# GPLD3D: Latent Diffusion of 3D Shape Generative Models by Enforcing Geometric and Physical Priors

Supplementary Material

## A. Proof of Theorem 1

As any probability distribution can be approximated by a mixture of Gaussian with a prescribed approximation error, it is sufficient to prove Theorem 1 in the setting where $p_{gt}$ is given a mixture of Gaussians.

**Theorem 3** *(**Formal statement of Theorem 1**) Consider a mixture of Gaussians with $K$ mixture components, where each component $(w_i, \boldsymbol{m}_i, \Sigma_i)$ is given by its mixture weight $w_i$, the mean $\boldsymbol{m}_i$, and the covariance matrix $\Sigma_i$:*

$$p_{gt}(\boldsymbol{z}) := \sum_{i=1}^{K} \frac{w_i}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\big(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m}_i)^T\Sigma^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)\big). \tag{9}$$

*Let $p_{\text{data}}$ be the empirical distribution defined by $n$ random samples from $p_{gt}(\boldsymbol{z})$. Then*

$$\mathop{\mathbb{E}}_{p_{\text{data}}} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{gt}} \|\nabla_{\boldsymbol{z}} p_{\text{data}}(\boldsymbol{z},\sigma) - \nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z},\sigma)\|^2 = O(\frac{1}{\sigma^4}). \tag{10}$$

*Proof of Theorem 3.* Note that

$$p_{gt}(\boldsymbol{z},\sigma) := \sum_{i=1}^{K} \frac{w_i \exp\big(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m}_i)^T\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)\big)}{(2\pi)^{\frac{d}{2}}|\Sigma_i(\sigma)|^{\frac{1}{2}}},$$

where

$$\Sigma_i(\sigma) := \Sigma_i + \sigma^2 \cdot I_d$$

Denote $p_{\text{data}} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n\}$. Then

$$p_{\text{data}}(\boldsymbol{z},\sigma) := \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(2\pi)^{\frac{d}{2}}\sigma^d} \exp(-\frac{\|\boldsymbol{z}\|^2}{2\sigma^2}).$$

It follows that

$$\nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z},\sigma)) = -\sum_{i=1}^{K} \frac{w_i\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)}{(2\pi)^{\frac{d}{2}} p_{gt}(\boldsymbol{z},\sigma)|\Sigma_i(\sigma)|^{\frac{1}{2}}} \tag{11}$$

$$\nabla_{\boldsymbol{z}} \log(p_{\text{data}}(\boldsymbol{z},\sigma)) = -\frac{\sum_{i=1}^{n} \exp\big(-\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2}\big)(\boldsymbol{z}-\boldsymbol{z}_i)}{\sigma^2 \sum_{i=1}^{n} \exp\big(-\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2}\big)} \tag{12}$$

To estimate the order of (10) in $\sigma$, we consider the regime of $\sigma$ where $\sigma \gg \max_{1\le i\le K}(\|\boldsymbol{m}_i\|, \|\Sigma_i\|)$. In this regime, we have when $\boldsymbol{z} \sim p_{gt}$,

$$p_{gt}(\boldsymbol{z},\sigma) \approx \sum_{i=1}^{K} \frac{w_i\big(1-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m}_i)^T\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)\big)}{(2\pi)^{\frac{d}{2}}|\Sigma_i(\sigma)|^{\frac{1}{2}}} \tag{13}$$

$$\exp(-\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2}) \approx 1 - \frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2} \tag{14}$$

Substituting (13) into (11), we have

$$\begin{aligned}
&\nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z},\sigma)) \\
&\approx -\frac{\sum_{i=1}^{K} \frac{w_i}{|\Sigma_i(\sigma)|^{\frac{1}{2}}} + \frac{1}{2}\sum_{i=1}^{K} \frac{w_i(\boldsymbol{z}-\boldsymbol{m}_i)^T\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)}{|\Sigma_i(\sigma)|^{\frac{1}{2}}}}{\Big(\sum_{i=1}^{K} \frac{w_i}{|\Sigma_i(\sigma)|^{\frac{1}{2}}}\Big)^2} \\
&\cdot \sum_{i=1}^{K} \frac{w_i\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)}{|\Sigma_i(\sigma)|^{\frac{1}{2}}}
\end{aligned} \tag{15}$$

In the large $\sigma$ regime, we have

$$\|\Sigma_i(\sigma)^{-2}\| = O(\frac{1}{\sigma^2}).$$

Therefore,

$$\nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z},\sigma)) \approx -\frac{\sum_{i=1}^{K} \frac{w_i\Sigma_i(\sigma)^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)}{|\Sigma_i(\sigma)|^{\frac{1}{2}}}}{\sum_{i=1}^{K} \frac{w_i}{|\Sigma_i(\sigma)|^{\frac{1}{2}}}} \tag{16}$$

Similarly, substituting (14) into (12), we have

$$\begin{aligned}
&\nabla_{\boldsymbol{z}} \log(p_{\text{data}}(\boldsymbol{z},\sigma)) \\
&\approx -\frac{1}{\sigma^2 n^2}\sum_{i,j=1}^{n} (1-\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2})(1+\frac{\|\boldsymbol{z}-\boldsymbol{z}_j\|^2}{2\sigma^2})(\boldsymbol{z}-\boldsymbol{z}_i) \\
&\approx -\frac{1}{\sigma^2 n}\sum_{i=1}^{n} (1-\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2}+\frac{\|\boldsymbol{z}-\boldsymbol{z}_i\|^2}{2\sigma^2})(\boldsymbol{z}-\boldsymbol{z}_i) \\
&= -\frac{\boldsymbol{z}}{\sigma^2} + \frac{1}{\sigma^2 n}\sum_{i=1}^{n} \boldsymbol{z}_i
\end{aligned} \tag{17}$$

Combing (16) and (17), we have

$$\|\nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z},\sigma)) - \nabla_{\boldsymbol{z}} \log(p_{\text{data}}(\boldsymbol{z},\sigma))\| = O(\frac{1}{\sigma^4}) \tag{18}$$

□

## B. Proof of Theorem 2

We then quantify the difference between $\nabla_{\boldsymbol{z}} \log(p_{\text{reg}}(\boldsymbol{z};\sigma))$ and $\nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z};\sigma))$. Suppose we have the following.

