s. Setting $\rho = 0.8$ achieves the lowest CD (0.0671) and improves all geometric metrics relative to the lower thresholds, while keeping total inference time below that at $\rho = 0.9$ (5.42s versus 5.67s). Therefore, we choose $\rho = 0.8$ in our main experiments to balance generation performance and computational efficiency. A detailed analysis of token retention across 10-step denoising intervals for different cumulative thresholds is provided in Appendix G.

Table 3: **Ablation study on the cumulative threshold (ρ).**

Setting	Metrics							
ρ	CD↓	F-Score↑	IoU↑	NC↑	N-FID↓	T_{denoise}↓	T_{decode}↓	T_{total}↓
0.6	0.0680	0.9111	0.5305	0.7235	16.62	**4.04s**	0.79s	**4.83s**
0.7	0.0679	0.9120	0.5339	0.7292	14.67	4.27s	0.77s	5.04s
0.8	**0.0671**	0.9125	0.5369	0.7343	13.07	4.67s	0.75s	5.42s
0.9	0.0672	**0.9127**	**0.5440**	**0.7379**	**12.31**	4.93s	**0.74s**	5.67s

Analysis of Sliding Window Size. Since the estimated clean latent $\hat{x}_0^{(t)}$ is derived from noisy intermediate states, relying solely on the current timestep ($W = 1$) may lead to fluctuating token-importance scores. As shown in Table 4, increasing W from 1 to 3 steadily reduces N-FID from 13.40 to 13.07. The $W = 3$ setting also achieves the lowest CD (0.0671), the highest F-Score (0.9125), and the highest IoU (0.5369). The improvements provide evidence that temporal aggregation enhances geometric fidelity. We therefore use $W = 3$ in the main experiments.

Table 4: **Ablation study on the sliding window size (W).**

Setting	Metrics							
W	CD↓	F-Score↑	IoU↑	NC↑	N-FID↓	T_{denoise}↓	T_{decode}↓	T_{total}↓
1	0.0676	0.9117	0.5363	**0.7347**	13.40	**4.60s**	0.76s	**5.36s**
2	0.0675	0.9115	0.5361	**0.7347**	13.28	4.65s	**0.75s**	5.40s
3	**0.0671**	**0.9125**	**0.5369**	0.7343	**13.07**	4.67s	**0.75s**	5.42s

Ablation on Initial Anchor Resolution. We investigate the impact of the initial anchor grid resolution r_c on geometric fidelity and inference efficiency. The number of initial candidate anchors scales as r_c^3. As shown in Figure 3, increasing r_c from 6 to 16 reduces N-FID from 13.86 to 13.07 while increasing denoising time from 4.62s to 4.67s. Beyond $r_c = 16$, N-FID changes little, ranging from 13.04 to 13.06 for $r_c \in \{20, 24, 32\}$, while denoising time reaches 4.76s at $r_c = 32$. To achieve a favorable balance between geometric quality and inference efficiency, we select $r_c = 16$ as our default setting.

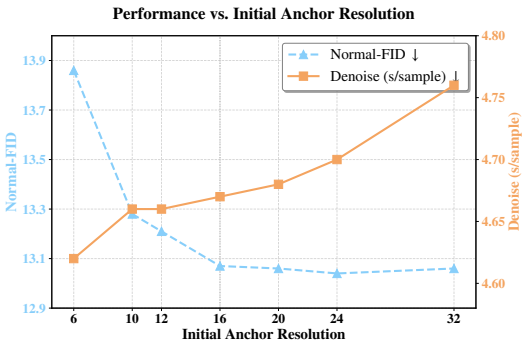

Figure 3: **Ablation analysis on the initial anchor resolution.** N-FID and denoising time for $r_c \in \{6, 10, 12, 16, 20, 24, 32\}$.

Beyond the above ablations, we further evaluate geometric fidelity on objects with thin structures and analyze the computational overhead of each proposed component. The corresponding analyses are presented in Appendix F and H, respectively.

6 Conclusion and Limitations

Conclusion. In this paper, we presented a unified acceleration framework for the popular VecSet-based 3D generation that leverages geometry preservation as a guiding principle. The key idea is to allocate computation according to geometric contribution: during denoising, Geometry-Aware Token Pruning retains the latent tokens most critical to the evolving surface, while during mesh extraction, curvature-adaptive volume decoding focuses high-resolution queries on geometrically complex regions. Extensive experiments on TripoSG and Hunyuan3D-2.0 validate the effectiveness of our approach, delivering overall inference speedups of $1.50\times$ and $1.48\times$, respectively, while largely preserving geometric fidelity in the generated 3D results.

