P2VOXEL: PYRAMID PIVOT VOXELIZATION FOR 3D MESH TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Triangle meshes provide explicit and accurate surface geometry, yet their irregular connectivity makes 3D mesh tokenization a geometric sampling problem: how to sample and organize geometric evidence into compact, structured and learnable tokens. Beyond field-centric volumetric sampling and edge-intersection surface sampling, we retarget mesh tokenization as *local surface evidence sampling*: representing each active voxel with compact local surface evidence that supports deterministic reconstruction. To this end, we introduce **P2Voxel**, a pyramid pivot voxelization framework for compact and reconstruction-aware mesh tokenization. P2Voxel is built on three key components. Under the *Local Planarity* assumption, Pivot Voxelization represents each active voxel with a surface pivot and an orientation sign, providing compact local evidence that can induce the corner values required for deterministic reconstruction. Under the *Spatial Complexity* assumption, Pyramid Pivot Voxelization exploits the spatial non-uniformity of real surfaces by allocating finer pivot tokens to geometrically complex regions while keeping smooth regions coarse and compact. Under the *Block Reconstructability* assumption, a Pyramid VAE learns compact multi-resolution latent codes over reconstructable pivot blocks, avoiding the need to model the entire high-resolution voxelized shape as a dense global field. Together, these designs convert meshes into compact, structured, and learnable pyramid pivot tokens, enabling efficient mesh reconstruction for downstream 3D tasks.

1 INTRODUCTION

Triangle meshes are the standard representation for 3D surfaces in graphics and geometry processing (Foley, 1996; Shirley et al., 2009; Yang et al., 2021), as they provide explicit and editable geometry for rendering, simulation, and downstream manipulation (Akenine-Moller et al., 2019; Saito & Takahashi, 1990; Zangi, 2004). Yet their irregular connectivity makes them poorly suited as direct inputs or targets for modern 3D representation learning and generative models (Zhang et al., 2023; Xiang et al., 2025b;a), which favor compact, structured, and learnable tokens. This leads to the problem of *3D mesh tokenization*: representing a complex surface by a finite set of geometric units that can be compressed, predicted, or generated. Rather than viewing tokenization as merely discretizing a shape, we view it as a problem of selecting the geometric evidence. Voxelization offers a natural spatial organization for such evidence by assigning local surface information to grid cells. The central challenge is therefore to design cell-wise tokens that are compact and regular, while still containing sufficient local surface evidence for deterministic high-quality mesh reconstruction.

Existing mesh tokenization methods can be viewed through the type of geometric evidence they preserve, including occupancy values (Wu et al., 2015), point samples (Nichol et al., 2022), implicit fields (Mildenhall et al., 2021; Gao et al., 2022), signed distance fields (SDF) (Oleynikova et al., 2016), and contouring constraints (Ju et al., 2002). For deterministic mesh reconstruction, grid-based SDFs and Dual Contouring represent two influential paradigms. Grid SDF methods (Oleynikova et al., 2016; Jones et al., 2006) store signed distances on spatial grids and extract surfaces through isosurface reconstruction, termed as *field-centric volumetric sampling*. Although this sampling produces regular tokens, it encodes a two-dimensional surface indirectly through scalar samples in 3D space, requiring redundant storage for near-surface points even in sparse variants (Deng et al., 2025). Dual Contouring (Ju et al., 2002) follows an *edge-intersection surface sampling* paradigm, using edge crossings and Hermite normals as boundary-level constraints or reconstruction rules that

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.

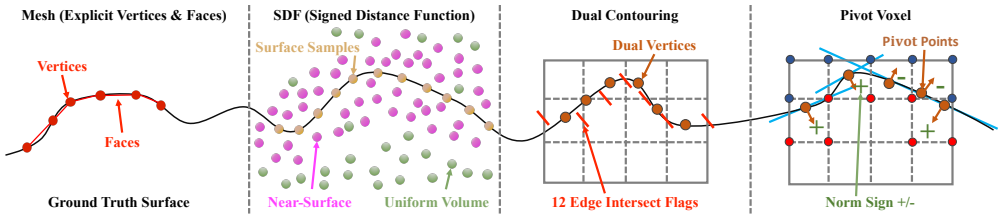


Figure 1: **Geometry Sampling for Mesh Tokenization**. (a) Meshes are explicit but irregular. (b) Grid SDFs use field-centric volumetric sampling by storing signed distance values. (c) Dual Contouring uses edge-intersection surface sampling with edge crossings and Hermite constraints. (d) Our Pivot Voxel uses local surface evidence sampling, with a pivot point and orientation sign.

guide subsequent dual-vertex placement. However, from a tokenization perspective, these edge-intersection samples are tied to sign-changing grid edges and mainly provide reconstruction cues for dual-vertex estimation, rather than defining a fixed and compact cell-wise token for the local surface patch itself. We therefore ask whether mesh tokenization can move beyond volumetric field samples and edge-tied Hermite evidence, and instead *represent each active voxel by the compact local surface evidence required for deterministic reconstruction*.

Beyond field-centric volumetric sampling and edge-intersection surface sampling, we formulate mesh tokenization as *local surface evidence sampling*: identifying the compact geometric evidence inside each active voxel that is sufficient for deterministic surface recovery. Our first observation is *local planarity*: at sufficiently high resolution, the surface patch inside a small active voxel can be approximated by a local plane. This observation is closely aligned with Marching Cubes (Lorensen & Cline, 1998; Dyken et al., 2008), where the surface topology and edge intersections within a voxel are determined by the signed scalar values at its corner vertices. However, if the local surface patch is approximated by a plane, these corner signs and distances can be derived from the plane geometry rather than stored independently. This motivates our **Pivot Voxelization**, where each surface-intersecting voxel is encoded by a pivot point on the surface and an orientation sign, which together define the local plane and induce the corner values required by Sparse Marching Cubes. Our second observation is *spatial complexity*: real surfaces are highly non-uniform, with smooth regions that can be represented coarsely and sharp features, thin structures, or high-curvature details that require finer spatial support. This motivates **Pyramid Pivot Voxelization**, which keeps each pivot token simple while adaptively allocating finer pivot voxels only to geometrically complex regions. Our third observation is *block reconstructability*: mesh recovery does not require learning the entire high-resolution voxelized shape as one monolithic object, because each pyramid block contains sufficient local surface evidence for deterministic reconstruction within its spatial extent. This motivates our **Pyramid VAE**, which learns compact multi-resolution latent codes over pyramid pivot blocks instead of consuming a dense global field. Together, these components form **P2Voxel**, a flexible framework that converts meshes into compact, structured, and learnable pivot tokens for token-efficient 3D mesh reconstruction. The contributions of this paper are summarized as follows:

- We formulate mesh tokenization as **local surface evidence sampling** and propose Pivot Voxelization, where a surface pivot and an orientation sign compactly represent the mesh.
- We introduce **Pyramid Pivot Voxelization**, which exploits spatial complexity by adaptively refining geometrically complex regions while keeping each pivot token simple and reconstructible.
- We then propose a **Pyramid VAE** based on block reconstructability, learning compact multi-resolution latent codes over locally pyramid pivot blocks for token-efficient mesh reconstruction.

2 Related Work

Explicit 3D Representations for Generation. Explicit representations, such as point clouds and meshes, are naturally discrete, making them intuitive candidates for Transformer tokenization. Point-based methods like Point-E (Nichol et al., 2022) and Shap-E (Jun & Nichol, 2023) generate sparse point clouds efficiently, but lack topological connectivity and often require post-processing such as Poisson reconstruction, which may fail on thin structures or sharp edges. Direct mesh generation methods, such as PolyGen (Nash et al., 2020) and MeshGPT (Siddiqui et al., 2024), predict

vertices and faces directly, but the combinatorial complexity of arbitrary mesh topology often limits them to specific categories or low-complexity shapes. Building on MeshAnything (Chen et al., 2024), MeshAnything V2 (Chen et al., 2025b) introduces Adjacent Mesh Tokenization (AMT) to reuse vertices across faces and shorten mesh sequences, while BPT (Weng et al., 2025) further improves scalability through block-wise indexing and patching for higher-resolution mesh generation.

Implicit and Hybrid Representations. Implicit representations, such as Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) and Signed Distance Fields (SDF) (Oleynikova et al., 2016), model geometry as continuous neural fields, while set-based approaches such as VecSet (Zhang et al., 2023) represent shapes as collections of surface elements. Recent generative models like TRELLIS (Xiang et al., 2025b) and DORA (Chen et al., 2025a) further employ structured latent codes, such as SLAT, to encode high-fidelity geometry. However, these methods usually rely on heavy neural decoders, typically MLPs, to extract explicit surfaces, introducing inference latency and often over-smoothing high-frequency details. Hybrid representations, such as FlexiCubes (Zhao et al., 2024) and Deep Marching Tetrahedra (DMTet) (Shen et al., 2021), bridge implicit and explicit representations, but still require extensive volumetric sampling beyond the actual surface to maintain field continuity.

Sparse and Explicit Representations. Sparse data structures, such as Octrees (Deng et al., 2025) and Hash Grids (Ju et al., 2002), are standard techniques for compressing voxel data. In geometry processing, Dual Contouring (DC) (Ju et al., 2002) allows for sharp feature preservation by storing a single vertex per active voxel. However, maintaining valid topology in these frameworks often requires dense structural overhead. Notable recent generative approaches, such as FaithC (Luo et al., 2025) and TRELLIS 2 (Xiang et al., 2025a), achieve high fidelity but rely on explicit edge-state encoding: they typically necessitate recording the intersection status across all 12 edges of a voxel to fully resolve topological ambiguities. This requirement increases the information density per token, diverting representational capacity toward structural bookkeeping.

3 METHODOLOGY

Problem Statement. Triangle meshes provide explicit and accurate surface geometry, but their irregular connectivity and non-uniform complexity make them poorly suited for compact and structured tokenization in representation learning. We therefore seek to convert a surface mesh $\mathcal{M}$ into a representation $\mathcal{T}$ that is (i) *token-efficient* with as few elements as possible, (ii) *structured* and easy to learn, and (iii) *recoverable* by standard geometric reconstruction operators (e.g., Marching Cubes (Lorensen & Cline, 1998)), while preserving accurate geometry recovery. Formally, given a watertight mesh $\mathcal{M}$, we seek a token set $\mathcal{T}$ and a deterministic decoder $\mathcal{D}$ such that

$$\hat{\mathcal{M}} = \mathcal{D}(\mathcal{T}), \qquad |\mathcal{T}| \text{ is minimized subject to } \hat{\mathcal{M}} \approx \mathcal{M}. \tag{1}$$

To this end, we propose **P2Voxel**, a voxelization-based tokenization framework that converts a watertight mesh $\mathcal{M}$ into a compact and structured token set $\mathcal{T}$, while enabling accurate reconstruction with standard geometry operators. P2Voxel contains three key components: Pivot Voxelization for sampling local surface evidence within active voxels, Pyramid Pivot Voxelization for adaptive multi-resolution token allocation, and a Pyramid VAE for learning compact latents over pivot blocks.

