Efficient and Scalable Point Cloud Generation with Sparse Point-Voxel Diffusion Models

Ioannis Romanelis, Vlassios Fotis, Athanasios Kalogeras, *Senior Member, IEEE*, Christos Alexakos, *Member, IEEE* , Konstantinos Moustakas, *Senior Member, IEEE*, Adrian Munteanu, *Member, IEEE*

***Abstract*—We propose a novel point cloud U-Net diffusion architecture for 3D generative modeling capable of generating high-quality and diverse 3D shapes while maintaining fast generation times. Our network employs a dual-branch architecture, combining the high-resolution representations of points with the computational efficiency of sparse voxels. Our fastest variant outperforms all non-diffusion generative approaches on unconditional shape generation, the most popular benchmark for evaluating point cloud generative models, while our largest model achieves state-of-the-art results among diffusion methods, with a runtime approximately 70% of the previously state-of-the-art PVD. Beyond unconditional generation, we perform extensive evaluations, including conditional generation on all categories of ShapeNet, demonstrating the scalability of our model to larger datasets, and implicit generation which allows our network to produce high quality point clouds on fewer timesteps, further decreasing the generation time. Finally, we evaluate the architecture's performance in point cloud completion and super-resolution. Our model excels in all tasks, establishing it as a state-of-the-art diffusion U-Net for point cloud generative modeling. The code is publicly available at https://github.com/JohnRomanelis/SPVD.git.**

***Index Terms*—Generative Modeling, Deep Learning, Point Clouds, Generation, Completion, Super-Resolution, Diffusion.**

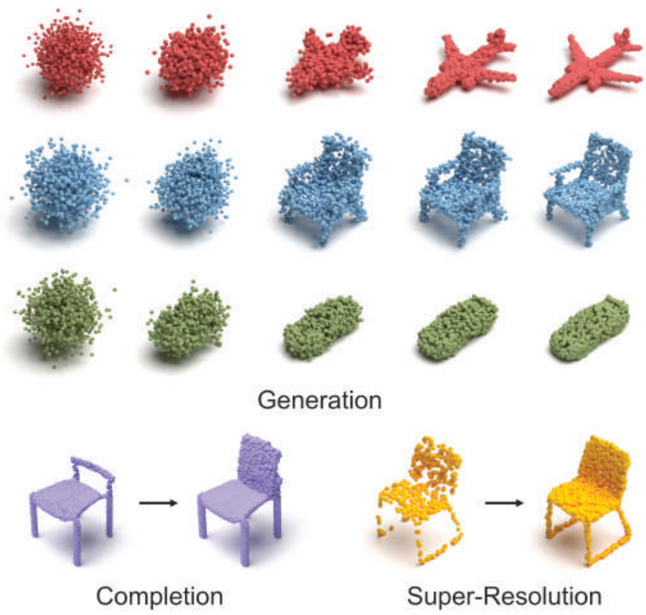

Fig. 1. The proposed Sparse Point-Voxel Diffusion (SPVD) is a novel diffusion architecture designed for efficient and scalable point cloud generation tasks. The generation process visualizes the gradual transformation of a noisy sample into a clean 3D shape. The completion and super-resolution tasks further demonstrate the capabilities of the proposed architecture.

I. Introduction

Generative models have emerged as powerful tools in the realm of artificial intelligence, offering significant advances in the automated generation of digital content across various modalities, such as text [1], music [2]–[4], image [5]–[7] and video [8]. Recently, the focus has extended to 3D generative models which hold promise for a wide array of applications like computer vision, computer graphics, and robotics. Among 3D data representations, point clouds are becoming increasingly common. This is primarily because they are the direct output of 3D sensors, increasing the availability of data, and can express a higher level of detail compared to other representations such as voxel grids. Additionally, a plethora of algorithms have been developed to transform point clouds into more sophisticated representations, such as 3D meshes [9]–[14].

Several methods have been proposed for point cloud generation, employing various architectures and pipelines, such as VAEs [15], [16], GANs [17]–[19], Gradient Fields [20], Normalizing Flows [21]–[23], and Diffusion models [24]–[26]. Diffusion models [5], recognized as the State-of-the-Art generative approach in visual computing (including image, video), are gaining increasing popularity in the Point Cloud domain due to their ability to produce high-fidelity and diverse shapes. These models operate by progressively denoising a noise sample from a Gaussian distribution to generate a clean novel shape. However, thousands of steps are required for this generation process [27], equaling to thousands of network activations.

Moreover, current state-of-the-art point-based Point Cloud models are inherently slow, with up to 90% of their runtime dedicated to structuring irregular data rather than actual feature extraction [28]. This structuring includes sampling and neighbor searching operations necessary for tasks like downsampling the point cloud, forming point neighborhoods, and interpolating point features. Unlike image pixels, which are organized in a fixed grid making these operations straightforward, point clouds require more complex algorithms such as furthest point sampling and k-nearest neighbor searches in

Ioannis Romanelis, Vlassis Fotis, and Konstantinos Moustakas are with the Department of Electrical and Computer Engineering, University of Patras. Ioannis Romanelis, Vlassis Fotis, Athanasios Kalogeras, and Christos Alexakos are with the Industrial Systems Institute (ISI) - Athena Research Center. Ioannis Romanelis and Adrian Munteanu are with the Department of Electronics and Informatics, Vrije Universiteit Brussel

continuous space. While voxel-based methods could potentially overcome these speed limitations, they suffer from significant information loss due to the aggressive downsampling needed to manage the cubical memory demands of voxel grids [29]. Additionally, voxel-based methods have been shown to produce poor generation results [24].

We propose SPVD, a novel UNet diffusion architecture that achieves state-of-the-art generation results while significantly reducing generation runtime compared to other diffusion methods. Our model, inspired by [29], combines a sparse voxel backbone designed to efficiently extract neighboring information with a high-fidelity point branch that preserves the fine details of the points. To minimize hard-to-interpret design choices, our sparse voxel backbone follows the DDPM [5] UNet architecture, incorporating only domain-specific adaptations. Additionally, to further decrease the generation time, we modify the voxelization pipeline from [29] to run entirely on the GPU. To put the numbers into perspective, PVD [24], the current state-of-the-art, requires more than 1 hour to generate 662 samples, which is the size of the ShapeNet - Chair category test set. In contrast, our model's fastest variant completes this task in less than 15 minutes, while the largest variant does not exceed 45 minutes[1]. Moreover, all versions of our network have been trained on a 24GB VRAM GPU, making it accessible for retraining and use by the academic community without the need for expensive and often unavailable equipment.

We quantitatively evaluate our model, following the paradigm of previous works [17], [20]–[25], [27], by measuring the 1-NN metric for the unconditional generation results in the Car, Airplane and Chair categories of ShapeNet. To test the model's scalability with increasing amounts of data, we train a variant for conditional generation on all categories of ShapeNet. To further decrease the generation time we study the DDIM generation rule [6], which allows generation with fewer iteration, and its effect on the generation quality. Finally, we qualitatively test our model in other candidate tasks, such as completion and super-resolution.