Limitations. While this work mainly focuses on accelerating the geometry generation stage and evaluates the reconstructed meshes using standard geometric metrics, the acceleration of texture synthesis is not explicitly investigated. We leave this as a potential future direction, which may provide additional efficiency benefits for complete 3D generation pipelines.

AI USE STATEMENT

In this work, we used generative AI tools to edit selected portions of the manuscript. We did not use generative AI tools to generate synthetic datasets, clean or reformat datasets, or create or modify scientific figures or images. The authors reviewed all AI-assisted work and checked the accuracy of the manuscript sections edited with AI assistance. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4599–4603, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing 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. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16251–16261, 2025.

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, 2023.

Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18083–18092, 2025.

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. *arXiv preprint arXiv:2503.21732*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ting Jiang, Yixiao Wang, Hancheng Ye, Zishan Shao, Jingwei Sun, Jingyang Zhang, Zekai Chen, Jianyi Zhang, Yiran Chen, and Hai Li. Sada: Stability-guided adaptive diffusion acceleration. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. 2025.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025a.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025b.

Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dso: Aligning 3d generators with simulation feedback for physical soundness. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6772–6783. IEEE, 2025a.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5307–5317, 2025b.

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025c.

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*, 2025d.

Jiehong Lin, Xian Shi, Yuan Gao, Ke Chen, and Kui Jia. Cad-pu: A curvature-adaptive deep learning solution for point set upsampling. *arXiv preprint arXiv:2009.04660*, 2020.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. *arXiv preprint arXiv:2503.06923*, 2025b.

Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. *arXiv preprint arXiv:2502.10389*, 2025c.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37: 133282–133304, 2024a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

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.

Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Yanyu Li, Yifan Gong, Kai Zhang, Hao Tan, Jason Kuen, Henghui Ding, et al. Lazydit: Lazy learning for the acceleration of diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20409–20417, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

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.

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6211–6220, 2024.

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, 2024.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Philip Torr, Xun Cao, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

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.

Jiajian Xie, Hubery Yin, Chen Li, Zhou Zhao, and Shengyu Zhang. ETC: training-free diffusion models acceleration with error-aware trend consistency. *arXiv preprint arXiv:2510.24129*, 2025.

Mengyu Yang, Yanming Yang, Chenyi Xu, Chenxi Song, Yufan Zuo, Tong Zhao, Ruibo Li, and Chi Zhang. Fast3dcache: Training-free 3d geometry synthesis acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27030–27040, 2026.

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. *arXiv preprint arXiv:2503.22236*, 3:2, 2025.

Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. In *Advances in neural information processing systems*, 2024.

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

Evelyn Zhang, Jiayi Tang, Xuefei Ning, and Linfeng Zhang. Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9878–9886, 2025.

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 Transactions on Graphics (TOG)*, 43(4):1–20, 2024.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *International Conference on Learning Representations*, 2025a.

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 *Advances in neural information processing systems*, 2023.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025b.

Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

Zanwei Zhou, Taoran Yi, Jiemin Fang, Chen Yang, Lingxi Xie, Xinggang Wang, Wei Shen, and Qi Tian. Few-step flow for 3d generation via marginal-data transport distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13853–13861, 2026.

Chang Zou, Evelyn Zhang, Runlin Guo, Haohang Xu, Conghui He, Xuming Hu, and Linfeng Zhang. Accelerating diffusion transformers with dual feature caching. *arXiv preprint arXiv:2412.18911*, 2024.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *International Conference on Learning Representations*, 2025.

A Analysis on Existing 3D Diffusion Acceleration Methods

In this section, we discuss existing 3D generation acceleration methods, which broadly fall into training-free and training-based approaches. **Training-free methods** include Fast3Dcache Yang et al. (2026), which uses voxel-occupancy stability and token velocity dynamics to guide cache reuse during denoising. **Training-based methods**, including FlashVDM Lai et al. (2025b) and MDT-dist Zhou et al. (2026), instead reduce the number of sampling steps by distilling pretrained generators into student models.

To better illustrate the differences between these methods and our methods, Table 5 summarizes their representations, acceleration mechanisms, training requirements, and plug-and-play applicability.

