Appendix: 3D Shape Generation and Completion through Point-Voxel Diffusion

A. Additional Generation Metrics

We present additional generation metrics in Table 1, following PointFlow [9]. We report coverage (COV), which measures the fraction of point clouds in the reference set that are matched to at least one point cloud in the generated set. We further report minimum matching distance (MMD), which measures for each point cloud in a reference set, the distance to its nearest neighbor in the generated set. Note that these generation metrics can vary depending on implementation and do not necessarily correlate to generation quality, as discussed in [9].

Table 2 includes generation results on Airplane, Chair, Car compared with the voxel-diffusion model, Vox-Diff, as described in the main paper, evaluated using the 1-NN metric. By generating less noisy point clouds, PVD significantly outperforms Vox-Diff.

B. Point Cloud Generation Visualization

We additionally visualize some generation results for Airplane, Car, and Chair in terms of the generation process and the final generated shapes from all angles in Figures 3, 4, 5, 6, 7, 8, 9, and 10.

C. Derivation of the Variational Lower Bound

\[
E_q(x_0) \log p(x_0) = E_q(x_0) \left[ \log \int p(x_0, \ldots, x_T) \, dx_{1:T} \right] 
\geq E_q(x_0) \left[ \int q(x_1, \ldots, x_T|x_0) \log \frac{p(x_0, \ldots, x_T)}{q(x_1, \ldots, x_T|x_0)} \, dx_{1:T} \right] 
= E_q(x_0) \left[ \log \frac{p(x_0, \ldots, x_T)}{q(x_1, \ldots, x_T|x_0)} \right],
\]

where the inequality is by Jensen’s inequality.

D. Properties of the Diffusion Model

\{\beta_0, \ldots, \beta_T\} is a sequence of increasing parameters; \(\alpha_t = 1 - \beta_t\) and \(\hat{\alpha}_t = \prod_{s=1}^{t} \alpha_s\). Two following properties are crucial to deriving the final \(L_2\) loss.

Property 1. Tractable marginal of the forward process:

\[
q(x_t|x_0) = \int q(x_{t-1}|x_0) \, dx_{1:t-1} = \mathcal{N}(\sqrt{\hat{\alpha}_t}x_0, (1 - \hat{\alpha}_t)I).
\]

This property is proved in the Appendix of [4] and provides convenient closed-form evaluation of \(x_t\) knowing \(x_0\):

\[
x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, \tag{1}
\]

where \(\epsilon \sim \mathcal{N}(0, I)\).

Property 2. Tractable posterior of the forward process. We first note the Bayes’ rule that connects the posterior with the forward process,

\[
q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}.
\]

Since the three probabilities on the right are Gaussian, the posterior is also Gaussian, given by

\[
q(x_{t-1}|x_t, x_0) = \mathcal{N}(\frac{\sqrt{\hat{\alpha}_{t-1}\beta_t}x_0 + \sqrt{\hat{\alpha}_t(1 - \hat{\alpha}_{t-1})}x_t}{1 - \hat{\alpha}_t}(1 - \hat{\alpha}_{t-1})\beta_tI). \tag{2}
\]

E. Derivation of \(L_2\) Loss

We need to match generative transition \(p_\theta(x_{t-1}|x_t)\) with ground-truth posterior \(q(x_{t-1}|x_t, x_0)\), both of which are Gaussian with a pre-determined variance schedule \(\beta_1, \ldots, \beta_T\). Therefore, maximum likelihood learning is reduced to simple \(L_2\) loss of the form with two cases:

\[
L_t = \begin{cases} 
\left\| \frac{1}{\sqrt{\alpha_t}}(x_0 - \frac{\beta_t}{\sqrt{1 - \hat{\alpha}_t}}\epsilon) - \mu_\theta(x_t, t) \right\|^2, & t > 1 \\
\left\| x_0 - \mu_\theta(x_t, t) \right\|^2, & t = 1 
\end{cases}
\]

where \(\alpha_t = 1 - \beta_t\) and \(\hat{\alpha}_t = \prod_{s=1}^{t} \alpha_s\). The supervision target of case \(t > 1\) comes from Eqn. 2. We can further reduce the case when \(t > 1\) by substituting \(x_0\) as an expression of \(x_t\) using Eqn. 1 and arrive at

\[
L_t = \begin{cases} 
\left\| \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1 - \hat{\alpha}_t}}\epsilon) - \mu_\theta(x_t, t) \right\|^2, & t > 1 \\
\left\| x_0 - \mu_\theta(x_t, t) \right\|^2, & t = 1 
\end{cases}
\]

where \(\epsilon \sim \mathcal{N}(0, I)\).