To summarize our contributions are the following:

- We propose SPVD, a novel diffusion U-Net architecture that combines the point representations with sparse voxels for efficient point cloud processing.
- We achieve state-of-the-art results in the most common generative benchmark - unconditional generation on Airplane, Chair, Car categories of ShapeNet - while reducing generation time compared to the previous state-of-the-art diffusion model.
- We present extensive quantitative and qualitative results to demonstrate our model's ability to scale to larger datasets, generate shapes faster through implicit generation, and perform additional generative tasks such as shape completion and super-resolution.

II. Related Work

A. Deep Learning for Point Clouds

Designing an effective 3D generative model requires selecting a backbone architecture that balances execution speed with accuracy. PointNet [30] utilizes shared-MLPs across points to project them into higher dimensions and extract global features through pooling; however, it lacks on descriptive power due to the absence of neighbor feature aggregation. PointNet++ [31] addresses this by applying PointNets withing small point neighborhoods to extract local features. Subsequent studies [32]–[37] explore different kernels for neighborhood feature aggregation. While these methods achieve state-of-the-art results in classification and segmentation tasks, their execution time is constrained by the time required for point sampling and grouping operations.

Transformer-based models [38]–[43] create point patches and process them using transformer blocks [44]. Although these models reduce the number of point operations, computing the attention matrices remains a time-intensive process.

Point-Voxel CNNs [28] introduce a hybrid convolution approach, combining a high-resolution point branch with a low-resolution voxel branch to aggregate neighborhood information. In [28] the authors propose two network architectures. The first, PVCNN, employs a series of Point-Voxel layers to extract point features. The second variant, PVCNN++, inspired by PointNet++, also utilizes point operations. After each voxel convolution, points are sampled and grouped into neighborhoods, and collective features are extracted for the neighborhood centroids, creating a point encoder with a decreasing number of points. While this approach enhances the network's ability to understand 3D geometry, it introduces the aforementioned runtime bottlenecks due to the point operations.

Sparse Voxel Convolution models [45]–[47] address the cubic memory requirements of dense voxel grids, thereby enabling higher voxel densities while preserving fast execution times. In SPVNAS [29] the authors propose the use of Sparse Point-Voxel Convolutions, creating an effective module that combines the efficiency of the sparse voxel convolution and the high fidelity features that are propagated through the point branch.

B. 3D Shape Generation

3D shape generation involves creating synthetic models, most commonly represented as Point Clouds or Meshes. Early works primarily utilized Autoencoder architectures [15], [16], Generative Adversarial Networks (GANs) [17]–[19] and Gradient Fields [48]. PointFlow [21] proposed a probabilistic framework, utilizing continuous normalizing flows, to learn a two-level hierarchy of distributions: a distribution of shapes and of points given a shape. Subsequent flow-based research includes [22], [23].

[25] introduces a diffusion-based pipeline where points are projected into a latent space representation using a PointNet-like network, followed by a neural network that gradually denoises the latent space representations, resulting in novel shapes. In PVD [24] the authors study a diffusion network

[1] All time measurements were performed on the same machine using a NVIDIA RTX 3090 GPU.

that operates directly on the point space. The diffusion U-Net is based on the architecture of PVCNN++ [28]. PVD is a pivotal work, as it has enabled the creation of more complex generative pipelines. In [26] the authors design a latent space diffusion pipeline, where the original point cloud is projected to a latent space and a PVD U-Net is responsible for point cloud denoising. In [27] the authors retrain a PVD U-Net by optimizing the curvy trajectory of the diffusion denoising process into a straight path. Furthermore, they propose a distillation process to shorten this straigh path into one step, enabling a fast generation pipeline.

However, the point operations in the PVD U-Net limit both the runtime of the network and its scalability. Network hyperparameters, such as the size of the point neighborhoods, are highly affected by the total number of points. Consequently, a network architecture cannot be applied to higher density point clouds without significant architectural changes. The same issue applies to attention operations in the point space, as the size of the attention matrix is influenced by the number of points.

Recent advancements in 3D shape generation also include the generation of 3D meshes, rather than point clouds, utilizing mesh diffusion techniques [49], [50] or improving the quality of generated shapes through refining modules [51].

III. Sparse Point-Voxel Diffusion

A. Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) are a class of generative models inspired by thermodynamics [52], where novel shapes are generated by progressively denoising samples originating from a Gaussian distribution. The process is illustrated in Figure 2.

During the forward diffusion process, a sample from the original data distribution, denoted as $\mathbf{x}_0$, is progressively corrupted by adding Gaussian noise according to a predefined variance schedule $\beta_1, \ldots, \beta_T$. This results in a series of progressively noisier samples $\mathbf{x}_1, ..., \mathbf{x}_T$, of the same dimensionality as $\mathbf{x}_0$. This process can be formulated by stating that the approximate posterior $q(\mathbf{x}_{0:T})$ is defined as a Markov chain.

$$\begin{gathered} q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\ \mathbf{x}_{t-1}, \beta_t\ \mathbf{I}) \end{gathered} \tag{1}$$

Furthermore, by setting $\alpha_t = 1 - \beta_t$, $\overline{\alpha}_t = \prod_{s=0}^{t} \alpha_s$ and applying reparameterization [53] on equation 1, a closed-form equation is derived, allowing for direct computation of the sample at timestep t:

$$\begin{gathered} \mathbf{x}_t \sim \mathcal{N}(\sqrt{\overline{\alpha}_t}\ \mathbf{x}_0, (1-\overline{\alpha}_t)\ \mathbf{I}) \\ \mathbf{x}_t = \sqrt{\overline{\alpha}_t}\ \mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\ \epsilon, \quad \epsilon \sim \mathcal{N}(0,1) \end{gathered} \tag{2}$$

The reverse diffusion, which is the actual generation phase, involves a neural network predicting the inverse noise distribution, enabling a gradual transition to a clean sample that belongs to the original data distribution. This process is also formulated as a Markov chain with learned Gaussian transitions.

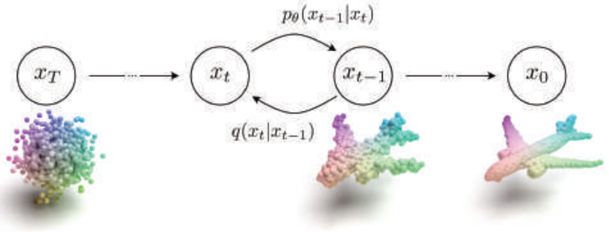

Fig. 2. Illustration of the forward and reverse diffusion processes in DDPM. Initially, a clean shape is progressively noisified through the forward diffusion process, resulting in increasingly noisy samples up to $\mathbf{x}_T$. These samples, generated via a predefined noise schedule, are utilized during the training phase. The reverse process, indicated by the arrows, involves a neural network tasked with estimating the inverse noise distribution to progressively denoise the samples, eventually reconstructing a clean shape $\mathbf{x}_0$.

$$\begin{gathered} p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), {\sigma_t}^2\ \mathbf{I}) \end{gathered} \tag{3}$$

where, $\mu_\theta(\mathbf{x}_t, t)$ denotes the shape predicted by the generative model at timestep $t-1$, ${\sigma_t}^2$ is the variance at timestep t; the specifics of how this variance is determined are discussed later in this section.