Table 5: **Comparison with existing 3D diffusion acceleration methods.**

Method	Representation / backbone	Accelerated component	Training-free	Training workload	Plug-and-play applicability
MDT-dist Zhou et al. (2026)	SLAT / TRELLIS	Diffusion distillation	No	Diffusion: 10K steps; Batch size: 256	No; backbone-specific training
FlashVDM Lai et al. (2025b)	VecSet / Hunyuan3D-2.0	Diffusion distillation + Efficient decoder + Hierarchical decoding	No	Diffusion: 53K steps; Decoder fine-tuning: up to 800K steps; Batch size: 256	No; backbone-specific training
Fast3Dcache Yang et al. (2026)	SLAT / TRELLIS, DSO	Structure-stage token caching	Yes	/	For **SLAT-based** structure generation
HGA3D (ours)	VecSet / TripoSG, Hunyuan3D-2.0	Token pruning + Curvature-adaptive decoding	Yes	/	For **Vecset-based** generation

All steps are optimization steps, and "/" denotes no additional training.

As summarized in Table 5, distillation-based methods Zhou et al. (2026); Lai et al. (2025b) require additional training to obtain student models for few-step sampling, whereas our methods and Fast3Dcache Yang et al. (2026) directly accelerate pretrained backbones without additional training. For example, MDT-dist Zhou et al. (2026) requires 10K optimization steps across its structure and SLAT models, while FlashVDM Lai et al. (2025b) uses 53K steps across its diffusion training stages. The full FlashVDM system additionally fine-tunes a redesigned lightweight decoder for up to 800K optimization steps; its hierarchical decoding strategy itself is training-free. The resulting distilled models are backbone-specific, so extending these methods to a different backbone requires corresponding training.

Although Fast3DcacheYang et al. (2026) and our HGA3D method share this training-free property, their plug-and-play applicability differs across representations. Specifically, Fast3Dcache is tailored to sparse-structure generation in SLAT-based models, using the stability of their explicit voxel occupancy to determine caching budgets. It supports plug-and-play acceleration of SLAT-based backbones such as TRELLIS Xiang et al. (2025b) and DSO Li et al. (2025a). In contrast, our method targets the VecSet representation and enables plug-and-play acceleration of pretrained VecSet backbones, including TripoSG Li et al. (2025d) and Hunyuan3D-2.0Zhao et al. (2025b).

We include FlashVDM's training-free hierarchical decoding strategy as a baseline in our decoding ablation (Section 5.3), providing a component-level comparison. At the full-system level, however, the differences in representation and training requirements discussed above complicate a fair comparison with our method. Our main experiments instead compare our method with token-wise acceleration methods adapted from 2D diffusion, using the same pretrained VecSet backbones. Specifically, we evaluate RAS Liu et al. (2025c) and the token-pruning method of SADA Jiang et al. (2025) to assess the efficiency–fidelity trade-off of our complete framework against existing token-wise accelerators.

B ALGORITHMS FOR PROPOSED ACCELERATION STRATEGIES.

In this section, we provide the detailed pseudocode for our proposed acceleration framework. Algorithm 1 outlines the Geometry-Aware Token Scoring Procedure, which identifies geometrically critical tokens by leveraging the cross-attention map between latent tokens and the dynamic surface. Furthermore, Algorithm 2 describes the Curvature-Adaptive Volume Decoding process, which concentrates computational resources on high-curvature geometric regions during mesh decoding.

Algorithm 1 Geometry-Aware Token Scoring

Input: Noisy Latent x_t at timestep t, Previous Anchors $\mathcal{A}_{t+1}$
Output: Importance Scores $S^{(t)}$, Updated Anchors $\mathcal{A}_t$
▷ Clean Data Approximation
Estimate clean latent: $\hat{x}_0^{(t)} = x_t - \frac{t}{T} v_\theta(x_t, \frac{t}{T} \mid c)$
▷ Surface Anchor Localization
Decode SDF values: $\text{SDF}(\mathcal{A}_{t+1}) \leftarrow \text{Decoder}(\mathcal{A}_{t+1}, \hat{x}_0^{(t)})$
Identify surface indicator δ_i for anchor $a_i \in \mathcal{A}_{t+1}$:
    $\delta_i \leftarrow (|\text{SDF}(a_i)| < \epsilon) \ \vee \ (\exists a_j \in \mathcal{N}(a_i), \text{sgn}(\text{SDF}(a_i)) \neq \text{sgn}(\text{SDF}(a_j)))$
▷ Dynamic Densification and Culling
Cull outliers: $\mathcal{O} \leftarrow \{a_i \in \mathcal{A}_{t+1} \mid \delta_i = 0\}$; $\mathcal{A}_{keep} \leftarrow \mathcal{A}_{t+1} \setminus \mathcal{O}$
Sample replacements: $\mathcal{A}_{new} \leftarrow \text{Sample}(\{a_i \mid \delta_i = 1\}, |\mathcal{O}|)$
Update anchor set: $\mathcal{A}_t \leftarrow \mathcal{A}_{keep} \cup \mathcal{A}_{new}$
▷ Geometry-Guided Token Scoring
Compute cross-attention: $\mathbf{C}^t \leftarrow \text{CrossAttention}(\mathcal{A}_t, \hat{x}_0^{(t)})$
Compute instantaneous scores $\tilde{S}_j^{(t)} \leftarrow \sum_{i=1}^{K_t} \mathbf{C}_{i,j}^{(t)}$
Apply sliding window smoothing: $S_j^{(t)} \leftarrow \frac{1}{L} \sum_{k=0}^{L-1} \tilde{S}_j^{(t+k)}$, where $L = \max(1, \min(T - t + 1, W))$
return $S^{(t)}, \mathcal{A}_t$

