A Remarks on Related Work

Human-aware Scene Synthesis. Although there are many scene synthesis works, a few of them consider human motions. To our best knowledge, MIME (Yi et al., 2023) and Ye et al. (2022) are the most related literature to ours. We compare our paper with other works in Table 5.

	Scene Representation	Text input	Customized	Editable
SUMMON (Ye et al., 2022)	Contact points	✗	✗	✗
MIME (Yi et al., 2023)	3D bounding box	✗	✗	✗
Ours	Point cloud	✓	✓	✓

Table 5: **Our work vs. related human-aware scene synthesis findings.**

We further remark on the metrics utilized in our study in comparison to other related works. In (Ye et al., 2022), two metrics, namely reconstruction accuracy and consistency score, are employed to evaluate the predicted contact labels against the ground truth, primarily focusing on the reconstruction of room objects. However, since our approach directly predicts the point cloud of the target object, metrics that specifically assess the reconstruction of 3D point clouds, such as CD, EMD, and F1, are more suitable for addressing the original objective outlined in (Ye et al., 2022).

Yi et al. (2023) adopt three metrics: interpenetration, IoU, and FID score. Although FID is a solid metric, we contend that the FID score is not adequate for our proposed task, as a single orthogonal top-down view may not capture sufficient details of 3D scenes (Li et al., 2022). Lastly, the IoU metric can be well-covered by the metrics we have employed in this study, namely CD, EMD, and F1. The metric of interpenetration is expanded in our 3D configuration in the following manner:

$$\text{3D IP} = \frac{\sum_{i=1}^{M} \sum_{p \in \hat{O}_{M+1}} \mathbb{I}_{p \in O_i} + \sum_{p \in \hat{O}_{M+1}} \mathbb{I}_{p \in H}}{|\hat{O}_{M+1}|}, \tag{11}$$

where we denote $\hat{O}_{M+1}$ as the target object predicted by any baseline, and $\mathbb{I}$ indicates whether a point p belonging to the predicted point cloud lies within the interior of the scene entity or not. Supplementary results on FID and IP metrics are shown in Appendix Sec. F.

Point Cloud Estimation. PointNet (Qi et al., 2017a) stands as one of the pioneering deep learning approaches for extracting point cloud data representations. Followed by this seminal work, a variety of methodologies have been utilized to tackle the problem of estimating point sets (Zhou et al., 2021): GNN-based (Wang et al., 2019b), GAN-based (Achlioptas et al., 2018; Cai et al., 2020; Li et al., 2021), flow-based (Yang et al., 2019), transformer-based (Zhao et al., 2021), etc. Overall, prior works have shown remarkable results on 3D deep learning tasks, such as semantic/part segmentation and shape classification (Zhao et al., 2021). Since designing point cloud neural networks that have the ability to create photorealistic, novel, and distinctive shapes remains a challenge (Li et al., 2021); therefore, in this paper, we do not establish a novel point cloud network but rather use existing baselines to evaluate the scene synthesis problem.

B Conditional Denoising Process with Guiding Points

Following the *Conditional Reverse Noising Process* (Dhariwal and Nichol, 2021), we use the similar definition of the conditional noising $\hat{q}$, given by:

$$\hat{q}(\mathbf{x}_0) \overset{\text{def}}{=} q(\mathbf{x}_0)\,, \tag{12}$$

$$\hat{q}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}) \overset{\text{def}}{=} q(\mathbf{x}_{t+1}|\mathbf{x}_t)\,, \tag{13}$$

$$q(\mathbf{y}|\mathbf{x}_0) = \text{Known per sample}\,, \tag{14}$$

$$\hat{q}(\mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{y}) \overset{\text{def}}{=} \prod_{t=0}^{T-1} \hat{q}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y})\,. \tag{15}$$

Proof of Proposition 1.