Training Objective: The training objective is to optimize the variational lower bound on the negative log likelihood of the data. This can be expressed as:

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right)\right] \tag{4}$$

where $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is the true posterior, for each timestep $1, ..., T$ and $p_\theta(\mathbf{x}_{0:T})$ is the model's approximation.

Since the noise at timestep t can be computed in closed form by equation 2, the network can be trained effectively by optimizing random terms of variational lower bound 4. By following the analysis in [5] the training objective can be simplified to:

$$\mathcal{L} = ||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2, \quad \epsilon \sim \mathcal{N}(0,1) \tag{5}$$

where $\epsilon_\theta(\mathbf{x}_t, t)$ is expressed by a neural network and ϵ is the added noise according to equation 2.

Generation Process: For the generation process, we follow the DDPM generation rule [5], using equation 6 for timesteps $T, ..., 1$.

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\ \epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z} \tag{6}$$

where $\mathbf{z} \sim \mathcal{N}(0,1)$ and σ_t can be either $\sqrt{\beta_t}$ or $\sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t}$. Our experiments indicate that using the latter value for σ_t yields slightly better results without a significant difference. During the last iteration, we do not add random noise; σ_1 is set to 0.

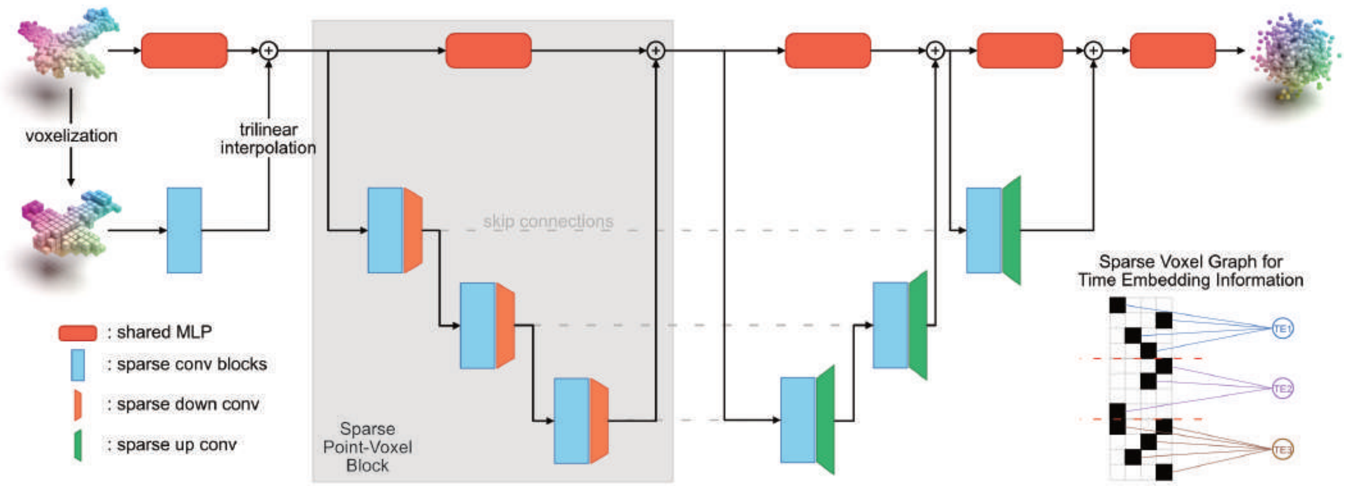

Fig. 3. Example architecture of the Sparse Point-Voxel U-Net. The initial point clouds are voxelized and sparse convolutions extract features incorporating neighborhood information. These features are propagated back to the point representation and are merged with the point features, extracted through shared-MLPs. This dual branch architecture is called Sparse Point-Voxel Block (SPVBlock). Note that, as shown, the voxel computations at each SPVBlock may vary, and the point branches in the encoder and decoder do not need to be symmetric. Additionally, we illustrate how sparse voxels and time embeddings can be linked as graph nodes to efficiently handle the varying number of sparse voxels in each point cloud in a batch.

Additionally, we explore the implicit generation rule introduced in DDIM [6], described by equation 7. This method introduces a non-Markovian generation process, that leverages the same training procedure as DDPM. The generation process in DDIM follows a deterministic trajectory, allowing for the sampling of fewer timesteps, which results in faster generation times.

$$\mathbf{x}_t = \sqrt{\overline{\alpha}_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\overline{\alpha}_t}\ \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\overline{\alpha}_t}} \right) + \sqrt{1-\overline{\alpha}_{t-1}}\ \epsilon_\theta(\mathbf{x}_t, t) \tag{7}$$

B. Sparse Point-Voxel Models

Based on [29], we propose a novel dual-branch architecture that combines the high fidelity of point representations with the effectiveness of voxel convolutions to extract features, leveraging neighborhood information. Given the complexity and space required to illustrate the entire model, we find it beneficial to present an example architecture that highlights all the key components of the network, as shown in figure 3.

Our model is structured around a series of Sparse Point-Voxel Blocks (SPVBlocks), where both the input and the output are point representations of shape B x N x F, with B representing the batch dimension, N the number of points per point cloud, and F the feature dimension. Initially, the input point cloud is voxelized into a sparse grid which undergoes processing through a combination of Residual Blocks, attention blocks, and either downsampling or upsampling convolution layers. Once processed, the final voxel features are projected back to the original points via trilinear interpolation and then added to the point features, which have been refined through an MLP.

In addition to the SPVBlocks, the combined blocks of the voxel branch form a U-Net network [54]. The architecture of this model follows the design of the DDPM U-Net [5], with appropriate adaptations for the sparse voxel domain. Further details are provided in the appendix.

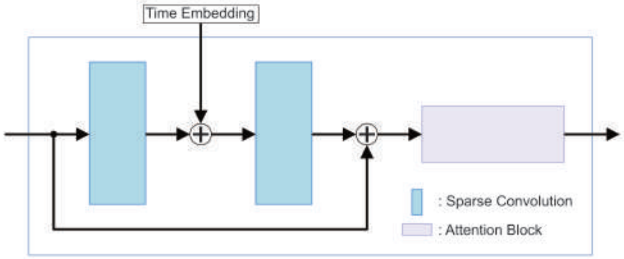

Fig. 4. Illustration of a Sparse Residual Convolutional Block. Time embedding information is integrated into the voxel features between two successive convolutional blocks. An optional attention block can further process the voxel features to incorporate global shape information.

