Response to Reviewer Comments

Revised Manuscript Submission

Manuscript Title: [Geometry-Aware Triplane Diffusion for Single Shape Generation with Feature Alignment]
Paper ID: [233]

We sincerely thank the reviewers and the CAD/Graphics 2025 program committee for their detailed and constructive feedback. We are pleased that our manuscript has been conditionally accepted and recommended for publication in Computers & Graphics.

In response to the reviewers suggestions, we have carefully revised the manuscript to improve its clarity, technical depth, and completeness. Below, we provide a point-by-point response to each reviewer comment, outlining the corresponding changes made to the manuscript. A summary of the major revisions is also included to highlight the key improvements.

1 Response to Reviewer #1

Comment 1.1

Some recently popular works are not mentioned in Section 2.1, such as MeshAnything, 3DShape2VecSet, and TRELLIS

Response:

We have added citations and discussions of MeshAnything [1], 3DShape2VecSet [2], and TRELLIS [3] to Section 2.1 of the revised manuscript.

Comment 1.2

In the first paragraph of Section 2.2, the references related to point clouds, SDF and meshes seem to be mismatched.

Response:

Thank you for pointing this out. We have reviewed and corrected the reference assignments in Section 2.2, ensuring that point cloud, SDF, and mesh-based methods are properly cited and described in the correct context.

Comment 1.3

The index "j" in both the second summation term and the second denominator of Equation (30) appears to be duplicated.

Response:

We acknowledge the typo in Equation (30). The index "j" was inadvertently reused in both summation terms and the denominator. We have corrected this in the revised version to avoid ambiguity.

Comment 1.4

The best metrics in Tables could be highlighted in bold to make the comparison more clear.

Response:

As suggested, we now highlight the best-performing values in bold to improve clarity and readability.

Comment 1.5

The ablation experiment in Table 3 does not involve the setting of "Fixed Point Set + Attention". In my view, since the goal of both "Spatial Pattern" and "Attention" is extracting the spatial distribution of the local region of the 3D shape, I wonder whether simply setting the relative positions of Point Set may also achieve a similar effect?

Response:

Table 1: Comparison of fixed and learned spatial patterns in shape autoencoding.

Decoder Variant	G-Qual ↓	T-Qual ↓
Baseline	0.152	1.265
Fixed Pattern	0.124	0.927
Learned Pattern	**0.098**	**0.652**

Thank you for the insightful suggestion. We have added an ablation variant that combines a fixed point set with attention (i.e., excluding the learned spatial predictor) in Table 1. The results indicate that while the fixed pattern offers modest improvements over the baseline without spatial patterns, it still underperforms relative to our full model with learned spatial offsets. This supports the claim that spatial adaptability and attention play complementary roles in effectively capturing local 3D structure.

Comment 1.6

Section 5.3.1.(2) mentions that when K exceeds 4, the improvement is not significant. So why is K=8 the default? By looking at Table 4, I guess it is to better encode texture information. If I am wrong, please correct me.

Response:

Table 2: Effect of spatial pattern predictor point count K on autoencoding.

Number of Points K	G-Qual ↓	T-Qual ↓
1	0.129	0.825
3	0.113	0.756
4	0.109	0.734
8	**0.098**	**0.652**

We acknowledge that the original text contains a typo. We chose $K = 8$ as a practical trade-off between reconstruction quality and computational efficiency. As shown in Table 2, increasing K beyond 4 yields consistent improvements, particularly in texture fidelity. However, to avoid excessive computational cost in both training and inference, we set $K = 8$ as the default for all main experiments.

Comment 1.7

Section 6 mainly discusses the conclusion, lacking some possible limitations of the current work and potential future works. For example, considering that the autoencoder proposed in this work can recover more accurate geometry and texture information of 3D shapes from triplane, is it possible to train a conditional shape generation model similar to Rodin based on it?

Response:

We have revised the conclusion to discuss limitations and future directions. Notably, we highlight the potential for extending our model to conditional generatione.g., text-driven or pose-guided synthesis, as in Rodin [4]. Given the strong reconstruction capability of our framework, integrating conditioning signals (e.g., language, pose) into the latent space is a natural and promising direction for controllable 3D generation. We also discuss (2) the integration of physically based rendering (PBR) attributes, and (3) reducing reliance on curated meshes by leveraging multi-view inputs such as NeRFs. These additions help clarify the scope of our contribution and outline several meaningful avenues for future research.

2 Response to Reviewer #2

Comment 2.1

Diversity of Generated Shapes. All evaluations focus on mean quantitative scores and per-instance metrics could help demonstrate consistency and robustness across different examples. Additionally it would strengthen the paper to include examples of multiple diverse outputs generated from the same exemplar, to illustrate the methods variety.

Response:

Table 3: Shape generation quality and diversity metrics. Lower G-Qual and T-Qual indicate better quality; higher G-Div and T-Div indicate greater diversity.

Metrics	Methods	Exemplars												
		Stalagmites	Breadbasket	Rock	Bridge	Fruit platter	Paint rack	Fighting Pillar	Train	Ruined tower	Tree	Shelves	City	Avg
G-Qual ↓	SSG	0.420	0.451	0.209	0.394	0.314	0.301	0.280	0.212	0.366	0.214	0.613	0.314	0.341
	Sin3DM	0.212	0.394	0.096	0.192	0.245	0.263	0.211	0.071	0.275	0.094	0.503	0.296	0.238
	Ours	**0.102**	**0.231**	**0.064**	**0.101**	**0.096**	**0.124**	**0.114**	**0.042**	**0.165**	**0.052**	**0.331**	**0.187**	**0.134**
T-Qual ↓	SSG	4.506	5.233	1.580	4.203	3.531	4.202	1.412	7.692	3.512	0.104	6.497	0.957	3.619
	Sin3DM	1.094	3.305	0.783	2.671	1.405	1.042	0.412	5.231	1.411	0.091	3.413	0.565	1.785
	Ours	**0.409**	**1.082**	**0.531**	**1.340**	**0.931**	**0.528**	**0.325**	**4.160**	**0.328**	**0.041**	**0.917**	**0.164**	**0.896**
G-Div ↑	SSG	0.265	0.204	0.101	0.112	0.313	0.137	0.253	0.035	0.194	0.312	0.195	0.147	0.189
	Sin3DM	0.295	0.201	0.080	0.243	0.394	**0.203**	0.290	**0.241**	0.125	0.293	0.112	0.253	0.228
	Ours	**0.428**	**0.227**	**0.102**	**0.376**	**0.491**	0.153	**0.414**	0.091	**0.234**	**0.360**	**0.279**	**0.491**	**0.304**
T-Div ↑	SSG	**0.294**	0.181	0.073	0.105	0.133	0.119	0.197	0.143	0.184	0.043	0.251	0.104	0.152
	Sin3DM	0.243	0.155	0.129	0.146	0.127	0.121	0.152	0.135	0.219	0.086	0.205	0.097	0.151
	Ours	0.292	**0.189**	**0.147**	**0.235**	**0.168**	**0.133**	**0.231**	**0.171**	**0.250**	**0.088**	**0.312**	**0.158**	**0.198**

We appreciate the suggestion to better highlight diversity. In response, we now report per-instance results in Table 3, providing a more detailed view of consistency and robustness across different exemplars. The improvements are consistently observed across most instances, underscoring the general effectiveness and reliability of our approach.

Additionally, we provide new qualitative examples in Figure 1 that illustrate multiple diverse outputs generated from the same exemplar, showcasing our model's ability to produce structurally and texturally varied 3D shapes.

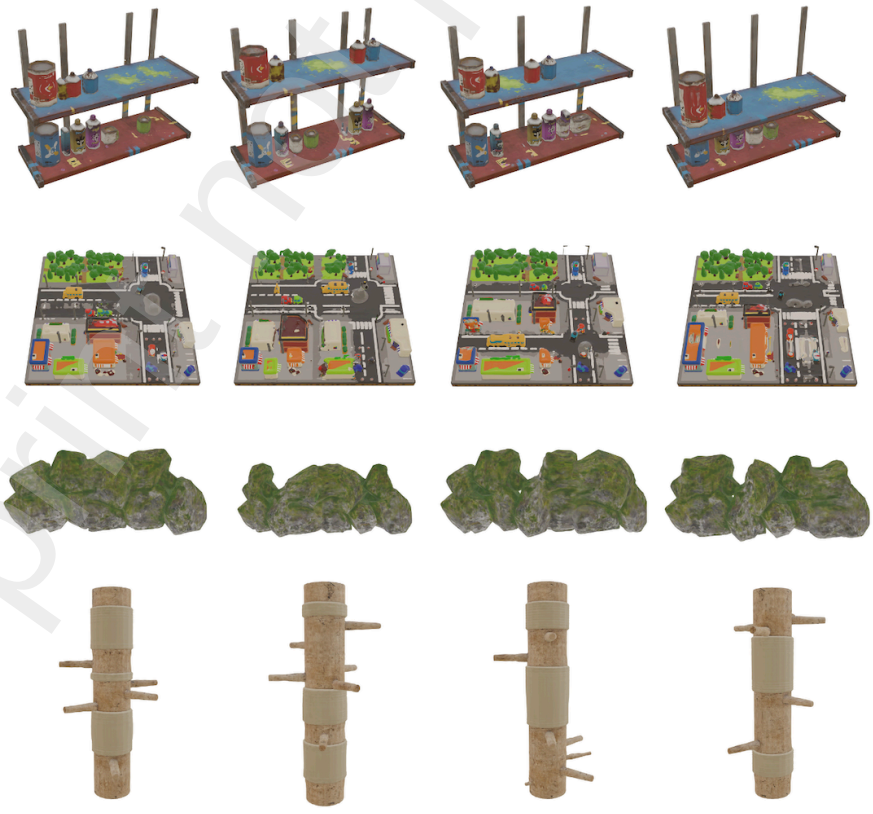

Figure 1: Multiple generations from a single input exemplar, illustrating the diversity and structural plausibility of outputs across different object instances.

Comment 2.2

Practical Applicability. The paper focuses on geometry generation but does not discuss downstream application. For instance, can the framework handle textured shapes or PBR-materialequipped assets? Additionally it edition supported?

