

MSVS-VAE: Multi-Scale Anchored VecSet for High-Fidelity 3D Reconstruction

Dehao Hao¹, Kaiyi Zhang¹, Tanghui Jia², Xiangjun Gao^{1✉}, Dongyu Yan³,
Weikai Chen⁴, Zeyu Hu⁴, Lingting Zhu⁴, Yingda Yin⁴, Runze Zhang⁴, Li
Yuan², Xin Wang⁴, and Long Quan¹

¹ The Hong Kong University of Science and Technology, Hong Kong SAR, China
xgaobq@connect.ust.hk

² Peking University, Beijing, China

³ The Hong Kong University of Science and Technology (Guangzhou), Guangzhou,
China

⁴ Lightspeed Studios, Shenzhen, China

Abstract. High-fidelity 3D generative modeling increasingly relies on the latent diffusion paradigm, where the reconstruction quality of the underlying 3D VAE becomes a primary bottleneck. Existing approaches largely follow two paradigms: *sparse voxel-based* representations achieve strong reconstruction quality but incur significant memory and computational overhead, while *set-based representations* are compact and continuous yet typically lag in fidelity due to latent sparsity and excessive global smoothness. We propose **MSVS-VAE**, a hierarchical set-based VAE that closes this fidelity gap without sacrificing compactness. Our key idea is to progressively densify *anchored VecSet* latents via *hierarchical point-shuffle upsampling*, increasing spatial capacity for fine-grained geometry modeling. To efficiently decode from the densified hierarchy, we replace global cross-attention with *AVS-Conv*, a geometry-aware local aggregation operator operating within local neighborhoods rather than the exhaustive latent set. We further introduce *multi-scale query decoding* to fuse coarse-to-fine latent features, where coarse scales provide stable global context, and fine scales refine localized geometry, reducing artifacts from overly local receptive fields. Extensive experiments on Objaverse, ABO, and in-the-wild benchmarks demonstrate that MSVS-VAE consistently outperforms prior set-based and voxel-based VAEs, delivering $\sim 10\times$ faster decoding than prior set-based methods and $\sim 10\times$ higher compactness than voxel-based baselines.

1 Introduction

The pursuit of a high-fidelity yet scalable 3D representation remains a fundamental challenge of 3D generative modeling [21, 45, 51]. As the field gravitates toward

✉ Corresponding author



Fig. 1: MSVS-VAE can achieve high-fidelity reconstruction using a compact vecset-based representation from a set of surface points.

the latent diffusion paradigm [8, 35], the performance of the 3D Variational Autoencoder (VAE) [15, 49] has become the primary bottleneck. A desirable 3D VAE must satisfy two conflicting requirements: it must be compact enough to allow efficient diffusion generative modeling, yet expressive enough to recover intricate geometric details during decoding.

Existing approaches largely fall into two representation paradigms. *Sparse voxel-based methods* [9, 10, 23, 34, 42, 44, 45, 54] build upon dense grids with localized 3D convolutions or sparse attention. Such representations naturally provide high spatial resolution and are effective at recovering sharp geometric details [9, 37]. However, sparse voxel structures typically require a large number of active voxels, leading to high memory and computational cost [3, 9]. Moreover, because the geometry is tied to a discrete grid, small prediction errors in the sparse structure may lead to brittle topology and a non-water-tight mesh [23].

In contrast, set-based representations [11, 17, 19–21, 49, 51] encode shapes as compact latent sets and decode them through continuous query operators. This formulation is elegant and highly compact, while naturally enabling arbitrary-resolution surface queries. Recent advances, such as Lattice [17], significantly improve the reconstruction capability of set-based pipelines by introducing an *anchored VecSet* representation—referred to as VoxSet—which augments VecSet tokens with explicit anchor positions (i.e., active voxel centers) to impose stronger spatial locality. Nevertheless, the reconstruction fidelity of set-based VAEs still lags behind voxel-based methods.

In this work, we argue that this fidelity gap arises from two fundamental limitations of existing VecSet-style representations. First, decoding typically relies on global cross-attention between query points and the entire latent set,

which implicitly imposes excessive *global smoothness* and weakens the model’s ability to capture *localized geometric details*. Second, VecSet representations are highly sparse—often containing tens of times fewer tokens than sparse voxel grids—which severely restricts the spatial resolution available during decoding. Although increasing the number of latent tokens could partially mitigate this issue, it incurs prohibitive computational overhead during both training and dense surface querying [18].

To address these limitations, we propose *Multi-Scale VecSet VAE (MSVS-VAE)*, a hierarchical set-based VAE that pushes the reconstruction capability of VecSet representations to a new level. Our key idea is to *progressively densify the anchored VecSet* to enhance the capability for geometric details through a hierarchical latent upsampling mechanism, thereby increasing the spatial capacity available for modeling fine geometric details. Specifically, we introduce a *Hierarchical Point-Shuffle Upsampling* module [32, 38] that expands each anchored latent token into multiple localized sub-latents, progressively increasing the spatial density of the latent set while maintaining a compact overall representation. This densified anchored latent set equips VecSet-style VAEs with the spatial resolution necessary to recover fine-grained surface structures.

To efficiently operate on this expanded latent hierarchy, we further replace expensive global cross-attention with a *geometry-aware local aggregation* operator, termed *AVS-Conv*. By querying only local neighborhoods in the latent space, AVS-Conv significantly reduces the computational cost of decoding while preserving the spatial inductive bias necessary for accurate surface reconstruction. Finally, we introduce a *Multi-Scale Query Decoding* module that aggregates information across multiple latent resolutions, allowing coarse latents to provide stable and long-range geometric context while fine latents capture localized geometric details [1, 22, 27]. Fig. 1 illustrates that our method reconstructs high-fidelity geometry from a compact anchored VecSet representation.