It follows from (Dhariwal and Nichol, 2021, pages 25-26) that

$$\hat{q}(\mathbf{x}_t) = q(\mathbf{x}_t)\,, \tag{16}$$

and

$$\hat{q}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t+1})\hat{q}(\mathbf{y}|\mathbf{x}_t)}{\hat{q}(\mathbf{y}|\mathbf{x}_{t+1})}\,. \tag{17}$$

By using Bayes' Theorem and Eq. (16), we have

$$\hat{q}(\mathbf{y}|\mathbf{x}_{t+1}) = \frac{\hat{q}(\mathbf{y}, \mathbf{x}_{t+1})}{\hat{q}(\mathbf{x}_{t+1})} = \frac{\hat{q}(\mathbf{y}, \mathbf{x}_{t+1})}{q(\mathbf{x}_{t+1})}\,. \tag{18}$$

We can progressively compute $q(\mathbf{x}_{t+1})$ as follows

$$q(\mathbf{x}_{t+1}) = \int_{\mathbf{S}} q(\mathbf{x}_{t+1}, \mathbf{x}_0)\mathrm{d}\mathbf{x}_0 = \int_{\mathbf{S}} q(\mathbf{x}_{t+1}|\mathbf{x}_0)q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 = \mathbb{E}\left[q(\mathbf{x}_{t+1}|\mathbf{x}_0)\right]\,. \tag{19}$$

Combining Eqs. (18) and (19) into Eq. (17), we can verify that

$$\hat{q}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t+1})\hat{q}(\mathbf{y}|\mathbf{x}_t)}{\hat{q}(\mathbf{y}, \mathbf{x}_{t+1})}\mathbb{E}\left[q(\mathbf{x}_{t+1}|\mathbf{x}_0)\right]\,. \tag{20}$$

Thus, Proposition 1 is then proved.

Proposition 2. *Let* $\mathbf{G}$ *be the convex hull of* $\mathbf{S}$ *and* d_0 *be the minimum distance from any point on a facet of* $\mathbf{G}$ *to the centroid* μ_0*. By denoting* $\mathbf{G}^\circ$ *as the interior of* $\mathbf{G}$ *and assuming* $\tilde{\mathbf{S}} = \{r_1, r_2, \ldots, r_L\}$*,* $r_i \sim \mathcal{N}(\mu_0, \sigma_0^2\mathbf{I})$*, and* $s^2 = \frac{1}{L}\sum_{i=1}^{L}\|r_i - \mu_0\|^2 > \frac{C\sigma_0^2}{L}$*, we show that*

$$\Pr\left(r_i \in \mathbf{G}^\circ\right) \geq \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{d_0}{O(s)}\right)\right)\,. \tag{21}$$

Proof. Since any point that has the distance to μ_0 less than d_0 must lie within the interior of $\mathbf{G}$ (Boyd and Vandenberghe, 2004). Therefore

$$\Pr\left(r_i \in \mathbf{G}^\circ\right) \geq \Pr\left(\|r_i - \mu_0\| < d_0\right)\,. \tag{22}$$

It is sufficient to prove the Proposition 2 with an alternative term $\Pr\left(\|r_i - \mu_0\| < d_0\right)$. Since r_i is drawn from $\mathcal{N}(\mu_0, \sigma_0^2\mathbf{I})$, we have that

$$\Pr\left(\|r_i - \mu_0\| < d_0\right) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{d_0}{\sigma_0\sqrt{2}}\right)\right)\,. \tag{23}$$

According to the definition of the standard deviation, the following is obtained

$$s^2 = \frac{1}{L}\sum_{i=1}^{L}\|r_i - \mu_0\|^2\,. \tag{24}$$

Since each $r_i \sim \mathcal{N}(\mu_0, \sigma_0^2\mathbf{I})$, we can write $r_i = \mu_0 + \sigma_0\epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \mathbf{I})$. The Eq. (24) can be rewritten as

$$s^2 = \frac{\sigma_0^2}{L}\sum_{i=1}^{L}\|\epsilon_i\|^2\,. \tag{25}$$

Eq. (25) indicates that $\frac{s^2 L}{\sigma_0^2}$ follows a chi-squared distribution with L degrees of freedom. Since $s^2 > \frac{C\sigma_0^2}{L}$, we infer that $\sigma_0 = O(s)$. Therefore, we can rewrite Eq. (23) as follows

$$\Pr\left(\|r_i - \mu_0\| < d_0\right) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{d_0}{O(s)}\right)\right)\,. \tag{26}$$