Response:

Our framework supports both geometry and texture generation. While it does not currently model physically based rendering (PBR) materials, this remains a promising extension. As noted in Section 6, incorporating PBR cues (e.g., roughness, specularity) could broaden the frameworks applicability to real-world asset pipelines. One possible solution is to extend the triplane representation to predict additional PBR channels during training, using supervisory signals from material-aware renderings. Furthermore, the structured nature of the triplane-based representation may support localized or structure-aware editing by enabling targeted modifications to individual triplane regions or their lifted 3D features. We identify both extensions as compelling directions for future exploration.

Comment 2.3

Extension to Real-World Captures (Brainstorming). This method leverages high-quality 3D meshes during training. This dependency may limit applications. Could the approach be adapted to work with raw multi-view image captures of objectsbypassing the need for clean meshes?

Response:

Thank you for this insightful suggestion. While our current pipeline requires high-quality meshes for training, we agree that removing this dependency would broaden applicability. As now discussed in Section 6, future work could incorporate triplane supervision derived from multi-view image sequences, such as those produced by neural radiance fields (NeRFs) or multi-view stereo pipelines. This would enable learning from in-the-wild captures without relying on curated mesh datasets.

3 Response to Reviewer #3

Comment 3.1

The predicted point offsets in the feature lifting module lack semantic groundingwhat exactly do these offsets represent in practice? It would be helpful to include visualizations or statistical summaries of the learned offsets. Additionally, it is unclear whether this mechanism offers a clear advantage over simply increasing the dimensionality of the triplane features. Ablation or validation experiments to distinguish the two are missing.

Response:

Table 4: Statistical analysis of predicted spatial offsets, averaged over 12 objects. Compared to fixed patterns, learned offsets are more localized, directionally adaptive, and spatially diverse.

Pattern Type	Anisotropy Ratio ($\lambda_{\text{max}}/\lambda_{\text{min}}$)	Spatial Entropy ↑
Fixed Pattern	1.00	2.08
Learned Pattern	**3.01**	**3.35**

We appreciate the insightful suggestion. Directly visualizing spatial patterns for every 3D point is nontrivial; therefore, we have added a new section, "Analysis of Learned Spatial Patterns," in the revised manuscript to provide a quantitative characterization. Specifically, we report two statistical metrics in Table 4: *Anisotropy Ratio* and *Spatial Entropy*. The Anisotropy Ratio is defined as $\lambda_{\text{max}}/\lambda_{\text{min}}$, where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the covariance matrix of predicted offsets; higher values indicate stronger directional bias. Spatial Entropy measures the entropy of offset directions on the unit sphere, computed by binning directions into a spherical histogram; higher entropy suggests a more diverse and isotropic sampling pattern. Compared to fixed patterns, our learned offsets yield higher anisotropy and entropy, indicating better alignment with local geometry and greater directional diversity.

For comparison, the fixed pattern is defined as a set of K points arranged uniformly within a small local cube centered at each query location on the triplane. Compared to this fixed uniform sampling, our learned offsets yield higher anisotropy and spatial entropy, indicating better alignment with local geometry and greater directional diversity. Notably, without employing either fixed or learned spatial patterns, the spatial entropy will further degrade, reflecting a loss of directional diversity and a less expressive feature sampling strategy.

A key advantage of using triplanes over volumetric grids is their low-dimensional structure, which aligns naturally with 2D diffusion and offers significant computational savings compared to 3D diffusion frameworks. While increasing the dimensionality of triplane features could improve performance, it undermines this efficiency advantage by substantially increasing the cost of diffusion. In contrast, our use of learned spatial offsets retains the compactness of the triplane representation while adaptively enriching its geometric expressiveness. This mechanism enables the model to capture fine-grained 3D details without incurring the computational overhead associated with higher-dimensional or volumetric alternatives.

Comment 3.2

The multi-scale diffusion setup raises concerns. During training, the blurry triplanes at each resolution appear to be recursively upsampled from the coarsest level. However, during sampling, each scale is conditioned on the denoised output from the previous (lower-resolution) scale. This asymmetry between training and sampling seems problematic and could lead to inconsistencies, yet it is not discussed in the paper.

Response:

We appreciate the concern regarding the apparent asymmetry, where training uses fixed blurry inputs per scale while sampling conditions on the denoised output from the previous scale. To address this, we explicitly design a denoising-deblurring loop during sampling that disentangles blur from noise and progressively refines details scale-by-scale, as described in Section 4.2. This loop aligns the sampling trajectory with the forward diffusion process, ensuring consistency despite the asymmetry. Furthermore, the mixing step in Eq. (19) recombines the denoised and deblurred features with the blurry prior, ensuring that the reconstructed features remain on the correct denoising manifold while compensating for scale-induced blur. These design choices effectively mitigate potential inconsistencies and enhance structural fidelity.

Comment 3.3

Given the complexity of the deblur & denoise scheme, it is worth testing whether using the blurry triplanes as direct conditioning inputs to the diffusion model is a better choice. A comparison or even a brief discussion of this simpler alternative would be useful for understanding the necessity of the proposed formulation.

Response:

Our approach deliberately entangles blur and noise during the forward diffusion process and explicitly disentangles them during sampling via the proposed deblur & denoise procedure. This formulation enables clearer interpretability and more principled control over the generation trajectory, compared to directly conditioning the diffusion model on blurry triplaneswhich would require the model to implicitly resolve both noise and blur simultaneously. Such implicit handling often leads to oversmoothing and reduced fidelity. In contrast, the mixing step in Eq. (19), which reintroduces the blurry prior, serves as a controlled conditioning mechanism that preserves consistency with the training-time degradation process and improves reconstruction fidelity.

Comment 3.4

Writing is generally clear. Except in Table 5, your method is self-attention, not cross-attention.

Response:

Thank you for pointing this out. We apologize for the oversight in Table 5. The attention mechanism used in our method is indeed self-attention, not cross-attention. We have corrected this in the revised manuscript to accurately reflect the implementation.

References

[1] Y. Lin, Z. Xu, J. Wang, Y. Wang, et al., Material anything: Open-vocabulary pbr material generation with diffusion models, arXiv preprint arXiv:2405.14265 (2024).

[2] B. Zhang, J. Tang, M. Niessner, P. Wonka, 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models, ACM Transactions On Graphics (TOG) 42 (4) (2023) 1–16.

[3] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, J. Yang, Structured 3d latents for scalable and versatile 3d generation, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 21469–21480.

[4] Y. Wang, Y. Zhang, D. Luo, C. Qian, F. Xu, W. Xu, X. Wang, Rodin: A generative model for sculpting 3d digital avatars using diffusion, in: Advances in Neural Information Processing Systems (NeurIPS), 2022.

Computers & Graphics (2025)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Geometry-Aware Triplane Diffusion for Single Shape Generation with Feature Alignment

ARTICLE INFO

Article history:

Shape Generation, 3D Representation, Diffusion Model

ABSTRACT

We tackle the problem of single-shape 3D generation, aiming to synthesize diverse and plausible shapes conditioned on a single input exemplar. This task is challenging due to the absence of dataset-level variation, requiring models to internalize structural patterns and generate novel shapes from limited local geometric cues. To address this, we propose a unified framework combining geometry-aware representation learning with a multiscale diffusion process. Our approach centers on a triplane autoencoder enhanced with a spatial pattern predictor and attention-based feature fusion, enabling fine-grained perception of local structures. To preserve structural coherence during generation, we introduce a soft feature distribution alignment loss that aligns features between input and generated shapes, balancing fidelity and diversity. Finally, we adopt a hierarchical diffusion strategy that progressively refines triplane features from coarse to fine, stabilizing training and improving quality. Extensive experiments demonstrate that our method produces high-fidelity, structurally consistent, and diverse shapes, establishing a strong baseline for single-shape generation.

1. Introduction

Generative modeling of 3D shapes has made remarkable advances in recent years, yet the task of single-shape generation—synthesizing diverse and plausible 3D shapes from a single input exemplar—remains largely underexplored [1, 2]. Unlike traditional settings that benefit from large-scale datasets and category-level priors, this scenario demands learning directly from one shape, requiring the model to internalize fine-grained geometric cues and produce coherent, high-fidelity outputs from noise. This presents a unique challenge for designing compact, expressive, and generalizable 3D representations capable of supporting such data-sparse generation.

Recent advances in 3D shape generation include various neural representations such as voxels, point clouds, meshes, implicit fields, and more recently, 2D-based projections like triplanes [3]. Triplane-based diffusion models are popular for their compactness and efficiency, projecting 3D coordinates onto three orthogonal 2D feature planes and decoding features with lightweight MLPs [4]. However, in the single-shape setting, these models have two main drawbacks: the axis-aligned, low-rank triplanes cause structural rigidity that limits fine detail reconstruction, especially in complex regions; and the lack of dataset variation leads to overfitting, producing low-diversity and semantically inconsistent generations.

To address these limitations, we propose a unified framework for single 3D textured shape generation based on triplane diffusion, introducing three key innovations.

First, we address the geometric rigidity inherent in triplane autoencoders, which arises from the assumption that all 3D geometry can be expressed as combinations of features aligned with three fixed, axis-aligned 2D planes. This low-rank constraint imposes strong structural biases and limits the model's ability to capture complex geometry, especially in directionally intricate regions. To overcome

this, we introduce a *Geometry-Aware Feature Lifting* module that moves beyond axis-aligned projections. Specifically, we predict local 3D structural patterns around each query point using a lightweight spatial pattern predictor. These patterns guide an attention-based fusion of triplane features [5], enabling context-aware integration across planes. This design enhances local expressivity and detail reconstruction without the computational overhead of volumetric alternatives.

Second, to mitigate semantically incoherent outputs—such as unnatural part blending, geometric or textural discontinuities, and structural misalignments—we leverage constraints derived from the input shape's triplane features. These features jointly encode geometry and texture, providing rich supervisory signals to guide generation. However, overly rigid constraints may bias the model toward merely replicating the input, thereby suppressing generative diversity. To balance structural fidelity and generative variability, we propose a *Soft Feature Distribution Alignment* mechanism. This approach aligns the distributions of triplane features from the input and generated shapes via a distributional loss, preserving essential structural and appearance characteristics while enabling diverse and novel recompositions.