Note that

$$\begin{aligned}&\nabla_{\boldsymbol{z}} \log(p_{\text{reg}}(\boldsymbol{z};\sigma))\\ =&\frac{1}{p_{\text{reg}}(\boldsymbol{z};\sigma)}\nabla_{\boldsymbol{z}} p_{\text{reg}}(\boldsymbol{z};\sigma)\\ =&\frac{\int_{\boldsymbol{y}} \nabla_{\boldsymbol{z}} p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})\text{exp}\big(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2}\big)d\boldsymbol{y}}{\int_{\boldsymbol{y}} p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})\text{exp}\big(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2}\big)d\boldsymbol{y}}\end{aligned} \tag{19}$$

Introduce

$$\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y}) = \frac{\nabla_{\boldsymbol{z}} p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})}{p_{\text{reg}}(\boldsymbol{z})} - \frac{\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y})}{p_{gt}(\boldsymbol{z})}, \tag{20}$$

$$e_2(\boldsymbol{z},\boldsymbol{y}) = \frac{p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})}{p_{\text{reg}}(\boldsymbol{z})} - \frac{p_{gt}(\boldsymbol{z}-\boldsymbol{y})}{p_{gt}(\boldsymbol{z})}. \tag{21}$$

Applying Taylor series, we obtain

$$\begin{aligned}\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y}) = \sum_{k=0}^{+\infty}\frac{(-1)^k}{k!}\Big(&\frac{1}{p_{\text{reg}}(\boldsymbol{z})}\langle\frac{\partial^{k+1}p_{\text{reg}}(\boldsymbol{z})}{\partial^{k+1}\boldsymbol{z}}, \otimes^k\boldsymbol{y}\rangle\\ &- \frac{1}{p_{gt}(\boldsymbol{z})}\langle\frac{\partial^{k+1}p_{gt}(\boldsymbol{z})}{\partial^{k}\boldsymbol{z}}, \otimes^{k+1}\boldsymbol{y}\rangle\Big),\end{aligned}$$

and

$$\begin{aligned}e_2(\boldsymbol{z},\boldsymbol{y}) = \sum_{k=1}^{+\infty}\frac{(-1)^k}{k!}\Big(&\frac{1}{p_{\text{reg}}(\boldsymbol{z})}\langle\frac{\partial^{k}p_{\text{reg}}(\boldsymbol{z})}{\partial^{k}\boldsymbol{z}}, \otimes^k\boldsymbol{y}\rangle\\ &- \frac{1}{p_{gt}(\boldsymbol{z})}\langle\frac{\partial^{k}p_{gt}(\boldsymbol{z})}{\partial^{k}\boldsymbol{z}}, \otimes^{k}\boldsymbol{y}\rangle\Big).\end{aligned}$$

Define

$$c_k(\boldsymbol{z}) := \|\frac{\frac{\partial^k p_{\text{reg}}(\boldsymbol{z})}{\partial^k \boldsymbol{z}}}{p_{\text{reg}}(\boldsymbol{z})} - \frac{\frac{\partial^k p_{gt}(\boldsymbol{z})}{\partial^k \boldsymbol{z}}}{p_{gt}(\boldsymbol{z})}\|.$$

It follows that

$$\|\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y})\| \le \sum_{k=0}^{+\infty}\frac{c_{k+1}(\boldsymbol{z})\|\boldsymbol{y}\|^k}{k!}, \tag{22}$$

and

$$|e_2(\boldsymbol{z},\boldsymbol{y})| \le \sum_{k=1}^{+\infty}\frac{c_k(\boldsymbol{z})\|\boldsymbol{y}\|^k}{k!}, \tag{23}$$

Rearranging (20) and (21), we arrive at

$$\begin{aligned}&\nabla_{\boldsymbol{z}} p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})\\ =&\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y})p_{\text{reg}}(\boldsymbol{z}) + \frac{p_{\text{reg}}(\boldsymbol{z})}{p_{gt}(\boldsymbol{z})}\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y}),\end{aligned} \tag{24}$$

and

$$\begin{aligned}&p_{\text{reg}}(\boldsymbol{z}-\boldsymbol{y})\\ =&\boldsymbol{e}_2(\boldsymbol{z},\boldsymbol{y})p_{\text{reg}}(\boldsymbol{z}) + \frac{p_{\text{reg}}(\boldsymbol{z})}{p_{gt}(\boldsymbol{z})}p_{gt}(\boldsymbol{z}-\boldsymbol{y}),\end{aligned} \tag{25}$$

Substituting (25) and (24) into (19), we obtain

$$\begin{aligned}&\nabla_{\boldsymbol{z}} \log(p_{\text{reg}}(\boldsymbol{z};\sigma))\\ =&\frac{\int_{\boldsymbol{y}} \big(\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y}) + p_{gt}(\boldsymbol{z})\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y})\big)\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}{\int_{\boldsymbol{y}} \big(p_{gt}(\boldsymbol{z}-\boldsymbol{y}) + p_{gt}(\boldsymbol{z})e_2(\boldsymbol{z},\boldsymbol{y})\big)\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}\end{aligned} \tag{26}$$

Introduce

$$\delta_k = \frac{1}{\sigma^{d+k}}\int_{\boldsymbol{y}\in\mathbb{R}^d}\|\boldsymbol{y}\|^k\text{exp}\big(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2}\big)d\boldsymbol{y}.$$

It is easy to check that

$$\delta_k = \int_{\boldsymbol{y}\in\mathbb{R}^d}\|\boldsymbol{y}\|^k\text{exp}\big(-\frac{\|\boldsymbol{y}\|^2}{2}\big)d\boldsymbol{y}$$

where $\Gamma$ is the Chi-distribution. Let

$$\boldsymbol{c}_g(\boldsymbol{z},\sigma) := \frac{1}{\sigma^d}\int_{\boldsymbol{y}}\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y})\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y} \tag{27}$$

$$c(\boldsymbol{z},\sigma) := \frac{1}{\sigma^d}\int_{\boldsymbol{y}} p_{gt}(\boldsymbol{z}-\boldsymbol{y})\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y} \tag{28}$$

Combing (22), (23), (27), (28), and (26), we have that in the small regime $\sigma$,

$$\begin{aligned}&\|\nabla_{\boldsymbol{z}} \log(p_{\text{reg}}(\boldsymbol{z};\sigma)) - \nabla_{\boldsymbol{z}} \log(p_{gt}(\boldsymbol{z};\sigma))\|\\ \le&\|\frac{\int_{\boldsymbol{y}} \big(\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y}) + p_{gt}(\boldsymbol{z})\boldsymbol{e}_1(\boldsymbol{z},\boldsymbol{y})\big)\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}{\int_{\boldsymbol{y}} \big(p_{gt}(\boldsymbol{z}-\boldsymbol{y}) + p_{gt}(\boldsymbol{z})e_2(\boldsymbol{z},\boldsymbol{y})\big)\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}\\ &- \frac{\int_{\boldsymbol{y}}\nabla_{\boldsymbol{z}} p_{gt}(\boldsymbol{z}-\boldsymbol{y})\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}{\int_{\boldsymbol{y}} p_{gt}(\boldsymbol{z}-\boldsymbol{y})\text{exp}(-\frac{\|\boldsymbol{y}\|^2}{2\sigma^2})d\boldsymbol{y}}\|\\ \le&p_{gt}(\boldsymbol{z})\frac{c(\boldsymbol{z},\sigma)c_1(\boldsymbol{z})\delta_0 + \sigma\big(\epsilon_2 c(\boldsymbol{z},\sigma) + \epsilon_1\|\boldsymbol{c}_g(\boldsymbol{z},\sigma)\|\big)}{c(\boldsymbol{z},\sigma)\big(c(\boldsymbol{z},\sigma) - p_{gt}(\boldsymbol{z})\sigma\epsilon_1\big)}\end{aligned} \tag{29}$$

where

$$\epsilon_1 := \sum_{k=1}^{\infty}\frac{c_k(\boldsymbol{z})}{k!}\sigma^{k-1}\delta_k$$

$$\epsilon_2 := \sum_{k=1}^{\infty}\frac{c_{k+1}(\boldsymbol{z})}{k!}\sigma^{k-1}\delta_k$$