From Eqs. (22) and (26), we conclude this proposition. □

The condition $s^2 > \frac{C\sigma_0^2}{L}$ is applicable, as we have proved in the Appendix that $\frac{s^2 L}{\sigma_0^2}$ follows a chi-squared distribution with L degrees of freedom. By considering the specific value $C = 1.0$, we can easily check that the probability of $\frac{s^2 L}{\sigma_0^2} > C$ is greater than $1 - 10^{-9}$ when L exceeds 20. We establish the following corollary:

Corollary 2.1.

$$\lim_{s \to 0} \Pr\left(r_i \in \mathbf{G}^\circ\right) = 1, \forall r_i \in \tilde{\mathbf{S}}\,. \tag{27}$$

Verification of Corollary 2.1 is straightforward, as $\mathrm{erf}(z)$ in Eq. (21) is known to be monotonically increasing to 1 when $z \to \infty$ (DeGroot and Schervish, 2012). Furthermore, this corollary implies that a smaller MSE (s^2) between the predicted guiding points $\tilde{\mathbf{S}}$ and μ_0 corresponds to a more accurate sampling set. Our experiments (In Sec. 4.3) also validate this implication.

C Implementation Details

Let N be the number of points in each scene entity. We begin with conducting point-level feature extraction from the given scene arrangement by $Q_0 = \text{HumanPoseBackbone}(H_0) \in \mathbb{R}^{N\times 3}$ and $[Q_1, Q_2, \ldots, Q_M] = \text{PointCloudBackbone}\left([O_1, O_2, \ldots, O_M]\right) \in \mathbb{R}^{M\times N\times 3}$.

Subsequently, the input text prompt e is embedded using a text encoder via: $\tilde{e} = \text{TextEncoder}(e) \in \mathbb{R}^D$. The dimension of the text encoder backbone, denoted as D, varies depending on the specific architecture. In order to standardize the output of all text encoders, we apply linear layers as follows: $e' = \text{LinearLayers}(\tilde{e}) \in \mathbb{R}^{d_{\text{text}}}$.

To obtain high-level translations for each scene entity Q_i, we employ a multi-head attention layer, where the input key is e' and the query as well as the value are extracted point features $[Q_0, Q_1, \ldots, Q_M]$. We revise the calculation of this layer as follows:

$$\begin{aligned}
\text{Attention}(\text{query}, \text{key}, \text{value}) &= \text{Softmax}(\frac{\text{query} \cdot \text{key}^\top}{\sqrt{n}})\text{value}, \\
\text{MultiheadAttention}(\text{query}, \text{key}, \text{value}) &= [\text{head}_1; \ldots; \text{head}_h]\mathbf{W}^O \\
\text{where head}_i &= \text{Attention}(\text{query} \cdot \mathbf{W}_i^Q, \text{key} \cdot \mathbf{W}_i^K, \text{value} \cdot \mathbf{W}_i^V)\,.
\end{aligned} \tag{28}$$

Upon passing through this layer, we obtain latent features denoted as $\mathbf{z}_i \in \mathbb{R}^{d_\mathbf{v}}$ and attention weight represented as $\mathbf{w}_i \in [0, 1]$. The high-level translation of each scene entity is given by: $[\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_M] = \text{LinearLayers}([e' \parallel \mathbf{z}_0, e' \parallel \mathbf{z}_1, \ldots, e' \parallel \mathbf{z}_M]) \in \mathbb{R}^3$, in which $\parallel$ denotes as the concatenate notation. The follow-up step in LSDM involves determining the transformation matrix for each point in the scene entities by reusing the multi-head attention: $[\mathbf{F}'_0, \ldots, \mathbf{F}'_M] = \text{MultiheadAttention}\left([\mathbf{v}_0, \ldots, \mathbf{v}_M], [Q_0, \ldots, Q_M], [Q_0, \ldots, Q_M]\right)$, where $\mathbf{F}'_i \in \mathbb{R}^{N\times d_\mathbf{F}}$. The output transformation matrices are computed given by: $[\mathbf{F}_0, \ldots, \mathbf{F}_M] = \text{LinearLayers}\left([\mathbf{F}'_0, \ldots, \mathbf{F}'_M]\right) \in \mathbb{R}^{(M+1)\times 12}$.