3.1 PIVOT VOXELIZATION

Assumption 1 (Local Planarity). At high resolution, most regular surface-intersecting voxels contain only a small local surface, which can be approximated by an *ideal plane*.

Sparse Marching Cubes (Sparse-MC) (Lorensen & Cline, 1998; Dyken et al., 2008; Tang, 2022) reconstructs watertight surfaces by extracting the zero-level set from signed distance values at the eight corners of surface-intersecting (*active*) voxels. These corner values jointly encode the topology and geometry of the enclosed surface patch $\mathcal{S}$. As the voxelization resolution increases (i.e., $h \to 0$), $\mathcal{S}$ becomes asymptotically locally planar. Following **Assumption 1**, we approximate each active voxel by an *ideal plane* $\hat{\mathcal{S}}$, which induces corner distances, as shown in Figure 2.

A plane in $\mathbb{R}^3$ is uniquely determined by a point and a normal direction. Given the voxelization prior, we encode each active voxel by a compact *pivot token* $\{\mathbf{p}, s\}$, where $\mathbf{p} \in \mathbb{R}^3$ is a pivot point

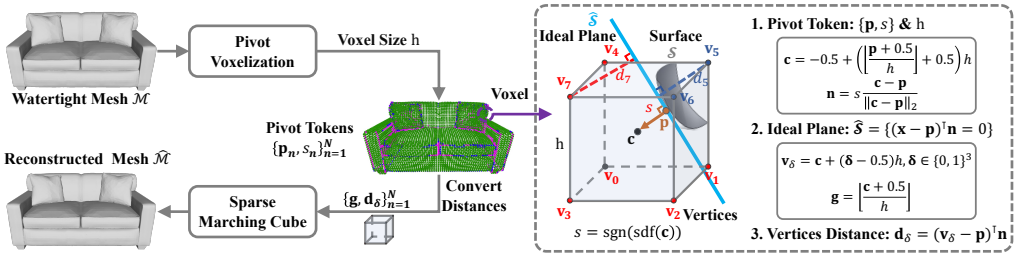


Figure 2: **Pivot Voxelization.** We voxelize a watertight mesh $\mathcal{M}$ with voxel size h and represent each active voxel by a pivot token $\{\mathbf{p}, s\}$, where $\mathbf{p}$ is the pivot point and $s = \text{sgn}(\text{sdf}(\mathbf{c}))$ indicates the inside/outside orientation. An ideal plane $\hat{\mathcal{S}}$ can evaluate vertex distances d_δ for Sparse-MC.

on the local surface patch and $s \in \{-1, +1\}$ is a binary inside/outside orientation cue. Concretely, $\mathbf{p}$ is obtained by projecting the voxel center $\mathbf{c}$ onto the mesh, i.e., the closest intersection point on $\mathcal{M}$, and s is given by the center SDF sign, $s = \text{sgn}(\text{sdf}(\mathbf{c}))$.

Conversely, given $\{\mathbf{p}, s\}$ and voxel size h, we recover the voxel center $\mathbf{c}$ by snapping $\mathbf{p}$ to its containing voxel, then compute the oriented normal $\mathbf{n}$ and ideal plane $\hat{\mathcal{S}}$:

$$\mathbf{c} = -\tfrac{1}{2} + \left(\left\lfloor \tfrac{\mathbf{p}+0.5}{h} \right\rfloor + 0.5\right)h. \tag{2}$$

Then the oriented normal and the induced ideal plane are given by

$$\mathbf{n} = s\frac{\mathbf{c}-\mathbf{p}}{\|\mathbf{c}-\mathbf{p}\|_2}, \qquad \hat{\mathcal{S}} = \left\{\mathbf{x} \in \mathbb{R}^3 \;\middle|\; (\mathbf{x}-\mathbf{p})^\top \mathbf{n} = 0\right\}. \tag{3}$$

With $(\mathbf{p}, \mathbf{n})$, the eight voxel corners and their plane-induced signed distances are computed as:

$$\mathbf{v}_{\boldsymbol{\delta}} = \mathbf{c} + (\boldsymbol{\delta} - 0.5)\, h, \qquad \boldsymbol{\delta} \in \{0,1\}^3, \tag{4}$$

and the signed distance at each corner is given by the plane SDF

$$d_{\boldsymbol{\delta}} = \text{SDF}(\mathbf{v}_{\boldsymbol{\delta}}) = (\mathbf{v}_{\boldsymbol{\delta}} - \mathbf{p})^\top \mathbf{n}. \tag{5}$$

The eight samples $\{d_{\boldsymbol{\delta}}\}$ provide the per-voxel corner SDF values. Since a grid vertex is shared by neighboring voxels, independently predicted corner values may be inconsistent. We therefore enforce a *vertex-consistent* scalar field by averaging predictions from incident active voxels $\Omega_{\mathbf{v}}$:

$$D(\mathbf{v}) = \frac{1}{|\Omega_{\mathbf{v}}|} \sum_{i \in \Omega_{\mathbf{v}}} (\mathbf{v} - \mathbf{p}_i)^\top \mathbf{n}_i, \tag{6}$$

where $(\mathbf{p}_i, \mathbf{n}_i)$ denotes the pivot token parameters of the i-th voxel. Sparse-MC is then applied to the unified field $D(\cdot)$, ensuring shared corner values across adjacent voxels and producing a seamless watertight reconstruction $\hat{\mathcal{M}}$. In summary, the pivot token provides a compact encoding for stable distance evaluation and deterministic reconstruction. We discuss the feasibility conditions and practical constraints of pivot construction in **Appendix A**.

3.2 Pyramid Pivot Voxelization

Assumption 2 (Spatial Complexity). *Geometric information density is non-uniform: high-frequency details are concentrated around high-curvature, sharp, or thin structures, where fine-grained sampling is necessary, while coarse blocks suffice elsewhere.*

Under **(Local Planarity)**, each active voxel admits a compact pivot representation, but this typically requires a sufficiently high resolution. Although a 512^3 grid can capture the overall geometry with high fidelity, it remains inefficient due to redundant sampling in smooth regions. **Assumption 2 (Spatial Complexity)** motivates adaptive sampling with a block-wise pyramid strategy: fine-grained resolution is allocated only to high-frequency regions, while coarser blocks represent smooth areas, presented in Figure 3.

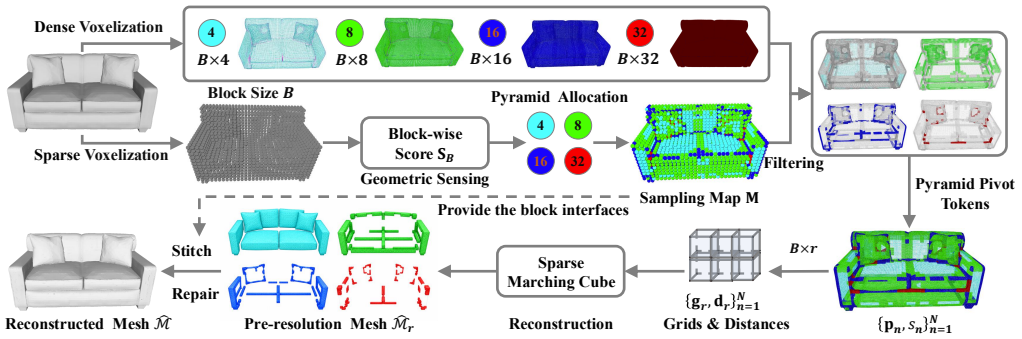


Figure 3: **Pyramid Pivot Voxelization.** We compute a block-wise score s_B over macro-blocks of size B to allocate adaptive sampling levels $\{4, 8, 16, 32\}$, producing a pyramid sampling map $\mathbf{M}$. After per-resolution reconstruction, the resulting meshes are stitched to recover the final mesh.

Pyramid Sampling Map. We partition the normalized space $[-0.5, 0.5]^3$ into a macro-block grid $\mathcal{B} \in \mathbb{Z}^{B\times B\times B}$ with block size $1/B$, and assign each block $\mathbf{g}$ a complexity score $\mathbf{S}_B(\mathbf{g})$. Surface samples $\mathbf{x}$ are mapped to blocks by $\mathbf{g}(\mathbf{x}) = \lfloor (\mathbf{x} + 0.5)\, B \rfloor$, and curvature- and normal-based cues are aggregated as

$$\mathbf{S}_B(\mathbf{g}) = \Big(\bar{c}(\mathbf{g}) + \lambda_{\text{var}}\, \sigma_c(\mathbf{g}) + \lambda_{\text{grad}}\, \|\nabla \bar{c}(\mathbf{g})\| + \lambda_{\text{n}}\, \bar{\eta}(\mathbf{g})\Big) \cdot \sqrt{N(\mathbf{g})}, \tag{7}$$

where $\bar{c}$ is the mean curvature magnitude, σ_c is curvature variation, $\|\nabla \bar{c}\|$ captures spatial curvature change, $\bar{\eta}$ measures local normal deviation, and $N(\mathbf{g})$ is the number of surface samples in the block. We discretize $\mathbf{S}_B$ into an allocation map

$$\mathbf{M} : \mathbf{g} \mapsto r \in \mathcal{R} = \{r_{min}, \ldots, r_{max}\}, \tag{8}$$

where $\mathcal{R}$ is a predefined set of sampling resolutions with quantile-based binning over blocks, and $\mathbf{M}$ is restricted to surface-intersecting blocks, with missing active blocks filled by resolution r_{min}.

Pyramid Sampling. Given $\mathbf{M} : \mathbf{g} \mapsto r$, we perform pivot voxelization with $h_r = 1/r$, producing

$$\mathcal{T}_r = \left\{\tau_n^{(r)}\right\}_{n=1}^{N_r}, \qquad \tau^{(r)} = \{\mathbf{p}^{(r)}, s^{(r)}\}, \tag{9}$$

where $\mathbf{p}^{(r)} \in \mathbb{R}^3$ is the pivot point and $s^{(r)} \in \{-1, +1\}$ is the orientation sign. According to $\mathbf{M}$, for each active macro-block $\mathbf{g}$, tokens from resolution $r = \mathbf{M}(\mathbf{g})$ form the final pyramid set

$$\mathcal{T}_{\text{pyr}} = \bigcup_{\mathbf{g}\in\mathcal{B}_{\text{active}}} \left\{\tau^{(\mathbf{M}(\mathbf{g}))} \;\middle|\; \tau \text{ lies in block } \mathbf{g}\right\}. \tag{10}$$

Although constructing $\{\mathcal{T}_r\}_{r\in\mathcal{R}}$ is not a single-pass adaptive implementation, it is practical because closest-point queries and triangle normals are provided by C++ backends (Zhou et al., 2018).