Together, these components enable set-based VAEs to achieve voxel-level *reconstruction fidelity* while maintaining *compactness* and improving the *decoding efficiency*. Our main contributions are summarized as follows:

1. We propose **MSVS-VAE**, a novel set-based 3D VAE that enables high-fidelity 3D reconstruction while maintaining compact latent representations. Our method achieves state-of-the-art reconstruction performance while providing $\sim 10\times$ faster decoding than prior VecSet methods and $\sim 10\times$ more compact representations than voxel-based baselines.
2. We introduce a *Hierarchical Point-Shuffle Upsampling* module together with a *Multi-Scale Query Decoding* module, which progressively increases the spatial resolution of the anchored VecSet and aggregates information across scales, significantly improving reconstruction quality while preserving global compactness.
3. We develop **AVS-Conv**, a geometry-aware local aggregation operator that replaces global cross-attention with efficient neighborhood-based decoding, substantially reducing computational cost while preserving fidelity.

2 Related Works

Various 3D representations have been explored [13, 28, 39–41, 47], but modern *native* 3D generation largely centers on two paradigms: voxel-based and set-based representations.

2.1 Voxel-based Representations

Voxel-based methods scale 3D generation by introducing explicit sparse structures and multi-stage pipelines. XCube [34] generates hierarchical sparse voxels in a coarse-to-fine manner, enabling high effective resolutions. Octfusion [46] compresses surface into an octree latents, and decodes continuous SDFs with MPU [31]. GaussianCube [50] constructs a grid-based 3DGS [14] representation via optimal transport. Trellis [45] proposes structured latents that jointly model appearance and geometry, where geometry is represented via differentiable FlexiCubes [37]. SparseFlex [9] further improves scalability with self-pruning and frustum-aware training, supporting high-resolution reconstruction and open-surface modeling. Direct3D-S2 [42] and SparC3D [23] emphasize representation consistency by symmetrically encoding and decoding SDF-like volumetric signals (or sparse cubes). Concurrently, Trellis2 [44] and FaithC [25] adopts a dual-contouring-inspired [4, 12] voxel representation to better capture complex topology and sharp features. Despite strong reconstruction fidelity, voxel-based pipelines often co-define topology via explicit sparse structure and implicit field, making them sensitive to structure errors (e.g., holes).

2.2 Set-based Representations

Set-based methods [48, 49] encode 3D surfaces into compact latent sets and decode SDFs via continuous, query-based operators. 3DILG [48] queries latents through inverse-distance weighting, whereas VecSet [49] replaces handcrafted interpolation with cross-attention, enabling single-stage generation. Michelangelo [53] aligns the latent set with pretrained CLIP [33] model for conditional generation, while Craftsman3D [19] refines the generated model with enhanced normals. Dora [2] improves geometric quality via importance sampling to prioritize informative queries. OAT [7] introduces octree-based adaptive tokenization that encodes 3D shape into a variable-length set aligned with local geometric complexity. CLAY [51], TripoSG [21], seed3d [36], and the Hunyuan3D series [11, 16, 52] further demonstrate the scalability of set-based latents, improving fidelity by scaling to larger models and datasets. To push fidelity further, LAT-TICE [17] introduces a two-stage refinement scheme that re-injects explicit positional information into the VecSet diffusion framework, mirroring voxel-based pipelines to better recover fine-grained details. Nevertheless, set-based methods remain bottlenecked by VAE reconstruction quality, which still trails voxel-based alternatives [9, 23, 42] in capturing fine-scale geometry.

3 Preliminaries: Set-based 3D Representations

Let a 3D shape be characterized by its continuous Signed Distance Function (SDF), denoted as $S(q) \in \mathbb{R}$ for any spatial query coordinate $q \in \mathbb{R}^3$. In contrast to methods that discretize space into explicit sparse grids, set-based representations encode the underlying 3D geometry into a discrete, permutation-invariant set of 1D latent vectors $\mathcal{Z} = \{z_i \in \mathbb{R}^C\}_{i=1}^M$. In this framework, M specifies the number of latent tokens and C defines the feature dimensionality. By abstracting the surface into a compact collection of latents, this representation naturally enables resolution-independent queries and provides a more flexible foundation for downstream generative modeling compared to grid-tied alternatives.

The VAE typically consists of an encoder that maps the input geometry into $\mathcal{Z}$, and a continuous, query-based decoder $\mathcal{D}$ that reconstructs the geometry. For any given spatial coordinate q , the SDF value is predicted by aggregating information from the global latent set:

$$S(q) \approx \mathcal{D}(q, \mathcal{Z})$$

In standard cross-attention-based frameworks (e.g., VecSet [49]), this decoding is achieved by treating the spatial query q —often enriched via positional encoding [26]—as the attention query, while the latent set $\mathcal{Z}$ serves as the keys and values. The extracted query feature is subsequently passed through a Multi-Layer Perceptron (MLP) to output the final SDF. During inference, it receives discrete grid queries and extracts mesh with iso-surface algorithms [24, 29].

4 Methodology

In this section, we present **MSVS-VAE**, a hierarchical set-based VAE for high-fidelity 3D reconstruction. We begin with a pipeline overview in Section 4.1, followed by detailed descriptions of the encoder in Section 4.2 and the decoder in Section 4.3.

4.1 Overview: MSVS-VAE

An overview of the proposed pipeline is illustrated in Fig. 2. Given an input point cloud and a set of voxel queries, we first extract a compact latent set via *AVS-Conv* from the input point cloud. The resulting tokens are further refined by several self-attention blocks to produce anchored VecSet latents. The compact anchored latent set is subsequently densified by *Hierarchical Point-Shuffle Upsampling*, which progressively increases latent spatial density over multiple stages. Finally, the *Multi-Scale Query Decoding* module predicts SDF values at dense grid queries by aggregating features from neighboring latents across all upsampling scales.