Algorithm 2 Curvature-Adaptive Volume Decoding

Input: Denoised Latent $\hat{x}_0$, Resolutions $\{R_1, \ldots, R_L\}$, Curvature Threshold τ_{curve}, SDF Thresholds ϵ, ϵ_b
Output: Final SDF Volume $\mathcal{V}_{final}$
Initialize query points $\mathcal{V}^{(1)}$ on coarse grid R_1
for $l = 1$ **to** L **do**
    ▷ SDF Prediction
    Query Decoder: $\text{SDF}(\mathcal{V}^{(l)}) \leftarrow \text{Decoder}(\mathcal{V}^{(l)}, \hat{x}_0)$
    if $l < L$ **then**
        ▷ Surface Region Localization
        Identify near-surface voxels $\mathcal{V}_s^{(l)}$:
            $\mathcal{V}_s^{(l)} \leftarrow \{v \in \mathcal{V}^{(l)} \mid |\text{SDF}(v)| < \epsilon \vee \text{SignChange}(\mathcal{N}(v))\}$
        ▷ Adaptive Query Selection
        if $l < L - 1$ **then**
            // Intermediate levels: Global surface reconstruction
            Upsample all surface points: $\mathcal{V}^{(l+1)} \leftarrow \text{Upsample}(\mathcal{V}_s^{(l)})$
        else
            // Final level (R_L): Curvature-adaptive sampling
            Compute discrete Laplacian: $\kappa(v) \leftarrow |\Delta\text{SDF}(v)| \approx |\sum_{v_n \in \mathcal{N}(v)}(\text{SDF}(v_n) - \text{SDF}(v))|$
            Identify high-curvature set $\mathcal{V}_c^{(l)}$: $\mathcal{V}_c^{(l)} \leftarrow \{v \in \mathcal{V}_s^{(l)} \mid \kappa(v) > \tau_{curve}\}$
            Identify buffer set $\mathcal{V}_b^{(l)}$: $\mathcal{V}_b^{(l)} \leftarrow \{v \in \mathcal{V}_s^{(l)} \mid |\text{SDF}(v)| < \epsilon_b\}$
            Construct final sparse query set: $\mathcal{V}^{(L)} \leftarrow \text{Upsample}(\mathcal{V}_c^{(L-1)} \cup \mathcal{V}_b^{(L-1)})$
        end if
    end if
end for
return Interpolate $\text{SDF}(\mathcal{V}^{(L)})$ to dense volume V_{final}

C THEORETICAL MOTIVATION FOR GEOMETRY-AWARE TOKEN SCORING

In this section, we analyze how removing latent tokens from decoder attention affects the reconstructed SDF for a given latent representation. The resulting bound relates reconstruction perturbation to the attention mass of the removed tokens, motivating the use of cumulative attention importance for token selection. We then analyze how approximating dense-query importance with anchor-based scores affects the reconstruction error bound.

Definition C.1 (Attention-aggregated query feature)**.** Let $Z = \{z_i\}_{i=1}^N$ denote the latent tokens. For a spatial query point q, let $\alpha_i(q) \geq 0$ denote the normalized decoder cross-attention weight assigned to token z_i, satisfying

$$\sum_{i=1}^{N} \alpha_i(q) = 1. \tag{11}$$

Let $v_i = V(z_i)$ denote the projected value vector of token z_i. The attention-aggregated query feature is defined as

$$v_q = \sum_{i=1}^{N} \alpha_i(q) v_i. \tag{12}$$

Definition C.2 (Dense-query token importance and discarded attention mass)**.** Let $\mathcal{Q}$ denote the dense query points used for final SDF decoding, and let μ_Q be the empirical distribution over these queries. The dense-query token-importance distribution is defined as