Note that when \(t = 1\), \(\hat{\alpha}_1 = \alpha_1\) so that the supervision target of the first case above evaluated at \(t = 1\) becomes:

\[
\frac{1}{\sqrt{\alpha_1}}(x_1 - \frac{\beta_1}{\sqrt{1 - \alpha_1}}\epsilon) = \frac{1}{\sqrt{\alpha_1}}(x_1 - \sqrt{1 - \alpha_1}\epsilon) = x_0, \tag{3}
\]

where the last equality is by rewriting Eqn. 1. Therefore, in fact, the two cases are equivalent.

The final \(L_2\) loss is

\[
L_t = \left\| \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1 - \hat{\alpha}_t}}\epsilon) - \mu_\theta(x_t, t) \right\|^2.
\]
the following form: since it similarly adds scaled noise outputs from the model. Langevin dynamics \[3, 8\] used in energy-based models, 
\[z \sim \mathcal{N}(0, I)\]. Both processes are Markovian, shifting the previous output by a model-dependent term and a noise term. The scaled model output of our model can also be seen as an approximation of gradients of an energy function.

\[\nabla \log p_\theta(x)\] 

F. Point Cloud Generation Process

Since the transition mean \(\mu(x_t, t)\) of \(p_\theta(x_{t-1}|x_t)\) is calculated by Eqn. 4, the generative process is performed by progressively sampling from \(p_\theta(x_{t-1}|x_t)\) as \(t = \cdots, 1\): 
\[
x_{t-1} = \frac{1}{\alpha_t} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right) + \sqrt{\beta_t} z_t, \tag{6}
\]
where \(z \sim \mathcal{N}(0, I)\). This approach is also similar to Langevin dynamics \[3, 8\] used in energy-based models, since it similarly adds scaled noise outputs from the model to current samples. Specifically, Langevin dynamics is in the following form:
H. Training Details

H.1. Model Architecture

Same as in [7], our point-voxel CNN architecture is modified from PointNet++, where we replace the PointNet substructure with point-voxel convolution, as shown in Figure 2. We specify our architecture in Table 3, Table 4, and Table 5. Table 3 shows details of a single set abstraction (SA) module. Table 4 shows details of a single feature propagation (FP) module. Table 5 shows how these modules are combined together.

In particular, we concatenate the temporal embeddings with point features before sending input into the Set Abstraction or the Feature Propagation modules. To obtain temporal embeddings, we used a sinusoidal positional embedding, commonly used in Transformers. Given a time \( t \) and an embedding dimension \( d \), the time embedding consists of pairs of \( \sin \) and \( \cos \) with varying frequencies, \((\sin (\omega_1 t), \cos (\omega_1 t), \ldots, \sin (\omega_d/2 t), \cos (\omega_d/2 t))\), where \( \omega_k = 1 / (10000^{2k/d}). \)

We use the same architecture for both generation and completion tasks. For shape completion specifically, the model takes as input a 200-point partial shape and 1,848 points sampled from noise, totaling 2048 points. At each step, the first 200 of the 2,048 points sampled by the model are replaced with the input partial shape. The updated point set is then used as input in the next time step.

H.2. Choices of \( \beta_t \) and \( T \)

For both hyper-parameters, we follow [4]. For Car and Chair, we set \( \beta_0 = 10^{-4}, \beta_T = 0.01 \) and linearly interpolate other \( \beta \)'s. For Airplane, we interpolate between \( \beta_0 = 10^{-5} \) and \( \beta_T = 0.008 \) for the first 90% steps and then fix \( \beta_T = 0.008 \). We also set \( T = 1000 \) for all experiments and we generally notice that lower timesteps (e.g., 100) are not enough for the model to construct shapes.

H.3. Training Parameters

We use Adam optimizer with learning rate \( 2 \times 10^{-4} \) for all experiments.

References

Figure 2: Model architecture diagram.

<table>
<thead>
<tr>
<th>Set Abstraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Feature Size: $N_{input} \times C_{input}$</td>
</tr>
<tr>
<td>Input Time Embedding Size: $N_{input} \times E_t$</td>
</tr>
<tr>
<td>Output Feature Size: $N_{output} \times C_{output}$</td>
</tr>
<tr>
<td>Voxelization Resolution: $D$</td>
</tr>
<tr>
<td>Number of Point-Voxel Convolution (PVConv) blocks: $L$</td>
</tr>
<tr>
<td>Whether to use attention mechanism: use_attn</td>
</tr>
</tbody>
</table>

Table 3: Set Abstraction Layer. Input is first fed through $L$ PVConv modules, then to an MLP module, and finally through the Sampling & Grouping module.
### Feature Propagation

