TIGER: Time-Varying Denoising Model for 3D Point Cloud Generation with Diffusion Process Supplymentary Materials

Zhiyuan Ren, Minchul Kim, Feng Liu, Xiaoming Liu
Michigan State University
{renzhiy1, kimminc2, liufeng6, liuxm}@msu.edu

1. Complete Definition of Diffusion

This section gives a brief and complete definition of diffusion process [1] as a complement to Sec.3.1 of our main paper.

Forward Diffusion Process. Given a data point $\mathbf{X}_0$ sampled from data distribution $q(\mathbf{X})$, the diffusion process will gradually add gaussian noise with small scale to the $\mathbf{X}_0$ in T steps. Controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$, we can get a sequence of noisy samples $\{\mathbf{X}_t\}_{t=1}^{T}$.

$$q(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \sqrt{1-\beta_t}\mathbf{X}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

$$q(\mathbf{X}_{1:T}|\mathbf{X}_0) = \prod_{t=1}^{T} q(\mathbf{X}_t|\mathbf{X}_{t-1}) \tag{2}$$

Using the reparameterization technique and supposing that $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$ and $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$:

$$\begin{aligned} \mathbf{X}_t &= \sqrt{\alpha_t}\mathbf{X}_{t-1} + \sqrt{1-\alpha_t}\epsilon \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{X}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon \\ &= ... \\ &= \sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon. \end{aligned} \tag{3}$$

A nice property we can find here is that we can sample $\mathbf{X}_t$ with one step from $\mathbf{X}_0$.

Reverse Diffusion of Process. We learn a model p_θ to approximate the conditional probability for reverse diffusion process.

$$p_\theta(\mathbf{X}_{0:T}) = p(\mathbf{X}_T)\prod_{t=1}^{T} p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t) \tag{4}$$

$$p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t) = \mathcal{N}(\mathbf{X}_{t-1}; \mu_\theta(\mathbf{X}_t), t), \Sigma_\theta(\mathbf{X}_t, t)) \tag{5}$$

The reverse condition probability will be tractable when adding a new condition $\mathbf{X}_0$:

$$q(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{X}_0) = \mathcal{N}(\mathbf{X}_{t-1}; \tilde{\mu}(\mathbf{X}_t, \mathbf{X}_0), \tilde{\beta}_t\mathbf{I}). \tag{6}$$

Using Bayes' rule, we can get the following:

$$\begin{aligned} q(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{X}_0) &= q(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{X}_0)\frac{q(\mathbf{X}_{t-1}|\mathbf{X}_0)}{q(\mathbf{X}_t|\mathbf{X}_0)} \\ &\propto e^{-\frac{1}{2}(\frac{(\mathbf{X}_t-\sqrt{\alpha_t}\mathbf{X}_{t-1})^2}{\beta_t}+\frac{(\mathbf{X}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{X}_0)^2}{1-\bar{\alpha}_{t-1}}-\frac{(\mathbf{X}_t-\sqrt{\bar{\alpha}_t}\mathbf{X}_0)^2}{1-\bar{\alpha}_t})} \\ &= e^{-\frac{1}{2}\left((\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{X}_{t-1}^2-(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{X}_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{X}_0)\mathbf{X}_{t-1}+C\right)}. \end{aligned} \tag{7}$$

According to Eq. 7, the mean and variance can be parameterized as follows:

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{8}$$

$$\begin{aligned} \tilde{\mu}_t(\mathbf{X}_t, \mathbf{X}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{X}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{X}_0 \\ &= \frac{1}{\sqrt{\alpha_t}}(\mathbf{X}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t). \end{aligned} \tag{9}$$

Following VAE [2], we could apply the variational lower bound to optimize the negative log-likelihood:

$$-\log p_\theta(\mathbf{X_0}) \leq \mathbb{E}_q[\log\frac{q(\mathbf{X}_{1:T}|\mathbf{X}_0)}{p_\theta(\mathbf{X}_{0:T})}] = L_{VLB}. \tag{10}$$

Based on Eq. 10, we can approximate this and get the training objective:

$$\begin{aligned} L_{VLB} \approx L_t &= \mathbb{E}_{\mathbf{X}_0,\epsilon_t}[\frac{1}{2\left\|\Sigma_\theta\right\|_2^2}\left\|\tilde{\mu}_t-\mu_\theta\right\|^2] \\ &\approx \mathbb{E}_{\mathbf{X}_0,\epsilon_t}[\left\|\epsilon_t-\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{X}_0+\sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\right\|^2]. \end{aligned} \tag{11}$$

2. Linear Representation of Relative Position

We will show that our proposed position encoding method Phase Shift Position Encoding (PSPE) can possess the property of linear representation of relative position.