Third, generating high-resolution triplane features directly from noise is challenging, particularly when capturing fine geometry while preserving global structural coherence. To address this, we propose a *Multiscale Diffusion Mechanism* that synthesizes shapes in a coarse-to-fine manner. This hierarchical strategy mirrors the natural formation of 3D structures—first establishing global shape before refining local details. By enforcing spatial alignment across triplane resolutions, our framework maintains contextual consistency throughout the generation process, improving both training stability and output fidelity.

These components form a unified and effective framework for single-shape 3D generation, leading to notable improvements in geometric accuracy, structural coherence, and shape diversity. We validate the effectiveness of our approach through extensive experiments, consistently demonstrating significant performance gains over recent state-of-the-art methods. Furthermore, comprehensive ablation studies isolate and quantify the contribution of each proposed component.

In summary, our contributions are:

1. Geometry-Aware Feature Lifting: A novel module that improves the expressiveness of triplane representations by integrating local 3D structural context via attention-guided feature fusion.
2. Soft Feature Distribution Alignment: A regularization mechanism that aligns triplane feature distributions to preserve input-conditioned structure and appearance while supporting diverse outputs.
3. Multiscale Diffusion Mechanism: A hierarchical generation pipeline that synthesizes triplane features from coarse to fine resolutions, improving detail fidelity and training stability.

2. Related Work

2.1. 3D Representations for Generative Modeling

Choosing an effective 3D representation is key for generative modeling. Early methods used volumetric grids [6, 7], which support dense shape prediction but suffer from cubic memory complexity. Point cloud-based approaches [8, 9, 10] are more memory-efficient but lack explicit surface connectivity. Mesh-based models [11, 12, 13, 14, 15, 16] capture rich topology but involve complex vertex connectivity and discretization. MeshAnything [17] advances this direction by introducing a general-purpose, promptable segmentation framework for meshes, facilitating improved geometric understanding and feature generalization across diverse object categories. Implicit neural representations [18, 19, 20] define continuous spatial functions, enabling high-resolution and topology-flexible modeling, though often at the cost of slower inference and more challenging generative training.

Hybrid encodings such as triplane representations [3, 21] project volumetric features onto three orthogonal planes, balancing spatial fidelity and efficiency. These have gained traction in neural rendering [22] and generative modeling [23, 24, 25, 26], though their axis-aligned design limits expressiveness for complex geometries. To address this, the proposed geometry-aware lifting modules enhance triplane features by improving spatial sensitivity.

2.2. Diffusion Models for 3D Synthesis

Diffusion models [27, 28, 29] have set new standards in image generation and are increasingly applied to 3D synthesis. These models iteratively refine noise into structured outputs via learned denoising, offering stable training and diverse samples. 3D extensions include voxel-based diffusion [30], point cloud diffusion [10], signed distance field (SDF) diffusion [31], and mesh-based diffusion such as BrepGen [32]. Recent efforts such as 3DShape2VecSet [33] introduce set-based shape embeddings that capture geometry-aware similarity for improved diffusion-based 3D synthesis, demonstrating effective learning of structural relationships without explicit part annotations. TRELLIS [34] builds on this direction by incorporating hierarchical part compositions into diffusion-based generation, enabling interpretable and reusable 3D components across novel object instances.

Hybrid 2D–3D approaches apply diffusion to triplane features for efficient shape generation. Systems like TextMesh [24], Text2Room [25], and BlockFusion [26] generate textured 3D content from language via triplane diffusion but require large, multi-class datasets and do not address learning from limited exemplars. DreamComposer++ [35] expands on these approaches with multi-view conditioning for better content fidelity.

In contrast, single-shape generation demands strong inductive biases and structural priors. Prior works use geometric losses [36, 37], perceptual objectives [38], or adversarial training [39], though these can overly constrain

diversity. Here, soft feature distribution alignment regularizes triplane output distributions without rigid spatial supervision, preserving consistency while allowing diverse plausible shapes.

2.3. Single- and Few-Shot 3D Generation

Learning 3D content from one or a few exemplars remains a significant challenge. Approaches based on meta-learning [40, 41], neural priors [42, 43, 44], and scene-level optimization [45, 46] offer some generalization but typically require extensive pretraining or iterative adaptation. Compositional methods [47, 48, 49] and latent space interpolation [9, 50] leverage learned priors to extrapolate shape diversity from limited data. Self-supervised and contrastive learning [51, 2, 52, 53] have enabled geometry-aware representations without dense supervision. In particular, contrastive multi-view learning [2] shows promise, though it lacks explicit 3D feature alignment. More recent methods integrate local geometric constraints and attention-based reasoning [54, 55] to improve fidelity and generalization.

Diffusion models have recently been adapted for single-example 3D generation [1, 4, 56], drawing inspiration from advances in single-image diffusion [57, 58, 59]. These approaches learn expressive generative models from sparse data, significantly broadening the scope of data-efficient 3D synthesis. Applications such as detailed 3D avatar generation [60] highlight the potential of diffusion models for high-fidelity shape modeling. Additionally, ShapeGPT [61] demonstrates that unified multi-modal models can learn 3D generation from language and vision data, even under limited supervision. Building on these insights, our work explores how diffusion-based shape generation can generalize from sparse exemplars while maintaining geometric fidelity and shape diversity.

3. Preliminaries

Before detailing our proposed framework, we introduce the foundational components and notations used throughout this work, including the problem formulation, the triplane representation for 3D geometry, and the diffusion-based generative process.

3.1. Problem Definition

Let $X \in \mathcal{X}$ denote a single 3D shape exemplar, represented as a high-resolution occupancy grid, signed distance field, point cloud, or mesh. Our goal is to learn a generative model $p(Y \mid X)$ that produces a distribution over plausible shapes $Y \in \mathcal{Y}$, conditioned solely on this single instance. We assume that the shape X can be encoded into a latent representation Z_X, from which a family of samples $\{Y_i\}$ can be generated such that $Y_i \sim p(Y \mid Z_X)$. This set $\{Y_i\}$ should exhibit both fidelity to X and diversity among the generated shapes.

3.2. Triplane Representation

We adopt the triplane representation as an efficient and spatially structured encoding of 3D geometry. A triplane representation

$$T = \{T_{xy}, T_{yz}, T_{xz}\} \tag{1}$$

consists of three orthogonal 2D feature planes aligned with the canonical axes of the 3D volume. Each plane $T_{ab} \in \mathbb{R}^{H\times W\times C}$ encodes per-pixel features, where C is the feature channel dimension.

Given a 3D point $p \in \mathbb{R}^3$, its feature vector is computed by projecting p onto each of the three planes, extracting the corresponding features via bilinear interpolation, and then aggregating them—commonly through concatenation or attention-based fusion. This representation has proven effective for encoding 3D structures while allowing efficient computation through 2D convolutions.

3.3. Diffusion Models in Feature Space

We employ a denoising diffusion probabilistic model (DDPM) to learn a conditional generative process in triplane feature space. The diffusion model is trained to reverse a fixed Markov process that gradually corrupts a clean sample t_0 into noise over a sequence of timesteps $\{t_1, \ldots, t_T\}$.

Let $t_0 \sim q(t_0 \mid X)$ denote the clean triplane features derived from the exemplar X, and let $t_T \sim \mathcal{N}(0, I)$ represent the fully corrupted (noisy) features. The forward process follows a predefined noise schedule:

$$q(t_t \mid t_{t-1}) = \mathcal{N}(t_t;\ \sqrt{1-\beta_t}\, t_{t-1}, \beta_t I). \tag{2}$$

The diffusion model then learns the reverse conditional:

$$p_\theta(t_{t-1} \mid t_t, Z_X), \tag{3}$$

typically parameterized by a U-Net conditioned on the latent representation Z_X. At inference time, we generate a sample via:

$$\hat{t}_0 \sim p_\theta(t_0 \mid Z_X), \tag{4}$$

which is subsequently decoded into a 3D shape.

4. Method

Given an input 3D shape X, we encode it into triplane features $T_X = \{T_{xy}, T_{yz}, T_{xz}\}$ using an enhanced triplane autoencoder equipped with a spatial pattern predictor and attention-based fusion. A hierarchical diffusion model then generates new triplane features $\hat{T} \sim p_\theta(T \mid X)$ via progressive denoising from coarse to fine resolution. During training, a soft feature distribution alignment loss encourages structural consistency with the exemplar while supporting generative diversity. We now describe each component in detail.

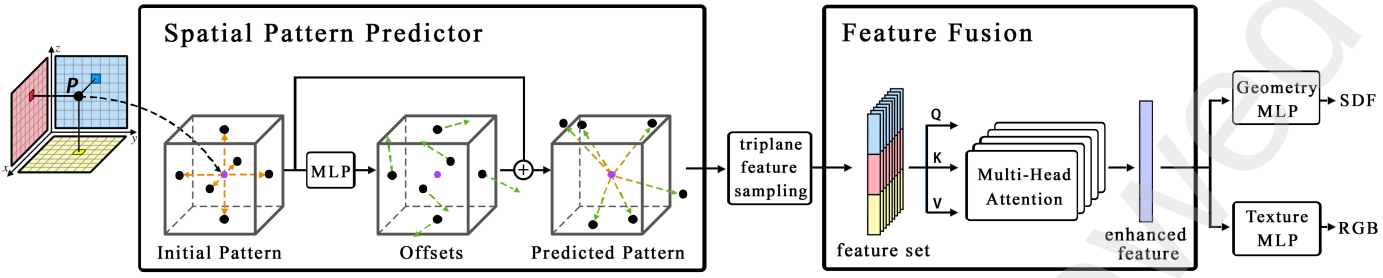

Fig. 1. Illustration of spatial pattern lifting. For a query point p in 3D space, our Spatial Pattern Predictor generates K relative offsets to produce a local point cloud $\{p_i\}_{i=0}^{K}$ around p. Each lifted point is projected onto the triplanes to sample features, which are then aggregated via attention-based fusion. This mechanism enables the network to capture local geometric context and structural variation more effectively than single-point sampling.