Guiding points inferred from each point cloud, is calculated as $\overline{\mathbf{S}}_i = \{\mathbf{F}_{i;j} Q_{i;j}^\intercal | j = \overline{0; N}\} \in \mathbb{R}^{N\times 3}, \forall i = \overline{0; M}$. For simplicity, we represent the application of the transformation matrix to point $Q_{i;j}$ as $\mathbf{F}_{i;j} Q_{i;j}^\intercal$, followed by the equation presented in Eq. (10). The weighted guiding points are then obtained via a reduction using $\mathbf{w}$ as follows: $\tilde{\mathbf{S}} = \sum_{i=0}^{M} \overline{\mathbf{S}}_i \mathbf{w}_i \in \mathbb{R}^{N\times 3}$.

In order to align the hidden features of the current timestep with the size of the point cloud, we replicate them N times: $t' = \text{Repeat}(\text{LinearLayers}(t + 1)) \in \mathbb{R}^{N\times d_t}$. Finally, the denoised point cloud is calculated by the following process:

$$\begin{aligned}
\mathbf{x}'_{t+1} &= \text{LinearLayers}(\mathbf{x}_{t+1} \parallel t') \in \mathbb{R}^{N\times 3}, \\
\mathbf{x}'_t &= \mathbf{x}'_{t+1} + \tilde{\mathbf{S}} \in \mathbb{R}^{N\times 3}, \\
\mathbf{x}_t &= \text{LinearLayers}(\mathbf{x}'_t) \in \mathbb{R}^{N\times 3}\,.
\end{aligned} \tag{29}$$

During training, we use the loss function similar to Tevet et al. (2022):

$$\mathcal{L} = \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2 . \tag{30}$$

Architecture Summarization. As illustrated in the main paper, our network architecture contains: (*i*) a human pose backbone, (*ii*) a point cloud backbone, (*iii*) a text encoder followed by standardized MLP layers, (*iv*) a multi-head attention followed by MLP layers to compute $\mathbf{v}$, (*v*) a multi-head attention followed by MLP layers to compute $\mathbf{F}$, (*vi*) MLP layers for transformation operations, and (*vii*) an MLP encoder-decoder pair. We summarize the neural architecture as well as hyperparameters of LSDM as in Table 6 and 7.

Component	Description	Input size	Output size
(*i*)	A pcd. backbone extracting features from the human pose	$[N, 3]$	$[N, 3]$
(*ii*)	A pcd. backbone extracting features from M objects	$[M, N, 3]$	$[M, N, 3]$
(*iii*-a)	A text encoder (CLIP or BERT)	Any	$[D]$
(*iii*-b)	MLP layers	$[D]$	$[d_{\text{text}}]$
(*iv*-a)	Multi-head attention to calculate translation vectors $\mathbf{v}$	$[d_{\text{text}}], [M+1, N, 3]$	$[M+1, d_{\mathbf{v}}]$
(*iv*-b)	MLP layers	$[M+1, d_{\mathbf{v}}]$	$[M+1, 3]$
(*v*-a)	Multi-head attention to calculate transformation matrix $\mathbf{F}$	$[M+1, 3], [M+1, N, 3]$	$[M+1, N, d_{\mathbf{F}}]$
(*v*-b)	MLP layers	$[M+1, N, d_{\mathbf{F}}]$	$[M+1, N, 12]$
(*vi*)	Transformation operations	$[M+1], [M+1, N, 12], [M+1, N, 3]$	$[M+1, N, 3]$
(*vii*)	MLP encoder-decoder pair	$[d_{\text{time}}], [M+1, N, 3], [M+1, N, 3]$	$[M+1, N, 3]$

Table 6: **Architecture specifications.**

Hyperparameter	Value
N	1024
M	8
D_{CLIP} of (iii)	512
D_{BERT} of (iii)	768
d_{text} of (iii)	128
$d_{\mathbf{v}}$ of (iv)	32
$d_{\mathbf{F}}$ of (v)	128
d_{time} of (vii)	32
Num. attention layers	12
Num. attention heads	8

Table 7: **Hyperparameter details.**

D Dataset Construction

PRO-teXt. While PROXD (Hassan et al., 2019) lacks semantic segmentation of the scenes, PROXE (Zhang et al., 2020) resolves this limitation by adding semantic point clouds for room objects. In our study, we utilize the semantic scenes provided by PROXE and integrate text prompts into each human motion from the PROXD dataset. Since a human sequence mainly contains movements and only interacts with the scene at a handful of moments; therefore, for each motion, we extract 3-5 human poses and generate a text prompt that describes the interaction between the current pose and the scene. Because we add text to PROXD, we reformulate the name of this dataset as *PRO-teXt*. In total, we adapt the alignments of human motions across 12 scenes, spanning 43 sequences, resulting in a total of 200 interactions.