$$p_Q(i) = \mathbb{E}_{q \sim \mu_Q}[\alpha_i(q)]. \tag{13}$$

Given a retained token subset $\mathcal{S} \subseteq \{1, \ldots, N\}$, we define the discarded dense-query attention mass as

$$\epsilon_Q = \sum_{i \notin \mathcal{S}} p_Q(i). \tag{14}$$

Assumption C.3 (Bounded value vectors)**.** There exists a constant $M > 0$ such that the projected value vectors are uniformly bounded:

$$\|v_i\|_2 \leq M, \quad \forall i \in \{1, \ldots, N\}. \tag{15}$$

Assumption C.4 (Local Lipschitz continuity of the decoder)**.** Let f_θ denote the decoder mapping from the attention-aggregated query feature to the predicted SDF value:

$$f_\theta : v_q \mapsto \mathrm{SDF}(q). \tag{16}$$

For each query q, f_θ denotes the remaining decoder mapping with other inputs held fixed. We assume a common Lipschitz constant over all queries in $\mathcal{Q}$ and a feature region containing the original features and those obtained from the retained token subsets considered below. That is, there exists a constant $L > 0$ such that

$$|f_\theta(v_q) - f_\theta(\tilde{v}_q)| \leq L \|v_q - \tilde{v}_q\|_2. \tag{17}$$

In this attention-aggregation model, token removal leaves the queries, keys, and values of retained tokens unchanged and only renormalizes their attention weights.

Lemma C.5 (Per-query feature perturbation)**.** *For a final decoding query $q \in \mathcal{Q}$, define the retained attention mass as*

$$S_q = \sum_{i \in \mathcal{S}} \alpha_i(q). \tag{18}$$

Assume $S_q > 0$, which holds when the retained set is non-empty under softmax attention. After pruning, the retained attention weights are re-normalized, and the pruned query feature is given by

$$\tilde{v}_q = \sum_{i \in \mathcal{S}} \frac{\alpha_i(q)}{S_q} v_i. \tag{19}$$

Under Assumption C.3, the feature perturbation $\Delta v_q = v_q - \tilde{v}_q$ satisfies

$$\|\Delta v_q\|_2 \leq 2M(1 - S_q). \tag{20}$$

Proof. By definition, the original query feature can be decomposed as

$$v_q = \sum_{i \in \mathcal{S}} \alpha_i(q) v_i + \sum_{j \notin \mathcal{S}} \alpha_j(q) v_j. \tag{21}$$

The pruned query feature is

$$\tilde{v}_q = \sum_{i \in \mathcal{S}} \frac{\alpha_i(q)}{S_q} v_i. \tag{22}$$

Therefore,

$$\Delta v_q = \sum_{i \in \mathcal{S}} \left(\alpha_i(q) - \frac{\alpha_i(q)}{S_q} \right) v_i + \sum_{j \notin \mathcal{S}} \alpha_j(q) v_j. \tag{23}$$

Taking the ℓ_2 norm and applying the triangle inequality yields

$$\|\Delta v_q\|_2 \le \sum_{i \in \mathcal{S}} \alpha_i(q) \left| \frac{1}{S_q} - 1 \right| \|v_i\|_2 + \sum_{j \notin \mathcal{S}} \alpha_j(q) \|v_j\|_2 \tag{24}$$

$$\le M \sum_{i \in \mathcal{S}} \alpha_i(q) \frac{1 - S_q}{S_q} + M \sum_{j \notin \mathcal{S}} \alpha_j(q). \tag{25}$$

Since $\sum_{i \in \mathcal{S}} \alpha_i(q) = S_q$ and $\sum_{j \notin \mathcal{S}} \alpha_j(q) = 1 - S_q$, we have

$$\|\Delta v_q\|_2 \le M S_q \frac{1 - S_q}{S_q} + M(1 - S_q) = 2M(1 - S_q). \tag{26}$$

This completes the proof. □

Proposition C.6 (Expected feature perturbation)**.** *Under Assumption C.3, the expected feature perturbation over the final dense decoding queries is bounded by*

$$\mathbb{E}_{q \sim \mu_Q} \|\Delta v_q\|_2 \le 2M \epsilon_Q. \tag{27}$$

Proof. Taking expectation in Lemma C.5 over $q \sim \mu_Q$ gives

$$\mathbb{E}_{q \sim \mu_Q} \|\Delta v_q\|_2 \le 2M \mathbb{E}_{q \sim \mu_Q}[1 - S_q] \tag{28}$$

$$= 2M \left(1 - \mathbb{E}_{q \sim \mu_Q} S_q\right). \tag{29}$$

By the definition of S_q,

$$\mathbb{E}_{q \sim \mu_Q} S_q = \mathbb{E}_{q \sim \mu_Q} \left[\sum_{i \in \mathcal{S}} \alpha_i(q) \right] \tag{30}$$

$$= \sum_{i \in \mathcal{S}} \mathbb{E}_{q \sim \mu_Q}[\alpha_i(q)] = \sum_{i \in \mathcal{S}} p_Q(i). \tag{31}$$