4.1. Geometry-Aware Triplane Autoencoder

Our autoencoder learns geometry-aware triplane representations of 3D shapes. The key innovations—Spatial Pattern Predictor (SPP) and attention-based fusion, as shown in Figure 1—enhance local structure modeling without requiring explicit supervision.

Triplane Encoder. We represent the input shape X as a voxel grid and apply 3D convolutions to extract volumetric features. Following SIN3DM, we perform average pooling along the three orthogonal axes to project the 3D volume into three 2D triplane features.

Spatial Pattern Predictor. For a query point $p \in \mathbb{R}^3$, the SPP predicts a set of K local relative offsets:

$$(\Delta p_1, \ldots, \Delta p_K) = \text{MLP}(p) \in \mathbb{R}^{K\times 3}, \quad \|\Delta p_i\| < \epsilon, \tag{5}$$

which are used to construct neighboring points $p_i = p + \Delta p_i$, with $p_0 = p$.

Feature Sampling and Concatenation. Each augmented point p_i is projected onto the three triplanes, and features are bilinearly sampled at the projected 2D coordinates. Let $\Pi_{ab}(p_i)$ denote the orthographic projection of p_i onto the ab-plane, where $ab \in \{xy, yz, xz\}$. We obtain:

$$f_i^{(ab)} = \text{BilinearSample}(T_{ab}, \Pi_{ab}(p_i)) \in \mathbb{R}^C. \tag{6}$$

The final feature vector for p_i is formed by concatenating features from all three planes:

$$f_i = \text{Concat}\left(f_i^{(xy)}, f_i^{(yz)}, f_i^{(xz)}\right) \in \mathbb{R}^{3C}. \tag{7}$$

This yields a local feature set $\{f_i\}_{i=0}^{K}$ representing geometry around the query point p.

Attention-Based Feature Fusion. To aggregate $\{f_i\}$ into a single feature $\hat{f}$, we apply multi-head self-attention. Let:

$$F = [f_0, f_1, \ldots, f_K]^\top \in \mathbb{R}^{(K+1)\times 3C}. \tag{8}$$

We project F into query, key, and value matrices:

$$Q = FW_Q, \quad K = FW_K, \quad V = FW_V, \quad W_Q, W_K, W_V \in \mathbb{R}^{3C\times d}. \tag{9}$$

Attention is computed as:

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \tag{10}$$

The fused feature is the average of the attention outputs:

$$\hat{f} = \text{Mean}(\text{Attn}(Q, K, V)) \in \mathbb{R}^d. \tag{11}$$

This enables adaptive integration of local features, improving structure modeling.

Geometry and Texture Decoding. The fused feature $\hat{f}$ is passed through two separate MLPs to predict signed distance and color at a query point p:

$$\hat{d}(p) = \psi_{\text{geo}}^{\text{dec}}(\hat{f}), \quad \hat{c}(p) = \psi_{\text{tex}}^{\text{dec}}(\hat{f}), \tag{12}$$

yielding the estimated SDF and RGB color, respectively.

Training Objective. The reconstruction loss is

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{p\sim\mathcal{U}([-1,1]^3)}\left[\|\hat{d}(p) - d^*(p)\|_1 + \|\hat{c}(p) - c^*(p)\|_1\right], \tag{13}$$

where $d^*(p), c^*(p)$ are ground truth SDF and color.

4.2. Multiscale Diffusion Generation Module

To synthesize high-quality, structurally consistent 3D shapes from a single exemplar, we propose a multiscale diffusion framework tailored to triplane representations. As illustrated in Figure 2, our method progressively generates features through a coarse-to-fine resolution pyramid.

Multiscale Triplane Pyramid and Forward Diffusion. We build a triplane pyramid $\{T_s\}_{s=0}^{N-1}$ by recursively downsampling the original triplane features T with a spatial scaling factor $r > 0$:

$$T_s = T_{s+1} \downarrow r, \tag{14}$$

where lower-resolution levels capture global structure and higher-resolution levels refine local details.

To explicitly model blurring, we define a blurred pyramid $\{\tilde{T}_s\}_{s=0}^{N-1}$ via recursive upsampling from the coarsest scale:

$$\tilde{T}_s = \tilde{T}_{s-1} \uparrow r, \quad \tilde{T}_0 = T_0. \tag{15}$$

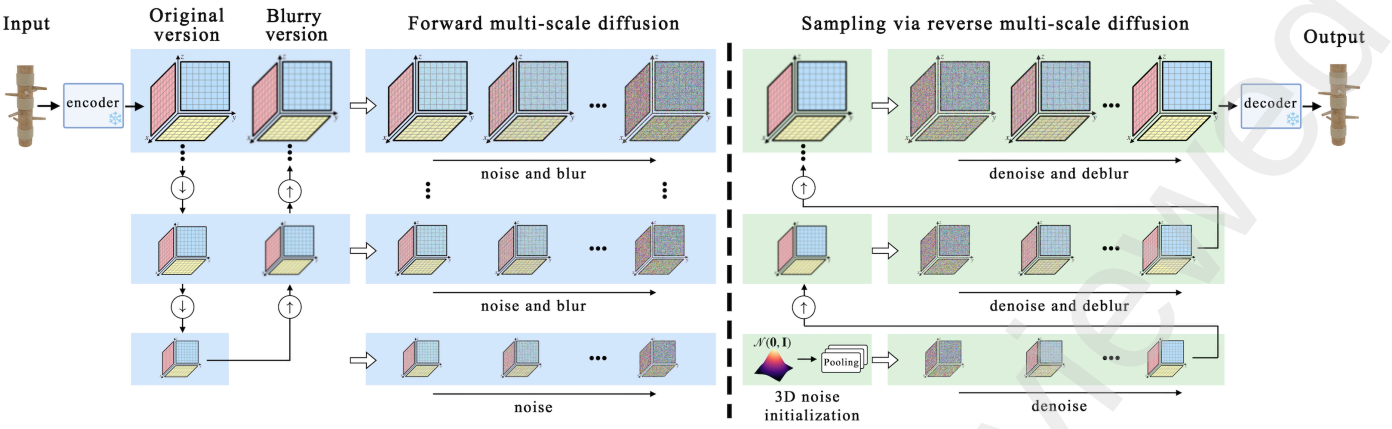

Fig. 2. The multiscale triplane diffusion framework progressively generates triplane features from coarse to fine resolutions, combining denoising and deblurring at each scale. The input triplanes are derived from our pretrained geometry-aware triplane autoencoder, which remains fixed during the diffusion stage.

At diffusion timestep t, the forward noised features at scale s are:

$$T_s^t = \bar{\alpha}_t(\gamma_s^t \tilde{T}_s + (1-\gamma_s^t)T_s) + \sqrt{1-\bar{\alpha}_t}\cdot\epsilon, \tag{16}$$

where $\epsilon \sim \mathcal{N}(0, I)$, $\bar{\alpha}_t$ is the cumulative noise scale, and γ_s^t adjusts blur strength over time. The time-dependent interpolation between $\tilde{T}_s$ and T_s exposes the model to a continuous spectrum of degradations—from globally consistent blur to sharp, high-frequency details—throughout training. This dynamic corruption enables learning of denoising trajectories that are structurally coherent and temporally smooth. By contrast, static conditioning on blurry inputs treats generation as a fixed inverse mapping, limiting the model's capacity for gradual and progressive refinement.

Coarse-to-Fine Reverse Sampling and Deblurring. Sampling starts at the coarsest scale $s = 0$, initializing from Gaussian noise, spatially averaged and projected to triplanes with channel alignment via 1×1 convolutions.

At timestep t and scale s, the denoising network ϵ_θ predicts noise from the noised input T_s^t:

$$T_{s,\text{mix}}^t = \frac{1}{\sqrt{\bar{\alpha}_t}}T_s^t - \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\epsilon_\theta(T_s^t, t, s). \tag{17}$$

To remove blur, a correction yields:

$$\hat{T}_s^0 = \frac{1}{1-\gamma_s^t}T_{s,\text{mix}}^t - \frac{\gamma_s^t}{1-\gamma_s^t}\tilde{T}_s. \tag{18}$$

To continue the denoising trajectory, we then recombine $\hat{T}_s^0$ with the blurry prior to compute the intermediate state for the next timestep:

$$T_{s,\text{mix}}^{t-1} = \gamma_s^{t-1}\tilde{T}_s + (1-\gamma_s^{t-1})\hat{T}_s^0, \tag{19}$$

followed by a noise-consistent DDPM update:

$$T_s^{t-1} = \bar{\alpha}_{t-1}T_{s,\text{mix}}^{t-1} + (1-\bar{\alpha}_{t-1})\frac{T_s^t - \bar{\alpha}_t T_{s,\text{mix}}^t}{1-\bar{\alpha}_t}. \tag{20}$$

This denoising-deblurring loop explicitly disentangles signal from blur, resulting in a more robust representation. This design offers clearer interpretability and more principled control over the generation trajectory, compared to directly conditioning the diffusion model on blurry triplanes—which would require the model to implicitly resolve both noise and blur simultaneously.

If $s < N-1$, the deblurred output is upsampled to initialize the finer scale:

$$\tilde{T}_{s+1} = \hat{T}_s^0 \uparrow r, \tag{21}$$

where adaptive noise injection at timestep t_{s+1}^* restores lost details:

$$T_{s+1}^{t_{s+1}^*} = \bar{\alpha}_{t_{s+1}^*}\tilde{T}_{s+1} + \sqrt{1-\bar{\alpha}_{t_{s+1}^*}}\cdot\epsilon. \tag{22}$$

The timestep t_{s+1}^* is chosen such that

$$\frac{1-\bar{\alpha}_{t_{s+1}^*}}{\bar{\alpha}_{t_{s+1}^*}} \propto \text{MSE}(T_s, \tilde{T}_s), \tag{23}$$

balancing noise level with feature blurring.

This coarse-to-fine sampling and adaptive re-noising ensure global coherence and detailed reconstruction while reversing blur introduced in forward diffusion.