HUMANISE. The scene meshes used in HUMANISE (Wang et al., 2022) are derived from ScanNet V2 (Dai et al., 2017). As ScanNet V2 provides excellent instance segmentation, and HUMANISE aligns human motions with scenes sufficiently, we directly leverage the text prompts provided by the authors. Although these prompts are action-driven, they still contain spatial relations between target objects and scene entities and are thus suitable for our problem. As HUMANISE is a large-scale dataset, we limit our study to 160 interactions, as our paper does not primarily focus on human actions, which is the original purpose of HUMANISE (Wang et al., 2022).

In summary, we present the object distribution of both datasets in Fig. 9. Our statistics reveal that there is a diverse range of room objects in both datasets, with chairs being the most prevalent asset

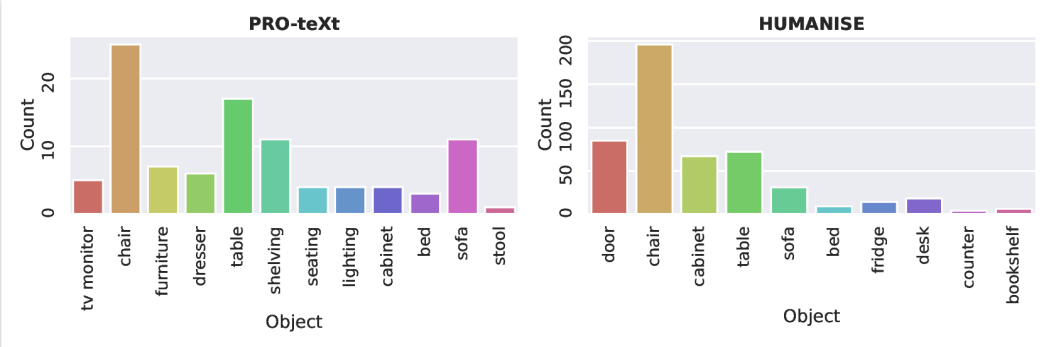


Figure 9: **Object distribution of two datasets.**

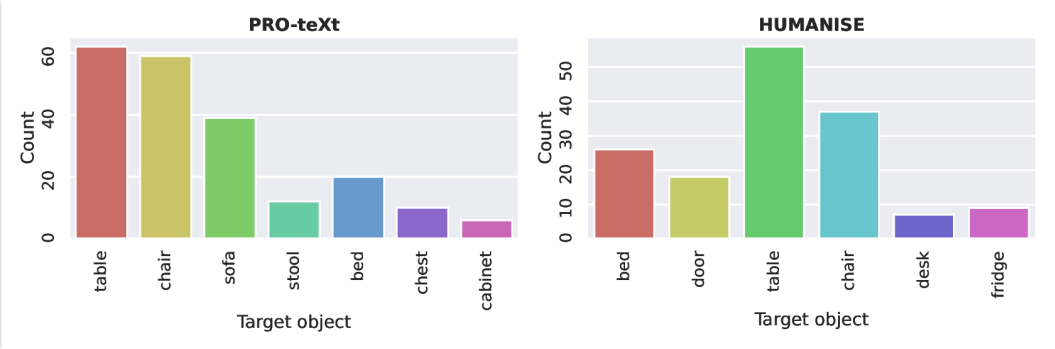


Figure 10: **Objects counted by appearances in textual commands.**

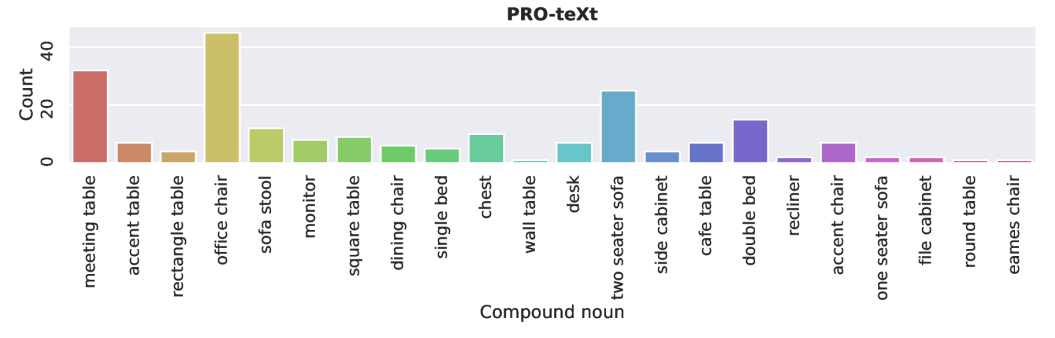


Figure 11: **Compound nouns distribution of PRO-teXt.**

in both cases. Fig. 10 gives information about the distribution of target objects. A major difference between our dataset and HUMANISE is that we describe objects as compound nouns. As a result, our text prompts comprise information about the shape of objects, which also enables scene editing operations that are not feasible with HUMANISE. We illustrate the distribution of compound nouns in Fig. 11, which demonstrates that PRO-teXt has a broad spectrum of object shapes.

E Scene Editing Formulation

In this section, we further elaborate on the procedure of each editing operation. We utilize a pre-trained model that was previously trained for the scene synthesis task and evaluate its performance on unseen editing interactions to assess its generalizability.

Editing operation	Fitness ↑	Inlier MSE ↓	Correspondent percentage (%) ↑
Object replacement	0.6815	0.0411	76.37
Shape alternation	0.6130	0.0340	60.05

Table 8: **Ground truth evaluation.** We assess the process of constructing ground truth in editing operations.