Thus,

$$\mathbb{E}_{q \sim \mu_Q}[1 - S_q] = 1 - \sum_{i \in \mathcal{S}} p_Q(i) = \sum_{i \notin \mathcal{S}} p_Q(i) = \epsilon_Q. \tag{32}$$

Therefore,

$$\mathbb{E}_{q \sim \mu_Q} \|\Delta v_q\|_2 \le 2M \epsilon_Q. \tag{33}$$

This completes the proof. □

Theorem C.7 (Expected SDF perturbation bound)**.** *Suppose Assumptions C.3 and C.4 hold. Then the expected SDF perturbation over the final dense decoding queries satisfies*

$$\mathbb{E}_{q \sim \mu_Q} \left[|\Delta \mathrm{SDF}(q)| \right] \le 2LM \epsilon_Q, \tag{34}$$

where

$$\Delta \mathrm{SDF}(q) = f_\theta(v_q) - f_\theta(\tilde{v}_q). \tag{35}$$

Proof. By Assumption C.4, for any $q \in \mathcal{Q}$, we have

$$|\Delta\mathrm{SDF}(q)| = |f_\theta(v_q) - f_\theta(\tilde{v}_q)| \le L\|v_q - \tilde{v}_q\|_2 = L\|\Delta v_q\|_2. \tag{36}$$

Taking expectation over $q \sim \mu_Q$ gives

$$\mathbb{E}_{q\sim\mu_Q}\left[|\Delta\mathrm{SDF}(q)|\right] \le L\mathbb{E}_{q\sim\mu_Q}\|\Delta v_q\|_2. \tag{37}$$

Applying Proposition C.6, we obtain

$$\mathbb{E}_{q\sim\mu_Q}\left[|\Delta\mathrm{SDF}(q)|\right] \le 2LM\epsilon_Q. \tag{38}$$

This completes the proof. □

Remark C.8 (Token selection under a fixed budget). For a given latent representation and query distribution, define the upper bound in Theorem C.7 as

$$U(\mathcal{S}) = 2LM\left(1 - \sum_{i\in\mathcal{S}} p_Q(i)\right). \tag{39}$$

Under the stated assumptions, the same constants L and M apply to the candidate retained subsets. For a token budget $m \in \{1, \dots, N\}$, selecting the m tokens with the largest $p_Q(i)$ maximizes the retained importance and therefore minimizes $U(\mathcal{S})$ over all subsets of size m. This gives a direct rationale for importance-ranked selection in the decoder-side model.

Dense-query importance provides the reference criterion in this analysis. Anchor-based scoring offers a less expensive proxy, and the following corollary accounts for its discrepancy from that reference.

Corollary C.9 (A sufficient condition via anchor-estimated importance)**.** *Let $\mathcal{A}$ denote the spatial queries used for anchor-based scoring, and let μ_A be their empirical distribution. For the same latent representation used to define p_Q, define the anchor-induced token-importance distribution as*

$$p_A(i) = \mathbb{E}_{a\sim\mu_A}[\alpha_i(a)]. \tag{40}$$

If the retained token subset $\mathcal{S}$ preserves at least ρ cumulative anchor-estimated importance,

$$\sum_{i\in\mathcal{S}} p_A(i) \ge \rho, \tag{41}$$

and the anchor-induced importance distribution differs from the dense-query importance distribution by at most η,

$$\|p_A - p_Q\|_1 \le \eta, \tag{42}$$

then the expected SDF perturbation satisfies

$$\mathbb{E}_{q\sim\mu_Q}\left[|\Delta\mathrm{SDF}(q)|\right] \le 2LM(1 - \rho + \eta). \tag{43}$$