Pyramid Mesh Stitching. Let $\mathcal{T}_{\text{pyr}}^{(r)}$ denote the tokens in $\mathcal{T}_{\text{pyr}}$ assigned to resolution $r \in \mathcal{R}$. At each resolution, we recover active voxels and corner distances using the vertex-consistent SDF:

$$\mathbf{V}_r = \{v_k^{(r)}\}_{k=1}^{K_r}, \qquad \mathbf{D}_r = \left\{\mathbf{d}_r(v_k^{(r)})\right\}_{k=1}^{K_r}, \qquad \mathbf{d}_r(v_k^{(r)}) = \left[D_r(\mathbf{v}_{k,\boldsymbol{\delta}}^{(r)})\right]_{\boldsymbol{\delta}\in\{0,1\}^3}, \tag{11}$$

where D_r is the shared-vertex field computed from $\mathcal{T}_{\text{pyr}}^{(r)}$. Sparse-MC then reconstructs each resolution separately:

$$\hat{\mathcal{M}}_r = \text{SparseMC}(\mathbf{V}_r, \mathbf{D}_r), \qquad r \in \mathcal{R}. \tag{12}$$

We stitch the per-resolution meshes across block interfaces across the sampling map and repair residual boundaries (Details are present in **Appendix C**):

$$\hat{\mathcal{M}} = \text{Repair}\Big(\text{Stitch}_{\mathbf{M}}\left(\{\hat{\mathcal{M}}_r\}_{r\in\mathcal{R}}\right)\Big). \tag{13}$$

In summary, Assumption 2 reserves fine sampling for high-curvature, sharp, or thin structures while keeping smooth regions coarse. Cross-resolution stitching combines these levels into a coherent mesh, retaining detail without the redundant vertices and faces of a uniformly dense grid.

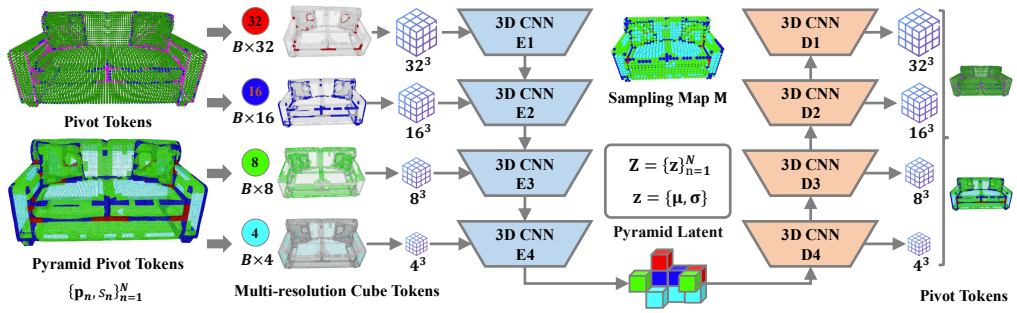


Figure 4: **Block-wise Pyramid VAE.** Given pivot tokens at adaptive sampling levels $\{4, 8, 16, 32\}$, the encoder aggregates multi-resolution block features into a compact pyramid latent code, then reconstructs pivot tokens with the guidance of the sampling map $\mathbf{M}$.

3.3 Pyramid VAE

Assumption 3 (Block Reconstructability). *Surface reconstruction is locally composable: each pyramid block can recover its local surface patch from its pivot tokens, and the full mesh can be obtained by enforcing consistent corner values across neighboring blocks.*

A straightforward strategy is to learn pivot tokens by decoding the entire shape on a dense global voxel grid. However, this quickly becomes prohibitive at high resolutions: even a compact pivot field with three pivot coordinates and one orientation sign per voxel requires approximately 2.0 GiB and 16.0 GiB of FP32 memory at 512^3 and 1024^3 resolutions, respectively. In 3D CNN-based models, the actual training cost is much larger due to multi-channel feature maps, activations, gradients, and optimizer states. Thus, directly learning a global high-resolution voxel field is memory-inefficient and difficult to scale, even with sparse convolutional techniques (Graham et al., 2018).

Based on **Assumption 3**, we instead design a **Pyramid VAE**, presented in Figure 4. The key idea is that pivot tokens are locally reconstructable: each block contains sufficient surface evidence to recover its local patch through the induced plane and Sparse-MC corner values. Thus, the model does not need to encode the entire high-resolution shape as a monolithic dense tensor. Instead, we organize pyramid pivot tokens into multi-resolution block tensors using the pyramid sampling map $\mathbf{M}$, where each level uses a sampling resolution such as $\{4, 8, 16, 32\}$ within B-sized macro-blocks.

The encoder uses hierarchical 3D CNN branches to aggregate block-wise geometric features from coarse to fine levels and maps them into a shared pyramid latent representation

$$\mathbf{z} = \{\mathbf{z}_n\}_{n=1}^N, \qquad \mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathcal{T}_{\text{pyr}}, \mathbf{M}),$$

where $\mathbf{M}$ provides the spatial allocation of sampling resolutions and $\mathcal{T}_{\text{pyr}}$ provides the corresponding local surface evidence. The decoder mirrors this hierarchy to reconstruct pivot-token grids at multiple resolutions. After decoding, reconstructed at their assigned resolutions, and stitched across block interfaces for final Sparse-MC reconstruction. The pyramid representation separates *where* to allocate resolution from *what* surface evidence to store. It only adds a lightweight sampling map $\mathbf{M}$, which guides each active macro-block to reconstruct at its assigned resolution and can reduce to standard Pivot Voxelization when all blocks use the finest level. For generation, $\mathbf{M}$ can be learned as a coarse layout prior, while pivot tokens model local geometry, naturally supporting a coarse-to-fine generative process. Details are presented in **Appendix D**.

4 Experiments

4.1 Implementation Details

Baselines. We benchmark P2Voxel against representative baselines covering two paradigms: (i) SDF-based sampling, including Vecset (Zhang et al., 2023), Dora (Chen et al., 2025a), and Hunyuan3D-2.1 (Zhao et al., 2025); and (ii) Dual Contouring, represented by FaithC (Luo et al.,

Table 1: **Quantitative evaluation of geometry reconstruction quality.** We report Chamfer Distance (CDL1/CDL2), Earth Mover's Distance (EMD), and F-score ($\tau = 0.002$) on ABO, Objaverse, and Wild datasets at **512** resolution (1024 Results in **Appendix**). We evaluate our three configurations: Pivot-512 (single-scale 512) with hierarchical Pyramid-512 ($128/256/512$) and Pyramid-1024 ($128/256/512/1024$). **Pyramid-512 provides an efficiency-oriented operating point, whereas Pyramid-1024 prioritizes reconstruction quality while retaining a clear complexity advantage. Bold** indicates the best performance, and underlined denotes the second best. Qualitative results are provided in **Figure 11 of Appendix H**.

Dataset	Metrics	SDF			Dual Contouring		Ours		
		Dora	Vecset	Hy3D	Trellis 2	FaithC	Pivot-512	Pyramid-1024	Pyramid-512
Voxel	Dim	4	4	4	7	18	4	4	4
ABO	Num ↓	731,072	500,000	624,640	908,296	820,445	880,134	385,151	**135,736**
	CDL1 ↓	2.1858	2.2699	2.4972	2.1205	2.1175	2.1166	**2.0624**	2.2519
	CDL2 ↓	0.0121	0.0145	0.0188	0.0110	0.0110	0.0110	**0.0106**	0.0130
	EMD ↓	3.4081	3.4962	3.7561	3.3425	3.3388	3.3364	**3.2680**	3.3954
	F-score ↑	0.4654	0.4387	0.3843	0.4866	0.4869	0.4889	**0.5083**	0.4377
Objaverse	Num ↓	731,072	500,000	624,640	787,174	710,859	762,776	358,357	**106,814**
	CDL1 ↓	2.1798	2.3196	3.2233	2.0218	2.0240	2.0362	**2.0121**	2.3986
	CDL2 ↓	0.0122	0.0153	0.0842	0.0104	0.0105	0.0105	**0.0103**	0.0167
	EMD ↓	3.3885	3.5664	4.8408	3.1973	3.2025	3.2096	**3.1716**	3.9019
	F-score ↑	0.4866	0.4442	0.3679	0.5357	0.5357	0.5311	**0.5375**	0.4416
Wild	Num ↓	731,072	500,000	624,640	416,054	380,769	403,306	234,999	**72,862**
	CDL1 ↓	1.7899	1.9798	2.3475	1.4525	1.4514	1.5090	**1.4357**	1.8539
	CDL2 ↓	0.0089	0.0114	0.0184	0.0055	0.0055	0.0058	**0.0053**	0.0095
	EMD ↓	2.9302	3.2289	3.8451	2.4976	2.4949	2.5602	**2.4358**	3.4419
	F-score ↑	0.6324	0.5766	0.4895	0.7552	0.7546	0.7370	**0.7746**	0.5997

Table 2: **Quantitative evaluation of geometry reconstruction quality with VAE compression on ABO, Objaverse, and Wild datasets at 512 resolution. Bold** indicates the best performance, and underlined denotes the second best. Qualitative results are provided in **Figure 12 of Appendix H**.

Dataset	Metrics	SDF			DC	Ours		
		Dora	Vecset	Hunyuan3D	Trellis 2	Pivot-512	Pyramid-1024	Pyramid-512
ABO	CDL1 ↓	2.2585	3.4681	2.5242	2.1597	2.1908	**2.1060**	2.2997
	CDL2 ↓	0.0138	0.0404	0.0355	0.0115	0.0116	**0.0108**	0.0133
	EMD ↓	3.4906	4.8132	3.9635	3.3759	3.4202	**3.3183**	3.5158
	F-score ↑	0.4475	0.3125	0.4108	0.4778	0.4606	**0.4900**	0.4221
Objaverse	CDL1 ↓	2.3785	3.0965	2.8737	2.1071	2.1019	**2.0956**	2.5122
	CDL2 ↓	0.0159	0.0299	0.0671	0.0114	**0.0112**	0.0116	0.0196
	EMD ↓	3.6592	4.4018	4.6346	3.2889	3.2926	**3.2435**	4.1456
	F-score ↑	0.4487	0.3345	0.4028	**0.5134**	0.5077	0.5112	0.4016
Wild	CDL1 ↓	1.9505	3.0526	2.2502	1.7363	1.6202	**1.6024**	2.1045
	CDL2 ↓	0.0116	0.0280	0.0233	0.0070	0.0068	**0.0064**	0.0145
	EMD ↓	3.1651	4.1121	3.7488	2.8409	2.6872	**2.5326**	3.8965
	F-score ↑	0.5879	0.2427	0.5238	0.7100	0.6885	**0.7112**	0.5429

2025) and TRELLIS 2 (Xiang et al., 2025a). We focus on *sampling strategy and surface reconstruction quality*, excluding downstream generative components (e.g., VAE tokenization or diffusion). For fairness, all methods sample SDFs from identical watertight meshes and reconstruct using a neural implicit network (Sitzmann et al., 2020), as detailed in **Appendix B**.