Object Replacement. We first select 10 interactions out of the test set for the object replacement as well as other operations. For every interaction, we adjust the text prompt e to include information about the new object category. The resulting modified prompt e^* captures this information.

In the next step, we proceed to compute the ground truth object through the editing operation by selecting an object O^*_{M+1} that matches the description of the new text prompt e^*. The alignment of O^*_{M+1} with the original object O_{M+1} in terms of position and rotation is performed using the ICP algorithm, subject to the constraint of fixing the z-rotation of the objects, as presented in (Rusinkiewicz and Levoy, 2001).

We evaluate the quality of transformation in Table 8. The metrics include: *i) Fitness score:* measures the overlapping area between the inlier correspondences and the original point cloud O_{M+1}; *ii) Inlier MSE:* calculates the MSE of all inlier correspondences; and *iii) Correspondent percentage:* is the ratio of the correspondence set and O_{M+1}. Our ground truth construction for the object replacement operation achieves significantly high scores in all metrics, in comparison to their maximum values. It is crucial to note that due to the fundamental dissimilarities in category and shape between O^*_{M+1} and O_{M+1}, the metrics are infeasible to reach their maximum values. Based on the ground truth assessment, the position of O^*_{M+1} following transformations almost matches that of the original object's position, hence can serve as a reliable ground truth for our operation.

Finally, we proceed to evaluate the object replacement operation similar to the former scene synthesis task, utilizing the established ground truths. The process involves denoising a noisy datapoint at timestep T, conditioning on the human pose, given objects, and the new modal command e^*, until we ultimately obtain the generated object $\mathbf{x}_0$. The results are presented in the main paper.

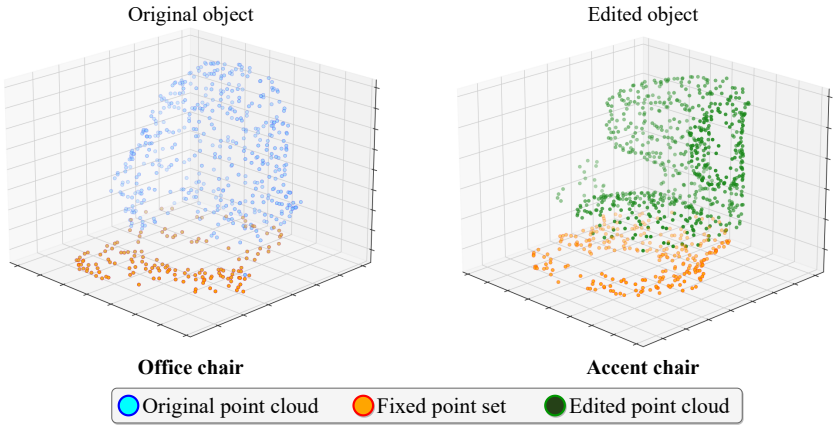


Figure 12: **Shape alternation formulation.** We fix 25% of the point that is closest to the ground (orange) then diffuse the rest 75% (blue) to obtain a new shape (green). We keep the object and spatial description in the text but change the adjective of the object (office → accent).