A key challenge in this sparse voxel approach is incorporating the time embedding information into the network architecture. The time embedding represents the current step in the denoising process. To integrate the time embedding with the voxel features F, we project it to a $scale$ and a $shift$ through the use of a multi-layer perceptron (MLP).

$$F' = scale * F + shift \tag{8}$$

However, the sparse voxels are stored sequentially for the entire batch, with their numbers varying among the individual point clouds. Furthermore, not all point clouds share the same time embedding, especially during training. This raises the need for an algorithm that would link each sparse voxel with the correct $scale$ and $shift$.

To avoid an iterative process, we implement a novel approach where each sparse voxel and each time embedding are represented as graph nodes and are connected accordingly as illustrated in Figure 3. For our implementation, we utilize PytorchGeometric [55], a framework designed for efficient graph

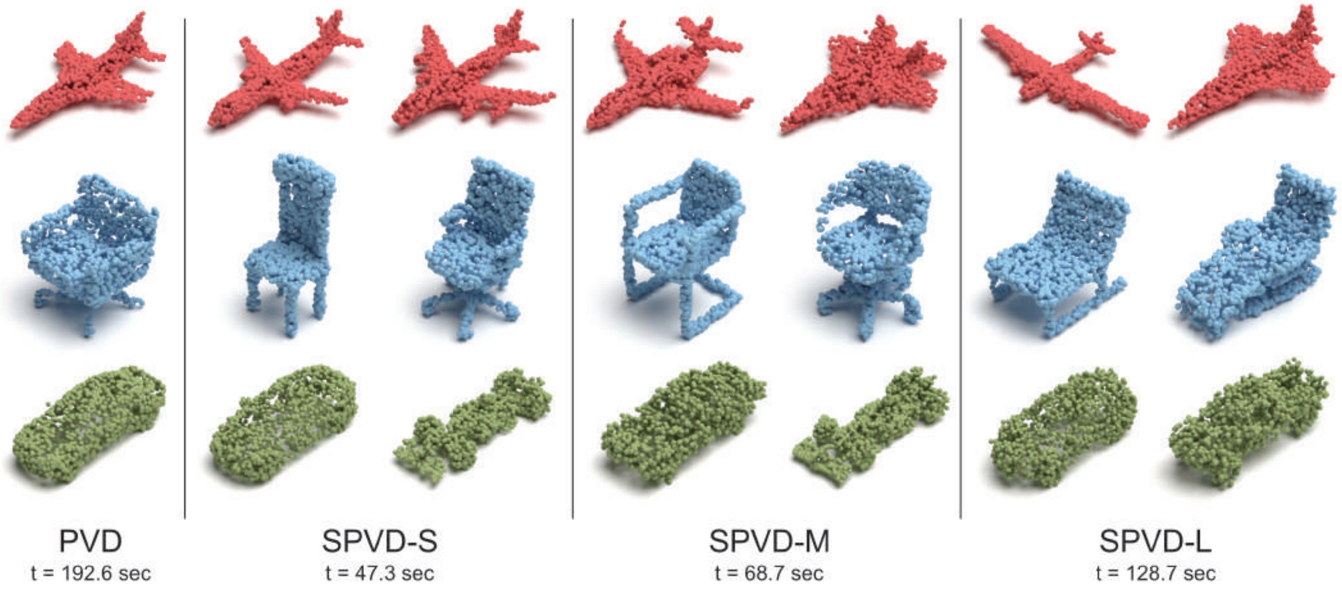

Fig. 5. Results of unconditional generation using our three model variants compared to PVD [24]. While all models produce high-quality point clouds, our largest models can generate more unique shapes with coarser features, whereas PVD has lower shape diversity. For each model we report the generation time of a batch with 32 samples.

processing. Additionally, the tensor storing the sparse voxels and graph nodes follows the same structure, allowing this transition without any computational overhead. This graph-based approach significantly boosts network execution and enables efficient batch processing. In figure 4 we illustrate the design of a sparse convolutional block. Time embedding information is incorporated between two successive convolutional blocks. The attention block is an optional component.

Finally, it is important to mention that the proposed architecture can process various point densities without any architectural changes, since the point operations are limited to the Shared-MLPs.

IV. Experiments

A. Unconditional Shape Generation

For comparative evaluations, we demonstrate our network's results on the ShapeNet [56] Airplane, Chair, and Car categories, following the paradigm of previous works [17], [20]–[25], [27]. To ensure fair comparisons, we use the same dataset and preprocessing methods proposed in PointFlow [21].

Unlike other methods, where the model is trained for 10k epochs and interval checkpoints are separately evaluated, we adopt a more efficient approach due to time and energy required for such extensive training and evaluation. We train our model using a one-cycle learning rate scheduler [57] for a smaller number of epochs and evaluate only the final checkpoint. For each checkpoint, we conduct three evaluation tests and report the best results. This approach acknowledges the stochasticity in the evaluation process, recognizing that running multiple evaluations can increase the chances of producing better results. However, rather than testing hundreds of interval checkpoints, we limit our evaluations to three, which strikes a balance between thoroughness and efficiency.

TABLE I
Comparative evaluations on shape generation using the 1-NN metrics for ShapeNet Airplane, Chair, and Car categories, employing Chamfer Distance (CD) and Earth Mover Distance (EMD) as distance metrics. Lower (↓) scores indicate better generation quality and shape diversity.

	Airplane		Chair		Car	
	CD	EMD	CD	EMD	CD	EMD
r-GAN [17]	98.40	96.79	83.69	99.70	94.46	99.01
l-GAN (CD) [17]	87.30	93.95	68.58	83.84	66.49	88.78
l-GAN (EMD) [17]	89.49	76.91	71.90	64.65	71.16	66.19
Shape-GF [20]	80.00	76.17	68.96	65.48	63.20	56.53
PointFlow [21]	75.68	70.74	62.84	60.57	58.10	56.25
SoftFlow [22]	76.05	65.80	59.21	60.05	64.77	60.09
DPF-Net [23]	75.18	65.55	62.00	58.53	62.35	54.48
SPVD-S (*ours*)	**73.82**	**64.56**	**57.10**	**55.97**	**56.39**	**53.83**
DPM [25]	76.42	86.91	60.05	74.77	68.89	79.79
PVD [24]	73.82	64.81	56.26	53.32	**54.55**	53.83
SPVD-M (*ours*)	73.95	63.08	56.11	57.10	70.88	52.98
SPVD-L (*ours*)	**73.21**	**61.97**	**55.36**	**52.56**	70.74	**52.41**

Evaluation results are presented in Table I. The 1-NN metric [21] is used to evaluate the performance of different methods, while Chamfer Distance (CD) and Earth Mover Distance (EMD) are employed as distance metrics. Lower scores indicate better generation quality and shape diversity. We observe that the smallest variant of SPVD outperforms non diffusion methods while producing results equivalent to PVD. Furthermore, SPVD-L variant achieves state-of-the-art results across all methods while still maintaining a faster runtime that PVD. Qualitative results of our networks and PVD are presented in Figure 5, along with the generation time for a batch of 32 samples. Details of the network variants and

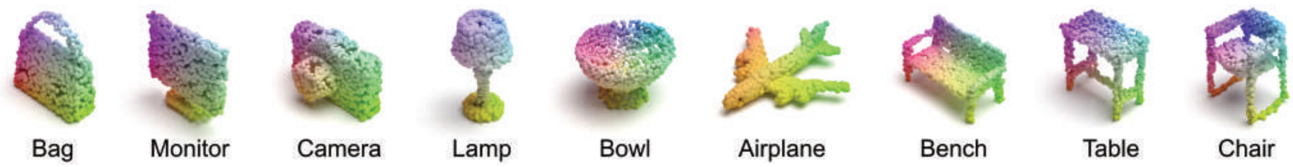

Fig. 6. Point clouds generated using the conditional SPVD-L model trained on all categories of ShapeNet. The use of the conditional embedding allows us to specify the class of the generated objects. Our model demonstrates its scalability by generating clean shapes across various categories.