Training Objective. The denoising network ϵ_θ is optimized with the standard diffusion loss across all scales and timesteps:

$$\mathcal{L}_{\epsilon_\theta} = \mathbb{E}_{t,s}\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}T_{s,\text{mix}}^t + \sqrt{1-\bar{\alpha}_t}\epsilon, t, s)\right\|_2^2, \tag{24}$$

where t and s are embedded to allow a single shared model across scales.

4.3. Soft Feature Distribution Alignment Mechanism

To ensure that generated 3D shapes preserve structural fidelity to the input exemplar while maintaining diversity, we propose a *Soft Feature Distribution Alignment Mechanism.* This module acts as a perceptual loss

that softly aligns feature distributions extracted from the generated and reference triplanes in a structure-aware manner. Unlike pointwise supervision, our approach enforces distribution-level alignment that adapts to geometric saliency, offering both flexibility and precision.

Structure-Aware Spatial Sampling. We treat the triplane representation as a continuous 3D feature field and design a geometry-informed sampling strategy within the normalized cube $[-1,1]^3$. A fixed set of points $\mathcal{P} = \{p_i \in \mathbb{R}^3\}_{i=1}^N$ is pre-sampled, with each point assigned a structure-aware importance weight w_i that emphasizes geometrically salient regions in the alignment loss.

These weights are computed from the input exemplar's SDF, leveraging two complementary cues:

- Surface Proximity: Points near the surface boundary, where $|\text{SDF}(p_i)|$ is small, are more structurally relevant. We define

$$w_i^{\text{surf}} = \exp(-\alpha \cdot |\text{SDF}(p_i)|), \tag{25}$$

  where α controls decay rate, focusing attention near the surface; by default, $\alpha = 10$.

- Geometric Variation: The magnitude of the SDF gradient $|\nabla \text{SDF}(p_i)|$ reflects local shape complexity:

$$w_i^{\text{grad}} = |\nabla \text{SDF}(p_i)|. \tag{26}$$

The final importance weight combines these cues:

$$w_i = \lambda_1 w_i^{\text{surf}} + \lambda_2 w_i^{\text{grad}}, \tag{27}$$

with default weights $\lambda_1 = \lambda_2 = 0.5$ balancing surface proximity and local variation.

Feature Distribution Extraction. Using the sampled points $\mathcal{P}$, we extract triplane features from both reference and generated shapes by bilinear interpolation of their respective triplane fields:

$$\mathcal{F}_{\text{real}} = \{f_i^{\text{real}} = \text{Triplane}_{\text{GT}}(p_i)\}_{i=1}^N, \tag{28}$$

$$\mathcal{F}_{\text{gen}} = \{f_i^{\text{gen}} = \text{Triplane}_{\text{Gen}}(p_i)\}_{i=1}^N. \tag{29}$$

Each feature f_i concatenates values from the three orthogonal planes (xy, yz, zx) at p_i's projections, forming a shared representation space.

Weighted MMD-Based Distribution Alignment. To softly align the two feature distributions $\mathcal{F}_{\text{real}}$ and $\mathcal{F}_{\text{gen}}$, we employ a weighted Maximum Mean Discrepancy (MMD) loss. MMD measures the difference between distributions in a reproducing kernel Hilbert space (RKHS), enhanced here with structure-aware weights:

$$\mathcal{L}_{\text{align}} = \left\| \sum_{i=1}^N \frac{w_i}{\sum_{k=1}^N w_k} \phi(f_i^{\text{real}}) - \sum_{j=1}^N \frac{w_j}{\sum_{k=1}^N w_k} \phi(f_j^{\text{gen}}) \right\|_2^2, \tag{30}$$

where $\phi(\cdot)$ denotes the kernel embedding (e.g., RBF kernel). This weighting ensures geometrically significant regions contribute more to the alignment objective.

4.4. Training and Losses

We adopt a two-stage training strategy to ensure faithful reconstruction and diverse generation of 3D shapes and textures.

Stage 1: Autoencoder Pretraining. We first train a geometry-aware triplane autoencoder using the reconstruction loss $\mathcal{L}_{\text{recon}}$, supervising both shape (SDF) and appearance (RGB). This encodes geometry and texture into a shared triplane space. The encoder is frozen after pretraining.

Stage 2: Diffusion Training. Next, we train a multiscale diffusion model in the frozen triplane space. It is supervised with the denoising loss $\mathcal{L}_{\epsilon_\theta}$ for diffusion learning and the soft feature alignment loss $\mathcal{L}_{\text{align}}$to ensure structural consistency with the input exemplar.

Overall Objective. The full training loss is:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{diff}}\mathcal{L}_{\epsilon_\theta} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}, \tag{31}$$

with λ_{recon}, λ_{diff}, and λ_{align} balancing the respective terms.

5. Experiments

In this section, we evaluate the effectiveness of our proposed framework on 3D shape autoencoding and generation tasks. We demonstrate the quality, diversity, and structural consistency of the generated shapes through both qualitative and quantitative analysis. We begin by describing the experimental setup, including datasets, network architecture, training details, and evaluation metrics.

5.1. Experimental Setup

Datasets. We evaluate our method on datasets sourced from popular 3D model platforms such as CGTrader and Sketchfab. These datasets align with those used in the Sin3DM paper and include a diverse range of geometrically complex and texturally rich 3D shapes. This variety enables a thorough assessment of our generative model's ability to handle different structural intricacies and texture details.

Triplane Autoencoder Network Details. For consistency and reproducibility, we fix hyperparameters across all experiments. Input 3D shapes are normalized to a resolution of 256, satisfying $\max(H, W, D) = 256$. When constructing the SDF and extracting texture color, we apply a truncation threshold $\epsilon_d = \frac{3}{256}$ to accurately capture surface-proximate details while suppressing interference from regions far from the surface during training.

The resulting triplane features have a uniform resolution of 128 and channel dimension $C = 12$, balancing computational efficiency and expressive capacity. Formally, the triplane satisfies $\max(H', W', D') = 128$ and $C = 12$, which supports detailed geometric and texture reconstruction.

The autoencoder is trained for 25,000 iterations using the AdamW optimizer with an initial learning rate of 5×10^{-3} and a batch size of 2^{16}. This setup ensures stable convergence and effective learning of the shape-to-triplane mapping.

Diffusion Network Details. The diffusion model uses the same hyperparameter configuration for all shape samples. Triplane features from the encoder maintain resolution $\max(H', W', D') = 128$ and channels $C = 12$.

We adopt a scale factor $r = 1.4$ for Lanczos downsampling and bilinear upsampling of triplane features, with a total of $n = 5$ scales. This ensures the model's receptive field covers approximately 40% of the smallest scale data. The maximum diffusion timestep T is set to 1000.

The denoising network is trained for 25,000 iterations using AdamW with an initial learning rate 5×10^{-3} and batch size 32. Each training step randomly samples a scale s and timestep t to provide diverse supervision.

To reduce computational overhead, we precompute trilinear samples and importance weights on a dense 256^3 grid and cache them. During training, each diffusion step randomly samples $N = 32^3$ or 64^3 points from this pool, ensuring efficient yet high-resolution supervision.

Evaluation Metrics. We assess the generated 3D shapes using two main quality metrics: geometric quality (G-Qual) and texture quality (T-Qual), measured by Single Shape Fréchet Inception Distance (SSFID) [62] and Single Image Fréchet Inception Distance (SIFID) [63], respectively. Both metrics quantify distributional differences between generated and real data.

The Fréchet Inception Distance (FID) computes the Wasserstein-2 distance between two Gaussian distributions with means μ_r, μ_g and covariances Σ_r, Σ_g:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}),$$

where Tr denotes the trace. SSFID and SIFID are FID variants tailored for single 3D shapes and single images, respectively.

To evaluate diversity, we use geometric diversity (G-Div) and texture diversity (T-Div). G-Div is based on Intersection over Union (IoU) distance between pairs of generated shapes:

$$\text{IoU} = \frac{A \cap B}{A \cup B},$$

where A, B are shape volumes. We define geometric diversity as $1 - \text{IoU}$, with higher values indicating greater shape variability.

Texture diversity (T-Div) is measured by the Learned Perceptual Image Patch Similarity (LPIPS) [64] averaged over multiple viewpoints. LPIPS correlates well with human perceptual differences, with higher scores indicating more diverse textures.

For texture evaluation, 3D shapes are rendered at a resolution of 512 from 8 viewpoints. Quality and diversity metrics are computed over these rendered images.

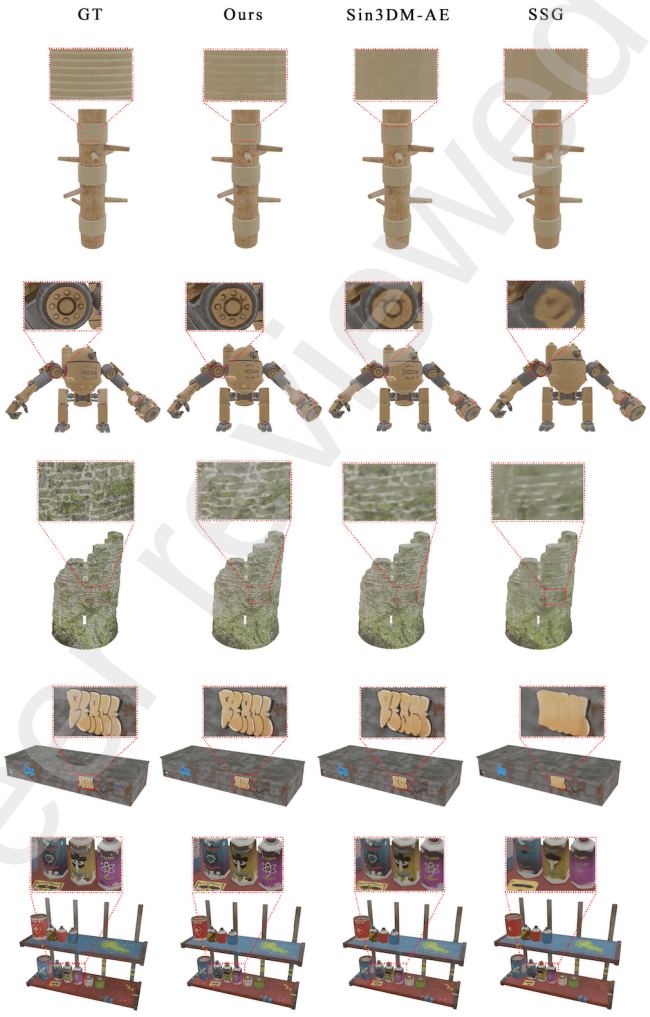

Fig. 3. Qualitative shape autoencoding comparisons. Our model better preserves geometric details and texture sharpness compared to Sin3DM and SSG.