Datasets and Metrics. We evaluate on three disjoint test sets ($\sim$400 shapes each) sampled from **ABO** (Collins et al., 2022), **Objaverse** (Deitke et al., 2023), and an in-the-wild (**Wild**) collection. Shapes are preprocessed into watertight surfaces using Dora's pipeline (UDF-to-SDF followed by MCs), ensuring topological consistency. Quantitative metrics include Chamfer Distance (L1/L2), Earth Mover's Distance (EMD) (Rubner et al., 2000), and F-score (Sturm et al., 2012) ($\tau = 0.002$).

Setups. We configure two voxelization strategies: a single-resolution baseline **Pivot-512** and our hierarchical Pyramid-R approach with $R_{max} = 32$. We employ a 3-level hierarchy $\{128, 256, 512\}$

Table 3: **Mesh complexity.** Averages faces (F) and vertices (V) before and after VAE compression. 1,373,722 faces and 686,835 vertices on ABO original dataset.

	w/o VAE (Tab. 1)		w/ VAE (Tab. 2)	
Method	F↓	V↓	F↓	V↓
TRELLIS 2	1,779,341	853,312	1,816,800	908,296
Pivot-512	1,613,553	793,655	1,721,815	905,282
Pyramid-1024	1,129,338	553,197	1,173,541	599,235
Pyramid-512	**388,846**	**192,401**	**403,727**	**208,165**

Table 4: **Sampling-ratio ablation** at 512 resolution over levels $\{128, 256, 512\}$ on ABO dataset. S1–S4 use $[0.85, 0.12, 0.03]$, $[0.85, 0.10, 0.05]$, $[0.80, 0.10, 0.10]$, and $[0.70, 0.20, 0.10]$, respectively.

Config.	Num	CDL1↓	CDL2↓	EMD↓	F-score↑
S1	116,992	2.3154	0.0134	3.6121	0.4234
S2	135,736	2.3045	0.0134	3.6001	0.4292
S3	189,783	2.2533	0.0126	3.5270	0.4437
S4	207,885	2.2229	0.0122	3.4801	0.4547

with sampling allocation ratios of $[0.85, 0.10, 0.05]$ for **Pyramid-512**, while extending to a 4-level hierarchy $\{128, 256, 512, 1024\}$ with ratios $[0.50, 0.35, 0.12, 0.03]$ for **Pyramid-1024**. Unless otherwise stated, hyperparameters are fixed at $\lambda_{var} = 0.5$, $\lambda_{grad} = 0.3$, and $\lambda_n = 3.0$. All experiments are conducted on a server with an AMD EPYC 7543 32-Core Processor and eight NVIDIA RTX 4090 GPUs (24GB VRAM). Detailed descriptions and runtime are provided in **Appendix B**.

4.2 Main Results

Quantitative Evaluation of P2Voxel. Table 1 reports reconstruction quality on ABO, Objaverse, and Wild using CDL1/CDL2, EMD, and F-score ($\tau = 0.002$). Overall, **P2Voxel** (Dim 4) achieves a strong efficiency–fidelity trade-off against both SDF- and dual-contouring baselines. **Pyramid-1024** achieves the best results across all four geometric metrics on all three datasets while using substantially fewer samples than Pivot-512; on ABO, for example, it uses $2.3\times$ fewer samples (385k vs. 880k). In contrast, **Pyramid-512** provides the most efficiency-oriented operating point, using 5.5–$7.1\times$ fewer samples than Pivot-512 on Objaverse and Wild, at the cost of reduced fidelity.

Quantitative Evaluation of PyramidVAE. Table 2 shows that **Pyramid-1024** achieves the best results on 10 of 12 metrics across ABO, Objaverse, and Wild, demonstrating that the multi-resolution latent representation preserves strong geometric fidelity after VAE compression. **Pivot-512** also consistently outperforms the SDF-based baselines and remains competitive with TRELLIS 2, outperforming it on Objaverse CDL1/CDL2 and Wild CDL1/CDL2/EMD. Notably, the Pyramid VAE is trained only on the ABO training split, without using Objaverse or Wild, yet generalizes effectively across datasets. While the more compact **Pyramid-512** sacrifices some reconstruction fidelity, it provides a more compact operating point at the cost of reconstruction fidelity.

Mesh Complexity. Table 3 compares the face and vertex counts of the reconstructed meshes with the original geometry. Pyramid-512 achieves the most compact meshes, while Pyramid-1024 substantially reduces mesh complexity while maintaining higher reconstruction fidelity. VAE compression slightly increases the resulting mesh complexity, but the relative efficiency advantage of the pyramid representations is preserved.

4.3 Ablation Study and Analysis

Allocation Evaluation. Table 4 shows how sampling-map cut ratios control the sampling budget at R=512 under S1–S4. Relaxing the thresholds activates more fine voxels (116,992 $\rightarrow$ 207,885) and consistently improves quality (CDL1: 2.3154 $\rightarrow$ 2.2229; F-score: 0.4234 $\rightarrow$ 0.4547). Figure 8 further visualizes the block-level distribution: increasing the budget activates more fine blocks and raises per-block sample counts, while most blocks remain sparse. This indicates that additional primitives are selectively allocated to geometrically demanding regions, such as thin and high-curvature boundaries, rather than uniformly increasing all areas. Overall, the sampling map provides a simple fidelity knob through uniform cut-ratio settings; detailed statistics are provided in Appendix E.2.

Resolution Evaluation. Table 5 presents an ablation study on pivot sampling resolution (64$\rightarrow$1024) on ABO. Increasing resolution consistently improves reconstruction quality: CDL1/CDL2 decrease from 4.8159/0.1003 to 1.6463/0.0072, EMD drops from 7.2071 to 2.2793, and F-score rises from 0.1937 to 0.6139. Meanwhile, artifact rates are greatly reduced, with TopoC and HighCurv decreasing from 14.85%/17.39% to 0.22%/0.87%, and overall defects (*All*) dropping from 19.09% to

Table 5: **Ablation on pivot sampling resolution** on ABO. TopoC, HighCurv, and All denote topological, high-curvature, and overall artifact rates, respectively; higher resolution consistently improves reconstruction quality and reduces artifacts. More analysis shows in **Appendix A**.

Resolution	Num	CDL1↓	CDL2↓	EMD↓	F-score↑	TopoC (%)	HighCurv (%)	All (%)
64	12,856	4.8159	0.1003	7.2071	0.1937	14.85	17.39	19.09
128	52,944	2.6730	0.0205	4.1791	0.3651	7.82	10.17	10.69
256	217,834	2.3806	0.0151	3.6894	0.4299	3.42	5.27	5.50
512	880,134	2.1166	0.0111	3.3364	0.4889	0.63	1.86	1.89
1024	3,524,752	1.6463	0.0072	2.2793	0.6139	0.22	0.87	0.89

Table 6: **Downstream learnability of Pyramid VAE latents.** Masked latent prediction with a Transformer consistently outperforms zero filling under random and region masking.

Mask Setting	Method	Latent Error ↓	CDL1 ↓	CDL2 ↓	F-score ↑	EMD ↓
–	GT latent	–	2.2405	0.0130	0.4537	3.6587
Random 20%	Zero-filling	0.9910	3.3089	0.0546	0.3381	5.1296
	Transformer	**0.3570**	**2.6944**	**0.0263**	**0.3882**	**4.5943**
Random 50%	Zero-filling	0.9921	8.5210	0.4841	0.2025	13.8788
	Transformer	**0.3967**	**4.2195**	**0.0957**	**0.2745**	**6.9491**
Region 20%	Zero-filling	1.0036	16.5010	3.8993	0.2008	12.7050
	Transformer	**0.4457**	**3.8144**	**0.1331**	**0.3260**	**5.4577**
Region 50%	Zero-filling	1.0022	45.7827	19.5491	0.1591	34.2470
	Transformer	**0.6025**	**13.1314**	**2.0524**	**0.2259**	**13.7525**

0.89%. Notably, the marginal gains diminish at higher resolutions (e.g., 1024→2048), making **1024** a practical trade-off between reconstruction fidelity and sampling cost. Appendix A further analyzes the local-planarity assumption and visualizes the artifact distributions in Figure 6.

Downstream Learnability of Pyramid VAE Latents. To verify that the learned Pyramid VAE latents support downstream learning beyond reconstruction, we freeze the VAE and train a 4-layer Transformer for **masked latent completion**. Using non-overlapping ABO splits of $7{,}000/500/500$ meshes, the Transformer predicts 20% or 50% masked latent blocks from visible latents and spatial/pyramid cues under random and contiguous-region masking, with zero-filling as the baseline. As shown in Table 6, learned prediction consistently reduces latent error by roughly 40–64% and improves all decoded geometry metrics. The larger gains under region masking provide evidence that the latents preserve spatially structured, predictable information, while this controlled setup isolates token-space learnability from any particular generative architecture.

5 Conclusion

We propose **P2Voxel**, a compact mesh tokenization framework that reformulates voxel-based mesh representation as *local surface evidence sampling*. Under local planarity, each active voxel is represented by a pivot point and an orientation sign, forming a compact Dim 4 descriptor that induces the corner values required for deterministic Sparse Marching Cubes reconstruction. Building on spatial complexity, Pyramid Pivot Voxelization adaptively allocates fine tokens to geometrically complex regions while keeping smooth regions coarse, avoiding the redundancy of uniformly high-resolution sampling. Based on block reconstructability, the Pyramid VAE further compresses these multi-resolution pivot blocks into structured latent codes while preserving local geometric recovery. Experiments on ABO, Objaverse, and Wild demonstrate favorable reconstruction–primitive trade-offs, strong fidelity after VAE compression, cross-dataset generalization, and predictable latent structure in downstream masked completion. Together, P2Voxel provides a compact, structured, and learnable token space for high-fidelity 3D mesh representation. Future work will explore its integration with autoregressive and diffusion-based shape generation.

AI USE STATEMENT

In this work, we used generative AI tools to edit and polish the manuscript for clarity, readability, and language quality, and to assist with modifying and debugging research code. The research ideas, methodology, experimental design, analysis, and scientific conclusions were developed by the authors. All AI-assisted text and code were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and experimental results produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on 3D geometry representation, tokenization, and reconstruction, and does not involve human subjects, private personal data, or high-risk deployment scenarios. P2Voxel may reduce the storage and computational cost of high-resolution 3D processing, benefiting applications such as digital content creation, simulation, robotics, and 3D generation. At the same time, large-scale use of 3D assets and generative pipelines should respect data provenance, copyright, licensing requirements, and responsible use of generated content.

REPRODUCIBILITY STATEMENT

We provide the implementation details necessary to reproduce our experiments in the main paper and appendices, including dataset preprocessing and evaluation protocols, voxelization and reconstruction settings, Pyramid VAE architecture, training objectives, optimization hyperparameters, hardware configurations, and runtime measurements. Additional qualitative results and ablation studies are also provided in the appendix. We plan to release the source code and necessary implementation resources upon publication to facilitate reproducibility and future research.

REFERENCES

Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering*. AK Peters/crc Press, 2019.

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, 2025a.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024.

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13922–13931, October 2025b.

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

Matt Deitke, 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.