Given two 3D positions pos^n and $pos^m = pos^n + \delta$, we can formulate them following Eq. 7 of the main paper:

$$\mathbf{P}^n_{emb}(pos^n_j, 6i + 2*(j-1)) = \sin(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \tag{12}$$

$$\mathbf{P}^n_{emb}(pos^n_j, 6i{+}1{+}2*(j-1)) = \cos(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \tag{13}$$

$$\mathbf{P}^m_{emb}(pos^m_j, 6i + 2*(j-1)) = \sin(\frac{pos^m_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \tag{14}$$

$$\mathbf{P}^m_{emb}(pos^m_j, 6i{+}1{+}2*(j-1)) = \cos(\frac{pos^m_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}), \tag{15}$$

where where pos_j is j-th element in each point's position, specifically $pos_1 = x$, $pos_2 = y$, $pos_3 = z$, and i is the dimension.

We plug $pos^m = pos^n + \delta$ into Eq. 14 and Eq. 15:

$$\begin{aligned}
&\mathbf{P}^m_{emb}(pos^m_j, 6i + 2*(j-1)) \\
&= \sin(\frac{pos^n_j + \delta}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \\
&= \sin(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \cos(\frac{\delta}{10000^{\frac{2i}{D}}}) \\
&+ \cos(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \sin(\frac{\delta}{10000^{\frac{2i}{D}}}) \\
&= \mathbf{P}^n_{emb,2i} \cos(\frac{\delta}{10000^{\frac{2i}{D}}}) + \mathbf{P}^n_{emb,2i+1} \sin(\frac{\delta}{10000^{\frac{2i}{D}}})
\end{aligned} \tag{16}$$

$$\begin{aligned}
&\mathbf{P}^m_{emb}(pos^m_j, 6i + 1 + 2*(j-1)) \\
&= \cos(\frac{pos^n_j + \delta}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \\
&= \cos(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \cos(\frac{\delta}{10000^{\frac{2i}{D}}}) \\
&- \sin(\frac{pos^n_j}{10000^{\frac{2i}{D}}} + (j-1)\frac{2\pi}{3}) \sin(\frac{\delta}{10000^{\frac{2i}{D}}}) \\
&= \mathbf{P}^n_{emb,2i+1} \cos(\frac{\delta}{10000^{\frac{2i}{D}}}) - \mathbf{P}^n_{emb,2i} \sin(\frac{\delta}{10000^{\frac{2i}{D}}}),
\end{aligned} \tag{17}$$

where $\sin(\frac{\delta}{10000^{\frac{2i}{D}}})$ and $\cos(\frac{\delta}{10000^{\frac{2i}{D}}})$ are constant for each i^{th} channel, which shows that our PSPE can linearly represent the relative position. According to [4], this property will make it easier for models to learn the position representation.

3. Implementation Details

Hyper-parameters. We mainly follow the setting in [5]: We set the learning rate $2e-4$ and the batch-size 16; The timestep in the diffusion process is $1,000$ and we use the linear schedule to gradually increase the β from $1e-4$ to 0.002; The number of the points N in $\mathbf{X}_t$ is $2,048$ and the number of the points M in downsampled $\hat{\mathbf{X}}_t$ is 256; The dimension d of the latent point cloud is 256 and the dimension D of the tokens is 512; We add dropout layer with probability 0.1 to the last layer of our TIGER model.