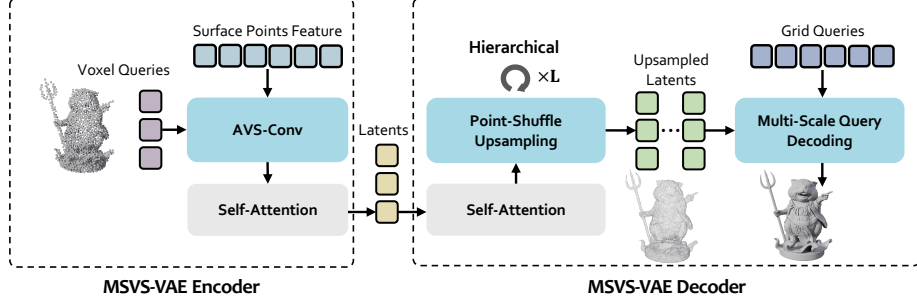


Fig. 2: Overview of the proposed *MSVS-VAE*. The encoder extracts compact latents from surface points via AVS-Conv and self-attention blocks, while the decoder progressively upsamples the anchored latents and predicts SDF using a multi-scale query decoding module.

4.2 MSVS-VAE Encoder

Given an input point cloud $\mathcal{X} = \{(x_n, a_n)\}_{n=1}^N$, where $x_n \in \mathbb{R}^3$ is the 3D position and $a_n \in \mathbb{R}^3$ is the surface normal, we first extract point-wise features $f_n = \Psi(a_n, x_n)$ using a lightweight PointNet.

Following Lattice [17], we sample latent anchor points $\mathcal{C} = \{c_m\}_{m=1}^M$ as voxel queries from the centers of active voxels intersected by the mesh, obtained from an off-the-shelf generative model (e.g., HY3D2.1 [11]).

Treating $\mathcal{C}$ as the query set and $\mathcal{X}$ as the support set, we apply AVS-Conv to aggregate point-wise features from the local neighborhood of each query, producing latent tokens $\mathcal{Z} = \{(c_m, z_m)\}_{m=1}^M$. These tokens are subsequently refined by a stack of residual self-attention blocks. The AVS-Conv is defined as follows:

AVS-Conv. Different from previous VecSet-based methods [17, 49], which rely on global cross-attention between latent queries and all input points, incurring $\mathcal{O}(MN)$ complexity and mixing information across the entire set. In contrast, we propose a geometry-aware local query-to-support operator whose cost scales more favorably with the number of queries while preserving locality.

As shown in Fig. 3(a), for each query point q_i in the query set, we retrieve its K nearest neighbors $\mathcal{N}(i)$ from the support set under the Euclidean metric $\|p_j - q_i\|_2$. This explicit KNN retrieval defines a localized receptive field and eliminates the need for dense attention. Inspired by PointConv [43], we aggregate features over $\mathcal{N}(i)$ by predicting dynamic weights from relative offsets $\Delta p_{ij} = p_j - q_i$. Concretely, we compute the weight as:

$$w_{ij} = \text{MLP}(\gamma(\Delta p_{ij})), \quad j \in \mathcal{N}(i), \quad (1)$$

where $\gamma(\cdot)$ denotes a positional embedding applied to the relative offset Δp_{ij} . The resulting AVS-Conv is formulated as:

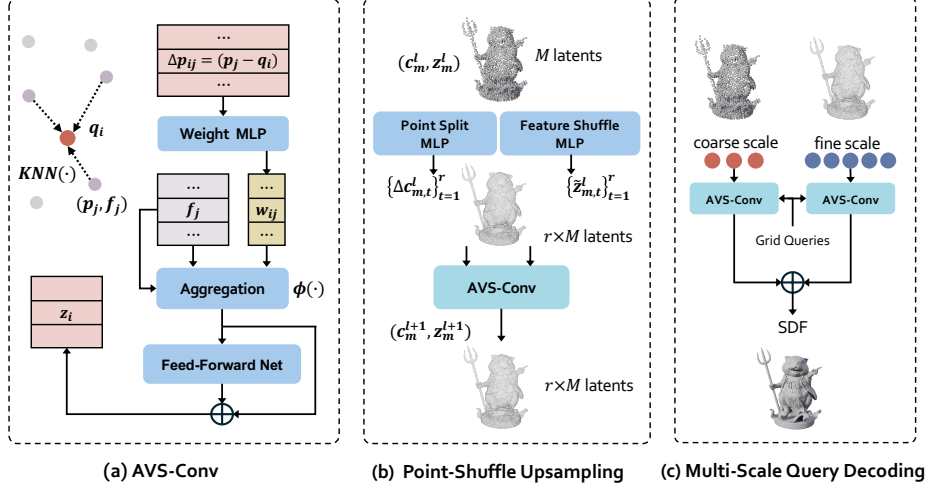


Fig. 3: Illustrations of the three core modules in MSVS-VAE. (a) **AVS-Conv**: aggregates local support features for each query via dynamic KNN-weighted convolution and a feed-forward residual block. (b) **Point-Shuffle Upsample**: expands each latent into r child latents by splitting both its geometry and feature, followed by AVS-Conv refinement. (c) **Multi-Scale Query Decoding**: fuses features independently queried from coarse and fine latent scales to predict the SDF.

$$VC(q_i, \mathcal{N}(i)) = \sum_{j \in \mathcal{N}(i)} w_{ij} \odot f_j, \quad (2)$$

where f_j is the support feature and $\odot$ denotes channel-wise multiplication. To improve optimization stability, we further incorporate a residual shortcut implemented as lightweight mean pooling over the same neighborhood:

$$\phi(q_i; \mathcal{S}) = VC(q_i, \mathcal{N}(i)) + \text{Mean}(q_i, \mathcal{N}(i)). \quad (3)$$

Together, these components define a geometry-aware local aggregation operator $\phi(\cdot)$. Finally, we adopt a Transformer-style residual refinement with LayerNorm and a feed-forward network to produce the output feature z_i :

$$z_i = \phi(q_i; \mathcal{S}) + \text{FFN}(\text{LN}(\phi(q_i; \mathcal{S}))). \quad (4)$$

Note that AVS-Conv is employed in both the encoder and decoder. In the decoder, it serves as a fundamental operator in both the Hierarchical Point-Shuffle Upsampling module and the Multi-Scale Query Decoding module. The main distinction lies in how the query and support sets are instantiated: in the encoder, the queries are VecSet anchors, and the support set comprises surface points; in the decoder, the queries are dense grid points, and the support set is the progressively upsampled anchored latent set.

