ShapeMorph: 3D Shape Completion via Blockwise Discrete Diffusion

Jiahui Li[1], Pourya Shamsolmoali[1*], Yue Lu[1], Masoumeh Zareapoor[2]
[1]East China Normal University, [2]Shanghai Jiao Tong University

1. Experiment Details

1.1. Training Details

We train our encoder for $\varepsilon = 800$ epochs with 32 batch size. The learning rate is linearly increased to $lr_{max} = 1e-3$ in the first $\epsilon_0 = 40$ epochs. Then gradually decreased using the cosine decay schedule $lr_{max} * 0.5^{1+\cos\frac{\epsilon-\epsilon_0}{\varepsilon-\epsilon_0}}$, until the minimum value of $lr_{min} = 1e-6$. The training process for the blockwise diffusion model is set to $\varepsilon = 400$ epochs with a batch size of 32. The learning rate is progressively increased to $lr_{max} = 2e-4$ within the initial $\epsilon_0 = 20$ epochs. Subsequently, it is progressively reduced to $lr_{min} = 1e-6$, guided by a monitoring system that tracks the frequency of no decrease in the loss value, with a threshold set at 5k iterations. Adamw optimizer is used with β = [0.9, 0.96].

1.2. Dataset Details

For single-view shape completion, we follow the benchmark provided by GenRe [40], training our model on [airplane, chair, car] categories, in which each shape containing 20 views of rendering. For ShapeNet [4], we train our model using the objects of 13 categories, including [airplane, bench, cabinet, car, chair, display, lamp, speaker, rifle, sofa, table, telephone, vessel]. For PartNet [24], we adopt the training approach used in other works [33, 35, 43], focusing on the categories [chair, lamp, table]. We extract occupancy value similar to [22], including 50k volume query points from the bounding volume ($[-1, 1]^3$) and 50k near query points from near surface region.

1.3. Baseline Comparison

We retrain all baselines with their original implementations and hyperparameters for a fair comparison.

cGAN [33] first introduces the concept of multiple point cloud completions using a GAN-based model, defining this approach as multi-modal shape completion. We reproduce their work using their official implementation while following our incompleteness setting.

PVD [43] operates directly on raw point clouds and proposes a completion method based on a continuous diffusion model. We follow their setting, using 2048 points as complete point cloud and 512 partial points as condition to implement their work.

SFormer [35] approximate the input into a variable-length discrete sequence and model the sequence completion based on an autoregressive model. This model takes point cloud as input and produce the output as deep implicit function, specifically occupancy value [22]. We use 2048 points as input to retrain SFormer following its official implementation.

AutoSDF [23] represents each 3D shape as 64^3 Truncated-SDF (TSDF) and learns latent patch priors using P-VQ-VAE along with a transformer-based autoregressive model for 3D shape completion. Similarly, we extract the TSDF described as [15], and reproduce it with its official implementation. Note that, the TSDF and the occupancy value (our representation) both belong to the realm of deep implicit functions, they can both represent shapes as implicit surfaces. During testing, we sample points on the surface to calculate errors with the ground truth point cloud.

3DQD [20] represents 3D shapes as 64^3 Truncated-SDF (TSDF), using the same P-VQ-VAE as AutoSDF but adopting a discrete diffusion model for 3D shape completion. We retrained the model using the official implementation.

1.4. Different settings on ShapeNet

Different to other baselines [33, 35, 43], which can accept point cloud from arbitrary views, AutoSDF and 3DQD only accept voxelized TSDF as input. It means the input is the voxel grids ($X \in \mathbb{R}^{n\times n\times n}$). Thus, these two works can only perform incomplete shapes through manually masking input grids and fail to accept arbitrary incomplete shapes as cGAN, PVD, and SFormer. In order to perform precise comparison with these two works, we follow their experiment setting, removing all points from the top half of shapes (Bottom Half) as incomplete shape to showcase the ability of our method.