Proof. Since $\sum_{i\in\mathcal{S}} p_A(i) \ge \rho$ and $\|p_A - p_Q\|_1 \le \eta$, we have

$$\sum_{i\in\mathcal{S}} p_Q(i) = \sum_{i\in\mathcal{S}} p_A(i) + \sum_{i\in\mathcal{S}} (p_Q(i) - p_A(i)) \tag{44}$$

$$\ge \sum_{i\in\mathcal{S}} p_A(i) - \sum_{i\in\mathcal{S}} |p_Q(i) - p_A(i)| \tag{45}$$

$$\ge \rho - \|p_Q - p_A\|_1 \ge \rho - \eta. \tag{46}$$

Therefore,

$$\epsilon_Q = \sum_{i\notin\mathcal{S}} p_Q(i) = 1 - \sum_{i\in\mathcal{S}} p_Q(i) \le 1 - \rho + \eta. \tag{47}$$

Substituting this into Theorem C.7 yields

$$\mathbb{E}_{q\sim\mu_Q}\left[|\Delta\mathrm{SDF}(q)|\right] \le 2LM(1 - \rho + \eta). \tag{48}$$

This completes the proof. □

The discrepancy term η quantifies how well anchor-induced importance represents dense-query importance. The bound accounts for both the retained anchor importance and this discrepancy, providing a rationale for cumulative selection based on anchor scores.

Remark C.10 (Connection to stable mesh extraction). Under the stated assumptions, Theorem C.7 bounds the expected SDF perturbation in terms of the discarded attention mass. During Marching Cubes extraction, sign flips at grid vertices are a direct source of mesh instability. For a grid vertex x, if the original SDF value satisfies $|\mathrm{SDF}(x)| > E$ and the perturbation satisfies $|\Delta\mathrm{SDF}(x)| \leq E$, then the sign of that vertex is preserved. Thus, reducing the SDF perturbation can mitigate sign-flip risks during mesh extraction.

Remark C.11 (Interpretation of the retention threshold). The sufficient condition in Corollary C.9 links the retention threshold ρ to the SDF perturbation bound. The term $1 - \rho$ bounds the discarded anchor importance, while η accounts for its discrepancy from dense-query importance. Together, these terms give a quantitative interpretation of cumulative importance thresholding and its relation to the decoder-side reconstruction bound.

D Implementation Details.

In this section, we provide the detailed hyperparameters utilized during the inference stage across our proposed acceleration modules. To limit drift during velocity reuse, we periodically evaluate all latent tokens and refresh their cached output velocities. Full computation also initializes the cache and is used outside the acceleration range. Empirically, we set this full-computation interval to 3 for TripoSG and 4 for Hunyuan3D-2.0. The hierarchical resolutions for the decoding stage are set to (64, 128, 256, 512), and consistent with the official configurations of the respective backbones, the classifier-free guidance (CFG) scale is set to 7.0 for TripoSG and 5.0 for Hunyuan3D-2.0.

E Visualization Results.

Figure 4 presents qualitative comparisons on the TripoSG benchmark. As demonstrated, our method produces high-fidelity 3D assets that are visually comparable to the vanilla baseline. Crucially, it maintains structural integrity and fine details without the degradation typically associated with acceleration, thereby achieving a superior trade-off between inference efficiency and generation fidelity.

F Thin Structure Evaluation.

A potential concern with curvature-adaptive decoding is that thin structures with low local curvature may receive insufficient high-resolution query coverage. However, our method also retains near-surface buffer points, which supplement the curvature-guided queries and are intended to mitigate this risk. To evaluate the complete framework on such objects, we conduct a targeted evaluation on objects containing representative thin structures.

Specifically, we construct a test subset from ShapeNetCore Chang et al. (2015), comprising three categories with representative thin structures: knives (blades), guitars (body plates and necks), and lamps (lampshades and slender supports). We randomly select up to 500 models from each category. We then apply a consistent aesthetic-score filter across all categories to exclude low-quality models, following TripoSG Li et al. (2025d). This procedure yields 764 test samples: 144 knives, 398 guitars, and 222 lamps.

As shown in Table 6, our method achieves the lowest end-to-end latency on both backbones, with speedups of $1.43\times$ on TripoSG and $1.45\times$ on Hunyuan3D-2.0. On TripoSG, it also obtains the best CD, F-Score, IoU, and NC among the acceleration methods, although SADA achieves a lower N-FID. On Hunyuan3D-2.0, our method achieves the best CD and F-Score among the acceleration methods, while ranking second on IoU, NC, and N-FID. These results show that the efficiency gains extend to categories containing thin structures while maintaining competitive geometric fidelity across both backbones.



Figure 4: **Visualization results of different methods on TripoSG.**

Table 6: **Quantitative evaluation on ShapeNetCore categories containing thin structures on the TripoSG Li et al. (2025d) and Hunyuan3D-2.0 Zhao et al. (2025b) benchmark.** The **best** results are bolded and underlined, and the second-best results are underlined (excluding Vanilla).