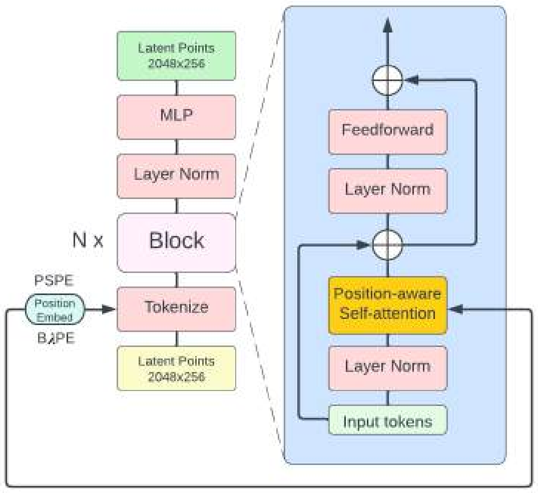


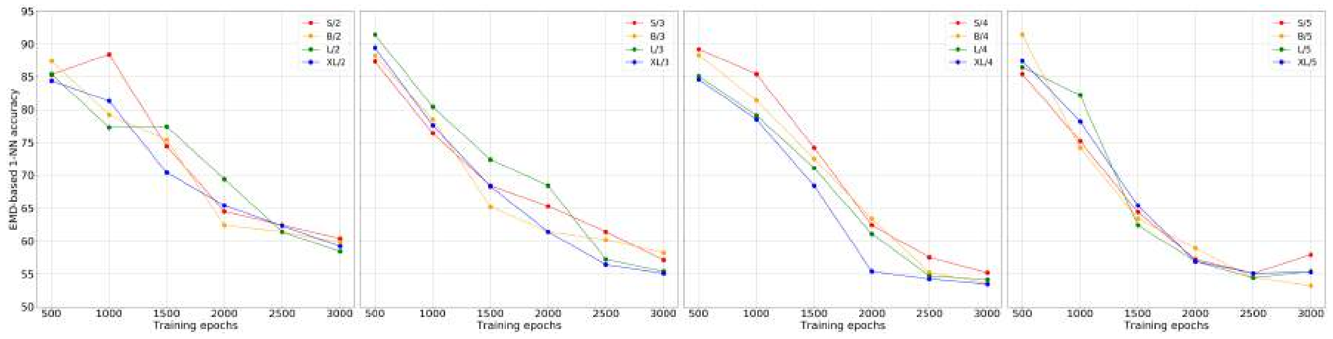
Figure 1. Overview of our latent point Transformer.

Architecture of Latent Point Cloud Transformer. As shown in Fig. 1, we first transform the latent point cloud into tokens, added by our proposed position embedding (PSPE/BλPE). Then we will perform N times global feature extraction by Transformer blocks, which replace the standard self-attention module with our position-aware self-attention module. Following [3], we also add time-shift and time-scale operations to fuse the timestep condition. Overall, we aim to retain the standard Transformer architecture and highlight the effectiveness of the proposed position encoding methods and position-aware self-attention module.

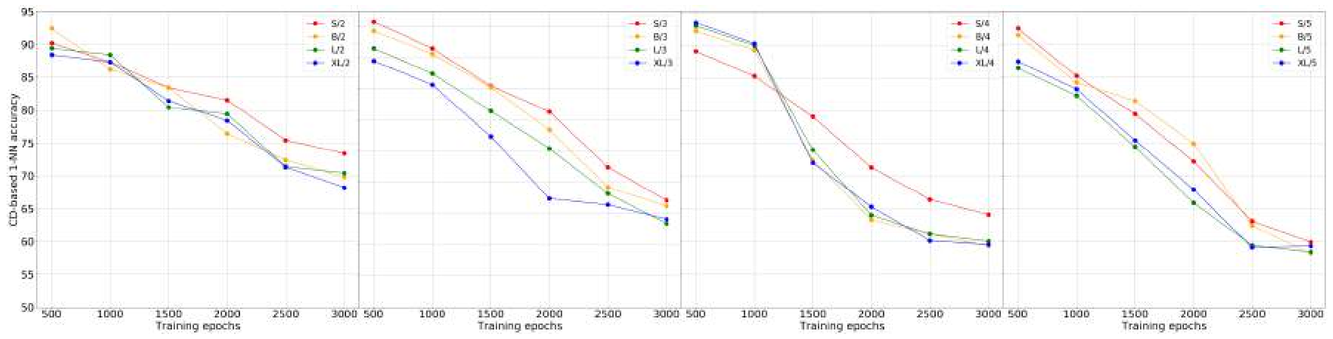
Computation of ConvNet Importance. This section gives the mathematical explanation for the description in line 787 of the main paper. As discussed in Eq.17 and Eq.18 of the main paper, we have two time-masks $\mathbf{M}_c$, $\mathbf{M}_{tr} \in \mathbb{R}^D$ for CNN and Transformer respectively. For each timestep, the importance of ConvNet can be computed by counting how many elements in $\mathbf{M}_c$ is larger than that in $\mathbf{M}_{tr}$:

$$\text{ConvNet Importance} = \frac{\text{sum}[\mathbf{M}_c > \mathbf{M}_{tr}]}{D}, \tag{18}$$

where $\text{sum}[\cdot]$ is the function: $\mathbb{R}^D \rightarrow \mathbb{R}^1$ and the importance of the ConvNet ranges from 0 to 1. Accordingly, since $\mathbf{M}_{tr} = 1 - \mathbf{M}_c$, the reduction of the ConvNet's importance implies the improvement of the Transformer's importance.