Kangle Deng, Hsueh-Ti Derek Liu, Yiheng Zhu, Xiaoxia Sun, Chong Shang, Kiran S Bhat, Deva Ramanan, Jun-Yan Zhu, Maneesh Agrawala, and Tinghui Zhou. Efficient autoregressive shape generation via octree-based adaptive tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11685–11696, 2025.

Christopher Dyken, Gernot Ziegler, Christian Theobalt, and Hans-Peter Seidel. High-speed marching cubes using histopyramids. In *Computer Graphics Forum*, volume 27, pp. 2028–2039. Wiley Online Library, 2008.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

James D Foley. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.

Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9224–9232, 2018.

Mark W Jones, J Andreas Baerentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on visualization and Computer Graphics*, 12(4):581–599, 2006.

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 339–346, 2002.

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

Eric Lengyel. Transition cells for dynamic multiresolution marching cubes. *Journal of Graphics, GPU, and Game Tools*, 15(2):99–122, 2010.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Yihao Luo, Xianglong He, Chuanyu Pan, Yiwen Chen, Jiaqi Wu, Yangguang Li, Wanli Ouyang, Yuanming Hu, Guang Yang, and ChoonHwai Yap. Faithful contouring: Near-lossless 3d voxel representation free from iso-surface. *arXiv preprint arXiv:2511.04029*, 2025.

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. *Communications of the ACM*, 65(1):99–106, 2021.

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

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

Helen Oleynikova, Alexander Millane, Zachary Taylor, Enric Galceran, Juan Nieto, and Roland Siegwart. Signed distance fields: A natural representation for both mapping and planning. In *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016.

Mehmet K Ozkan, A Murat Tekalp, and M Ibrahim Sezan. Pocs-based restoration of space-varying blurred images. *IEEE Transactions on Image Processing*, 3(4):450–454, 1994.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 197–206, 1990.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. AK Peters/CRC Press, 2009.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19615–19625, 2024.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 573–580. IEEE, 2012.

Jiaxiang Tang. cubvh: A cuda mesh bvh acceleration toolkit. `https://github.com/ashawkey/cubvh`, 2022.

Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 311–318, 1994.

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Scaling mesh generation via compressive tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11093–11103, 2025.

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

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

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

Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 34:22483–22497, 2021.

Ronen Zangi. Water confined to a slab geometry: a review of recent computer simulation studies. *Journal of Physics: Condensed Matter*, 16(45):S5371, 2004.

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.

Ruowen Zhao, Zhengyi Wang, Yikai Wang, Zihan Zhou, and Jun Zhu. Flexidreamer: single image-to-3d generation with flexicubes. *arXiv preprint arXiv:2404.00987*, 2024.

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*, 2025.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

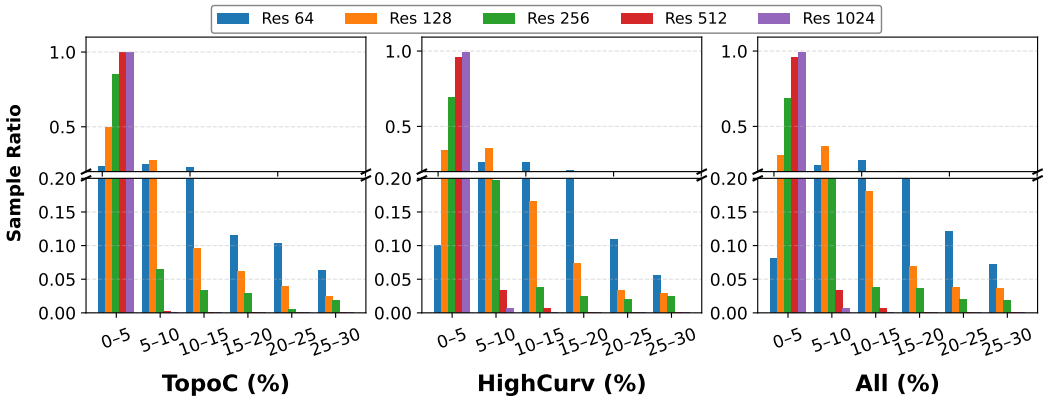


Figure 6: **Artifact ratio distribution across pivot sampling resolutions for Table 5.** We visualize the percentage of samples affected by topological complexity (TopoC), high-curvature artifacts (HighCurv), and their union (*All*) under different pivot sampling resolutions (64–1024) on ABO. Artifact ratios consistently decrease as resolution increases, while improvements beyond **512** become less pronounced, suggesting **512** as a practical trade-off between robustness and sampling cost.

A Pivot Feasibility within a Voxel

Sufficient Resolution. Assumption 1 is only valid when the voxel size is sufficiently small such that each active voxel contains a *single* and *nearly planar* surface patch. To ensure this, we perform a local complexity test by probing multiple sub-voxel locations within each candidate voxel and querying their closest surface points and corresponding triangle normals. A voxel is labeled as *ambiguous* if the normals exhibit strong directional inconsistency (suggesting multi-sheet or topologically complex regions) or if the probed points deviate noticeably from a single plane (indicating high curvature). In our implementation, we use a normal-consistency threshold of 0.9 (approximately a 25° tolerance) for the topological check and a planarity tolerance of 5% of the voxel size for high curvature, which provides a conservative filter for valid locally planar voxels, where Figure 5 demonstrates the trends of scaling voxelization resolution.

Pivot Feasibility within a Voxel. To make the pivot token well-defined and numerically stable, the pivot point must lie strictly inside its associated voxel (otherwise, the voxel index recovered from $\mathbf{p}$ becomes ambiguous), and it should not collapse to the voxel center $\mathbf{c}$ (otherwise, the normal direction becomes ill-conditioned). Given an initial pivot estimate on the plane, we refine it by solving a constrained closest-point problem via *alternating projections* (projection onto convex sets, POCS (Ozkan et al., 1994)): we first project $\mathbf{c}$ onto the plane, then alternately enforce *in-voxel* feasibility (by clamping to the voxel bounds with a small ϵ margin) and *on-plane* feasibility (by re-projecting onto the plane) for a few iterations, and finally apply a hard in-voxel constraint to guarantee that the stored pivot remains inside the voxel.

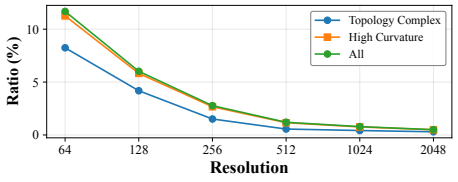


Figure 5: **Effect of voxelization resolution on the mesh shown in Figure 2.** The ratios of topology-complex and high-curvature regions decrease with increasing resolution, especially beyond $r \geq 1024$. Please also refer to Table 5 and Figure 6.

Artifact Ratio Distribution across Resolutions. We analyze the effect of pivot sampling resolution on reconstruction robustness using ABO. For resolutions from 64 to 1024, we report the percentage of samples affected by topological complexity (TopoC), high-curvature artifacts (HighCurv), and their union (*All*). The artifact ratios decrease consistently as resolution increases, showing that finer pivot sampling better captures challenging local geometry. However, the gains become marginal beyond resolution 1024, suggesting that 512 provides a practical balance between robustness and sampling cost. Accordingly, we use 512 as the default maximum resolution in our experiments.

B Implementation Details

Baselines. We benchmark P2Voxel against representative baselines from two mainstream geometry discretization paradigms: SDF-based sampling methods, including 3DShape2VecSet (VecSet) (Zhang et al., 2023), DORA (Chen et al., 2025a), and Hunyuan3D-2.1 (Zhao et al., 2025); and Dual Contouring-based methods, including FaithC (Luo et al., 2025) and TRELLIS 2 (Xiang et al., 2025a). Our evaluation focuses on the geometry discretization stage, namely the sampling strategy and its resulting surface reconstruction fidelity, while excluding downstream generative components such as VAE tokenization, latent modeling, or diffusion synthesis. For a controlled comparison, all SDF-based baselines sample or downsample points from the same watertight input mesh and reconstruct surfaces using the same neural implicit representation (Sitzmann et al., 2020), so that performance differences mainly reflect the effect of the sampling strategy rather than reconstruction-network variations.

Datasets and Metrics. We conduct experiments on three disjoint test sets, each containing approximately 400 shapes. The **ABO** dataset (Collins et al., 2022) contains product models from a catalog of over 147k items and 7,953 artist-designed meshes. The **Objaverse** dataset (Deitke et al., 2023) is sampled from a large-scale repository of over 800k 3D assets, covering diverse categories such as animals, humans, vehicles, and everyday objects. The **Wild** dataset includes challenging internet-collected shapes with noisy geometry, irregular sampling artifacts, and complex topology. All shapes are converted into watertight reference surfaces using DORA's UDF-to-SDF pipeline, followed by Marching Cubes and largest connected component extraction. This provides consistent ground-truth surfaces for evaluation while reducing ambiguity from disconnected fragments and non-manifold artifacts. We evaluate reconstruction quality using complementary geometric metrics. Chamfer Distance under L1 and L2 norms (CDL1 and CDL2) (Fan et al., 2017) measures bidirectional nearest-neighbor discrepancy between reconstructed and reference point sets. Earth Mover's Distance (EMD) (Rubner et al., 2000) captures global distribution mismatch and structural distortion. We also report F-score (Sturm et al., 2012) at a threshold of 0.002, which balances precision and recall to indicate reconstruction completeness and surface correctness.

Setups. We evaluate two voxelization configurations to study the trade-off between reconstruction fidelity and token efficiency: a single-resolution baseline **Pivot-512** and our hierarchical **Pyramid-R** strategy with $R_{\max} = 32$. In **Pivot-512**, we perform pivot-based voxelization at a fixed resolution of 512^3, serving as a strong single-scale reference. For the hierarchical setting, we instantiate two hierarchical variants that allocate pivot tokens across multiple resolutions. **Pyramid-512** adopts a 3-level hierarchy $\{128, 256, 512\}$ with allocation ratios $[0.85, 0.10, 0.05]$, prioritizing coarse coverage while reserving fine samples for structurally informative regions. **Pyramid-1024** extends this to a 4-level hierarchy $\{128, 256, 512, 1024\}$ with allocation ratios $[0.50, 0.35, 0.12, 0.03]$, enabling higher-resolution reconstruction with limited fine-scale sampling and sufficient mid-scale support. Unless otherwise specified, we keep all scoring hyperparameters fixed across datasets and baselines, setting $\lambda_{\text{var}} = 0.5$, $\lambda_{\text{grad}} = 0.3$, and $n = 3.0$, which control curvature variance, curvature gradient magnitude, and normal deviation in the block-level importance score, respectively. All experiments are conducted on a workstation equipped with an AMD EPYC 7543 32-Core Processor and eight NVIDIA RTX 4090 GPUs with 24GB VRAM under implementation settings for fair comparison.