4.3 MSVS-VAE Decoder

As discussed in Section 1, the sparsity of VecSet representations—often tens of times more compact than sparse voxels—limits their ability to recover fine geometric details. Therefore, we propose a hierarchical VecSet latent upsampling module to enhance the capability of VecSet.

Hierarchical Point-Shuffle Upsampling. Given a compact latent set $\mathcal{Z} = \{(c_m, z_m)\}_{m=1}^M$ in Fig. 3(b), we progressively increase the spatial density of VecSet latents while reducing per-latent channel capacity. Specifically, we apply L upsampling stages to construct a hierarchy of latent supports

$$\mathcal{Z}^{(l)} = \{(c_m^{(l)}, z_m^{(l)})\}_{m=1}^{M_l}, \quad l = 0, \dots, L,$$

where $\mathcal{Z}^{(0)} = \mathcal{Z}$ and M_l grows exponentially with l .

For each stage $l \rightarrow l+1$, we apply a point-shuffle upsampling step. Specifically, for each latent feature $z_m^{(l)}$ in $\mathcal{Z}^{(l)}$, we split it into r sub-features $\{z_{m,t}^{(l)}\}_{t=1}^r$ through a feature shuffle MLP with reduced channel dimension $C_{l+1} = C_l/2$. In parallel, we predict r geometric offsets $\{\Delta c_{m,t}^{(l)}\}_{t=1}^r$ from $z_m^{(l)}$ using a point split MLP and form new centers

$$c_{m,t}^{(l+1)} = c_m^{(l)} + \Delta c_{m,t}^{(l)}, \quad z_{m,t}^{(l+1)} = z_{m,t}^{(l)}.$$

This transforms M_l latents into $M_{l+1} = r \times M_l$ latents while keeping the total feature budget limited.

We then refine the upsampled latent set $\mathcal{Z}^{(l+1)}$ using AVS-Conv over the updated centers $c^{(l+1)}$ and features $z^{(l+1)}$. Here, AVS-Conv is applied in a self-aggregation manner, where the query set and the support set are both the upsampled latent set $\mathcal{Z}^{(l+1)}$. Moreover, the residual shortcut in $\phi(\cdot)$ is an identity connection on the latent features, rather than the mean pooling in the MSVS-VAE encoder stage.

Multi-Scale Query Decoding. A fundamental distinction of our decoder from standard VecSet architectures is that we eliminate global cross-attention. In Fig. 3(c), To predict the signed distance function (SDF) at an arbitrary spatial query $q_i \in \mathbb{R}^3$, we retrieve features from its K -NN latents using AVS-Conv. This local querying strategy avoids attending to all latents globally, and thus remains efficient even when the upsampled latent set becomes dense.

Relying only on the *finest latent scale* can lead to high-frequency artifacts in smooth regions, since the receptive field becomes overly local. To mitigate this, we propose a multi-scale query fusion mechanism that aggregates information from both coarse and fine latent scales. Specifically, we query features from all L progressive latent scales using separate AVS-Conv operators, concatenate the resulting features, and apply a lightweight fusion head:

$$h_i^{(l)} = \text{AVS}^{(l)}(q_i; \mathcal{Z}^{(l)}), \quad h_i = \text{Fuse}([h_i^{(1)} \parallel \dots \parallel h_i^{(L)}]) \quad (5)$$

The final SDF value is predicted via $\hat{s}_i = \text{MLP}_{\text{out}}(h_i)$. This multi-scale design lets coarse latents supply *stable, long-range geometric context*, while fine latents

sculpt intricate local residuals, enabling voxel-level fidelity within a continuous set-based representation.

Training Objectives. Following standard practices [21] for 3D VAEs, we supervise the reconstruction using the Truncated Signed Distance Function (TSDF) to prioritize near-surface geometry. The raw ground truth SDF s is truncated and normalized as $\bar{s} = \text{clamp}(s/\tau, -1, 1)$, where the truncation distance is empirically set to $\tau = 1/128$. The overall training objective combines a geometric reconstruction loss and a Kullback-Leibler (KL) divergence regularizer:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{recon}} \left(\|\hat{s} - \bar{s}\|_1 + \|\hat{s} - \bar{s}\|_2^2 \right) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} \quad (6)$$

where $\hat{s}$ denotes the predicted TSDF, and $\mathcal{L}_{\text{KL}}$ regularizes the latent tokens towards a standard normal distribution $\mathcal{N}(0, I)$.

5 Experiments

5.1 Implementation Details

We implement our MSVS-VAE using PyTorch and Triton kernels. For nearest-neighbor search, we employ the BVH acceleration structure in cuBQL [30], which significantly speeds up K -NN computation. We train our MSVS-VAE on 400k high-quality meshes filtered from Objaverse [6].

Training Strategy. To overcome the memory bottleneck of high-fidelity continuous surface learning, we adopt a progressive chunk-based training scheme. We first pre-train on full 3D assets with a fixed latent budget of $M=4096$ tokens to capture global topology and low-frequency geometry. We then fine-tune with localized random 3D crops: the same 4K tokens are allocated only to the cropped region (with a small spatial padding for context), effectively increasing local spatial resolution for high-frequency detail refinement. During the localized stage, we supervise SDF only on near-surface query points, as far-field SDF values depend on global topology beyond the sampled chunk and are therefore ill-posed under strictly local latents.

Training Settings. We train on 8 GPUs with a batch size of 4 per GPU for both stages. The global pre-training runs for 300k steps, followed by 100k steps of localized chunk fine-tuning. We use AdamW with a learning rate of 1×10^{-4} , $\beta = (0.9, 0.99)$, $\epsilon = 10^{-6}$, and weight decay 10^{-2} .