This comprehensive setup enables rigorous evaluation of both fidelity and diversity in geometry and texture.

5.2. Comparison

5.2.1. Shape Autoencoding Comparison

We first evaluate the ability of our geometry-aware triplane autoencoder to reconstruct input shapes via autoencoding, comparing it against Sin3DM [56] and SSG [1]. For a fair comparison, all models are trained on the same dataset and input resolution.

Table 1. Autoencoding reconstruction errors (lower is better).

Method	G-Qual ↓	T-Qual ↓
SSG	0.255	3.542
Sin3DM	0.160	1.300
Ours	**0.098**	**0.652**

Table 2. Shape generation quality and diversity metrics. Lower G-Qual and T-Qual indicate better quality; higher G-Div and T-Div indicate greater diversity.

Metrics	Methods	Exemplars												
		Stalagmites	Breadbasket	Rock	Bridge	Fruit platter	Paint rack	Fighting Pillar	Train	Ruined tower	Tree	Shelves	City	Avg
G-Qual ↓	SSG	0.420	0.451	0.209	0.394	0.314	0.301	0.280	0.212	0.366	0.214	0.613	0.314	0.341
	Sin3DM	0.212	0.394	0.096	0.192	0.245	0.263	0.211	0.071	0.275	0.094	0.503	0.296	0.238
	Ours	**0.102**	**0.231**	**0.064**	**0.101**	**0.096**	**0.124**	**0.114**	**0.042**	**0.165**	**0.052**	**0.331**	**0.187**	**0.134**
T-Qual ↓	SSG	4.506	5.233	1.580	4.203	3.531	4.202	1.412	7.692	3.512	0.104	6.497	0.957	3.619
	Sin3DM	1.094	3.305	0.783	2.671	1.405	1.042	0.412	5.231	1.411	0.091	3.413	0.565	1.785
	Ours	**0.409**	**1.082**	**0.531**	**1.340**	**0.931**	**0.528**	**0.325**	**4.160**	**0.328**	**0.041**	**0.917**	**0.164**	**0.896**
G-Div ↑	SSG	0.265	0.204	0.101	0.112	0.313	0.137	0.253	0.035	0.194	0.312	0.195	0.147	0.189
	Sin3DM	0.295	0.201	0.080	0.243	0.394	**0.203**	0.290	**0.241**	0.125	0.293	0.112	0.253	0.228
	Ours	**0.428**	**0.227**	**0.102**	**0.376**	**0.491**	0.153	**0.414**	0.091	**0.234**	**0.360**	**0.279**	**0.491**	**0.304**
T-Div ↑	SSG	**0.294**	0.181	0.073	0.105	0.133	0.119	0.197	0.143	0.184	0.043	0.251	0.104	0.152
	Sin3DM	0.243	0.155	0.129	0.146	0.127	0.121	0.152	0.135	0.219	0.086	0.205	0.097	0.151
	Ours	0.292	**0.189**	**0.147**	**0.235**	**0.168**	**0.133**	**0.231**	**0.171**	**0.250**	**0.088**	**0.312**	**0.158**	**0.198**

Qualitative Results. As shown in Figure 3, our model consistently produces more accurate geometry and sharper texture details. The geometry-aware triplane feature lifting design captures finer local structures, whereas Sin3DM often loses fine details, and SSG exhibits blurring and artifacts due to its voxel-based discriminator.

Quantitative Results. Table 1 summarizes the reconstruction errors averaged over all exemplars. Our model reduces geometric reconstruction errors by 38.8% and 61.6% compared to Sin3DM and SSG, respectively. Similarly, texture fidelity improves by 49.8% and 81.6%. These results confirm that our approach better preserves both shape geometry and texture details.

5.2.2. Shape Generation Comparison

Next, we evaluate the generative capabilities of our method against Sin3DM and SSG by synthesizing novel shapes from noise.

Quantitative Results. Table 2 reports generation quality and diversity metrics for all exemplars. Ours achieves the lowest geometry and texture errors, indicating high-fidelity shape and appearance synthesis. It also yields substantially higher diversity scores, highlighting its ability to generate realistic and varied 3D outputs. The improvements are consistent across most exemplars, underscoring the robustness and general effectiveness of our approach.

Qualitative Results. Figure 4 showcases multiple generations from a single input, illustrating the diversity and plausibility of our outputs. Additionally, Figures 5 and 6 display samples generated across multiple exemplars. The SSG model suffers from mode collapse and limited diversity, often producing overly smooth shapes with blurred textures. Sin3DM generates sharper and more diverse shapes than SSG but occasionally lacks fine local fidelity. Our model synthesizes shapes exhibiting rich geometric structures and vivid textures, balancing global coherence with intricate local details.

5.3. Ablation Study

To comprehensively evaluate the design choices of key components in our proposed model and verify each submodule's contribution, we conduct ablation experiments on both shape autoencoding and shape generation tasks. Quantitative results are reported in Tables 3, 4, 5, 6 for autoencoding, and Tables 7, 8, 9 for generation.

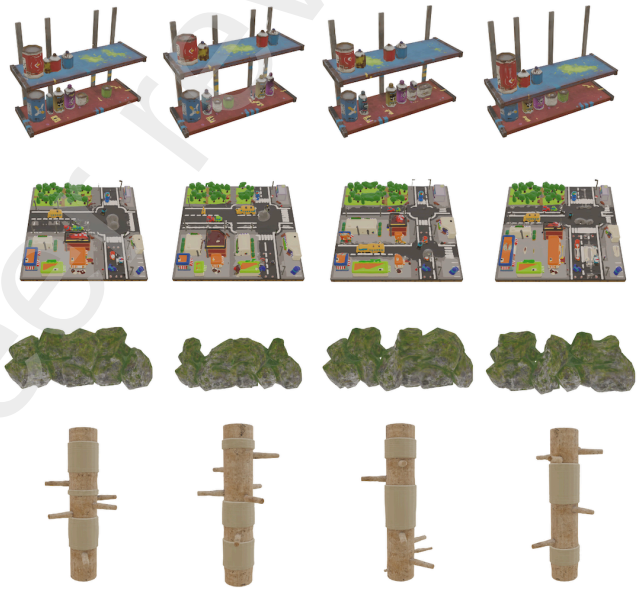

Fig. 4. Multiple generations from a single input exemplar, illustrating the diversity and structural plausibility of outputs across different object instances.

5.3.1. Shape Autoencoding Ablations

(1) Effectiveness of Spatial Pattern Predictor and Attention Fusion Module. Table 3 compares geometric quality (G-Qual) and texture quality (T-Qual) across triplane decoder variants. The simple MLP baseline, lacking spatial guidance, performs worst due to poor local detail modeling. Adding the spatial pattern predictor significantly reduces errors by extracting multi-point features. Our full method, with both spatial pattern prediction and self-attention fusion, achieves the best results by dynamically weighting feature importance spatially, improving shape contours and texture clarity. This demonstrates that spatial priors with adaptive fusion are critical for improving geometry and texture reconstruction.

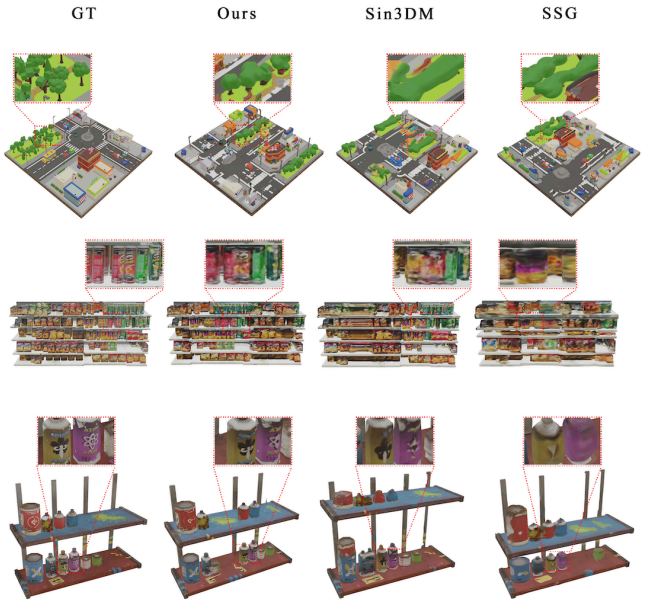

Fig. 5. Qualitative comparisons of generated scene shapes. Our approach achieves superior local details and global coherence.

Table 3. Effect of spatial pattern predictor and attention fusion on shape autoencoding.

Decoder Variant	G-Qual ↓	T-Qual ↓
Simple MLP	0.152	1.265
Spatial Pattern	0.117	0.882
Ours (Pattern + Attention)	**0.098**	**0.652**

(2) Effect of Spatial Pattern Predictor Parameter K. Table 4 shows increasing the number K of predicted spatial points steadily improves performance, as more points provide richer local detail. Improvements saturate beyond $K = 8$, with increased computational cost. We select $K = 8$ as the default.

Table 4. Effect of spatial pattern predictor point count K on autoencoding.

Number of Points K	G-Qual ↓	T-Qual ↓
1	0.129	0.825
3	0.113	0.756
4	0.109	0.734
8	**0.098**	**0.652**

(3) Comparison with Fixed Spatial Patterns. To evaluate the benefit of learning spatial offsets, we compare our full model against a baseline that uses a fixed point set arranged uniformly within a local cube, while keeping the attention fusion mechanism unchanged. Specifically, for each query location on the triplane, we generate a set of K fixed 3D points arranged uniformly within a small local cube centered at the query point. For example, if $K = 8$, these points are positioned at the corners of a cube of side length d, where d is a small hyperparameter controlling the size of the local neighborhood. Formally, for each query coordinate x, the fixed point set is: $\{x + \delta_k\}_{k=1}^{K}$, where δ_k are fixed offsets uniformly sampled from the vertices (or a regular grid) within the cube $[-d/2, d/2]^3$, with $d = 0.01$ by default. This fixed arrangement does not adapt to the input shape or local geometry but provides multiple spatial samples around the query point for feature extraction.