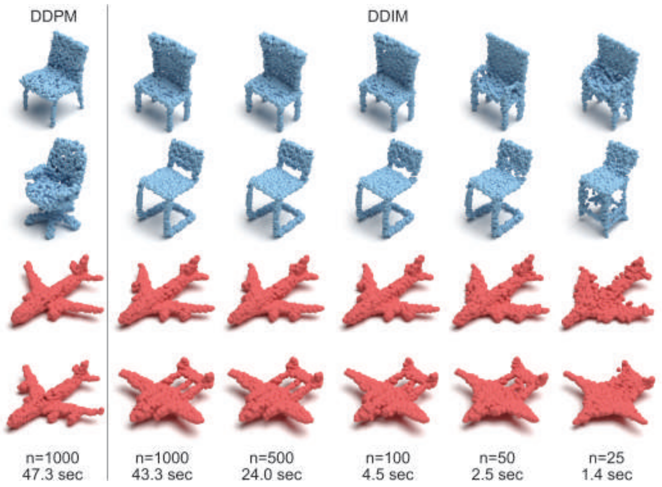

Fig. 7. Comparison of probabilistic (DDPM) and implicit (DDIM) generation. Each row displays shapes generated from the same initial random noise. For each shape, the number of sampling steps and the generation time for a batch of 32 point clouds are reported. The implicit model consistently converges to the same shape, demonstrating that DDIMs follow a deterministic trajectory during the denoising process. Additionally, the generative model can produce high-quality samples even with 100 sampling steps, reducing the initial sampling time to one-tenth.

TABLE II
1-NN METRIC AND GENERATION TIMES FOR THE SPVD VARIANTS USING THE DDIM GENERATION RULE WITH 100 AND 1000 STEPS. THE DDPM METRICS ARE ALSO INCLUDED FOR COMPARISON.

DDIM steps	Gen (sec)	Airplane		Chair		Car	
		CD	EMD	CD	EMD	CD	EMD
SPVD-S - **23M** parameters							
100	4.5	85.80	65.92	64.73	60.50	66.19	57.81
1000	43.3	75.18	65.18	61.63	59.74	60.36	60.22
DDPM	47.3	73.82	64.56	57.10	55.97	56.39	53.83
SPVD-M - **33M** parameters							
100	6.7	84.19	75.80	75.37	77.26	79.97	77.41
1000	68.4	84.56	79.75	78.32	79.07	87.21	83.38
DDPM	68.7	73.95	63.08	56.11	57.10	70.88	52.98
SPVD-L - **88M** parameters							
100	12.8	84.07	71.85	69.48	71.48	78.12	75.00
1000	127.6	80.00	78.02	70.77	71.97	80.25	76.42
DDPM	128.7	73.21	61.97	55.36	52.56	70.74	52.41

additional evaluation metrics are provided in the Appendix.

B. Conditional Generation

It is important for a general-purpose generative model, akin to the novel generative models in image synthesis, to be capable of generating various objects from a wide range of categories. To test our model's ability to generalize across multiple categories, we train it on all categories of ShapeNet. To select the object category for generation, we use a conditional class embedding, which is incorporated into the pipeline by adding it to the time embedding.

In figure 6, we present generation results of objects from different categories. The results indicate that the model succeeds in generating shapes from diverse categories, demonstrating that the proposed backbone can scale to accommodate more data.

C. Implicit Generation

In this section, we evaluate the results of our network when using the DDIM rule for shape generation. Implicit generation, as proposed by [6], suggests that the generation process follows a deterministic trajectory and allows for generation with fewer sampling steps. The aim of this experiment is to decrease the point cloud generation time while observing the effect on generation quality.

In Figure 7, we illustrate the generation results for various sampling steps, all starting from the same random noise. We present results for the chair and airplane categories, chosen for their distinguishable features. The model used in this experiment is the SPVD-S variant, trained for unconditional generation. It is evident that our model can generate high-quality shapes with just 100 steps, equivalent to one-tenth of the initial generation time. This speed-up could enable a range of applications where execution speed is prioritized over absolute accuracy.

In Table II we present the 1-NN metric for the DDIM generation rule using 100 and 1000 generation steps along with the corresponding runtime.The qualitative results of Figure 7 are quantitatively verified for the SPVD-S variant, demonstrating high generation quality even with just 100 steps. Interestingly, the largest models experience a significant performance drop. This may be due to the larger number of parameters and the lack of random noise during generation, which favors variability.

D. Shape Completion

We further test our model's ability to complete incomplete shapes by proposing a new task called Part Completion. The