5.2 Evaluation Settings

Dataset. Following SparC3D [23] and Dora [2], we curate an evaluation benchmark by selecting challenging shapes from Objaverse [6], ABO [5], and in-the-wild Internet sources. Our test set focuses on shapes with thin structures and complex topology, from which qualitative examples are sampled. We further evaluate on the Level-3 and Level-4 subsets of Dora-Bench [2], which provide a widely used benchmark for challenging shapes with rich geometric details.

Table 1: VAE reconstruction evaluation on Objaverse, ABO, and in-the-wild benchmarks for voxel-based and set-based methods. † denotes our re-implementation.

Method	#tokens	Objaverse			ABO			in-the-wild		
		MD ↓	F1@0.01 ↑	F1@0.001 ↑	MD ↓	F1@0.01 ↑	F1@0.001 ↑	MD ↓	F1@0.01 ↑	F1@0.001 ↑
<i>Voxel-based methods</i>										
Direct3D-S2	~74k	3.301	99.955	95.100	2.091	99.987	97.181	2.771	99.928	97.020
SparC3D [†]	~50k	2.463	99.986	97.788	1.515	99.947	99.354	1.947	99.999	98.137
SparseFlex-512	~50k	3.745	99.867	93.182	2.017	99.958	97.449	4.404	99.915	90.634
SparseFlex-1024	~210k	2.165	99.999	97.327	1.092	100.00	99.902	2.014	99.998	98.944
<i>Set-based methods</i>										
Dora	4k	17.352	98.541	48.450	7.450	99.952	73.770	14.641	99.645	44.110
HY3D2.1	4k	11.485	99.717	63.085	5.384	99.978	84.036	11.895	99.948	57.096
Lattice [†]	4k	11.502	99.273	82.365	2.941	99.982	97.107	10.703	99.584	74.764
Lattice [†]	10k	4.146	99.944	92.046	2.445	99.996	98.426	4.814	96.968	87.519
Lattice [†]	20k	3.183	99.860	95.373	2.283	99.997	99.088	3.737	99.996	91.872
Lattice [†]	40k	2.838	99.991	96.787	2.239	99.997	98.675	3.200	100.00	94.088
MSVS-VAE	4k	5.644	99.714	86.779	1.885	99.956	98.902	4.589	99.984	87.207
MSVS-VAE	10k	2.422	99.994	96.458	1.485	99.999	99.843	2.416	99.999	95.842
MSVS-VAE	20k	1.695	99.996	98.930	1.432	99.999	99.927	1.642	100.00	98.775
MSVS-VAE	40k	1.395	99.992	99.621	1.216	99.995	99.924	1.231	100.00	99.726

Baselines. We compare our method with the strongest prior *set-based* approaches, including Lattice [17], HY3D 2.1 [11], and Dora [2], as well as state-of-the-art *voxel-based* methods, including SparseFlex [9], SparC3D [23], and Direct3D-S2 [42], where Lattice and SparC3D are our re-implementations following their original papers.

Metrics. We evaluate reconstruction quality using standard surface metrics. Mesh Distance (MD, $\times 10^4$) is computed as the bidirectional point-to-mesh distance. We also report the F1-score ($\times 10^2$) at thresholds 0.01 and 0.001, which measures the harmonic mean of precision and recall based on whether bidirectional surface samples fall within the given distance threshold. To better evaluate sharp geometric details, we additionally report sharp-edge-specific metrics on Dora-Bench, including one-sided Mesh Distance and F-score evaluated on ground-truth sharp points, denoted as S-MD and S-F, respectively.

We additionally benchmark **query latency** against prior cross-attention-based methods to validate the efficiency of our AVS-Conv. Specifically, we report the average wall-clock time in milliseconds for **chunk-wise** point queries, using a fixed chunk size of 200k points per forward pass. We evaluate three token budgets, 10k, 20k, and 40k, for both our method and Lattice.

5.3 Evaluation Results

Quantitative Comparisons. As shown in Tab. 1, our method substantially and consistently surpasses the Lattice [17] baseline across all token budgets, benchmarks, and evaluation metrics, yielding a large improvement in geometry reconstruction quality.

Furthermore, using only 20K latent tokens, our method already outperforms competing voxel-based approaches on both Objaverse and in-the-wild benchmarks. It requires only **~10%** (Ours: 20K *vs* SparseFlex-1024: 210k) of the

Table 2: Evaluation on the Level-3 and Level-4 subsets of Dora-Bench for challenging shapes with sharp geometric details. MD and S-MD are scaled by 10^{-4} ; F and S-F are reported as percentages.

Method	Level-3				Level-4			
	MD↓	F↑	S-MD↓	S-F↑	MD↓	F↑	S-MD↓	S-F↑
Direct3D-S2	3.99	95.6	5.37	91.3	2.64	98.4	4.26	93.2
SparC3D	1.65	99.0	2.80	96.4	2.14	97.4	3.22	94.4
SparseFlex	1.74	99.1	3.21	95.6	1.91	99.1	3.08	96.2
MSVS-VAE-20K	1.14	99.7	3.22	96.4	1.58	99.3	3.58	95.5
MSVS-VAE-40K	1.06	99.9	2.99	97.4	1.44	99.7	3.15	97.9

tokens used by SparseFlex-1024, and $\sim 40\%$ of those required by SparC3D and Direct3D-S2. As shown in Tab. 2, MSVS-VAE also achieves strong performance on Dora-Bench Level-3 and Level-4 subsets, especially on sharp-edge-specific metrics. These quantitative comparisons demonstrate that our method achieves state-of-the-art geometric reconstruction quality with significantly more compact latents compared with the voxel-based method.

With only $4K$ tokens—the same budget as Dora and HY3D2.1—our method still outperforms them by a large margin on Objaverse (*e.g. on Mesh Distance*, Ours: 5.644 *vs* Dora: 17.352 *vs* HY3D2.1: 11.485). The reported token numbers are averaged across all evaluation datasets.