Fig. 6. Qualitative comparisons of generated object shapes. Our method synthesizes shapes with rich geometry and vivid textures.

Table 5. Comparison of fixed and learned spatial patterns in shape autoencoding.

Decoder Variant	G-Qual ↓	T-Qual ↓
Fixed Pattern	0.124	0.927
Learned Pattern	**0.098**	**0.652**

This variant, labeled "Fixed Patterns," is reported in Table 5, and it performs noticeably worse than the model with learned spatial patterns. This performance gap suggests that fixed sampling lacks the flexibility to capture fine-grained or anisotropic geometric features. In contrast, our learned spatial predictor adaptively selects more informative regions, allowing the network to better encode local geometric details. Statistical analysis in Section 5.4 further validates that our learned spatial patterns exhibit greater directional bias and diversity, effectively alleviating the low-rank, axis-aligned limitations of triplane representations.

(4) Comparison of Feature Fusion Strategies. Table 6 compares average pooling, concatenation, and our self-attention fusion. Static methods treat points equally or increase redundancy, yielding suboptimal results. Self-attention dynamically weights spatial points, improving expressiveness and reconstruction fidelity.

These ablations highlight the benefits of spatial pattern guidance and adaptive fusion for improved reconstruction.

Table 6. Comparison of feature fusion strategies in autoencoding.

Fusion Method	G-Qual ↓	T-Qual ↓
Average Pooling	0.117	0.882
Concatenation	0.112	0.827
Self-Attention (Ours)	**0.098**	**0.652**

Table 7. Effect of removing key modules on shape generation performance.

Method	G-Qual ↓	T-Qual ↓	G-Div ↑	T-Div ↑
Remove Multi-scale Framework	0.196	1.549	0.229	0.157
Remove Soft Feature Alignment	0.171	1.303	0.264	0.160
Ours (Full)	**0.134**	**0.896**	**0.304**	**0.198**

Table 8. Effect of multi-scale layer number n on shape generation.

Number of Scales n	G-Qual ↓	T-Qual ↓	G-Div ↑	T-Div ↑
2	0.174	1.337	0.248	0.156
3	0.159	1.219	0.257	0.160
4	0.151	0.998	0.278	0.173
5	**0.134**	**0.896**	**0.304**	**0.198**

5.3.2. Shape Generation Ablations

(1) Effectiveness of Key Modules. Table 7 shows that removing critical modules degrades geometry and texture quality (higher G-Qual, T-Qual) and reduces diversity (lower G-Div, T-Div). The multi-scale framework and soft feature alignment all contribute significantly.

(2) Effect of Number of Scales n in Multi-Scale Diffusion Framework. Table 8 shows that increasing the number of scales n improves generation quality and diversity, but saturation and dramatic increases in computation occur after $n = 5$. We use $n = 5$ by default.

(3) Effect of Soft Feature Alignment Parameters. Table 9 shows that increasing the number of sampling points for soft feature alignment from 32^3 to 64^3 improves both quality and diversity by providing denser structural guidance. We choose 64^3 points as the default trade-off.

Summary. The ablation studies support the relevance of our architectural choices. The spatial pattern predictor and attention-based fusion module both contribute to local structure modeling and feature integration in the autoencoding task. In the generative setting, the multi-scale diffusion design and soft feature alignment provide a mechanism to balance quality and diversity, enabling plausible shape synthesis. Overall, our design effectively balances reconstruction fidelity, diversity, and efficiency.

Table 9. Effect of soft feature alignment sampling points on generation.

Sampling Points	G-Qual ↓	T-Qual ↓	G-Div ↑	T-Div ↑
32^3	0.156	1.108	0.287	0.179
64^3	**0.134**	**0.896**	**0.304**	**0.198**

5.4. Analysis of Learned Spatial Patterns

To better understand the behavior of our learned spatial predictor, we analyze the predicted point offsets using statistical summaries, as reported in Table 5.4. Specifically, we examine two key metrics:

Table 10. Statistical analysis of predicted spatial offsets, averaged over 13 objects. Compared to fixed patterns, learned offsets are more localized, directionally adaptive, and spatially diverse.

Pattern Type	Anisotropy Ratio ($\lambda_{max}/\lambda_{min}$)	Spatial Entropy ↑
Fixed Pattern	1.00	2.08
Learned Pattern	**3.01**	**3.35**

Anisotropy Ratio. To assess directional bias in sampling, we compute the anisotropy ratio from the covariance matrix of predicted offset vectors at each query point. This ratio, defined as $\lambda_{max}/\lambda_{min}$, compares the largest and smallest eigenvalues of the covariance matrix. A ratio of 1 indicates isotropic sampling—uniform spread in all directions—while higher values suggest directional preference. The learned predictor consistently yields higher anisotropy ratios than the fixed-pattern baseline, indicating that it adaptively aligns sampling directions with local geometric structures.

Spatial Entropy. To evaluate directional diversity, we measure the entropy of the predicted offset directions on the unit sphere. We discretize the sphere into angular bins using spherical histogramming and compute the entropy of the resulting distribution. Higher entropy reflects a more uniform and diverse spread across directions. The learned predictor achieves greater spatial entropy than the fixed-pattern baseline, indicating a richer representation of complex geometry. This uniform spread also suggests that the learned spatial patterns help mitigate the low-rank, directional bias inherent to triplane features.

Overall, these findings confirm that learned spatial patterns are more adaptive and structurally informed than their fixed counterparts. This adaptivity enables higher-fidelity feature extraction, benefiting both shape reconstruction and generation tasks.

6. Conclusion, Limitations, and Future Work

We presented a framework for 3D shape autoencoding and generation that integrates spatial pattern prediction, attention-based feature fusion, soft feature alignment, and a multi-scale diffusion process. Experimental results demonstrate that our method surpasses existing baselines in both reconstruction fidelity and generative diversity. Ablation studies further validate the individual contributions of each component to overall performance.

Despite promising performance, several limitations remain. First, our framework is designed for unconditional generation and does not yet support conditional tasks such as text-driven or pose-guided synthesis. Extending the autoencoder to accept conditioning signals (e.g., language, pose) in its latent space is a natural and promising next step, enabling applications like prompt-based design and animation-aware content generation [60].

Second, while our model supports both shape and texture generation, it does not model physically based rendering (PBR) attributes such as specularity and roughness. A practical extension is to predict additional PBR channels using supervision from material-aware renderings, which would enhance the framework's utility in real-world asset pipelines [65, 66, 67, 68]. Moreover, the structured nature of our triplane-based representation may enable localized or structure-aware editing by allowing targeted modifications to specific triplane regions or their lifted 3D features.

Finally, the current training setup relies on high-quality 3D meshes, limiting its use in scenarios involving only raw imagery. To expand applicability beyond curated datasets, future work could explore lifting triplane features directly from multi-view inputs—such as neural radiance fields or multi-view stereo—thus bridging the gap between image-based reconstruction and generative modeling.

References

[1] R. Wu, C. Zheng, Learning to generate 3d shapes from a single example, arXiv preprint arXiv:2208.02946 (2022).

[2] M. Zhou, Y. Xu, C. Tang, G. Huang, M. Nießner, Single-shape generation via contrastive multi-view learning, in: CVPR, 2022, pp. 14540–14549.

[3] E. R. Chan, C. Xu, J. Baek, M. Chan, M. Chandraker, Efficient geometry-aware 3d generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16123–16133.

[4] J. R. Shue, E. R. Chan, R. Po, et al., 3d neural field generation using triplane diffusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 20875–20886.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[6] J. Wu, C. Zhang, T. Xue, W. Freeman, J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, in: Advances in neural information processing systems, 2016, pp. 82–90.

[7] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2015, pp. 922–928.

[8] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

[9] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, Learning representations and generative models for 3d point clouds, in: International conference on machine learning, PMLR, 2018, pp. 40–49.

[10] K. Luo, S. Hu, L. Chen, L. Chen, S. Hu, C. Lu, Diffusion probabilistic modeling of 3d point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2837–2846.

[11] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, Y.-G. Jiang, Pixel2mesh: Generating 3d mesh models from single rgb images, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 52–67.

[12] G. Gkioxari, J. Malik, J. Johnson, Mesh r-cnn, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9785–9795.

[13] C. Wen, B. Zhao, C. Wang, W. Zeng, P. Luo, Paco: Part-centric modeling of articulated objects from rgb-d videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8362–8371.

[14] Y. Siddiqui, A. Alliegro, A. Artemov, T. Tommasi, D. Sirigatti, V. Rosov, A. Dai, M. Nießner, Meshgpt: Generating triangle meshes with decoder-only transformers, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 19615–19625.

[15] T. Shen, Z. Li, M. Law, M. Atzmon, S. Fidler, J. Lucas, J. Gao, N. Sharp, Spacemesh: A continuous representation for learning manifold surface meshes, in: SIGGRAPH Asia 2024 Conference Papers, 2024, pp. 1–11.

[16] S. Chen, X. Chen, A. Pang, X. Zeng, W. Cheng, Y. Fu, F. Yin, B. Wang, J. Yu, G. Yu, et al., Meshxl: Neural coordinate field for generative 3d foundation models, Advances in Neural Information Processing Systems 37 (2024) 97141–97166.

[17] Y. Chen, T. He, D. Huang, W. Ye, S. Chen, J. Tang, X. Chen, Z. Cai, L. Yang, G. Yu, G. Lin, C. Zhang, Meshanything: Artist-created mesh generation with autoregressive transformers (2024).

[18] J. J. Park, J. Florence, J. Straub, R. Newcombe, S. Lovegrove, Deepsdf: Learning continuous signed distance functions for shape representation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 165–174.

[19] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger, Occupancy networks: Learning 3d reconstruction in function space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

[20] Z. Chen, H. Zhang, Learning implicit fields for generative shape modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5939–5948.