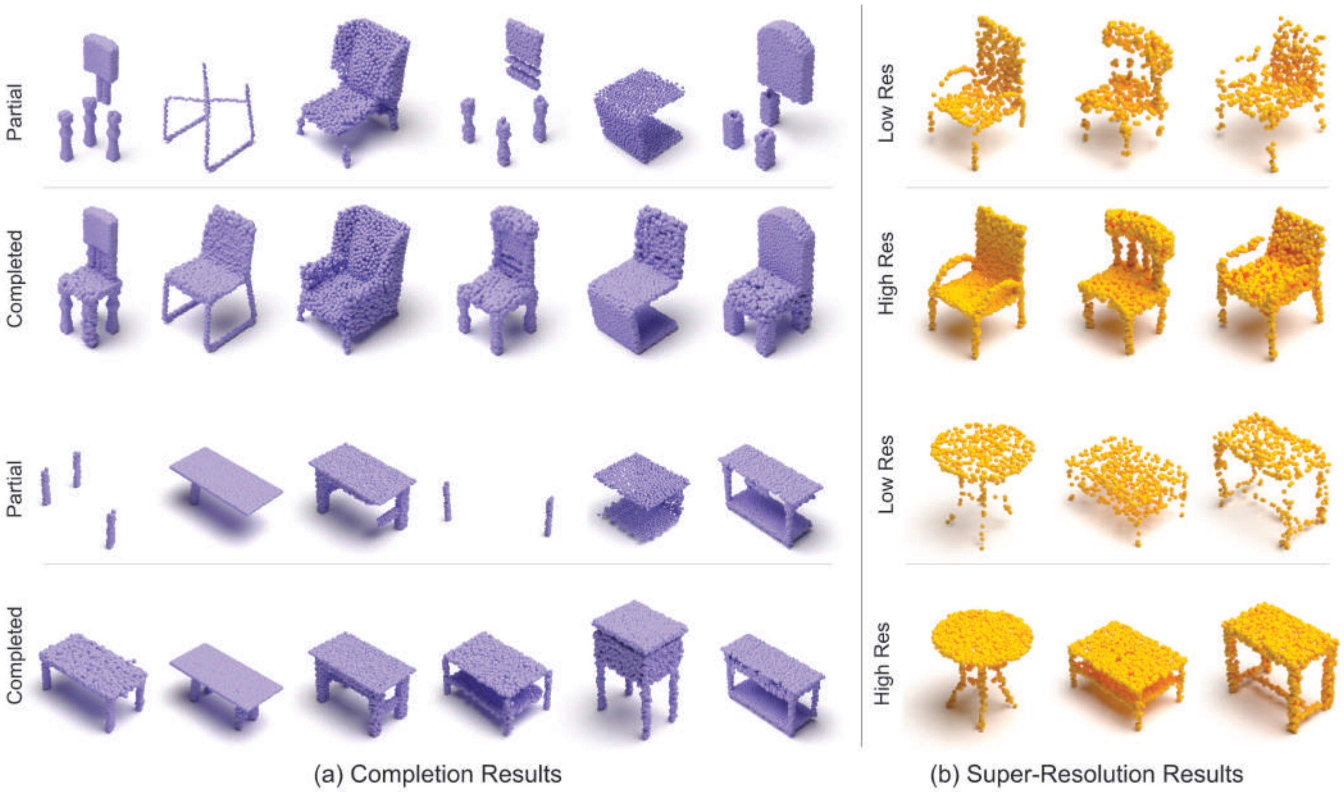

Fig. 8. Results of (a) completion and (b) super-resolution networks. (a) The completion network successfully fills in missing parts of the objects without any guidance, predicting only the missing points for input point clouds with varying point counts. (b) The super-resolution network not only increases point density but also adds details to shapes (such as the back of the chair) and fills gaps, like missing points in the chair handles or uneven chair or table legs.

input to the model consists of shapes from the PartNet [58] dataset, where random parts of the objects have been removed. More details are provided in the appendix. The task for the model is to reconstruct the missing areas. The key challenges of this task include the variable number of input points, as the selection and number of discarded parts are random, leading to varying input shapes. Additionally, there is no information about the location of the missing points, requiring the model to infer the underlying geometry.

During training, the input points remain constant while only the selected parts for removal are noisified. The model is tasked with estimating the added noise, similar to the generation pipeline. At inference, random noise is concatenated with the input, and the model gradually denoises this new noise to reconstruct the missing parts.

Completion results are presented in Figure 8 (a). We showcase results for the chair and table categories, chosen for their distinctive features. The network successfully completes the missing parts of the shapes.

E. Point Cloud Super Resolution

Super-Resolution is a well-known task in the image domain, but it has received limited attention in Point Clouds. Structurally, for diffusion models, super-resolution is similar to completion, with the primary difference being that in super-resolution, random points are noisified, while in completion, specific parts. For our experiment, we use a point cloud with 512 points as input and generate an output with 2048 points.

The results of the super-resolution task are shown in Figure 8 (b). We notice that the low sampling rate of the input results in many missing details, which can alter the shape, such as uneven chair legs or missing parts in the handles and back. The model not only increases the resolution of the shapes but also fills in the missing information.

V. Conclusion and Future Work

In this paper, we introduced SPVD, a novel diffusion U-Net architecture that enables efficient and scalable point cloud generation. Our approach achieves state-of-the-art results in point cloud generation with significantly shorter generation times compared to other high-fidelity diffusion models. Our experiments demonstrate that SPVD can scale to larger datasets and achieve even faster generation times through implicit generation. Additionally, it proves to be a strong candidate for tasks such as point cloud completion and super-resolution.

Fast generative models for 3D shapes are crucial for applications where user experience is just as important as generation quality. SPVD represents a significant step forward in addressing this need.

Future research will focus on developing a latent diffusion pipeline, similar to [59] in the image domain, where the SPVD model operates within the latent space to denoise latent representations. This pipeline will also facilitate the use of guidance, allowing images or text prompts to influence the generative results. Given the computational efficiency of our model, this latent variant could be extended beyond the

ShapeNet to more practical datasets such as Objaverse [60] and Objaverse-XL [61]. However, these extensive datasets require filtering and selection of suitable models for training a diffusion model, as well as preprocessing — a separate area for future work.

Another area of future research involves developing novel evaluation metrics that can better distinguish between generation quality and shape diversity. It may also involve identifying distance metrics more suitable than Chamfer and Earth Mover distances. An example from the image domain is the Fréchet Inception Distance (FID) [62], which compares feature maps from specific layers of InceptionNet [63] rather than the images themselves.

Acknowledgments

The present paper has been developed as part of the RRREMAKER project, funded by the European Union 2020 Research and Innovation program under the Marie Sklodowska Curie - RISE grant agreement no 101008060 and the framework of H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union – NextGenerationEU (H.F.R.I. Project Number: 16469.).

Appendix A
Network Desing Details

In this section, we provide the implementation details of the three network variants.

The SPVD-S variant uses only a single SPVD block, meaning that there is a point representation only at the start and end of the voxel U-Net. The SPVD-M and SPVD-L variants share the same architecture but differ in latent space dimensions. An overview of the architectures is presented in Table III.

A minor difference between the architectures concerns how the latent space is increased. In the SPVD-S variant, the latent space is increased during the first convolution of each down block. In contrast, for the SPVD-M and SPVD-L variants, the increase occurs during the downsampling convolution, that is the last layer of the block. This approach reduces the parameter count, allowing for larger feature dimensions.

Our experiments show that training larger models with the architecture of the SPVD-S variant did not yield better results. We believe that the point skip connections in the larger variants facilitate the propagation of gradients, enabling the successful training of larger architectures.

Appendix B
Additional generation evaluation metrics

In this section of the appendix, we provide a comparative evaluation against other methods following [24]. The metrics used are Coverage (Cov) and Minimum Matching Distance (MMD). These metrics are complementary: Cov evaluates the diversity of the generated shapes compared to the test set, while MMD measures the quality of the generated shapes.

Although these methods are considered less reliable than the 1-NN metric [21], our models still achieve state-of-the-art results, as shown in Table IV.

Appendix C
Part Completion

In the context of circular economy and recycling initiatives, the repair and reuse of older objects are highly encouraged. Generative models can contribute to this effort by reconstructing missing parts, facilitating the retrieval of replacement parts in databases, or enabling 3D printing solutions. Additionally, 3D model designers could benefit from automated algorithms that either complete their shapes or suggest alternatives for specific parts.