Qualitative Comparisons. Fig. 4 shows that our method faithfully reconstructs high-frequency surface details (*see col. 1*), complex structures (*see col. 2*), and delicate thin components (*see col. 3–4*). Compared with *set-based* methods, our approach achieves higher-fidelity reconstruction at both global and local levels, recovering intricate geometric details by leveraging the upsampled VecSet.

In addition, our method clearly outperforms most *voxel-based* (SparC3D and Direct3D-S2) baselines in reconstruction quality, producing sharper boundaries and substantially fewer surface sticking artifacts (*see col. 3*). Despite using only $\sim 10\%$ of the tokens, our results remain competitive with or slightly better than SparseFlex-1024. Moreover, our reconstructions are more topologically robust and watertight, avoiding the holes and broken parts observed in SparseFlex (*see the red circle in col. 4; zoom in for details*). Overall, these results highlight a key advantage of set-based representations and demonstrate that our approach pushes the reconstruction capability of VecSet to the next level.

Discussion about SparseFlex: Although the improvements over SparseFlex-1024 in both visual quality and quantitative metrics are modest, it is important to note that SparseFlex-1024 operates with a substantially denser latent space. Specifically, it uses a spatial resolution of $256 \times 256 \times 256$ with a compression ratio of only $4 \times 4 \times 4$, whereas most voxel-based pipelines adopt a downsampling factor of around 8 to keep the latent space manageable. Such a dense latent space poses significant challenges for generative modeling due to the increased computational and modeling complexity.

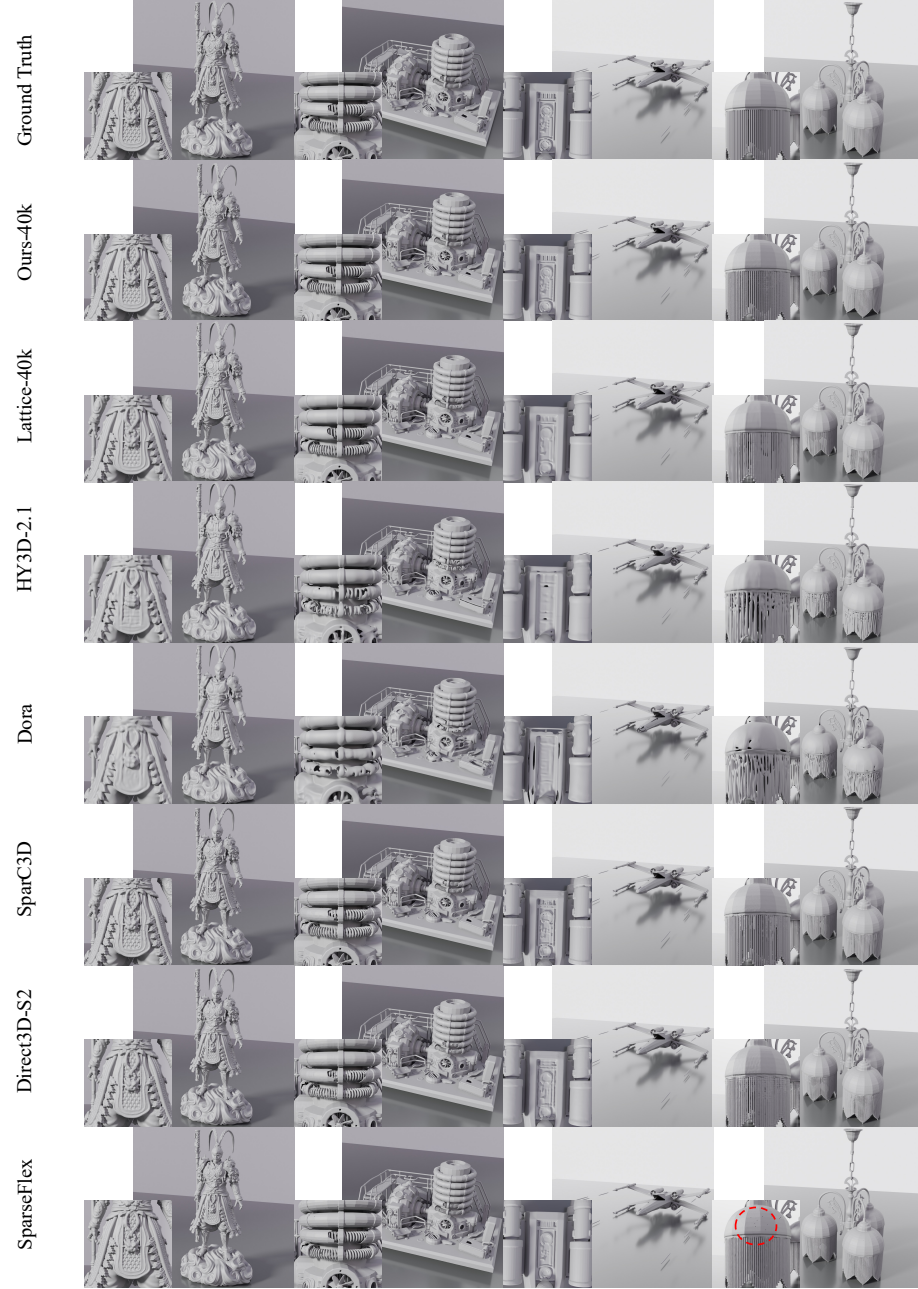


Fig. 4: Qualitative comparison of VAE reconstruction. *Best viewed with zoom-in.*

Table 3: Ablation study by component removal. Performance on Objaverse, ABO, and in-the-wild (MD↓, F1@0.01↑, F1@0.001↑) after removing MSQ or PSU from the full model. Both modules contribute, and PSU is the most critical.