□

## C. Details on the Optimal Hyper-Parameters

We adopt three principles from [56] to determine $\lambda_{\text{in}}(\sigma)$, $\lambda_{\text{out}}(\sigma)$, and $\lambda_{\text{skip}}(\sigma)$. First, we want the variance of the weighted inputs to be 1, i.e.,

$$\underset{\boldsymbol{z}\sim p_{\text{data}},\boldsymbol{n}}{\text{Var}}\big(c_{\text{in}}(\sigma)(\boldsymbol{z}+\boldsymbol{n})\big) + \frac{\lambda_{\text{reg}}(\sigma)}{\lambda_{\text{data}}(\sigma)}\underset{\boldsymbol{z}\sim p_{\text{reg}},\boldsymbol{n}}{\text{Var}}\big(c_{\text{in}}(\sigma)(\boldsymbol{y}+\boldsymbol{n})\big) = 1 \tag{30}$$

Note that

$$\underset{\boldsymbol{z}\sim p_{\text{data}},\boldsymbol{n}}{\text{Var}}\big(c_{\text{in}}(\sigma)(\boldsymbol{z}+\boldsymbol{n}) = c_{\text{in}}(\sigma)^2(\sigma_{\text{data}}^2+\sigma^2) \tag{31}$$

$$\underset{\boldsymbol{z}\sim p_{\text{reg}},\boldsymbol{n}}{\text{Var}}\big(c_{\text{in}}(\sigma)(\boldsymbol{y}+\boldsymbol{n}) = c_{\text{in}}(\sigma)^2(\sigma_{\text{reg}}^2+\sigma^2) \tag{32}$$

Substituting (31) and (32) into (30), we obtain

$$c_{\text{in}}(\sigma) = \frac{1}{\sqrt{\sigma_{\text{data}}^2+\sigma^2+\frac{c_{\text{ratio}}(c_{\text{reg}}^2+\sigma^2)}{\sigma^2(c_{\text{off}}+\sigma)^2}}} \tag{33}$$

Second, we constrain that the weighted fitting targets to have unit variance, i.e.,

$$\frac{1}{c_{\text{out}}(\sigma)^2}\Big(\underset{\boldsymbol{z}\sim p_{\text{data}},\boldsymbol{n}}{\text{Var}}\big(\boldsymbol{z}-c_{\text{skip}}(\sigma)(\boldsymbol{z}+\boldsymbol{n})\big) + \frac{\lambda_{\text{reg}}(\sigma)}{\lambda_{\text{data}}(\sigma)}\underset{\boldsymbol{z}\sim p_{\text{reg}},\boldsymbol{n}}{\text{Var}}\big(\boldsymbol{z}-c_{\text{skip}}(\sigma)(\boldsymbol{z}+\boldsymbol{n})\big)\Big) = 1 \tag{34}$$

Note that

$$\underset{\boldsymbol{z}\sim p_{\text{data}},\boldsymbol{n}}{\text{Var}}\big((\boldsymbol{z}-c_{\text{skip}}(\sigma)(\boldsymbol{z}+\boldsymbol{n})\big) = (1-c_{\text{skip}}(\sigma))^2\sigma_{\text{data}}^2 + c_{\text{skip}}(\sigma)^2\sigma^2 \tag{35}$$

$$\underset{\boldsymbol{z}\sim p_{\text{reg}},\boldsymbol{n}}{\text{Var}}\big(\boldsymbol{z}-c_{\text{skip}}(\sigma)(\boldsymbol{z}+\boldsymbol{n})\big) = (1-c_{\text{skip}}(\sigma))^2\sigma_{\text{reg}}^2 + c_{\text{skip}}(\sigma)^2\sigma^2 \tag{36}$$

Substituting (35) and (36) into (34), we obtain

$$c_{\text{out}}^2(\sigma) = (1-c_{\text{skip}}(\sigma))^2\big(\sigma_{\text{data}}^2+\frac{c_{\text{ratio}}\sigma_{\text{reg}}^2}{\sigma^2(c_{\text{off}}+\sigma)^2}\big) + c_{\text{skip}}(\sigma)^2\big(\sigma^2+\frac{c_{\text{ratio}}}{(c_{\text{off}}+\sigma)^2}\big) \tag{37}$$

Third, we select $c_{\text{skip}}(\sigma)$ to minimize $c_{\text{out}}(\sigma)$, i.e.,

$$c_{\text{skip}}(\sigma) = \underset{c_{\text{skip}}(\sigma)}{\text{argmin}} \quad c_{\text{out}}(\sigma)^2.$$

This leads to

$$c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}^2+\frac{c_{\text{ratio}}\sigma_{\text{reg}}^2}{\sigma^2(c_{\text{off}}+\sigma)^2}}{\sigma_{\text{data}}^2+\frac{c_{\text{ratio}}\sigma_{\text{reg}}^2}{\sigma^2(c_{\text{off}}+\sigma)^2}+\sigma^2+\frac{c_{\text{ratio}}}{(c_{\text{off}}+\sigma)^2}} \tag{38}$$

Substituting (38) into (37), we arrive at

$$c_{\text{out}}(\sigma) = \sqrt{\frac{\big(\sigma_{\text{data}}^2+\frac{c_{\text{ratio}}\sigma_{\text{reg}}^2}{\sigma^2(c_{\text{off}}+\sigma)^2}\big)\big(\sigma^2+\frac{c_{\text{ratio}}}{(c_{\text{off}}+\sigma)^2}\big)}{\sigma_{\text{data}}^2+\sigma^2+\frac{c_{\text{ratio}}(\sigma_{\text{reg}}^2+\sigma^2)}{\sigma^2(c_{\text{off}}+\sigma)^2}}} \tag{39}$$

Now we determine $\lambda_{\text{data}}(\sigma)$ and $\lambda_{\text{reg}}(\sigma)$. Note that their relative values are determined as

$$\frac{\lambda_{\text{reg}}(\sigma)}{\lambda_{\text{data}}(\sigma)} = \frac{c_{\text{ratio}}}{\sigma^4(c_{\text{off}}+\sigma)^2}.$$

Therefore, it remains to determine the relative values of $\lambda_{\text{data}}(\sigma)$ for different $\sigma$. To this end, we follow EDM (pp 29) [20] and enforce that

$$\lambda_{\text{data}}(\sigma)c_{\text{out}}(\sigma)^2 = 1.$$

This leads to

$$\lambda_{\text{data}}(\sigma) = \frac{\sigma_{\text{data}}^2+\sigma^2+\frac{c_{\text{ratio}}(\sigma_{\text{reg}}^2+\sigma^2)}{\sigma^4(c_{\text{off}}+\sigma)^2}}{\big(\sigma_{\text{data}}^2+\frac{c_{\text{ratio}}\sigma_{\text{reg}}^2}{\sigma^4(c_{\text{off}}+\sigma)^2}\big)\big(\sigma^2+\frac{c_{\text{ratio}}}{\sigma^2(c_{\text{off}}+\sigma)^2}\big)}.$$

## D. Correlations between the Quality Checker with Human Evaluations

Following [45], we utilize Pearson's Correlation Coefficient [35] to assess the correlation between the mesh quality score and the FPD score with human evaluations, measuring the linear relationship between metrics. We use other shape generators (i.e., NW [18], 3DILG [55], and LAS [59]) to generate shapes for the categories of chair, airplane, and table. Additionally, we employ shap-e [19] to generate objects in five unseen categories (umbrella, diamond ring, sword, shoes, and clothing). For each generator, we generate 200 synthetic objects for the known categories and 100 synthetic objects for the unseen categories, each using different random seeds. We asked 20 ordinary users to score the mesh quality of the generated shapes. An example of the user interface for human evaluation is shown in Figure 7. We evaluated the correlation of people with Quality Score and Fréchet PointNet++ Distance (FPD). FPD* indicates that PointNet++ used in this metric is trained with ShapeNetCore55 as well as additional labeled synthetic shapes. The 'Quality Score' is obtained from the quality prediction branch of the classifier.