(a) EMD-based 1-NN accuracy across training epochs.



(b) CD-based 1-NN accuracy across training epochs.

Figure 2. We show EMD-based 1-NN accuracy and CD-based 1-NN accuracy over training iterations for 16 of our TIGER models. For each figure, we hold the number of blocks in the encoder and decoder while scaling the depth of the Transformer.

4. Ablation on Scalability

Fig. 2 shows our TIGER model's mean accuracy under 16 different scales: (**S**mall/**B**ase/**L**arge/**X**-Large) / (**2**-blocks/**3**-blocks/**4**-blocks/**5**-blocks both in Encoder and Decoder), where **S/B/L/X** is used to describe the depth (number of blocks) in Transformer. As we increase the depth in Transformer, the model will gain some improvement in EMD-based 1-NN accuracy and CD-based 1-NN accuracy while **L** model only reaches similar results as **XL** model. If we fix the the depth in Transformer and only scale the number of blocks in the encoder and decoder, we find that the performance will increase from 2 to 4 but stop growing from 4 to 5. These results show that there are some bottlenecks in our model. Simply scaling the number of blocks either in Transformer or encoder and decoder will not boost the performance.

5. Distance Map Visualization

As shown in Fig. 3, the 1-nearest-neighbor of generated samples is half in the generation set and half in the reference set, and it is the same with the reference samples, which means the generated samples are so similar to the reference set that it is hard to distinguish them. It is worth mentioning that Fig. 3g shows the 1-nearest-neighbor of reference samples is less in the generation set, which is corresponding to our slightly worse performance in CD-based accuracy.

6. More Visualization Results

Fig. 4, 5, 6 show the airplane, chair, and car objects from different angles. Fig. 7 shows more generated samples from the unified model under the 55-class setting.

References

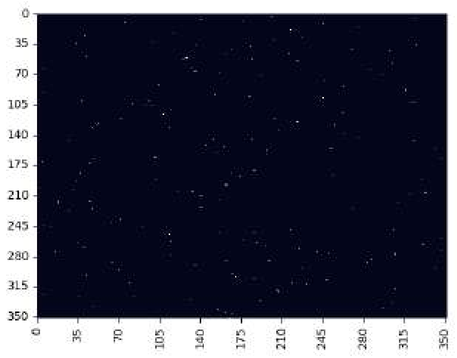
[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

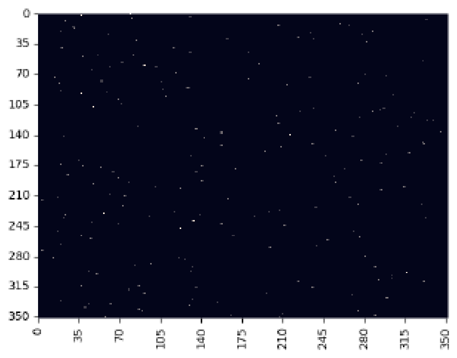
[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014. 1

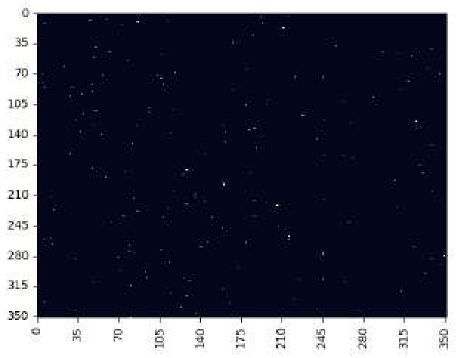
[3] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2

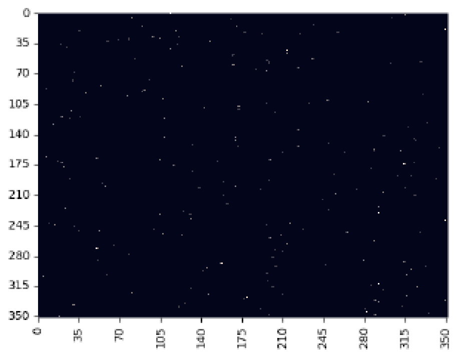
[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Neurips*, 2017. 2