Setting	MSQ	PSU	Objaverse			ABO			in-the-wild		
			MD↓	F1@0.01↑	F1@0.001↑	MD↓	F1@0.01↑	F1@0.001↑	MD↓	F1@0.01↑	F1@0.001↑
Full model	✓	✓	1.395	99.992	99.621	1.216	99.995	99.924	1.231	100.00	99.726
w/o MS-Query (MSQ)	✗	✓	1.690	99.959	99.322	1.439	99.967	99.890	1.626	99.977	99.416
w/o Upsample (PSU)	✗	✗	3.693	99.932	93.281	2.818	99.930	98.261	4.021	99.986	90.675

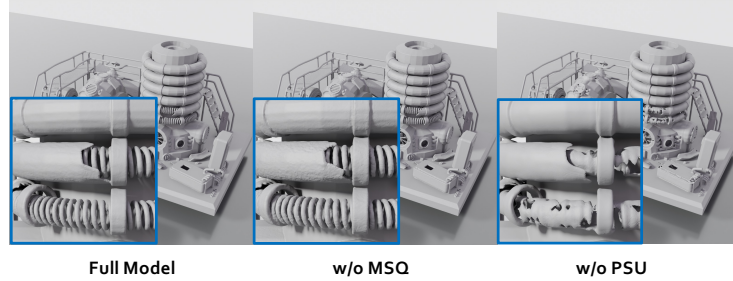


Fig. 5: Qualitative ablation results of the proposed components. Removing MSQ results in noisier surfaces (*Please zoom in for more details*), while removing PSU significantly degrades geometric reconstruction. The full model achieves smoother surfaces and more faithful reconstruction.

5.4 Ablation Studies

Multi-scale Query Decoding (MSQ). As shown in Tab. 3, multi-scale query decoding consistently improves over the single-scale variant, yielding lower MD and higher F-scores across all benchmarks. Fig. 5 further shows that MSQ produces smoother surfaces with fewer noise-induced artifacts, as fusing features across resolutions provides global context for structural consistency while preserving fine-scale cues for local details.

Hierarchical Point-Shuffle Upsampling (PSU). As shown in Tab. 3, PSU is the most impactful component: removing it causes a drastic degradation in both MD and F-scores across all benchmarks. Fig. 5 further shows that PSU substantially improves geometric detail fidelity by progressively densifying the VecSet latents, thereby strengthening local geometric modeling and improving the recovery of high-frequency details. This module is therefore essential for pushing set-based representations toward voxel-level reconstruction quality.

Decoding Efficiency. Moreover, we compare query decoding efficiency against Lattice in Tab. 4. Thanks to our geometry-aware local aggregation operator (*AVS-Conv*), decoding latency increases much more slowly with the number of tokens, yielding consistent speedups over Lattice of $\sim 5.5\times/9.4\times/14.4\times$ at 10k/20k/40k tokens, respectively. These results demonstrate that *AVS-Conv* significantly reduces the computational cost of query decoding, enabling substantially faster decoding than existing *set-based methods*.

Table 4: Query Latency Comparison. We report the average latency (ms) for SDF querying with a chunk size of 200k. Our localized *AVS-Conv* demonstrates significantly better scalability compared to the global-attention-based Lattice [17].

# Latent Tokens	Lattice (ms)	Ours (ms)	Speedup
10k	792.21	143.75	5.5×
20k	1419.27	151.22	9.4×
40k	3068.05	213.29	14.4×



Fig. 6: Qualitative test-time scaling with respect to token budget. Increasing the number of tokens at inference time progressively improves reconstruction quality, enabling more faithful recovery of fine-grained geometric details.

Test-Time Scaling (TTS). We also evaluate the test-time scaling ability of our MSVS-VAE by varying the latent set size at inference time. We report reconstruction metrics with 4K, 10K, 20K, and 40K tokens on Objaverse, ABO, and in-the-wild. As shown in Tab. 1, increasing the number of tokens consistently improves reconstruction quality. We additionally evaluate 30K, 60K, and 80K tokens on the in-the-wild benchmark and plot the results. Fig. 7 shows that, under the same token budget, our method consistently outperforms all competing approaches. Notably, using only *20K* tokens, our model can achieve state-of-the-art performance, despite voxel-based methods operating with substantially larger latent spaces (e.g., 80K or 210K tokens). Fig. 6 further illustrates test-time scaling: increasing the token budget consistently improves reconstruction quality.

6 Conclusion

We present MSVS-VAE, a hierarchical set-based VAE that significantly improves the reconstruction capability of VecSet representations. By progressively densifying anchored VecSet latents through hierarchical point-shuffle upsampling and decoding them with multi-scale query aggregation, our approach effectively increases the spatial capacity of compact latent sets while maintaining efficient decoding. In addition, the proposed AVS-Conv operator replaces global cross-attention with geometry-aware local aggregation, substantially reducing computational cost. Extensive experiments demonstrate that our method achieves

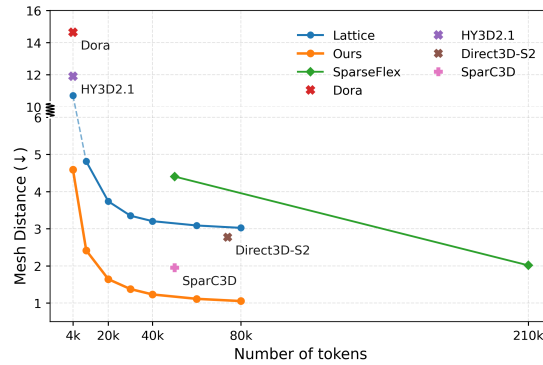


Fig. 7: Reconstruction quality improves as the token budget increases. Our method consistently outperforms Lattice across token scales and achieves strong performance with far fewer tokens than voxel-based methods. (Evaluated on in-the-wild benchmark)

state-of-the-art reconstruction performance across multiple benchmarks, while remaining significantly more compact and efficient than voxel-based alternatives. These results show that compact set-based representations can approach voxel-level reconstruction fidelity, offering a promising direction for scalable 3D generative modeling.