We evaluated FPD only at the model level, as it requires distributional statistics over multiple samples, preventing sample-level calculations. As shown in Table 3, Quality Score achieved 0.708 Pearson's in all samples, indicating a strong correlation with human evaluations. FPD* and FPD achieve 0.702 Pearson's and 0.625 Pearson's respectively, indicating FPD* correlates with human evaluations much stronger than alternative FPD. Note that Shap-e is not included in calculating FPD because there is no ground-truth data distribution.

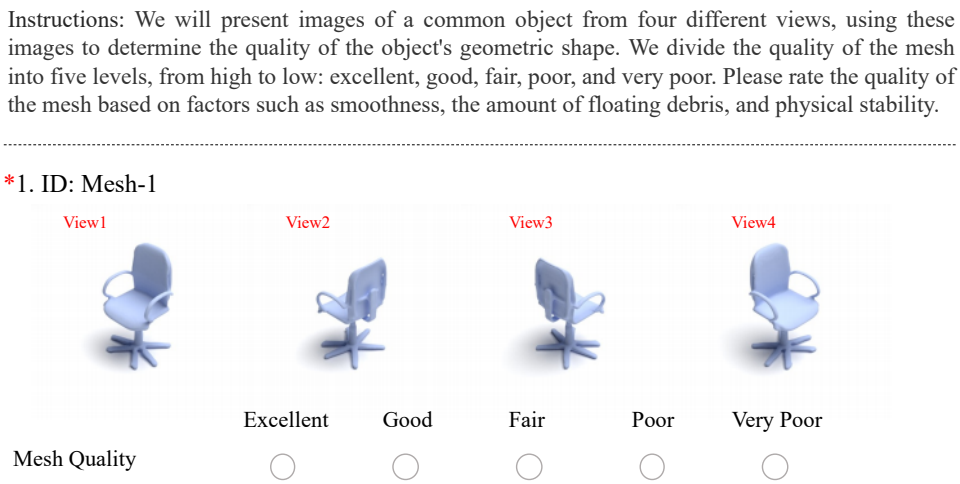


Figure 7. Rating selection UI. Instructions for raters in human evaluations are located above the rating selection.

| | Quality Score | FPD | FPD* |
|---|---|---|---|
| Model Level | - | -0.625 | -0.702 |
| Sample Level | 0.708 | - | - |

Table 3. Person correlations calculated with human judgments for Quality Score and FPD. Sample-level correlations are calculated on every individual samples, while model-level correlations are determined using all samples generated by a specific model.

# E. Implementation Details

## E.1. Quality Checker Computation Acceleration

To address this computational issue, we train neural networks to directly predict geometric feasibility and physical stability scores to avoid decoding a latent code during training time. We used a transformer to predict the score, the architecture of which is similar to that of the shape decoder. As shown in Figure 8, the transformer consists of a series of self-attention blocks. This approach avoids decoding the latent code into a shape and then sample a point cloud for classification. We trained different networks to predict the geometric feasibility and physical stability scores.

## E.2. Training Details

For category-conditioned generation, we use the ShapeNet-v2 dataset [6] as a benchmark and utilize the training/val splits in [55, 56]. And each shape is converted to a watertight mesh and then normalized to fit within its bounding box ($[-1, 1]^3$). We use 3DS2VS [56] as our backbone. For shape encoding, the point cloud of size 2048 is input to the pre-trained encoder to get the latent code $zt \in p_{\text{data}}$. For regularized diffusion, we use the 3DS2VS latent diffusion model [56] as our generator to obtain the latent code $z^{'}$, which is then fed into the quality checker to obtain a quality score $q^{\psi}(g^{\phi}(z))$. In this work, we pass the latent code $z^{'} \in p_{\text{reg}}$ when the quality score $q^{\psi}$ exceeds 0.9. We initialize our diffusion model with weights [56] and train it on 8 V100 using AdamW optimizer with batch size of 192 for 6,400 epochs. In each batch, we sample $2/3$ latent codes from $p_{\text{data}}$ and $1/3$ latent codes from $p_{\text{reg}}$. The learning rate is set to $4e-5$ and gradually decreased using the cosine decay schedule. For category-conditioned generation, objects from all categories are used for training, and the category label is mapped to an embedding. This embedding is then input into the denoising neural network as conditional information. We employ the default configurations for the EDM hyperparameters and set additional hyperparameters $c_{\text{ratio}} = 10^{-2}$ and $c_{\text{off}} = 1.0$ in this paper. During sampling, we obtain the final latent code via 40 denoising steps.

# F. Additional Results

## F.1. Image-conditioned Generation

For the single view object reconstruction, we use the 2D rendering dataset provided by 3D-R2N2 [13]. In this dataset, each shape is rendered in RGB images with dimensions of 224 × 224 pixels, captured from 24 randomly selected angles. We also adopt CLIP [31] as the image encoder and compare GPLD3D with 3DS2VS [56] and AtlasNet [17] in Figure 9.

## F.2. Text-conditioned Generation

For text-driven shape generation, we use the Text2shape dataset [8], which provides descriptions of the chair and table categories in ShapeNet. We adopt CLIP [31] as the text encoder and compare GPLD3D with 3DS2VS [56] and SDFusion [11]. For GPLD3D and 3DS2VS, we generate the

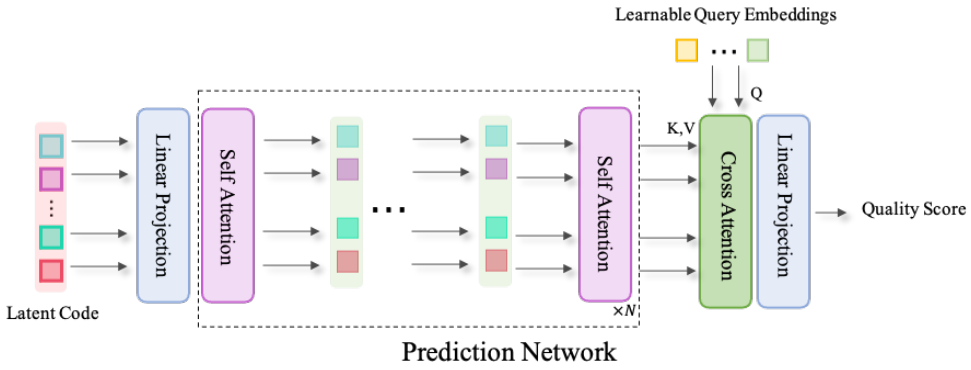


Figure 8. The architecture of the score prediction network.