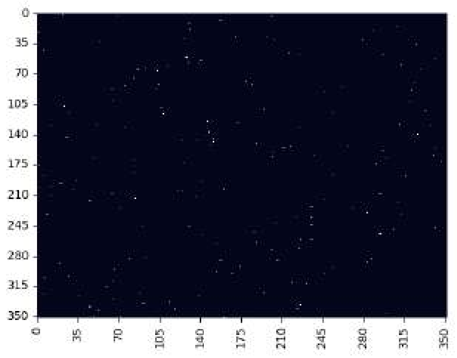
[5] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*. IEEE, 2021. 2

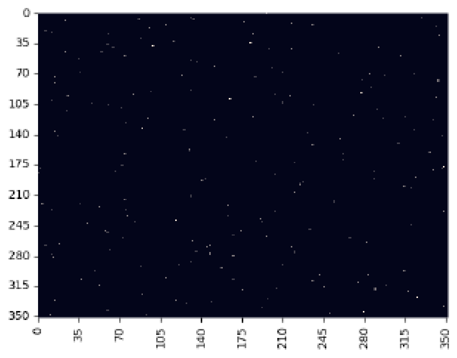

(a) Gen-Gen (EMD).

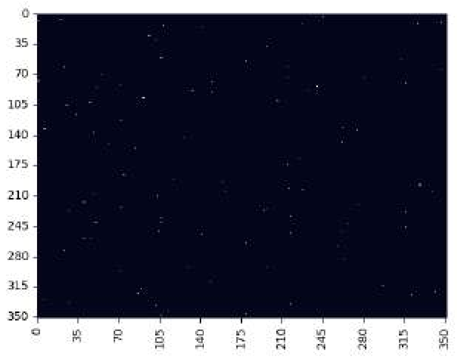

(b) Gen-Ref (EMD).


(c) Ref-Gen (EMD).


(d) Ref-Ref (EMD).


(e) Gen-Gen (CD).


(f) Gen-Ref (CD).


(g) Ref-Gen (CD).

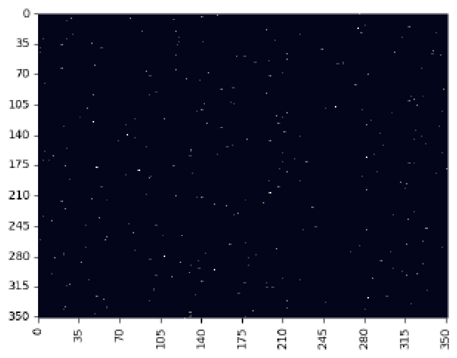

(h) Ref-Ref (CD).

Figure 3. Visualization of distance maps. Y-axis shows the index of query point clouds and X-axis shows the index of key point clouds. The white dot(x,y) means the y's 1-nearest-neighbor is x. [Key: Gen=Generated samples; Ref=Reference samples.]