[*] Correspondence to <pshams55@gmail.com>

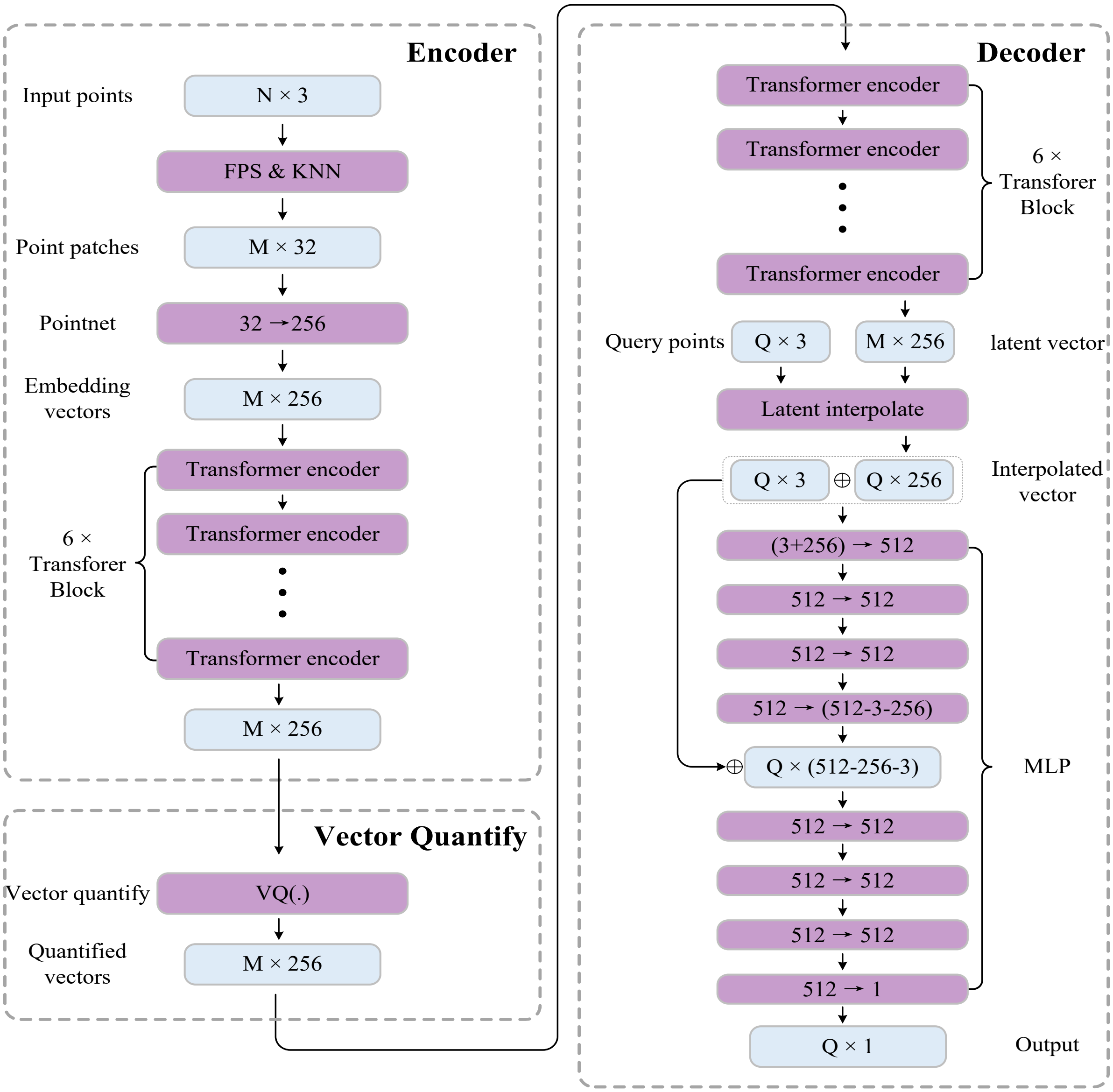

Figure 1. Network architecture of our shape encoding schedule. We show the detailed module inside our autoencoder.

1.5. Additional Instructions of Single-view Shape Completion

Since AutoSDF and 3DQD did not adopt point cloud as the condition in their official implementation, we directly take the depth map as a condition, following their approach in the single-view 3D reconstruction task.

2. Network Architecture of Shape Autoencoder

Our shape autoencoder consists of three components: an Encoder E, a decoder D and a vector quantizer VQ. Following [38], we represent each 3D shape as an irregular discrete latent sequence. The network architecture is shown in Fig. 1. Additionally, In our decoder D, we employ a MLP to predict the occupancy value as outlined in [22]. Specifically, we interpolate the quantified vectors z_i^q obtained from VQ to each query point by:

$$\mathrm{z_q} = \sum_{i=0}^{M-1} \frac{\exp(-\beta \|\mathrm{q} - \mathrm{c}_i\|^2)}{\sum_{j=0}^{M-1} \exp(-\beta \|\mathrm{q} - \mathrm{c}_j\|^2)} z_i^q, \qquad (1)$$

in which, q is the query point used for surface reconstruction, while c_i is the position of each quantified vector z_i^q (which has the same position as the latent vectors). β is a learnable parameter controlling the smoothness of interpolation. With above operation (Eq. (1)), we get interpolated vectors with the same size as query points q. With a MLP we can predict the occupancy value of each query point.

3. Architecture details of Flow transformer

Our denosing network consists of 17 layers of Flow transformer (FLOT) blocks, each block contains switch operation, self attention, cross attention, and feed forward network (FFN). The dimension of each block is 1024 and the FFN contains two linear layer, which expand the dimension to 4096 in the middle layer.