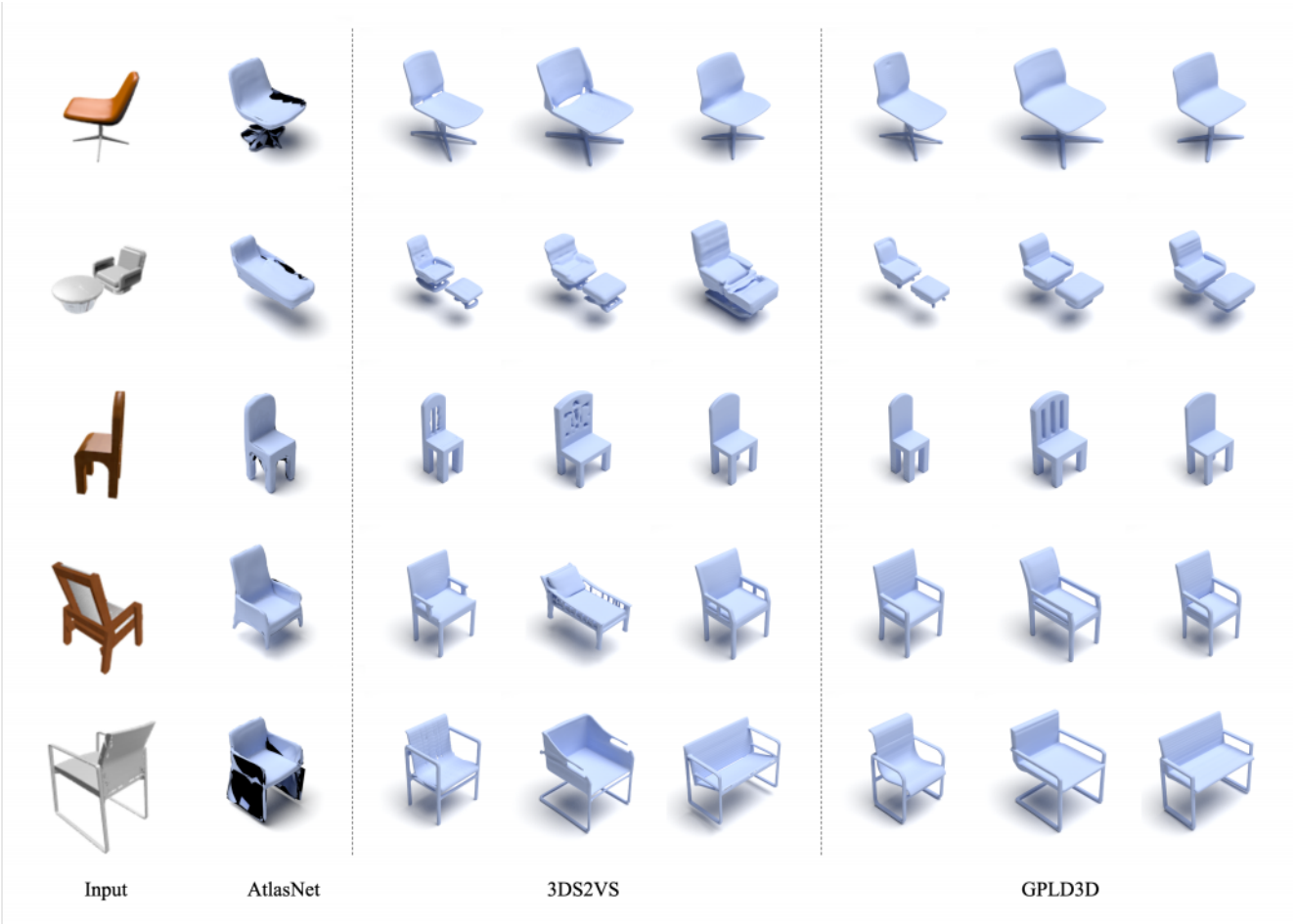


Figure 9. Image-conditioned 3D shape generation. We compare GPLD3D with AtlasNet [17] and 3DS2VS [56] on the 3D-R2N2 (chair) dataset. GPLD3D generates high-quality shapes with more details.

meshes by running Marching Cubes on a $128^3$ grid. The results of our text-conditioned generation model can be found in Figure 10. Qualitatively, GPLD3D generates shapes of better quality and with fewer geometric artifacts.

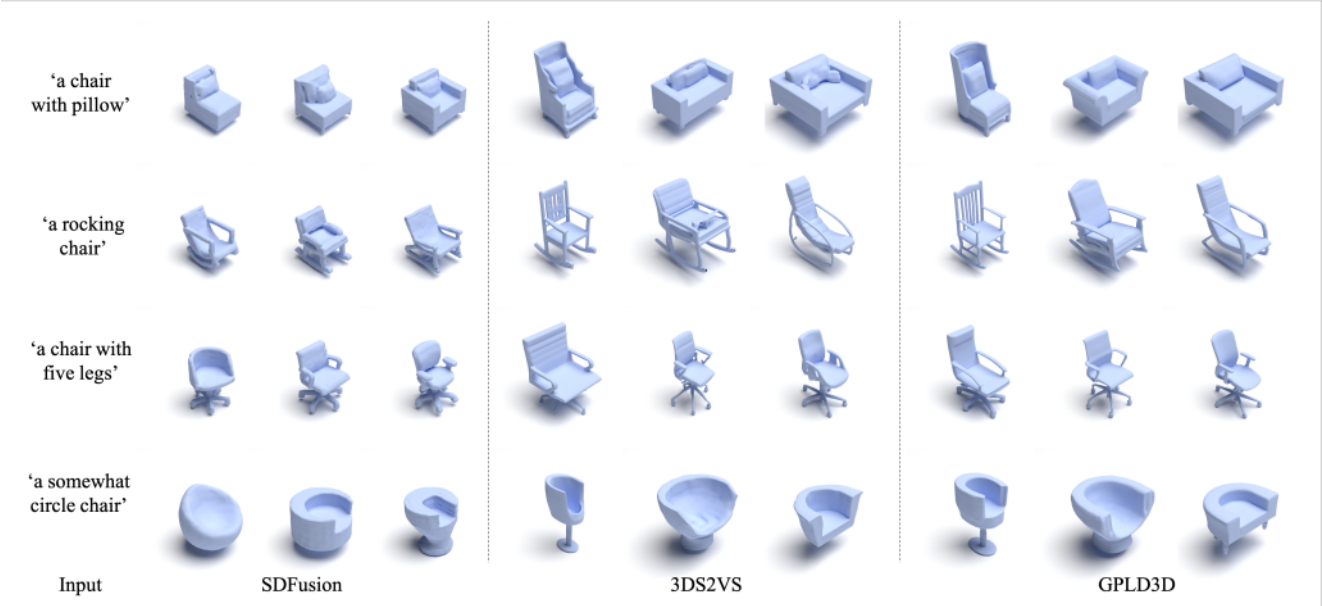


Figure 10. Text-conditioned 3D shape generation. We compare GPLD3D with SDFusion [11] and 3DS2VS [56] on the Text2Shape dataset for chairs. GPLD3D generates shapes with more details and fewer geometric issues, while complying with the description.