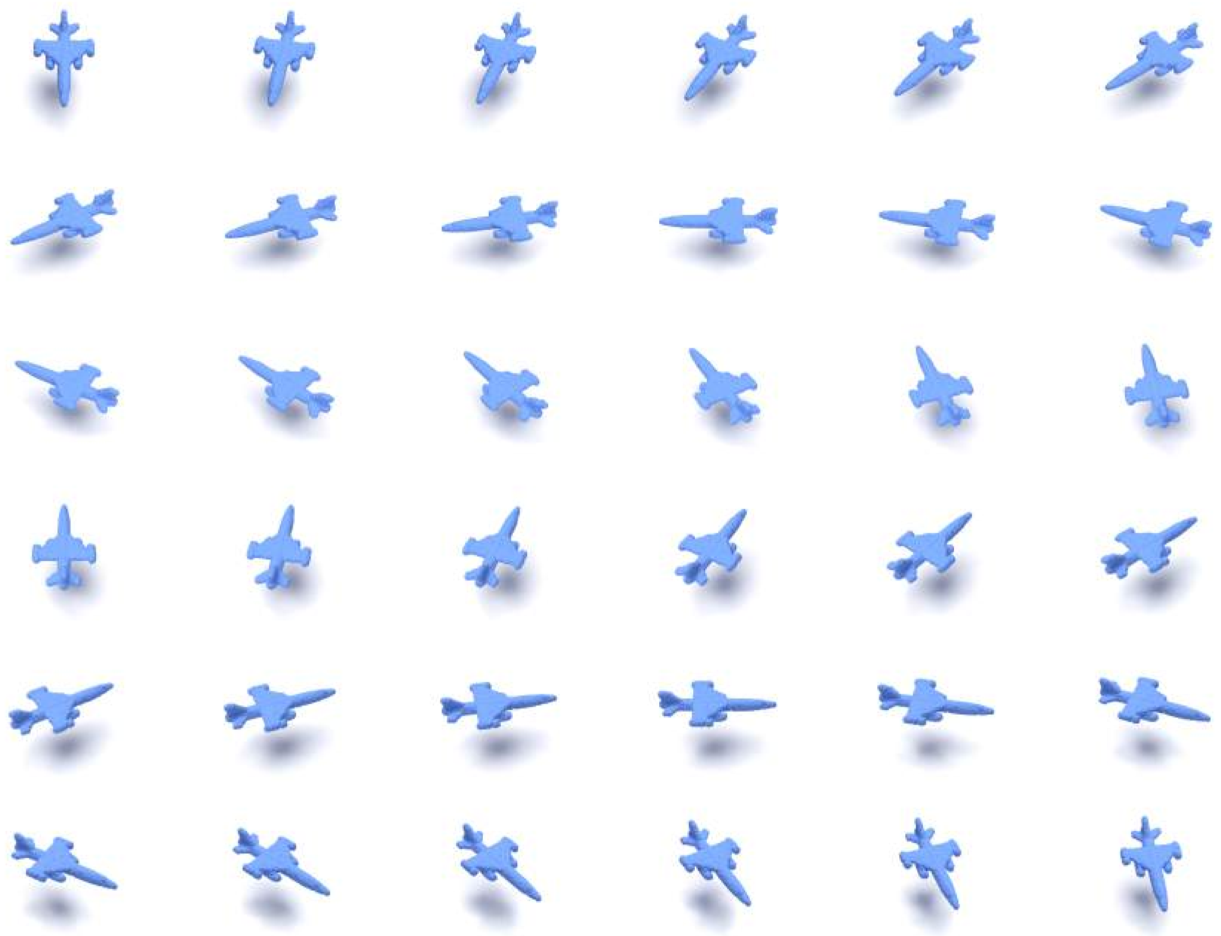


Figure 4. Airplane shape from different angles. All shapes are from just one point cloud.