Shape Alternation. Fig. 12 demonstrates how the shape alternation operation progresses. Like the previous operation, we first construct the data and ground truth. Since this operation aims to alter the original object's shape, we replace the adjective in the original text prompt e with a new shape adjective, as illustrated in the figure. The ground truth construction occurs in the same procedure as the object replacement. Table 8 exhibits the quality of the ground truth construction. With acceptable metrics, we adopt the transformed objects as supervisors for this operation.

The main difference between shape alternation and object replacement is that instead of diffusing the entire point cloud, only three-quarters of it are diffused. To ensure consistency in terms of the position and rotation of the edited object, we fix 25% of the point cloud with the lowest z-coordinate, *i.e.*, points that are closest to the floor (see Fig. 12). This fixation strategy is inspired by the editing application in (Tevet et al., 2022). We note that not all objects will support the shape alternation operation, however, this operation allows us to have more practical applications in real-world scenarios.

Object Displacement. The final operation introduced in this paper is object displacement. The formulation of this task involves modifying the spatial relation expressed in the textual input e, while keeping the rest of the information unchanged. For the ground truth of this operation, we manually set the correct objects semantically related to the editing text prompts.

F Extra Experiments

Baselines. We provide more information about the implementation of other baselines:

1) ATISS (Paschalidou et al., 2021). As stated in (Yi et al., 2023), ATISS does not take humans into account, therefore, we extend humans as an exceptional object. Furthermore, ATISS represents objects as bounding boxes defined by quadruple: position, size, angle, and category; consequently, we convert both the input and the output to bounding box representation by utilizing an algorithm from Open3D (Zhou et al., 2018). We even upscale the target object to its bounding box as ground truth so that redundant points within boxes predicted by ATISS are still counted. We utilize the same implementation of ATISS as in the original paper.

2) SUMMON (Ye et al., 2022). We use the pre-trained model provided by the authors for predicting contact points and freeze the model during training sessions. Next, we create a bounding box from the forecast contact points as a supplementary object and proceed similarly to ATISS.

3) MIME (Yi et al., 2023). We re-implement the neural architecture as presented in (Yi et al., 2023). MIME considers human motions as a bounding box with contact labels and iteratively generates the next object base on the quadruple representations of existing scene meshes (including human poses). Note that, by the time of our paper submission, both the code and dataset of MIME have not yet been made publicly available.

Backbone Variation. We measure the impact of different point cloud and human pose backbones in Table 9. For the point cloud backbones, we utilize PointNet++ (Qi et al., 2017b) and DGCNN (Wang et al., 2019b), two off-the-self approaches. We leverage state-of-the-art POSA (Hassan et al., 2021) and P2R-Net (Nie et al., 2022) to encode human pose. In case of representing human pose as SMPL parameters, we resolve this problem by employing a SMPL-X (Pavlakos et al., 2019) model to extract vertices from human body. Our method uses CLIP embedding (Radford et al., 2021) and BERT (Devlin et al., 2018) to encode text prompts.

Text encoder	Human backbone	Point cloud backbone	CD ↓	EMD ↓	F1 ↑
CLIP	POSA	PointNet++	0.5365	**0.5906**	**0.3686**
CLIP	POSA	DGCNN	**0.3499**	0.6811	0.3375
CLIP	P2R	PointNet++	0.6255	0.9392	0.1288
CLIP	P2R	DGCNN	<u>0.5131</u>	<u>0.6793</u>	<u>0.3496</u>
BERT	POSA	PointNet++	0.7050	0.9971	0.1148
BERT	POSA	DGCNN	0.9276	0.9621	0.2431
BERT	P2R	PointNet++	1.7615	1.2433	0.0320
BERT	P2R	DGCNN	0.9136	1.0472	0.0534

Table 9: **Backbone variation.** Experiments were conducted on PRO-teXt.