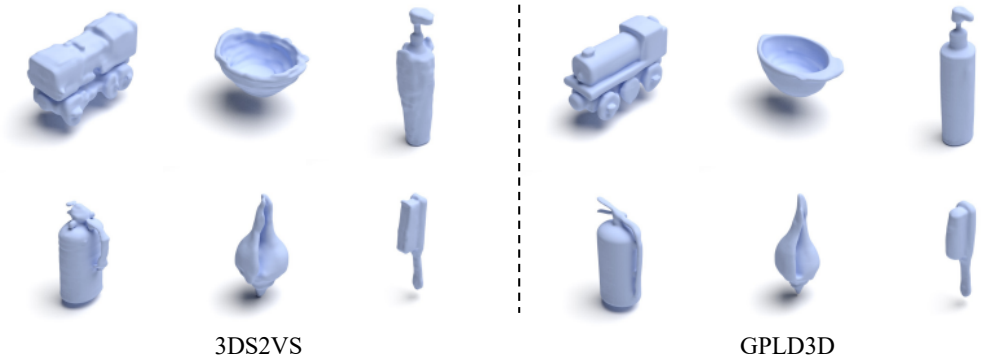


Figure 11. Unconditional generation between GPLD3D and 3DS2VS on OmniObject3D [51] dataset.

### F.3. Additional experiment on the OmniObject3D

We conducted an additional experiment on the OmniObject3D [51] dataset, which comprises 6,000 scanned objects in 190 categories. We train an unconditional model using this dataset and Fig. 11 shows some qualitative results compared with 3DS2VS [56].

### F.4. 3D Shape Texturing

We showcase an application that uses our text-conditioned GPLD3D F.2 to generate 3D shapes with detailed geometry, and then employs a depth-conditioned text-to-image 2D diffusion model [7, 57] to generate consistent textures. The results are shown in Figure 12, We can generate high-quality object meshes based on the input text prompts and obtain textured objects through the texture generation algorithm.

### F.5. More Visual Galleries on Different Categories

We provided more visual results for category-conditioned generation across different categories (Chair in Figure 13, Table in Figure 14, Airplane in Figure 15, Sofa in Figure 17, Car in Figure 16, Lamp in Figure 18).

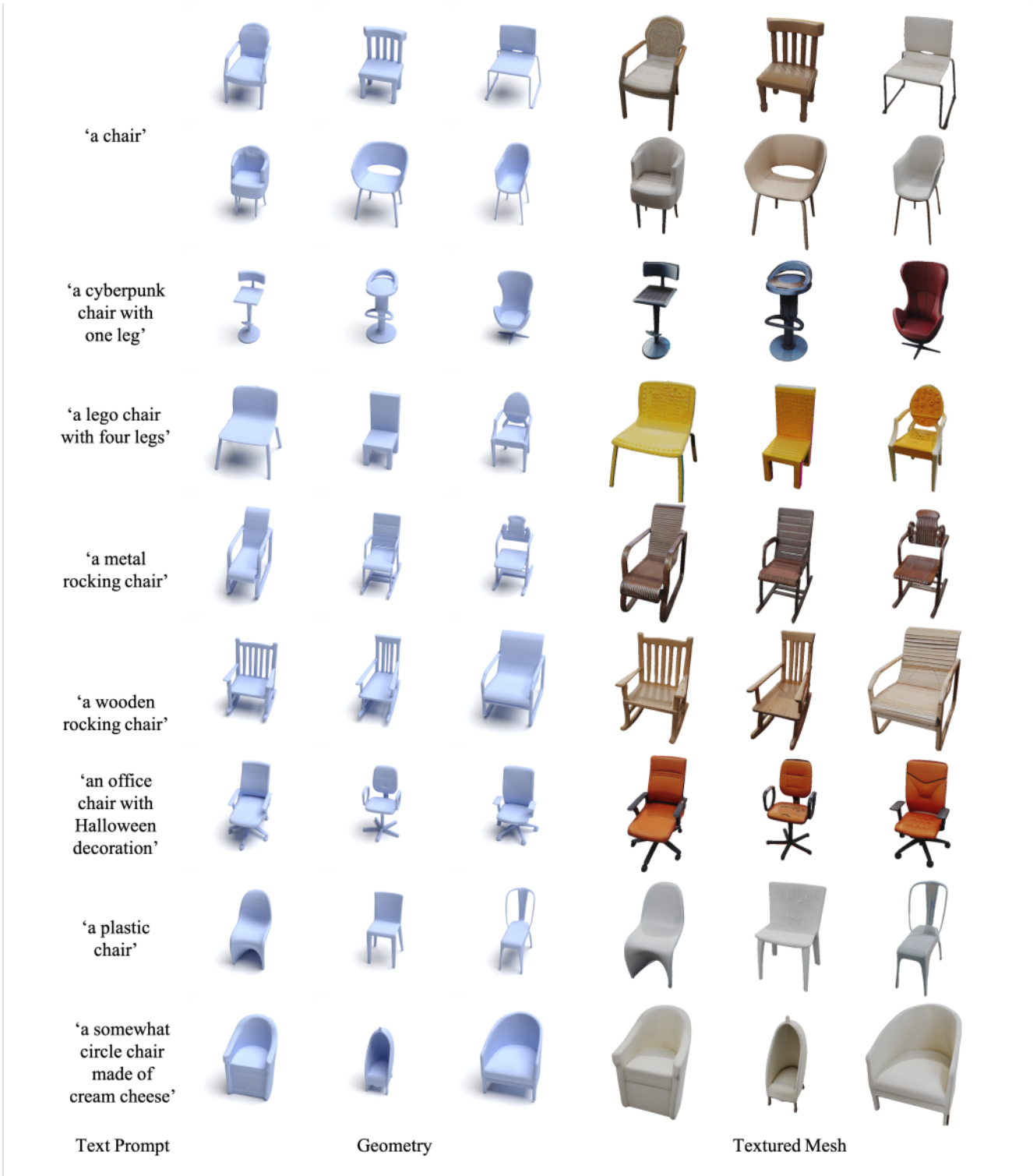


Figure 12. Text-conditioned 3D shape texturing. Firstly, we generate an object mesh using text-conditioned GPLD3D, and then texture the synthetic shapes with Text2Tex [7]. Given a text prompt, the pipeline can generate high-quality and diverse textured meshes.