C Cross-Resolution Stitching and Repair

The inputs are the per-resolution meshes $\{\hat{\mathcal{M}}_r\}_{r\in\mathcal{R}}$ and the sampling map $\mathbf{M}$. We concatenate the meshes while retaining a resolution label for each vertex and face. We then visit interfaces between adjacent macro-blocks assigned different resolutions. Near each interface, edges incident to only one face are assembled into boundary chains, and nearby coarse and fine chains are paired. This mesh-space procedure is related to mesh zippering (Turk & Levoy, 1994); transition-cell methods instead connect multiresolution surfaces during voxel extraction (Lengyel, 2010).

For each paired chain, nearby cross-resolution vertices are merged at the finer vertex position. When the chains nearly coincide, we retriangulate the adjacent coarse faces along the fine boundary. For larger gaps, we connect the chains with an interpolated strip. Let $\mathbf{q}_c(t)$ and $\mathbf{q}_f(t)$ denote points at matched normalized arc length on the two chains. Interior strip vertices are

$$\mathbf{q}_\alpha(t) = (1-\alpha)\mathbf{q}_c(t) + \alpha\mathbf{q}_f(t), \qquad t, \alpha \in [0, 1]. \tag{14}$$

Table 7: Training hyperparameters of the Pyramid VAE.

Data / Architecture		**Optimization / Objective**	
Hyperparameter	Value	Hyperparameter	Value
Dataset	ABO	Optimizer	AdamW
Representation	Pyramid tokens	Learning rate	5×10^{-6}
Global resolution	512^3	Epochs	500
Maximum block resolution $R_{\max}$	32	Gradient clipping	1.0
Training stages	$\{32, 16, 8\}$	Precision	FP32
Stage sampling weights	$\{5.0, 3.0, 2.0\}$	Mixed precision	Disabled
Input channels	4	KL weight	10^{-4}
Output channels	6	Sign CE weight	0.5
Latent channels C_z	16	Sign CE class weights	$[1.0, 0.25, 1.0]$
Encoder channels	$[32, 64, 128, 256]$	XYZ loss weight	1.0
Decoder channels	$[512, 256, 128, 64]$	Empty-space loss weight	0.01
Residual blocks per stage	2	SDF cosine loss weight	0
Middle residual blocks	2	Validation interval	5 epochs
Normalization	LayerNorm	Checkpoint interval	20,000 steps
Batch size	2	Maximum kept checkpoints	10
Block batch size	64	Random seed	1234

Table 8: Architectural details of the Pyramid VAE encoder. The encoder starts from the selected pyramid resolution $R \in \{32, 16, 8\}$ and progressively maps the token grid to a 4^3 latent grid. Each ResBlock3D uses LayerNorm, SiLU, and $3 \times 3 \times 3$ convolutions.

Stage	Input Resolution	Output Resolution	Block
Input-32	32^3	32^3	$\mathrm{Conv3D}(4, 32, 3)$
E_1	32^3	16^3	$[\mathrm{ResBlock3D}(32, 32)] \times 2,\ \mathrm{Down}(32, 64)$
Input-16	16^3	16^3	$\mathrm{Conv3D}(4, 64, 3)$
E_2	16^3	8^3	$[\mathrm{ResBlock3D}(64, 64)] \times 2,\ \mathrm{Down}(64, 128)$
Input-8	8^3	8^3	$\mathrm{Conv3D}(4, 128, 3)$
E_3	8^3	4^3	$[\mathrm{ResBlock3D}(128, 128)] \times 2,\ \mathrm{Down}(128, 256)$
Input-4	4^3	4^3	$\mathrm{Conv3D}(4, 256, 3)$
E_4	4^3	4^3	$[\mathrm{ResBlock3D}(256, 256)] \times 2$
Middle	4^3	4^3	$[\mathrm{ResBlock3D}(256, 256)] \times 2$
Posterior Head	4^3	4^3	$\mathrm{LayerNorm},\ \mathrm{SiLU},\ \mathrm{Conv3D}(256, 2C_z, 3)$

Table 9: Architectural details of the Pyramid VAE decoder. The decoder maps the latent grid with $C_z = 16$ channels to multi-resolution token predictions. Each output head predicts three relative coordinates and three sign logits.

Stage	Input Resolution	Output Resolution	Block
Latent Input	4^3	4^3	$\mathrm{Conv3D}(C_z, 512, 3)$
Middle	4^3	4^3	$[\mathrm{ResBlock3D}(512, 512)] \times 2$
D_4	4^3	4^3	$[\mathrm{ResBlock3D}(512, 512)] \times 2$
Head-4	4^3	4^3	$\mathrm{LayerNorm},\ \mathrm{SiLU},\ \mathrm{Conv3D}(512, 6, 3)$
D_3	4^3	8^3	$\mathrm{Up}(512, 256),\ [\mathrm{ResBlock3D}(256, 256)] \times 2$
Head-8	8^3	8^3	$\mathrm{LayerNorm},\ \mathrm{SiLU},\ \mathrm{Conv3D}(256, 6, 3)$
D_2	8^3	16^3	$\mathrm{Up}(256, 128),\ [\mathrm{ResBlock3D}(128, 128)] \times 2$
Head-16	16^3	16^3	$\mathrm{LayerNorm},\ \mathrm{SiLU},\ \mathrm{Conv3D}(128, 6, 3)$
D_1	16^3	32^3	$\mathrm{Up}(128, 64),\ [\mathrm{ResBlock3D}(64, 64)] \times 2$
Head-32	32^3	32^3	$\mathrm{LayerNorm},\ \mathrm{SiLU},\ \mathrm{Conv3D}(64, 6, 3)$

Consecutive rows are triangulated and oriented consistently with the neighboring faces. Matching distances are scaled by the local voxel sizes.

Finally, we inspect remaining single-face boundary edges. We first add triangles between compatible boundary-edge pairs, then examine missing triangular cycles and local four-vertex configurations. Candidate faces must have valid geometry and compatible orientation, and must not give an edge more than two incident faces. Degenerate and duplicate faces are removed. This local cleanup is related to boundary-based mesh hole filling.

D Details of Pyramid VAE

Representation and architecture. We train Pyramid VAE on the ABO training split using pyramid pivot tokens from watertight meshes voxelized at 512^3. Each block uses $R_{\max} = 32$ with stages $\mathcal{R} = \{32, 16, 8\}$. Each token contains a relative 3D pivot coordinate and a sign value, while the decoder predicts three coordinates and three sign logits over $\{-1, 0, +1\}$. The latent code has $C_z = 16$ channels. Training hyperparameters are summarized in Table 7, and encoder/decoder architectures in Tables 8 and 9.

Training. At each iteration, we sample one pyramid stage with probabilities 0.5, 0.3, and 0.2 for resolutions 32, 16, and 8, respectively. We train end-to-end with AdamW for 500 epochs using an initial learning rate of 5×10^{-6} with cosine decay, FP32 precision, gradient clipping at 1.0, batch size 2, block batch size 64, and random seed 1234.

Objective. Given target token $x_i = (\mathbf{p}_i, s_i)$ and prediction $\hat{x}_i = (\hat{\mathbf{p}}_i, \hat{\boldsymbol{\ell}}_i)$, we optimize

$$\mathcal{L} = \lambda_{\text{xyz}}\mathcal{L}_{\text{xyz}} + \lambda_{\text{empty}}\mathcal{L}_{\text{empty}} + \lambda_{\text{sign}}\mathcal{L}_{\text{sign}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}},$$

with $(\lambda_{\text{xyz}}, \lambda_{\text{empty}}, \lambda_{\text{sign}}, \lambda_{\text{KL}}) = (1.0, 0.01, 0.5, 10^{-4})$. Here, $\mathcal{L}_{\text{xyz}}$ is an ℓ_1 loss on occupied pivot coordinates, $\mathcal{L}_{\text{empty}}$ suppresses non-zero predictions in empty voxels, $\mathcal{L}_{\text{sign}}$ is weighted cross-entropy over $\{-1, 0, +1\}$ with class weights $(1.0, 0.25, 1.0)$, and $\mathcal{L}_{\text{KL}}$ regularizes the latent posterior. Validation is performed every 5 epochs.

E More Experimental Results

E.1 More Visualization of Efficiency-quality for Table 1

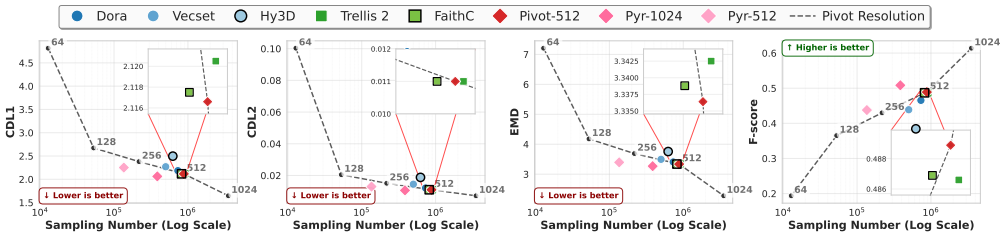
Figure 7 provides an overall analysis of the efficiency–quality trade-off corresponding to Table 1. It plots CDL1/CDL2, EMD, and F-score against the number of sampling primitives (log scale), where red diamonds, blue circles, and green squares denote our variants, SDF-based baselines, and dual-contouring methods, respectively. On **ABO** (Fig. 7a), Pyramid-1024 achieves the strongest reconstruction quality with substantially fewer primitives than Pivot-512, while Pyramid-512 provides a more aggressive efficiency-oriented operating point; Pivot-512 also compares favorably with naïve resolution scaling. Similar trends hold on **Objaverse** and **Wild** (Figs. 7b and 7c), confirming that the efficiency–quality advantage of the pyramid representation generalizes across datasets.

E.2 Block Activation Distribution of Table 4.

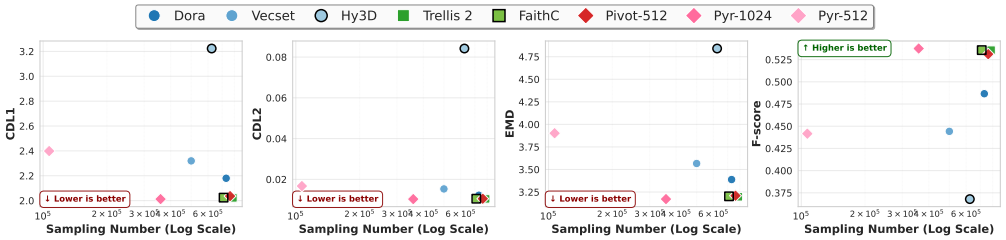
Figure 8 visualizes how the sampling map redistributes the sampling budget across pyramid blocks at the maximum resolution (R=512) **Table 4**. As the cut ratio is relaxed, more fine-resolution blocks become active, and the number of sampled primitives within active blocks also increases. Importantly, the distribution remains highly sparse: most blocks still contain only a small number of samples, while additional budget is concentrated in selected regions. This indicates that the sampling map does not uniformly densify the entire shape, but instead expands fine-level coverage in geometrically demanding areas, showing the effectiveness of region-dependent sampling allocation.