To this aim, we propose a new task called Part Completion. We use PartNet [58], a 3D model dataset with hierarchical part annotations. We preprocess the models by following the hierarchical structure of their parts and representing it as a tree. Since some parts may be too small, resulting in minimal scan information, we merge all leaves with a low point count into their parent nodes. We then save the resulting point cloud along with their per-point part labels.

During training and evaluation, we set m as the minimum number of parts that a partial object should have. We then randomly select a number between 1 and m to determine the parts of the object to discard.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] N. Zhang, "Learning adversarial transformer for symbolic music generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1754–1763, 2023.

[3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[4] M. W. Lam, Q. Tian, T. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song *et al.*, "Efficient neural music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[6] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[7] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[8] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.

[9] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, "Point2mesh: A self-prior for deformable meshes," *ACM Trans. Graph.*, vol. 39, no. 4, 2020.

[10] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[11] T. Hashimoto and M. Saito, "Normal estimation for accurate 3d mesh reconstruction with point cloud model incorporating spatial structure." in *CVPR workshops*, vol. 1, 2019, pp. 1–10.

[12] C. Lv, W. Lin, and B. Zhao, "Voxel structure-based mesh reconstruction from a 3d point cloud," *IEEE Transactions on Multimedia*, vol. 24, pp. 1815–1829, 2022.

[13] S. Peng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, "Shape as points: A differentiable poisson solver," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 032–13 044, 2021.

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

TABLE III

ARCHITECTURAL DETAILS OF THE PROPOSED ARCHITECTURES. THE FIRST ROW LISTS THE LAYERS OF THE VOXEL U-NET ARCHITECTURE. FOR EACH ARCHITECTURE, WE REPORT THE LAYERS USED, THE FEATURE DIMENSIONS, WHETHER THE OUTPUT OF A BLOCK IS PROJECTED TO THE POINT SPACE (INDICATING THE END OF AN SPVD BLOCK AND THE START OF A NEW ONE), IF THERE IS AN UPSAMPLING OR DOWNSAMPLING CONVOLUTION IN THOSE LAYERS, AND IF THERE IS A VOXEL ATTENTION BLOCK. THE SYMBOLS USED ARE AS FOLLOWS: ✓INDICATES THE PRESENCE OF THE FEATURE IN THE BLOCK, ×INDICATES THAT THE FEATURE IS EXPLICITLY SET TO FALSE, **-** INDICATES THAT THE FEATURE IS NOT APPLICABLE TO THE SPECIFIC LAYER OR IS AN OPTIONAL FEATURE THAT IS SKIPPED, AND **NA** INDICATES THAT THE LAYER DOES NOT EXIST IN THE ARCHITECTURE, MAKING THE VALUE NOT APPLICABLE.

Blocks:	stem	down1	down2	down3	down4	down5	mid	up1	up2	up3	up4	up5
SPVD-S												
feature dim	32	32	64	128	256	na	256	256	128	64	32	na
project to points	-	-	-	-	-	na	-	-	-	-	✓	na
down/up sample	-	✓	✓	✓	×	na	-	✓	✓	✓	×	na
use attention	×	×	×	×	✓	na	×	✓	×	×	×	na
SPVD-M												
feature dim	32	64	128	192	192	256	na	256	192	128	64	32
project to points	✓	-	-	-	-	✓	na	-	✓	-	-	✓
down/up sample	-	✓	✓	✓	✓	×	na	✓	✓	✓	✓	×
use attention	×	×	×	×	✓	✓	na	✓	✓	×	×	×
SPVD-L												
feature dim	64	128	192	256	384	384	na	384	256	192	128	64
project to points	✓	-	-	-	-	✓	na	-	✓	-	-	✓
down/up sample	-	✓	✓	✓	✓	×	na	✓	✓	✓	✓	×
use attention	×	×	×	×	✓	✓	na	✓	✓	×	×	×

TABLE IV

ADDITIONAL GENERATION EVALUATION METRICS.

Model	Airplane				Chair				Car			
	MMD (↓)		COV (%, ↑)		MMD (↓)		COV (%, ↑)		MMD (↓)		COV (%, ↑)	
	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
r-GAN [17]	0.4471	2.309	30.12	14.32	5.151	8.312	24.27	15.13	1.446	2.133	19.03	6.539
l-GAN (CD) [17]	0.3398	0.5832	38.52	21.23	2.589	2.007	41.99	29.31	1.532	1.226	38.92	23.58
l-GAN (EMD) [17]	0.3967	0.4165	38.27	38.52	2.811	1.619	38.07	44.86	1.408	0.8987	37.78	45.17
Shape-GF [20]	2.703	0.6592	40.74	40.49	2.889	1.702	46.67	48.03	9.232	0.7558	**49.43**	50.28
PointFlow [21]	0.2243	0.3901	47.90	46.41	**2.409**	1.595	42.90	50.00	**0.9010**	0.8071	46.88	50.00
SoftFlow [22]	0.2309	0.3745	46.91	47.90	2.528	1.682	41.39	47.43	1.187	0.8594	42.90	44.60
DPF-Net [23]	0.2642	0.4086	46.17	48.89	2.536	1.632	44.71	48.79	1.129	0.8529	45.74	49.43
SPVD-S (*ours*)	**0.2281**	**0.3807**	**48.64**	**49.62**	2.545	**1.549**	**47.88**	**52.11**	0.9155	**0.7501**	45.73	**53.40**
PVD [24]	0.2243	0.3803	**48.88**	52.09	2.622	1.556	49.84	50.60	1.077	0.7938	41.19	50.56
SPVD (*ours*)	**0.2187**	**0.3661**	44.44	51.11	0.2589	1.579	48.94	51.51	0.9947	0.7855	**46.02**	**51.98**
SPVD-L (*ours*)	0.2247	0.3705	48.46	**53.33**	0.2562	**1.521**	**51.15**	**52.87**	**0.9782**	**0.7495**	45.45	51.70

[15] J. Kim, J. Yoo, J. Lee, and S. Hong, "Setvae: Learning hierarchical composition for generative modeling of set-structured data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 059–15 068.

[16] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215.

[17] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 40–49.

[18] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized generative models for 3d point clouds via graph convolution," in *International Conference on Learning Representations*, 2019.

[19] D. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3858–3867.

[20] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[21] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[22] H. Kim, H. Lee, W. H. Kang, J. Y. Lee, and N. S. Kim, "SoftFlow: Probabilistic framework for normalizing flow on manifolds," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., pp. 16 388–16 397.

[23] R. Klokov, E. Boyer, and J. Verbeek, "Discrete point flow networks for efficient point cloud generation," in *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, 2020.

[24] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Proceedings of the IEEE/CVF International*

Conference on Computer Vision (ICCV), October 2021, pp. 5826–5835.