Methods	CD↓	F-Score↑	IoU↑	NC↑	N-FID↓	$T_{denoise}$↓	T_{decode}↓	T_{total} (Speedup)↓
Framework: TripoSG								
Vanilla	0.0314	0.9732	0.5954	0.7528	38.68	6.52s	0.76s	7.28s (1.00×)
RAS	0.0516	0.9288	0.4641	0.6988	65.06	**4.51s**	0.65s	5.16s (1.41×)
SADA (Token Prune)	0.0322	0.9710	0.5867	0.7513	**39.46**	4.80s	0.75s	5.55s (1.31×)
Ours	**0.0315**	**0.9734**	**0.5923**	**0.7521**	40.75	4.69s	**0.39s**	**5.08s (1.43×)**
Framework: Hunyuan3D-2.0								
Vanilla	0.0314	0.9749	0.5939	0.7596	37.31	9.24s	0.42s	9.66s (1.00×)
RAS	0.0316	0.9752	0.5760	0.7464	41.36	6.63s	0.33s	6.96s (1.39×)
SADA (Token Prune)	0.0320	0.9746	**0.5867**	**0.7563**	**37.54**	6.51s	0.40s	6.91s (1.40×)
Ours	**0.0315**	**0.9767**	0.5812	0.7508	38.43	**6.43s**	**0.23s**	**6.66s (1.45×)**

G Analysis on Token Retention Ratios

In this section, we examine how token retention evolves during denoising for different cumulative thresholds on TripoSGLi et al. (2025d) with Toys4K DatasetStojanov et al. (2021). We group denoising steps into non-overlapping intervals of 10 steps and compute each object's average retention ratio over sparse steps within each interval. As shown in Figure 5, higher thresholds correspond to higher token retention, while retention decreases as denoising progresses. At the default $\rho = 0.8$, the average retention decreases from 79.03% in the first interval to 74.59% in the last, indicating more aggressive pruning during later denoising steps.

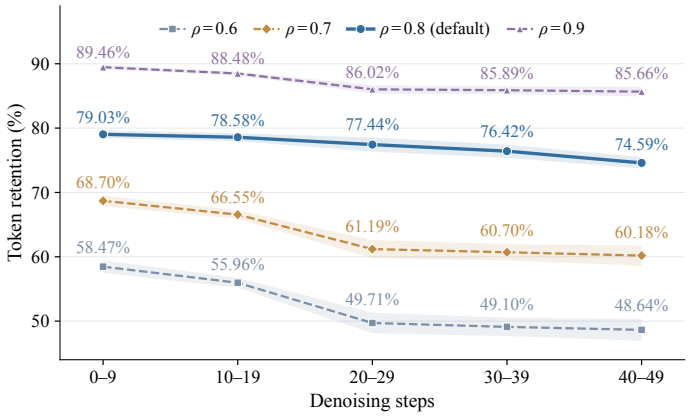

Figure 5: **Analysis on token retention ratios.** Shading denotes ± 1 standard deviation across objects, and full-compute steps are excluded.

H Analysis on Efficiency and Computational Overhead

To demonstrate the efficiency of our framework, we analyze the computational overhead introduced by the acceleration modules relative to the time saved. As detailed in Table 7, the additional costs for geometry scoring and curvature estimation are small compared to the substantial reduction in backbone processing on the TripoSG benchmark.

Specifically, in the denoising stage, the overhead from *Geometry Score Computation* accounts for approximately 1.09% of our total inference time (0.0591s out of 5.42s). This marginal cost allows the model to bypass extensive redundant computations. Similarly, in the decoding stage, *Curvature Estimation* via the fast discrete Laplacian operator incurs a minimal cost of 0.0038s. By guiding the decoder to prioritize high-curvature regions while retaining near-surface buffer points, this strategy achieves a 2.39× decoding speedup, reducing latency from 1.79s to 0.75s. In summary, the significant time savings far outweigh the minimal computational costs of these auxiliary modules.

Table 7: Detailed breakdown of inference latency on TripoSG. The overhead introduced by our method (Geometry Score and Curvature Est.) is negligible compared to the backbone acceleration. The time reduction compared to the Vanilla baseline is shown in red.

	$T_{\text{denoise_all}} \downarrow$		$T_{\text{decoder_all}} \downarrow$	
Vanilla	6.34s		1.79s	
	4.67s **(-1.67s)**		0.75s **(-1.04s)**	
Ours	Geometry Score	Denoising (Core)	Curvature Est.	Decoder (Core)
	0.0591s	4.6109s	0.0038s	0.7462s