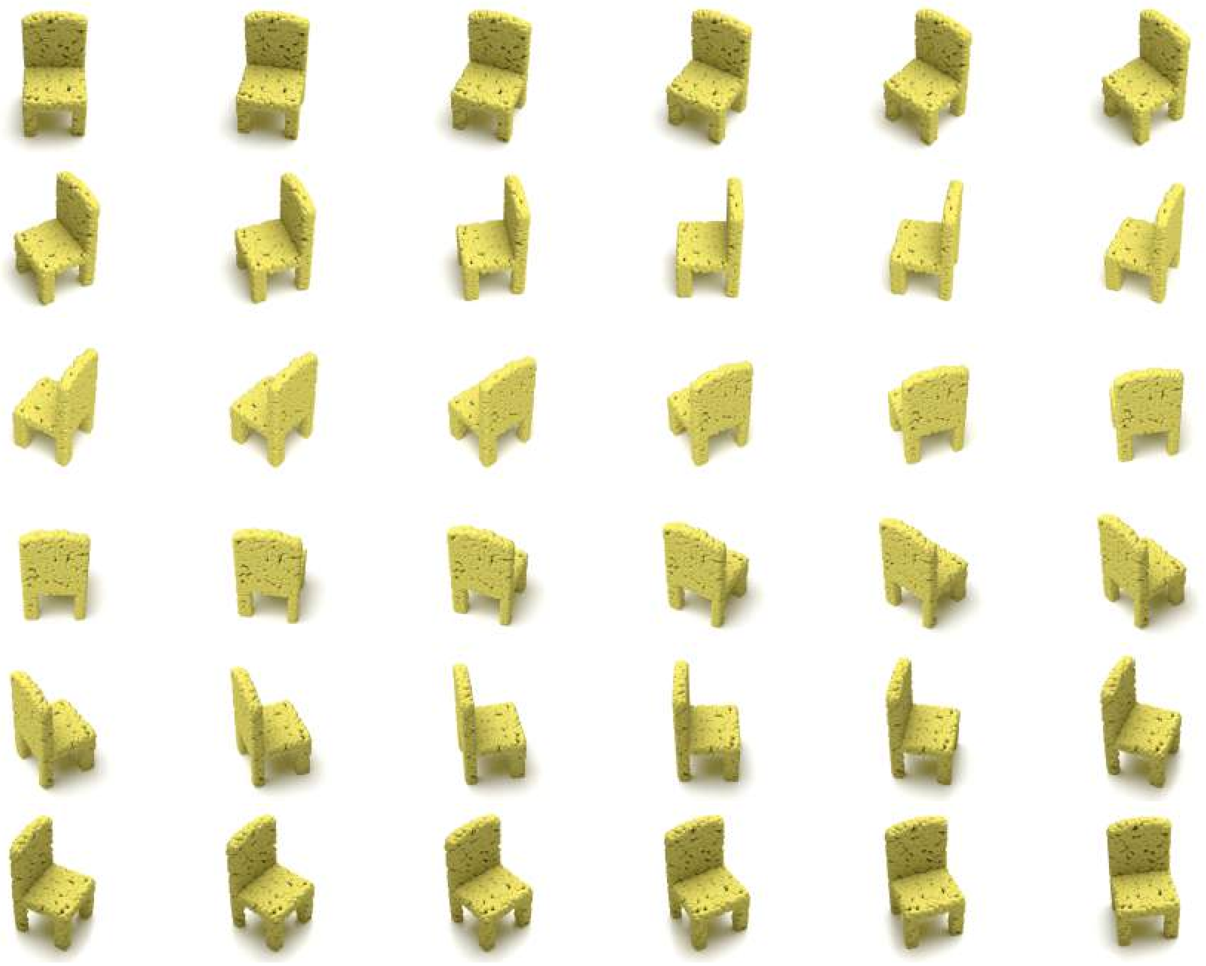


Figure 5. Chair shape from different angles. All shapes are from just one point cloud.