Limitations. Although MSVS-VAE achieves state-of-the-art performance on geometry reconstruction, several limitations remain. (1) Extracting sharp features remains challenging under a set-based continuous implicit formulation. (2) The decoding efficiency is still behind voxel-based methods. (3) Extending MSVS-VAE to texture modeling is an important direction for future work.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
2. Chen, R., Zhang, J., Liang, Y., Luo, G., Li, W., Liu, J., Li, X., Long, X., Feng, J., Tan, P.: Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16251–16261 (2025)
3. Chen, Y., Li, Z., Wang, Y., Zhang, H., Li, Q., Zhang, C., Lin, G.: Ultra3d: Efficient and high-fidelity 3d generation with part attention. arXiv preprint arXiv:2507.17745 (2025)
4. Chen, Z., Tagliasacchi, A., Funkhouser, T., Zhang, H.: Neural dual contouring. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022)
5. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., 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)

6. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13142–13153 (2023)
7. Deng, K., Liu, H.T.D., Zhu, Y., Sun, X., Shang, C., Bhat, K.S., Ramanan, D., Zhu, J.Y., Agrawala, M., Zhou, T.: Efficient autoregressive shape generation via octree-based adaptive tokenization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11685–11696 (2025)
8. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
9. He, X., Zou, Z.X., Chen, C.H., Guo, Y.C., Liang, D., Yuan, C., Ouyang, W., Cao, Y.P., Li, Y.: Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14822–14833 (2025)
10. Hui, K.H., Li, R., Hu, J., Fu, C.W.: Neural wavelet-domain diffusion for 3d shape generation. In: SIGGRAPH Asia 2022 conference papers. pp. 1–9 (2022)
11. Hunyuan3D, T., Yang, S., Yang, M., Feng, Y., Huang, X., Zhang, S., He, Z., Luo, D., Liu, H., Zhao, Y., et al.: Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. arXiv preprint arXiv:2506.15442 (2025)
12. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of hermite data. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques. pp. 339–346 (2002)
13. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023)
14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139–1 (2023)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
16. Lai, Z., Zhao, Y., Liu, H., Zhao, Z., Lin, Q., Shi, H., Yang, X., Yang, M., Yang, S., Feng, Y., et al.: Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. arXiv preprint arXiv:2506.16504 (2025)
17. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. arXiv preprint arXiv:2512.03052 (2025)
18. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Wang, F., Shi, H., Yang, X., Lin, Q., Huang, J., Liu, Y., et al.: Unleashing vecset diffusion model for fast shape generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2523–2533 (2025)
19. Li, W., Liu, J., Yan, H., Chen, R., Liang, Y., Chen, X., Tan, P., Long, X.: 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 (2025)
20. Li, W., Zhang, X., Sun, Z., Qi, D., Li, H., Cheng, W., Cai, W., Wu, S., Liu, J., Wang, Z., et al.: Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. arXiv preprint arXiv:2505.07747 (2025)
21. Li, Y., Zou, Z.X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.C., Liang, D., Ouyang, W., et al.: Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)

22. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8456–8465 (2023)
23. Li, Z., Wang, Y., Zheng, H., Luo, Y., Wen, B.: Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. arXiv preprint arXiv:2505.14521 (2025)
24. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
25. Luo, Y., He, X., Pan, C., Chen, Y., Wu, J., Li, Y., Ouyang, W., Hu, Y., Yang, G., Yap, C.: Faithful contouring: Near-lossless 3d voxel representation free from iso-surface. arXiv preprint arXiv:2511.04029 (2025)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
27. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
28. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
29. Nielson, G.M.: Dual marching cubes. In: IEEE visualization 2004. pp. 489–496. IEEE (2004)
30. Nvidia: cubql. <https://github.com/NVIDIA/cuBQL> (2025), accessed: 2026-06-30
31. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicit. In: Acm Siggraph 2005 Courses, pp. 173–es (2005)
32. Qian, G., Abualshour, A., Li, G., Thabet, A., Ghanem, B.: Pu-gcn: Point cloud upsampling using graph convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11683–11692 (2021)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
34. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4209–4219 (2024)
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
36. Seed, B.: Seed3d 1.0: From images to high-fidelity simulation-ready 3d assets (2025)
37. Shen, T., Munkberg, J., Hasselgren, J., Yin, K., Wang, Z., Chen, W., Gojcic, Z., Fidler, S., Sharp, N., Gao, J.: Flexible isosurface extraction for gradient-based mesh optimization. ACM Transactions on Graphics (ToG) **42**(4), 1–16 (2023)
38. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)

39. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4563–4573 (2023)
40. Wu, L., Wang, D., Gong, C., Liu, X., Xiong, Y., Ranjan, R., Krishnamoorthi, R., Chandra, V., Liu, Q.: Fast point cloud generation with straight flows. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9445–9454 (2023)
41. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems* **37**, 121859–121881 (2024)
42. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Yang, Y., Bao, Y., Qian, J., Zhu, S., Cao, X., Torr, P., et al.: Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412* (2025)
43. Wu, W., Qi, Z., Li, F.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9613–9622 (2019)
44. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., et al.: Native and compact structured latents for 3d generation. *arXiv preprint arXiv:2512.14692* (2025)
45. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21469–21480 (2025)
46. Xiong, B., Wei, S.T., Zheng, X.Y., Cao, Y.P., Lian, Z., Wang, P.S.: Octfusion: Octree-based diffusion models for 3d shape generation. In: *Computer Graphics Forum*. vol. 44, p. e70198. Wiley Online Library (2025)
47. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4541–4550 (2019)
48. Zhang, B., Nießner, M., Wonka, P.: 3dirl: Irregular latent grids for 3d generative modeling. *Advances in Neural Information Processing Systems* **35**, 21871–21885 (2022)
49. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)* **42**(4), 1–16 (2023)
50. Zhang, B., Cheng, Y., Yang, J., Wang, C., Zhao, F., Tang, Y., Chen, D., Guo, B.: Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. *arXiv preprint arXiv:2403.19655* (2024)
51. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)* **43**(4), 1–20 (2024)
52. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025)
53. Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., Fu, B., Chen, T., Yu, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems* **36**, 73969–73982 (2023)

54. Zheng, X.Y., Pan, H., Wang, P.S., Tong, X., Liu, Y., Shum, H.Y.: Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)* **42**(4), 1–13 (2023)