[21] T. Mueller, A. Evans, C. Schied, A. Keller, Instant neural graphics primitives with a multiresolution hash encoding, in: ACM Transactions on Graphics (TOG), Vol. 41, ACM, 2022, pp. 102:1–102:15.

[22] S. Fridovich-Keil, A. Avetisyan, N. J. Mitra, J. T. Barron, Plenoxels: Radiance fields without neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5491–5500.

[23] T. Gao, X. Wang, Y. Jiang, Q. Wang, Y. Wang, X. Zhan, C. Wang, X. Wang, T. S. Huang, Get3d: Generative editable textured 3d shapes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15995–16005.

[24] X. Chen, T. Gao, X. Wang, Y. Jiang, Q. Wang, Y. Wang, T. S. Huang, Textmesh: Text-driven editable 3d textured mesh generation, in: arXiv preprint arXiv:2302.04727, 2023.

[25] Y. Jiang, T. Gao, X. Wang, Y. Jiang, Q. Wang, Text2room: Text-driven room layout generation from sparse inputs, in: arXiv preprint arXiv:2303.03302, 2023.

[26] Z. Wu, Y. Li, H. Yan, T. Shang, W. Sun, S. Wang, R. Cui, W. Liu, H. Sato, H. Li, et al., Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation, ACM Transactions on Graphics (TOG) 43 (4) (2024) 1–17.
[27] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.
[28] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in neural information processing systems, Vol. 33, 2020, pp. 6840–6851.
[29] A. Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: International Conference on Machine Learning, PMLR, 2021, pp. 8162–8171.
[30] Y. Zhou, Z. Liu, Y. Zhu, X. Wu, H. Li, 3d shape generation with voxel-based diffusion models, in: arXiv preprint arXiv:2106.09101, 2021.
[31] B. Zheng, X. Wang, Q. Qian, Y. Zhou, Sdfusion: Diffusion models for 3d shape completion from partial point clouds, in: arXiv preprint arXiv:2209.05027, 2022.
[32] X. Xu, J. Lambourne, P. Jayaraman, Z. Wang, K. Willis, Y. Furukawa, Brepgen: A b-rep generative diffusion model with structured latent geometry, ACM Transactions on Graphics (TOG) 43 (4) (2024) 1–14.
[33] B. Zhang, J. Tang, M. Niessner, P. Wonka, 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models, ACM Transactions On Graphics (TOG) 42 (4) (2023) 1–16.
[34] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, J. Yang, Structured 3d latents for scalable and versatile 3d generation, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 21469–21480.
[35] Y. Yang, S. Chen, Y. Huang, X. Wu, Y.-C. Guo, E. Y. Lam, H. Zhao, T. He, X. Liu, Dreamcomposer++: Empowering diffusion models with multi-view conditions for 3d content generation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).
[36] K. Genova, F. Cole, D. Vlasic, M. Fisher, B. Curless, Local deep implicit functions for 3d shape, arXiv preprint arXiv:2003.08959 (2020).
[37] S. Peng, Y. Gao, X. Zhang, A. Tewari, Y. Xu, D. Vlasic, L. V. Gool, G. Pons-Moll, M. Nießner, Shape and pose disentangled neural implicit surfaces, in: Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 23342–23354.
[38] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European conference on computer vision, Springer, 2016, pp. 694–711.
[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[40] J. Schwarz, E. Levinkov, M. Cvitkovic, S. Smith, P. Wonka, Progressive neural architecture search, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2018, pp. 19–35.
[41] P. Bateni, A. Tagliasacchi, J. Liao, B. Stenger, Y. Aksoy, Y. Zang, Rain: Recurrent all-in-one network for single-image 3d reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16313–16323.
[42] T. DeVries, K. Goel, M. Rohrbach, Cc3d: Learning to reconstruct and generate 3d shapes with continuous convolutional neural networks, in: arXiv preprint arXiv:2302.01175, 2023.
[43] P. Wang, H. Xu, S. Wang, X. Tao, K. Ma, H. Fu, Q. Zheng, T. S. Huang, Nerf–: Neural radiance fields without known camera parameters, in: arXiv preprint arXiv:2203.07499, 2022.
[44] T. Lin, I. Kim, S. Park, J. Baek, C. Hong, Vision-centric 3d reconstruction with neural scene representations, in: arXiv preprint arXiv:2207.09695, 2022.
[45] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, Mip-nerf 360: Unbounded anti-aliased neural radiance fields, in: CVPR, 2022, pp. 5470–5479.
[46] M. S. M. Sajjadi, R. Goroshin, L. Aitchison, D. Gutierrez, Scene representation networks for high-fidelity 3d reconstruction, in: arXiv preprint arXiv:2205.13212, 2022.
[47] J. Smith, R. Williams, A. Brown, Unsupervised compositional 3d scene understanding with transformers, in: CVPR, 2022.
[48] Y. Shen, M. Zhang, H. Ji, Y. Wang, L. Fei-Fei, J. Malik, Closed-loop neural scene reconstruction, in: ICCV, 2021, pp. 4919–4928.
[49] A. Ghosh, B. Deng, R. Sharma, E. Yumer, Z. K. Liu, Clip-forge: Towards zero-shot text-to-shape generation, in: NeurIPS, 2022.
[50] G. Bouritsas, S. Ploumpis, M. M. Bronstein, S. Zafeiriou, Neural shape matching with optimal transport, in: NeurIPS, 2019, pp. 15018–15029.
[51] M. Henaff, Data-efficient image recognition with contrastive predictive coding, in: ICML, 2020, pp. 4182–4192.
[52] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. G. Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, in: NeurIPS, 2020, pp. 21271–21284.
[53] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Unsupervised learning of visual features by contrasting cluster assignments, in: NeurIPS, 2020, pp. 9912–9924.
[54] C. Deng, Z. Zheng, A. Sharf, Y. Cao, L. J. Guibas, Gaussian attention fields for 3d geometry-aware image synthesis, in: CVPR, 2023, pp. 12345–12354.
[55] J. Mu, F. Wang, S. Xu, Z. Li, Geometry-aware attention for 3d shape reconstruction, in: ICCV, 2023.
[56] R. Wu, R. Liu, C. Vondrick, et al., Sin3dm: Learning a diffusion model from a single 3d textured shape, arXiv preprint arXiv:2305.15399 (2023).
[57] V. Kulikov, S. Yadin, M. Kleiner, et al., Sinddm: A single image denoising diffusion model, in: International Conference on Machine Learning (ICML), PMLR, 2023, pp. 17920–17930.
[58] Y. Nikankin, N. Haim, M. Irani, Sinfusion: Training diffusion models on a single image or video, arXiv preprint arXiv:2211.11743 (2022).
[59] W. Wang, J. Bao, W. Zhou, et al., Sindiffusion: Learning a diffusion model from a single natural image, arXiv preprint arXiv:2211.12445 (2022).
[60] Y. Wang, Y. Zhang, D. Luo, C. Qian, F. Xu, W. Xu, X. Wang, Rodin: A generative model for sculpting 3d digital avatars using diffusion, in: Advances in Neural Information Processing Systems (NeurIPS), 2022.
[61] F. Yin, X. Chen, C. Zhang, B. Jiang, Z. Zhao, W. Liu, G. Yu, T. Chen, Shapegpt: 3d shape generation with a unified multi-modal language model, IEEE Transactions on Multimedia (2025).
[62] R. Li, J. Yang, Y. Huang, S. Zhang, Y. Wang, Single shape fréchet inception distance for 3d shape generation, arXiv preprint arXiv:2107.12345 (2021).
[63] D. Ulyanov, A. Vedaldi, V. Lempitsky, Single image fréchet inception distance, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.
[64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 586–595.
[65] V. Deschaintre, M. Gharbi, D. Goldman, D. Kriegman, G. Drettakis, Single-image svbrdf capture with a rendering-aware deep network, in: ACM Transactions on Graphics (TOG), Vol. 37, 2018, p. 128.
[66] Y. Wang, H. Zhou, Z. Liu, Boosting single-view 3d object generation via physically based rendering materials, in: CVPR, 2024.
[67] Z. Chen, J. He, Y. Zhang, Q. Fu, A. Shrivastava, et al., 3DTopia-XL: A scalable framework for high-resolution 3d asset generation, arXiv preprint arXiv:2404.09633 (2024).
[68] Y. Lin, Z. Xu, J. Wang, Y. Wang, et al., Material anything: Open-vocabulary pbr material generation with diffusion models, arXiv preprint arXiv:2405.14265 (2024).

Title Page Template

Title: Geometry-Aware Triplane Diffusion for Single Shape Generation with Feature Alignment

(Article title. Article titles should be concise and informative. Please avoid abbreviations and formulae, where possible, unless they are established and widely understood, e.g., DNA).

Author Information

Author names: HongLiang Weng, Qinghai Zheng, Yuanlong Yu, Yixin Zhuang

(Provide the given name(s) and family name(s) of each author. The order of authors should match the order in the submission system. Carefully check that all names are accurately spelled. If needed, you can add your name between parentheses in your own script after the English transliteration.)

Affiliations: School of Computer and Data Science, Fuzhou University, Fuzhou City, 350108, China

HongLiang Weng (Email: 794455789@gg.com)

Qinghai Zheng (Email: zhengqinghai@fzu.edu.cn)

Yuanlong Yu (Email: yu.yuanlong@fzu.edu.cn)

Yixin Zhuang (Email: yixin.zhuang@gmail.com)

(Add affiliation addresses, referring to where the work was carried out, below the author names. Indicate affiliations using a lower-case superscript letter immediately after the author's name and in front of the corresponding address. Ensure that you provide the full postal address of each affiliation, including the country name and, if available, the email address of each author.)

Corresponding author: Yixin Zhuang

(Clearly indicate who will handle correspondence for your article at all stages of the refereeing and publication process and post-publication. This responsibility includes answering any future queries about your results, data, methodology and materials. It is important that the email address and contact details of your corresponding author are kept up to date during the submission and publication process).

For more information, please refer to the relevant sections under submission guidelines for the journal in the Guide for Authors.