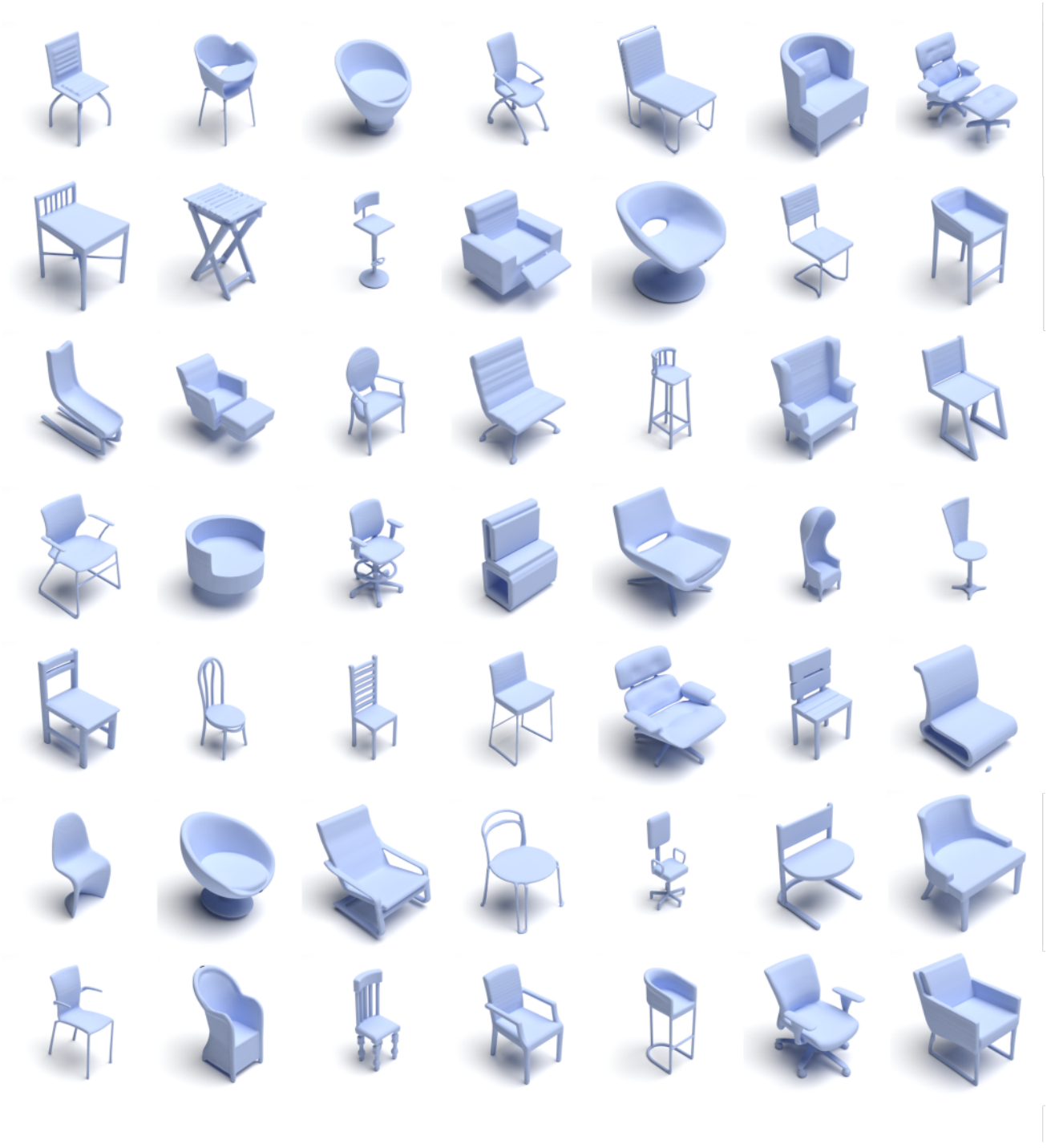

Figure 13. Gallery of chairs generated by our method.

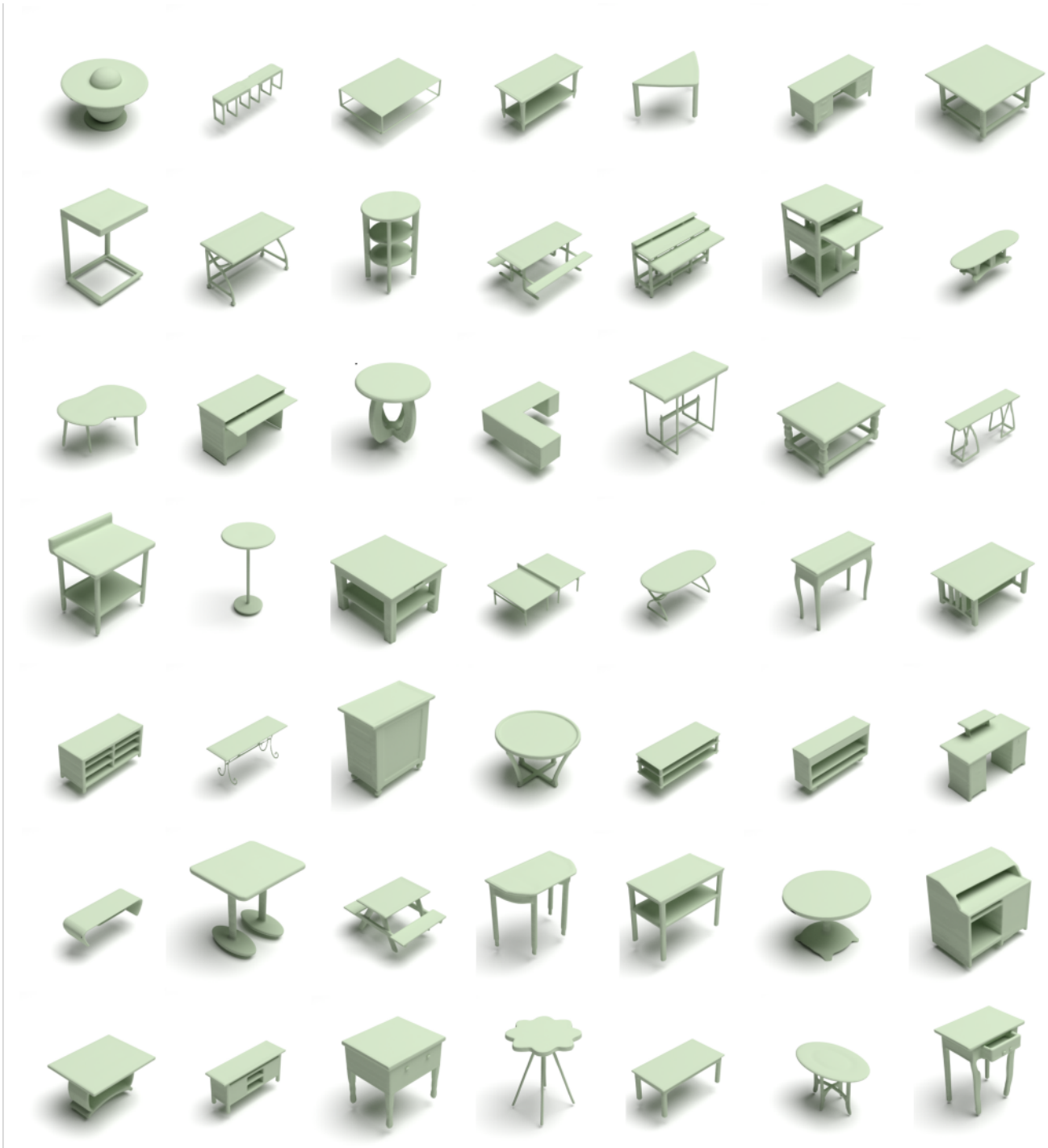

Figure 14. Gallery of tables generated by GPLD3D.