[25] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[26] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[27] L. Wu, D. Wang, C. Gong, X. Liu, Y. Xiong, R. Ranjan, R. Krishnamoorthi, V. Chandra, and Q. Liu, "Fast point cloud generation with straight flows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 9445–9454.

[28] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," in *Advances in Neural Information Processing Systems*, 2019.

[29] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020.

[30] C. R. Qi, H. Su, K. Mo, and 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.

[31] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[32] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[35] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, p. 90–105.

[36] C. Wen, J. Long, B. Yu, and D. Tao, "Pointwavelet: Learning in spectral domain for 3-d point cloud analysis," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2024.

[37] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268.

[38] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 19 313–19 322.

[39] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision – ECCV 2022*, 2022, pp. 604–621.

[40] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," *arXiv preprint arXiv:2205.14401*, 2022.

[41] I. Romanelis, V. Fotis, K. Moustakas, and A. Munteanu, "Exppoint-mae: Better interpretability and performance for self-supervised point cloud transformers," *IEEE Access*, vol. 12, pp. 53 565–53 578, 2024.

[42] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, and Y. Yue, "Pointgpt: Auto-regressively generative pre-training from point clouds," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[43] K. Abou Zeid, J. Schult, A. Hermans, and B. Leibe, "Point2vec for self-supervised representation learning on point clouds," 2023.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[45] B. Graham and L. van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[46] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018.

[47] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[48] H. Zhang, C. Wang, L. Yu, S. Tian, X. Ning, and J. Rodrigues, "Pointgt: A method for point-cloud classification and segmentation based on local geometric transformation," *IEEE Transactions on Multimedia*, pp. 1–12, 2024.

[49] Z. Liu, Y. Feng, M. J. Black, D. Nowrouzezahrai, L. Paull, and W. Liu, "Meshdiffusion: Score-based generative 3d mesh modeling," in *International Conference on Learning Representations*, 2023.

[50] S. Lee, S. Heo, and S. Lee, "Dmesh: A structure-preserving diffusion model for 3-d mesh denoising," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2024.

[51] Z. Lyu, Z. Kong, X. XU, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," in *International Conference on Learning Representations*, 2022.

[52] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, 2015, pp. 2256–2265.

[53] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[55] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[56] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, 2015.

[57] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[58] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[59] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[60] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," *arXiv preprint arXiv:2212.08051*, 2022.

[61] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi, "Objaverse-xl: A universe of 10m+ 3d objects," *arXiv preprint arXiv:2307.05663*, 2023.

[62] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[63] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

Ioannis Romanelis received his Electrical and Computer Engineering diploma in 2021 at the University of Patras. During the same year, he enrolled for a PhD at the same department under the supervision of Professor Konstantinos Moustakas and joined the Visualization and Virtual Reality (VVR) group. His main research interests include computer vision, deep learning, point cloud processing, 3D scene understanding, generative modeling, and explainable AI.

Vlassis Fotis received his Electrical Engineering and Computer Technology degree in 2021 at the university of Patras. During the same year he joined the VVR lab as a PhD candidate. His main research interests include (but are not limited to) computer vision, theoretical deep learning, 3D scene understanding and geometry processing.

Athanasios Kalogeras (Senior Member, IEEE) received the Diploma degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the University of Patras, Greece. He has been with the Industrial Systems Institute, ATHENA Research and Innovation Center, since 2000, where he currently holds a position of the Research Director. He has worked as an Adjunct Faculty at the Technological Educational Foundation of Patras. He has been a Collaborating Researcher at the University of Patras, the Computer Technology Institute and Press “Diophantus,” and the private sector. His research interests include cyber- physical systems, the Industrial IoT, industrial integration and interoperability, and collaborative manufacturing. Application areas include the manufacturing environment, critical infrastructure protection, smart buildings, smart cities, smart energy, circular economy, health, and tourism and culture. He has served as a program committee member for more than 30 conferences and as a reviewer in more than 40 international journals and conferences. He has been a Postgraduate Scholar of the Bodossaki Foundation. He is a member of the Technical Chamber of Greece. He is a Local Editor in Greece of ERCIM News.

Christos Alexakos (Member, IEEE) received the D.Eng. degree in computer engineering and informatics, the M.Sc. degree in computer science and engineering, and the Ph.D. degree from the Department of Computer Engineering and Informatics, University of Patras, Greece. He was the Technical Manager of the Pattern Recognition Laboratory for ten years and the project manager in four development projects carried out by the laboratory. In 2013, he co-founded InSyBio Ltd., a bioinformatics company that delivers a cloud-based software suite for intelligent analysis of biological big data. He is currently Principal Researcher with the Industrial Systems Institute, ATHENA Research and Innovation Center, Greece. He is collaborating as a Research Engineer with the Pattern Recognition Laboratory, Department of Computer Engineering and Informatics, University of Patras, and the Computer Technology Institute. He has multi-year experience in development of software applications, both web-based and standalone, including databases applications, GIS, and service-oriented applications. He is a highly experienced programmer and a software architect and has worked as a freelancer in ICT projects for both public and private sector, since 2003. He has published 15 articles in journals, 49 in conference proceedings, and five chapters in books. He also has a submitted patent to USPO. His main research interests include information systems architecture and integration in the fields of enterprise and manufacturing processes, bioinformatics, the IoT, cloud computing, and cloud manufacturing.

Konstantinos Moustakas (Senior Member, IEEE) received the Diploma degree and the PhD in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2003 and 2007 respectively. During 2007-2011 he served as a post-doctoral research fellow in the Information Technologies Institute, Centre for Research and Technology Hellas. He is currently a Professor at the Electrical and Computer Engineering Department of the University of Patras, Head of the Visualization and Virtual Reality Group, Director of the Wire Communications and Information Technology Laboratory and Director of the MSc Program on Biomedical Engineering of the University of Patras. He serves as an Academic Research Fellow for ISI/Athena research center. His main research interests include virtual, augmented and mixed reality, 3D geometry processing, haptics, virtual physiological human modeling, information visualization, physics-based simulations, computational geometry, computer vision, and stereoscopic image processing. He is a senior member of the IEEE, the IEEE Computer Society and member of Eurographics.

Adrian Munteanu (Member, IEEE) is professor at the Electronics and Informatics (ETRO) department of the Vrije Universiteit Brussel (VUB), Belgium. He received the MSc degree in Electronics and Telecommunications from Politehnica University of Bucharest, Romania, in 1994, the MSc degree in Biomedical Engineering from University of Patras, Greece, in 1996, and the Doctorate degree in Applied Sciences (Summa Cum Laudae) from Vrije Universiteit Brussel, Belgium, in 2003. In the period 2004-2010 he was post-doctoral fellow with the Fund for Scientific Research – Flanders (FWO), Belgium, and since 2007, he is professor at VUB. His research interests include image, video and 3D graphics compression, 3D video, deep-learning, distributed visual processing, error-resilient coding, and multimedia transmission over networks.