E.3 Higher Watertight Resolution

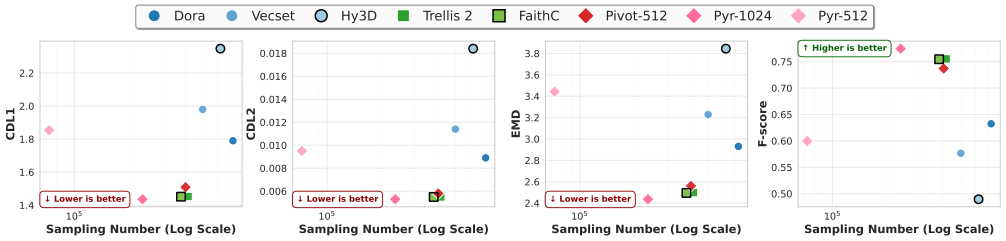
Table 10 reports reconstruction results on ABO, Objaverse, and Wild when the watertight ground-truth surfaces are generated at a higher resolution of **1024**. Compared with our main results under the default watertight setting, we observe that increasing the watertight resolution leads to only minor numerical variations across all metrics, indicating that our evaluation is not sensitive to the specific watertight extraction resolution. Under this stricter setting, our **Pivot-1024** (Dim 4) remains consistently competitive, achieving strong CDL1/CDL2 and EMD with the highest or near-highest F-score across datasets. Overall, these results suggest that the reported performance trends are stable and that further increasing the watertight resolution has limited impact on the comparative conclusions.



(a) **ABO**



(b) **Objaverse**



(c) **Wild**

Figure 7: **Sampling-quality trade-off for Table 1.** We report Chamfer Distance (L_1/L_2), EMD, and F-score against the number of sampling primitives (log scale). Red diamonds denote our methods, blue circles SDF-based methods, and green squares Dual Contouring methods. Overall, P2Voxel achieves favorable Pareto efficiency across ABO, Objaverse, and Wild.

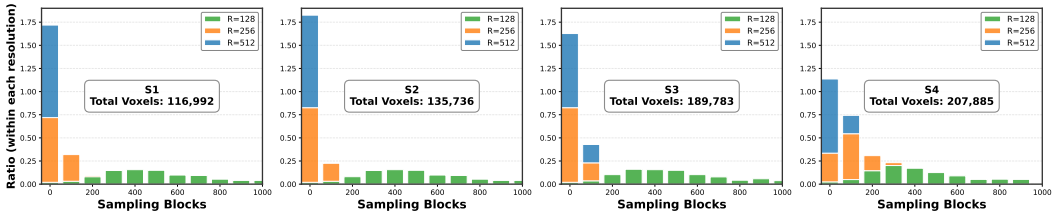


Figure 8: Visualization of sampling-map induced block activation (focus on R=512) for **Table 4**. We visualize how different sampling cut ratios redistribute the sampling budget across blocks. At the maximum resolution (R=512), increasing the total number of sampled primitives consistently activates more fine blocks and increases their per-block sample counts, resulting in clear qualitative improvements in reconstruction. This supports adaptive, region-dependent sampling allocation.

F Previous Resolution-Lifting Implementation

An earlier implementation of Pyramid Pivot Voxelization reconstructed the mesh by lifting all pyramid tokens to the maximum resolution before applying Sparse-MC. Specifically, a token at resolution r was expanded to $(r_{\max}/r)^3$ fine voxels, whose corner distances were evaluated from the token-induced local plane. Although this provides a simple unified reconstruction procedure, globally lifting coarse tokens to the finest grid can unnecessarily increase the number of reconstructed vertices and faces, partially offsetting the efficiency gained from pyramid tokenization. We therefore replaced this implementation with the revised reconstruction strategy used throughout the main

Table 10: **Quantitative evaluation of geometry reconstruction quality.** We report Chamfer Distance (CDL1/CDL2), Earth Mover's Distance (EMD), and F-score ($\tau = 0.002$) on ABO, Objaverse, and Wild datasets at watertight **1024** resolution. We evaluate our **Pivot** (single-scale 512) configuration. Considering the trade-off between voxel dimension capacity and reconstruction fidelity, our P2Voxel method (Dim 4) achieves the most efficient performance. Results are sorted by method category. **Bold** indicates the best performance, and underlined denotes the second best.

	Dataset	Dora	Vecset	Faithc	TRELLIS.2	Hy3D	Ours-Pivot
CDL1 ↓	ABO	2.3144	2.4034	2.1679	2.1606	2.3303	**2.1440**
	Objaverse	2.3599	2.5113	2.0615	2.0534	2.4324	**2.0091**
	Wild	1.9001	2.1017	1.6637	1.6515	2.0115	**1.5304**
CDL2 ↓	ABO	0.0139	0.0167	0.0115	0.0116	0.0151	**0.0113**
	Objaverse	0.0162	0.0203	0.0108	0.0107	0.0180	**0.0103**
	Wild	0.0103	0.0132	0.0073	0.0072	0.0117	**0.0061**
EMD ↓	ABO	3.5541	3.6460	3.3838	3.3765	3.5992	**3.3589**
	Objaverse	3.7049	3.8994	3.2250	3.2121	3.7182	**3.1674**
	Wild	3.3108	3.6483	2.7409	2.7314	3.4402	**2.5593**
F-score ↑	ABO	0.4338	0.4089	0.4713	0.4759	0.4211	**0.4796**
	Objaverse	0.4514	0.4121	0.5171	0.5199	0.4249	**0.5345**
	Wild	0.5961	0.5435	0.6581	0.6631	0.5760	**0.7171**

Table 11: Average runtime breakdown of pyramid voxelization at resolutions 512 and 1024. Values are averaged across ABO, Objaverse, and Wild datasets. Note that the reported times measure the end-to-end execution in **Python**, explicitly including the overall I/O (read/write) overhead.

Resolution	Sampling map	Pyramid Voxelization	Resolution Lifting	Sparse MC RTX 4090
512	8.39 s	5.96 s	0.01 s	0.60 s
1024	8.98 s	14.55 s	0.08 s	2.34 s

paper and all reported experiments. We retain the previous formulation here only for completeness with the reference to Figure 9.

Pyramid Resolution Lifting. Given $\mathcal{T}_{\text{pyr}}$ over multiple resolutions $\mathcal{R}$, we lift all tokens to the maximum resolution $r_{\max} = \max(\mathcal{R})$ with voxel size $h_{\max} = 1/r_{\max}$ for a unified Sparse-MC reconstruction. For each token $\tau^{(r)} = \{\mathbf{p}^{(r)}, s^{(r)}\} \in \mathcal{T}_{\text{pyr}}$, we recover its induced plane $\hat{\mathcal{S}}^{(r)} = (\mathbf{p}^{(r)}, \mathbf{n}^{(r)})$, subdivide the original voxel into $(r_{\max}/r)^3$ fine voxels, and evaluate their corner SDF samples to populate the sparse high-resolution field:

$$\mathbf{V}_{\max} = \{v_k\}_{k=1}^{K}, \qquad v_k \in \{0, \dots, r_{\max} - 1\}^3, \tag{15}$$

together with their corner-distance vectors

$$\mathbf{D}_{\max} = \left\{\mathbf{d}(v_k)\right\}_{k=1}^{K}, \qquad \mathbf{d}(v_k) = \left[d_{\boldsymbol{\delta}}(v_k)\right]_{\boldsymbol{\delta}\in\{0,1\}^3} \in \mathbb{R}^8, \tag{16}$$

where each component is given by

$$d_{\boldsymbol{\delta}}(v_k) = \left(\mathbf{v}_{k,\boldsymbol{\delta}} - \mathbf{p}^{(r)}\right)^{\top} \mathbf{n}^{(r)}. \tag{17}$$

Finally, Sparse-MC is applied to $(\mathbf{V}_{\max}, \mathbf{D}_{\max})$ to reconstruct the sparse watertight mesh $\hat{\mathcal{M}}$.

Runtime Breakdown. Table 11 summarizes the average end-to-end runtime of our pyramid voxelization pipeline at resolutions 512 and 1024, decomposed into four stages. The *sampling map* construction dominates the preprocessing cost and remains relatively stable across resolutions (8.39s→8.98s), while *pyramid voxelization* increases notably with resolution (5.96s→14.55s) due to the higher-density token generation and accumulation. In contrast, *resolution lifting* is lightweight

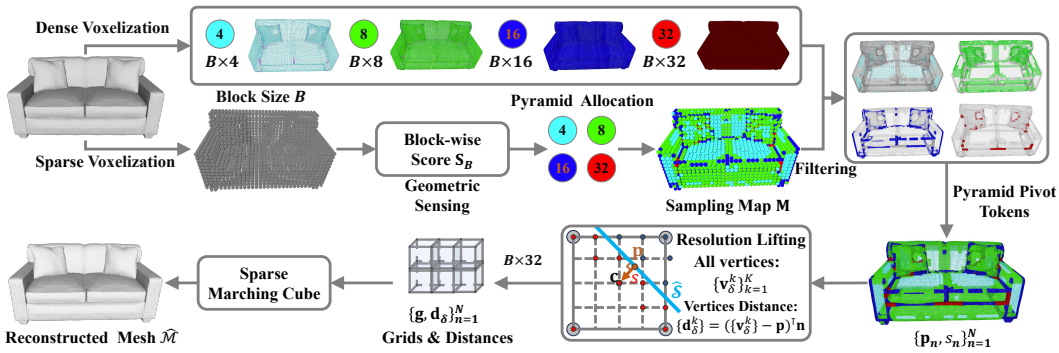


Figure 9: **Previous pyramid resolution-lifting implementation.** An earlier variant lifts multi-resolution pivot tokens to the maximum grid resolution before Sparse-MC reconstruction. This design is retained only for reference and is not used for the results reported in the main paper.

Table 12: Runtime benchmark for training and inference. Training speed is measured under $r_7 = 400{,}000$ and reported as wall-clock time per 1,000 steps, estimated total training time, and peak VRAM. Inference speed is reported as end-to-end reconstruction time per shape and peak VRAM.

Setting	GPU	Block Batch Size	Measured Runtime	Estimated Total Time	Peak VRAM
Train	RTX 4090	32	4m36s / 1,000 steps	12.8 days	12,299 MiB
Inference	RTX 4090	–	1.69 s / shape	–	4,119 MiB

in both settings (0.01s and 0.08s), indicating that multi-scale refinement introduces negligible overhead compared to voxelization itself. Finally, the *sparse Marching Cubes* stage scales with output resolution (0.60s→2.34s) and is the only component accelerated by the **CuBVH** CUDA implementation (Tang, 2022) on RTX 4090; all other stages are executed in **Python** and include I/O overhead. This suggests substantial room for further optimization by migrating the sampling and voxelization stages to GPU kernels and reducing Python-side bottlenecks.