Our findings indicate that CLIP is more effective when combined with both human backbone and PCD backbone in comparison to BERT. Each configuration of human backbone and PCD backbone, when integrated with CLIP, yields more comprehensive performance compared to the integration

with BERT. For instance, the combination of CLIP, P2R, and DGCNN achieves an F1 score nearly seven times higher than the combination of BERT, P2R, and DGCNN.

Moreover, it can be observed that POSA exhibits better evaluation metrics compared to P2R when paired with other components. This distinction is particularly observable when utilizing CLIP as the text encoder and PointNet++ as the PCD backbone. In this scenario, the F1 score achieved by POSA is nearly three times higher than that of P2R.

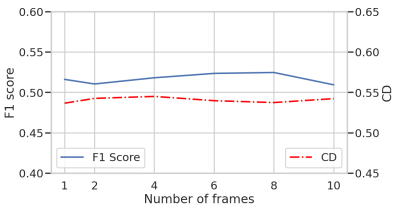
The effectiveness of PointNet++ and DGCNN is found to be comparable, with no noticeable differences observed upon replacing PointNet++ with DGCNN. Among all the possible combinations of components, the most comprehensive configuration involving CLIP, POSA, and PointNet++ achieves the the best EMD and F1 metrics, while also outputting a decent CD metric (only 0.1866 units worse than the best CD metric observed). Follow by this fact, we also select this combination as the default setting for LSDM.

	PRO-teXt		**HUMANISE**	
Baseline	3D IP ↓	FID ↓	3D IP ↓	FID ↓
ATISS (Paschalidou et al., 2021)	0.0652	319.24	<u>0.0672</u>	196.83
SUMMON (Ye et al., 2022)	0.0559	<u>163.98</u>	0.0719	127.12
MIME (Yi et al., 2023)	0.0620	257.82	**0.0626**	373.02
LSDM w.o text (Ours)	**0.0161**	167.97	0.0768	<u>94.16</u>
LSDM (Ours)	<u>0.0402</u>	**161.05**	0.0851	**83.96**

Table 10: **Supplementary results on the scene synthesis task.**

Supplementary Results. We provide results on auxiliary metrics from all baselines in Table 10. All baselines yield statistically insignificant values in terms of 3D IP, indicating that the generated objects generated by all methods exhibit limited collision with other scene entities. However, on the FID metrics, our method remarkably improves other baselines. This improvement is particularly considerable in the HUMANISE dataset, where our method outperforms other state-of-the-art benchmarks by at least 35%.

Impact of Number of Frames. We remark that our model can take either single-frame human pose or multi-frame human pose. We have also included a study regarding the impact of the number of frames on scene synthesis results in Fig. 13a. The experimental results indicate that the performance of our LSDM varies little when taking different number of human pose frames as the input. Consequently, using one frame is enough for our network.



(a) Impact of the number of human pose frames.

(b) Failure cases of guiding points.

Figure 13: **Additional experiments.**

Failure Cases of Guiding Point Network. We visualize failure cases of predicting $\tilde{\mathbf{S}}$ of our LSDM method in Fig. 13b. Although our guiding points fail to predict the object's position, they still span over the correct object's shape.

G User Study Details

This section presents the implementation of our user study in detail. We conduct a user study with the participation of 40 people with a variety of ages, and professions. The male/female ratio is 50/50. For each questionnaire, we provide 15 questions about preferences, corresponding to five baselines (ground truth, LSDM, ATISS, SUMMON, and MIME). For each baseline, we anonymize the name and ask participants to evaluate three categories:

1) Naturalness. Naturalness of the scene is the level of familiarity that the interviewers experience when viewing the pictures in comparison to actual real-life room arrangements.

2) Non-collision. This metric is assessed by measuring the degree to which participants perceive objects overlapping within the scene. A lower score is assigned to the scene when participants report a greater sense of overlap among objects within it.

3) Intentional Matching. The final criterion we use in this study assesses the extent to which the arrangement of objects within the scene aligns with the personalized intention outlined in the text prompt.

Overall, we visualize 8 scenarios from both PRO-teXt and HUMANISE datasets for each baseline. All images and videos were collated into a single question, and interviewees were asked to score the performance of each baseline based on the visual cues presented in the 8 demonstrations. The use of 8 distinct visualizations for each baseline ensured that the study is conducted with minimal bias. Each question is rated on a scale of 1 to 5. The results are presented in the main paper.