<table>
<thead>
<tr>
<th>Input Feature Size: $N_{\text{input}} \times C_{\text{input}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Feature Size: $N_{\text{output}} \times C_{\text{output}}$</td>
</tr>
<tr>
<td>Voxelization Resolution: $D$</td>
</tr>
<tr>
<td>Number of Point-Voxel Convolution (PVConv) blocks: $L$</td>
</tr>
<tr>
<td>Whether to use attention mechanism: use_attn</td>
</tr>
</tbody>
</table>

### Interpolation

**PVConv $\times L$**

<table>
<thead>
<tr>
<th>Layers</th>
<th>In-Out Size</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: $X$</td>
<td>$D \times D \times D \times C_{\text{input}}$</td>
<td>1</td>
</tr>
<tr>
<td>3x3x3 conv($C_{\text{output}}$), GroupNorm(8), Swish</td>
<td>$D \times D \times D \times C_{\text{output}}$</td>
<td>1</td>
</tr>
<tr>
<td>Dropout(0.1)</td>
<td>$D \times D \times D \times C_{\text{output}}$</td>
<td></td>
</tr>
<tr>
<td>3x3x3 conv($C_{\text{output}}$), GroupNorm(8)</td>
<td>$D \times D \times D \times C_{\text{output}}$</td>
<td></td>
</tr>
<tr>
<td>Attention(use_attn)</td>
<td>$D \times D \times D \times C_{\text{output}}$</td>
<td></td>
</tr>
</tbody>
</table>

**MLP**

<table>
<thead>
<tr>
<th>Layers</th>
<th>In-Out Size</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3x3 conv($C_{\text{output}}$), GroupNorm(8), Swish</td>
<td>$D \times D \times D \times C_{\text{output}}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: Feature Propagation Layer. Input is fed through Interpolation module, $L$ PVConv modules, and an MLP module.

### Time Embedding

<table>
<thead>
<tr>
<th>Layers</th>
<th>In-Out Size</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sinusoidal Embedding dim = 64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP(64, 64)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LeakyRelU(0.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP(64, 64)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Entire point-voxel CNN architecture. Input point clouds and time steps are sequentially passed through SA 1-4, FP 1-4, and an MLP to obtain output of the same dimension. At the start of each SA and FP module, time embedding and point features are first concatenated.
Figure 3: Airplane generation process.
Figure 4: Airplane results from all angles.
Figure 5: Car generation process.
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Car Results" /></td>
<td><img src="image2" alt="Car Results" /></td>
<td><img src="image3" alt="Car Results" /></td>
<td><img src="image4" alt="Car Results" /></td>
<td><img src="image5" alt="Car Results" /></td>
<td><img src="image6" alt="Car Results" /></td>
<td><img src="image7" alt="Car Results" /></td>
</tr>
<tr>
<td><img src="image8" alt="Car Results" /></td>
<td><img src="image9" alt="Car Results" /></td>
<td><img src="image10" alt="Car Results" /></td>
<td><img src="image11" alt="Car Results" /></td>
<td><img src="image12" alt="Car Results" /></td>
<td><img src="image13" alt="Car Results" /></td>
<td><img src="image14" alt="Car Results" /></td>
</tr>
<tr>
<td><img src="image15" alt="Car Results" /></td>
<td><img src="image16" alt="Car Results" /></td>
<td><img src="image17" alt="Car Results" /></td>
<td><img src="image18" alt="Car Results" /></td>
<td><img src="image19" alt="Car Results" /></td>
<td><img src="image20" alt="Car Results" /></td>
<td><img src="image21" alt="Car Results" /></td>
</tr>
<tr>
<td><img src="image22" alt="Car Results" /></td>
<td><img src="image23" alt="Car Results" /></td>
<td><img src="image24" alt="Car Results" /></td>
<td><img src="image25" alt="Car Results" /></td>
<td><img src="image26" alt="Car Results" /></td>
<td><img src="image27" alt="Car Results" /></td>
<td><img src="image28" alt="Car Results" /></td>
</tr>
<tr>
<td><img src="image29" alt="Car Results" /></td>
<td><img src="image30" alt="Car Results" /></td>
<td><img src="image31" alt="Car Results" /></td>
<td><img src="image32" alt="Car Results" /></td>
<td><img src="image33" alt="Car Results" /></td>
<td><img src="image34" alt="Car Results" /></td>
<td><img src="image35" alt="Car Results" /></td>
</tr>
</tbody>
</table>

Figure 6: Car results from all angles.
Figure 7: Chair generation process.
Figure 8: Chair results from all angles.
Figure 9: Chair generation process.
Figure 10: Chair results from all angles.