G Discussion

Scope of Evaluation. P2Voxel is designed as a geometry tokenization framework, while complete generative modeling represents a broader system-level setting. We therefore evaluate the representation independently of any particular generator, since end-to-end generation quality is jointly affected by the tokenizer, generator architecture, conditioning mechanism, training scale, and sampling strategy. Our evaluation isolates the properties of the token space along three complementary axes: deterministic geometric recovery, VAE-based compression and cross-dataset generalization, and downstream masked latent completion with the VAE frozen. The latter directly tests whether the learned pyramid latents support spatially structured information that can be predicted from context and decoded back into accurate geometry. Accordingly, the claims of this work concern the compactness, reconstructability, and learnability of the representation itself, rather than the performance of a specific downstream generative pipeline.

Future Work. P2Voxel provides a compact and reconstruction-aware token space by converting dense mesh geometry into locally reconstructable pyramid pivot blocks. An important future direction is to integrate this token space with large-scale autoregressive or diffusion-based 3D generators. Such integration is complementary to the representation-focused evaluation in this work and would additionally involve choices of generator architecture, conditioning, training scale, and sampling strategy. The pyramid sampling map naturally provides a coarse spatial layout, while local pivot tokens capture fine-grained surface evidence, suggesting a coarse-to-fine generation process that first predicts resolution allocation and then local geometry. Another direction is to learn the sampling map directly from data, allowing the token budget to adapt to object category, surface complexity, and downstream objectives. Finally, more efficient CUDA-based kernels and parallel reconstruction procedures could further improve scalability for large-scale 3D processing.

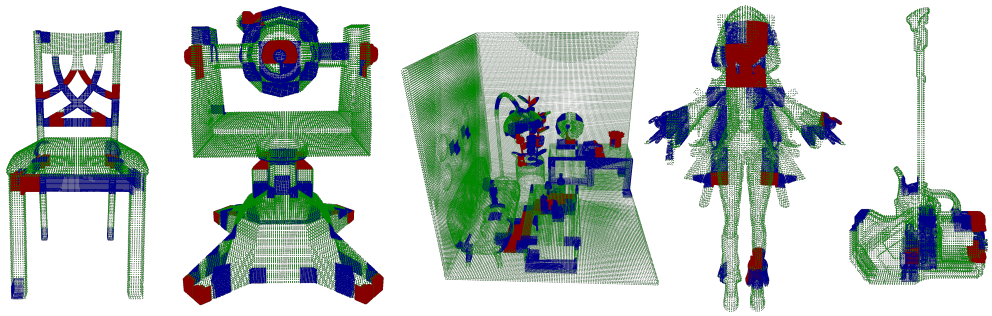


Figure 10: We visualize some pyramid token allocations across resolutions in Figure 11, where red/blue/green/cyan denote R=512/256/128, respectively. Lower-resolution tokens predominantly cover smooth and slowly varying regions, while higher-resolution tokens concentrate around silhouettes, sharp boundaries, and transition areas, demonstrating effective multi-scale budget allocation.

H Qualitative Visualization

We provide qualitative visualizations to further illustrate the behavior and reconstruction quality of P2Voxel. Figure 10 visualizes the pyramid sampling maps, showing how tokens are allocated across different resolutions according to local geometric complexity. Figure 11 compares mesh reconstruction results before VAE tokenization, focusing on the effectiveness of the pivot voxelization and pyramid sampling strategy itself. Figure 12 further presents reconstruction results after VAE compression, demonstrating that the compact pyramid latent representation preserves the main surface structure while supporting token-efficient learning. Together, these visualizations show that P2Voxel allocates high-resolution tokens to geometrically informative regions and maintains high-quality reconstruction under both direct and VAE-compressed settings.

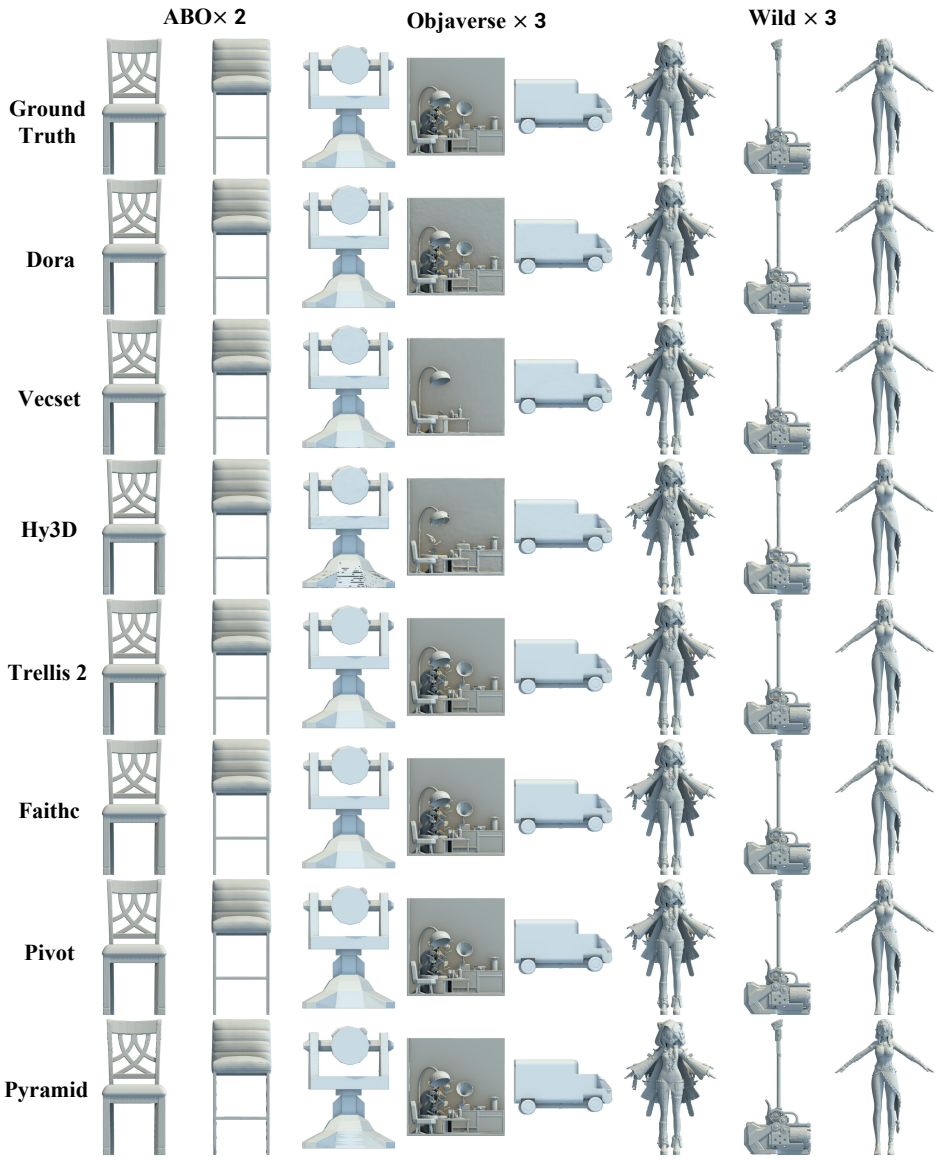


Figure 11: **Visualization results across different methods.** All comparisons are conducted purely at the mesh reconstruction level, without VAE tokenization or diffusion-based generation. With sufficiently dense sampling, all methods produce plausible reconstructions. Our P2Voxel achieves comparable quality with fewer active voxels (smaller token budget), and its compact Dim-4 representation also provides a structured token space for downstream learning.

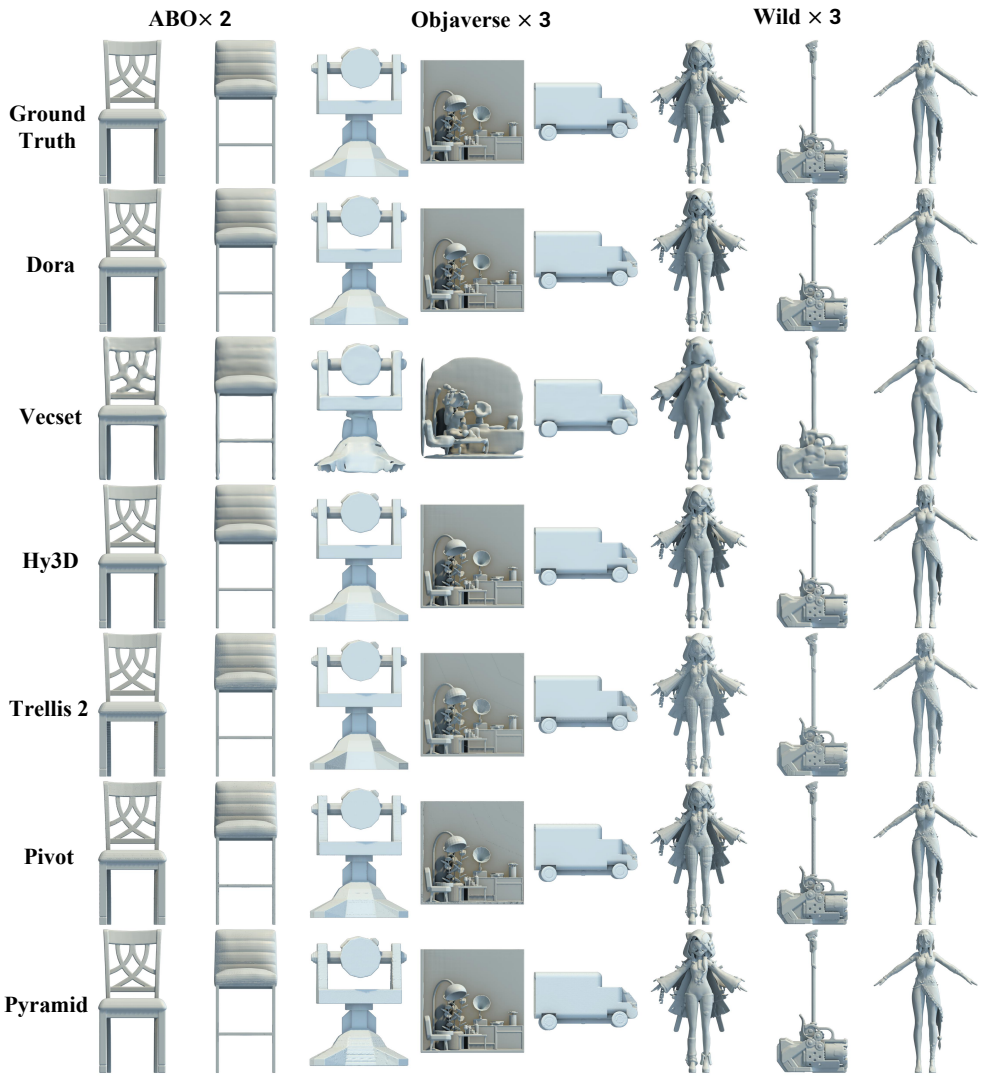


Figure 12: **Visualization results of VAE across different methods.** All comparisons are conducted purely at the mesh reconstruction level, without VAE tokenization or diffusion-based generation. With sufficiently dense sampling, all methods produce plausible reconstructions. Our P2Voxel achieves comparable quality with fewer active voxels (smaller token budget), and its compact Dim-4 representation also provides a structured token space for downstream learning.