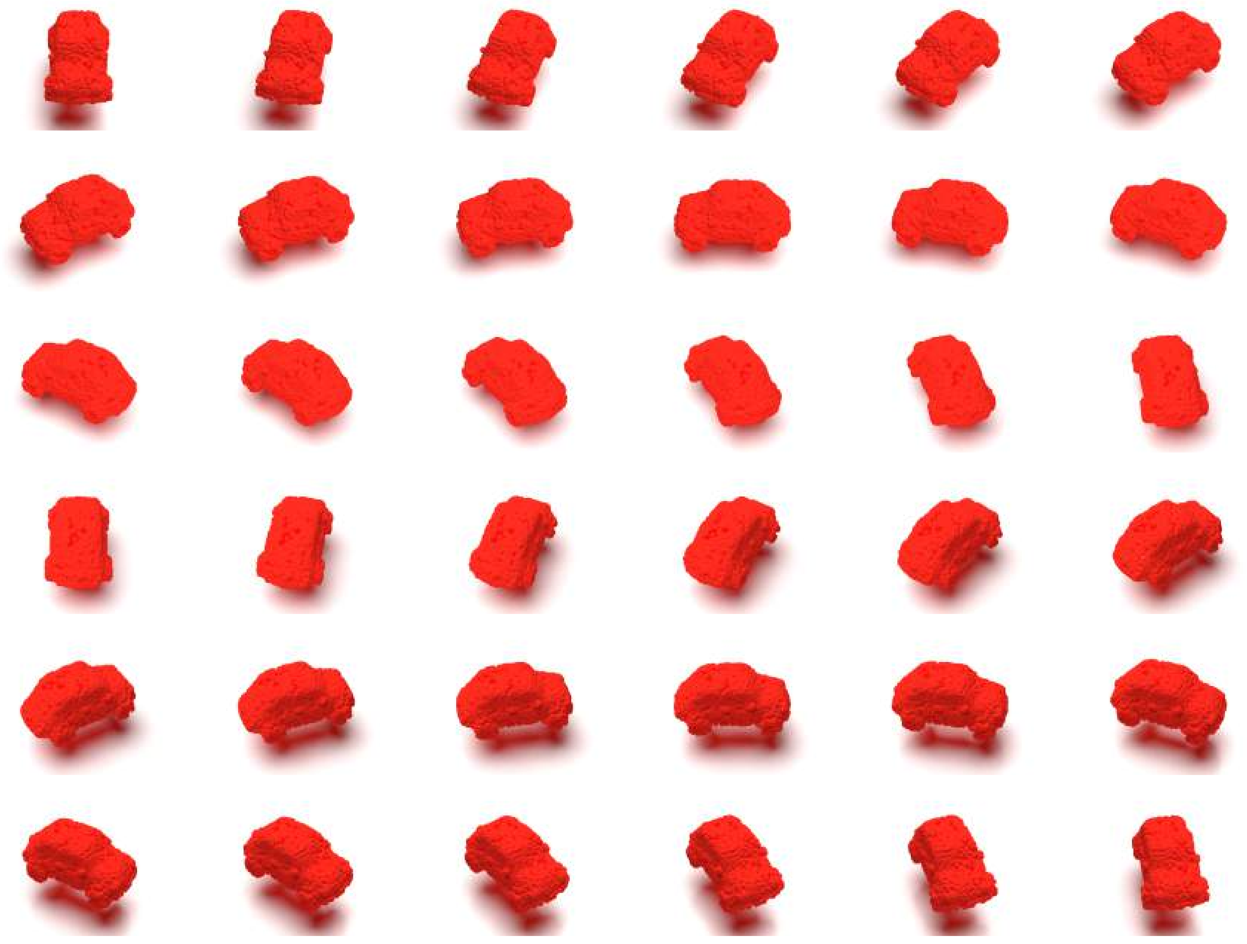


Figure 6. Car shape from different angles. All shapes are from just one point cloud.

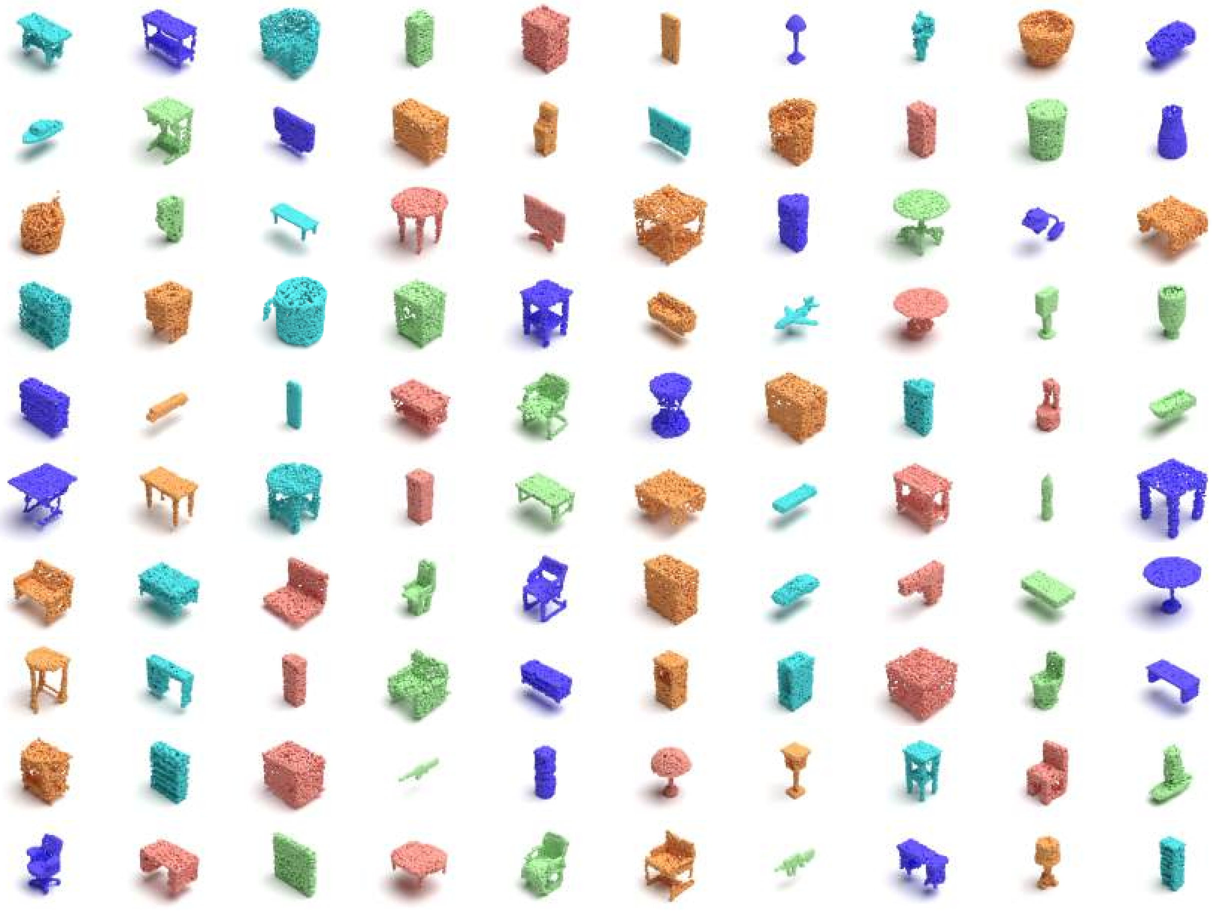


Figure 7. Visualization of our 55-class model.