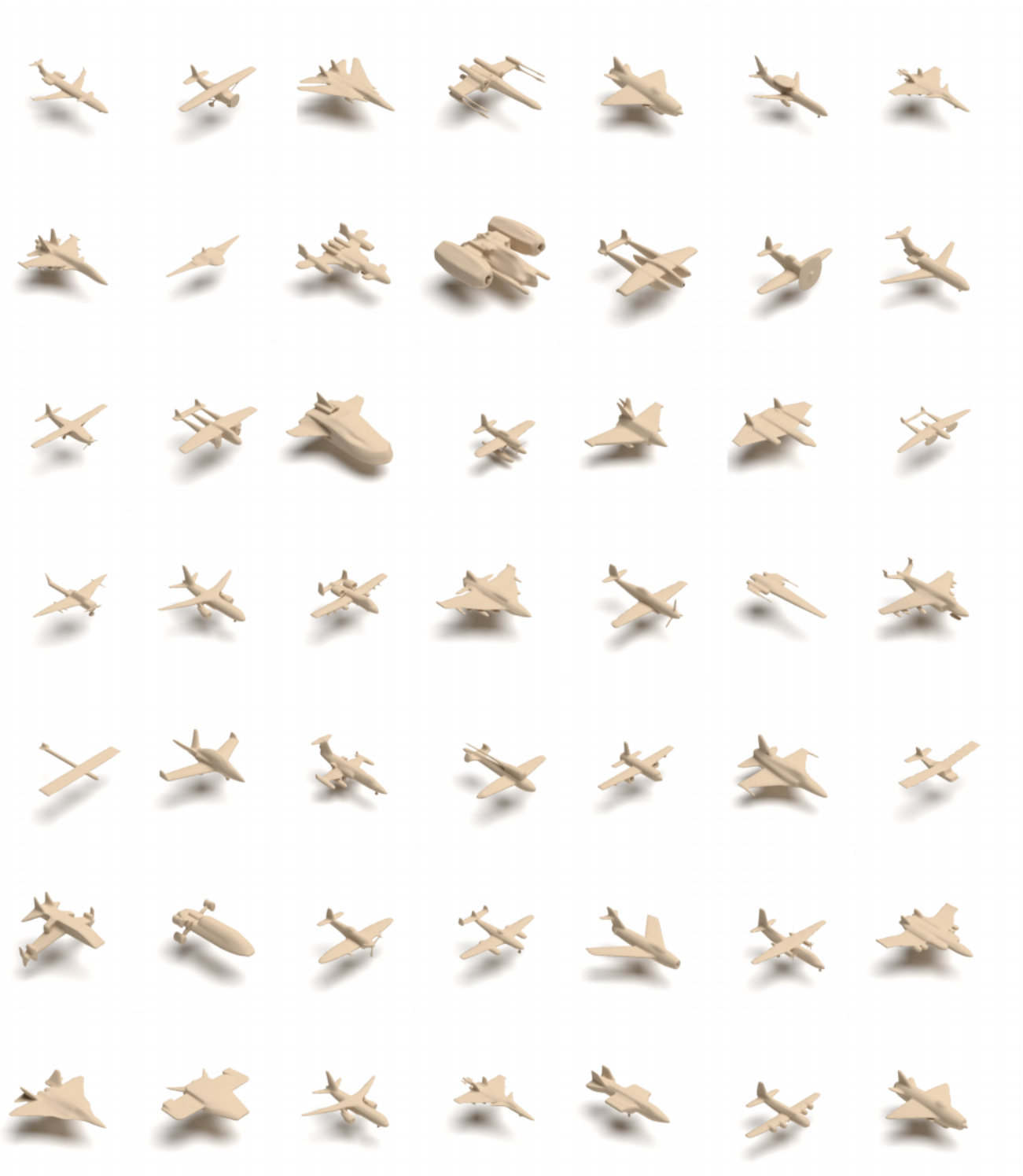

Figure 15. Gallery of airplanes generated by our method.

Figure 16. Gallery of cars generated by GPLD3D.

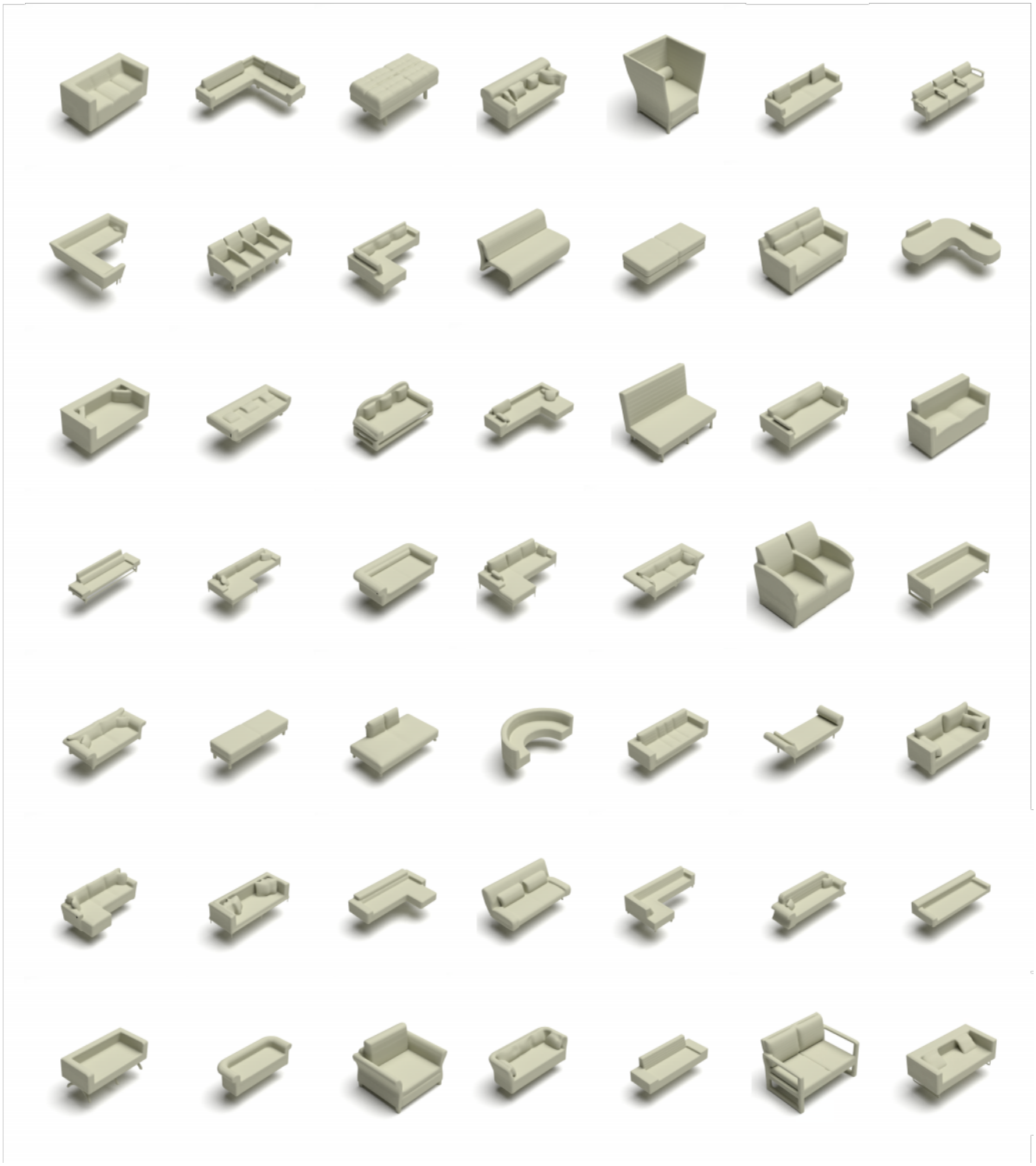

Figure 17. Gallery of sofas generated by GPLD3D.

Figure 18. Gallery of lamps generated by GPLD3D.