# MiDSummer: Multi-Guidance Diffusion for Controllable Zero-Shot Immersive Gaussian Splatting Scene Generation

Supplementary Material

## A. Further Implementation Details

### A.1. Multilevel Cross-Proposal Consistency

**Algorithm 1** Bounding Box Parameter Consistency

**Require:** Box parameters for object $i$: $\{\mathbf{b}_m^i\}_{m=1}^M$, variance threshold $\nu$, and scaling parameter $\tau$
**Ensure:** Bounding box consistency score $\mathbf{C}_B^{(i)}$
Compute mean bounding box: $\bar{\mathbf{b}}^i \leftarrow \frac{1}{M}\sum_{m=1}^M \mathbf{b}_m^i$
Compute variance: $\sigma^2_{\mathbf{b}^i} \leftarrow \frac{1}{M}\sum_{m=1}^M \left(\mathbf{b}_m^i - \bar{\mathbf{b}}^i\right)^2$
Convert variance to consistency:
$\mathbf{C}_B^{(i)} \leftarrow \frac{1}{1+\exp\left(\tau\left(\sigma^2_{\mathbf{b}^i} - \nu\right)\right)}$
**return** $\mathbf{C}_B^{(i)}$

**Algorithm 2** Edge Consistency

**Require:** Set of scene graph proposals $\{\mathcal{G}_m\}_{m=1}^M$ with edge sets $\mathcal{E}_m$, number of nodes $N$
**Ensure:** Pairwise relationship count matrix $\mathbf{E} \in \mathbb{R}^{N\times N}$
Initialize $\mathbf{E} \leftarrow \mathbf{0}_{N\times N}$
**for** $m = 1$ to $M$ **do**
  **for** each edge $e_{ij} \in \mathcal{E}_m$ **do**
    $\mathbf{E}(i,j) \leftarrow \mathbf{E}(i,j) + 1$
  **end for**
**end for**
**return** $\mathbf{E}$

**Algorithm 3** Clique Consistency

**Require:** Set of scene graph proposals $\{\mathcal{G}_m\}_{m=1}^M$ with clique groupings $\mathcal{C}_m$, number of nodes $N$
**Ensure:** Clique consistency vector $\mathbf{C}_G \in \mathbb{R}^N$
Initialize $\mathbf{G} \leftarrow \mathbf{0}_{N\times N}$
**for** $m = 1$ to $M$ **do**
  **for** each pair $(v_i, v_j)$ such that $(v_i, v_j) \in \mathcal{C}_m$ **do**
    $\mathbf{G}(i,j) \leftarrow \mathbf{G}(i,j) + 1$
  **end for**
**end for**
**for** $i = 1$ to $N$ **do**
  $\mathbf{C}_G(i) \leftarrow \frac{1}{N-1}\sum_{j=1}^N \mathbf{G}(i,j)$
**end for**
**return** $\mathbf{C}_G$

### A.2. Per-Stage Runtime Breakdown

The average runtime per scene is broken down as follows: layout generation (3.1.1) takes 510s for 10 layout proposals using Claude [2], with an additional 67s for GDM diffusion (3.1.2); assets (3.1.3) require around 90s per object with GaussianDreamer [13] and less than 3s with DiffSplat [96]. Registration, assembly and refinement takes approximately 258s per scene assuming the number of cameras $V$ can fit into one batch.

| Planning Stage | | Runtime (s) |
|---|---|---|
| Layout | Claude ($M = 10$) | 510 |
| | EchoLayout | 67 |
| Asset | Per Object | $(3N, 90N)$ |
| **Assembly Stage** | | 258 |
| **Total Runtime** | | **15-30 min** |

Table 6. Per-scene runtime breakdown across different stages.

## B. Scalability Analysis

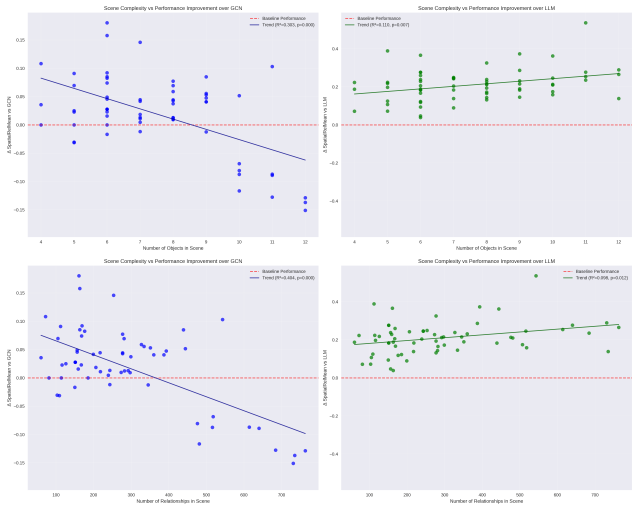


Figure 3. Relative performance gain from self-consistency. We plot changes in mean spatial relationships accuracy relative to GDM (left) and LLM (right) baselines, shown as a function of scene complexity (quantified by the number of nodes and edges in scene graphs). Our method's advantage against GDM is more pronounced in simpler scenes. Conversely, the positive trend against the LLM baseline indicates that the performance benefit of our method grows as scene complexity increases.


Figure 4. Comparative scalability of base models and self-consistency method with respect to scene complexity. LLM baseline performance deteriorates with increasing numbers of objects (left) and relationships (right), whereas GDM performance exhibits relative stability. Our consistency method (green) exhibits intermediate scalability characteristics as a result of the integration of both guidance mechanisms.

## C. Preliminary Probing Experiments: Out-of-Distribution Generalizability of GDMs

Table 7. Effect of OOD node features and scene graphs on spatial adherence and realism. We evaluate EchoLayout under an naive zero-shot open-vocabulary setting, where GDM models are forced to generalize beyond their training distribution. Results indicate that obfuscating node features leads to a significant degradation in physical validity, as evidenced by increased volume estimation error, to a degree that matches or even exceeds the impact of OOD $\mathcal{G}$. However, spatial relationships remain relatively intact: GDM can still adhere to scene graph constraints to a large extent despite OOD node features causing a drop in performance.

| Config | Δ Performance | | | | | |
|---|---|---|---|---|---|---|
| | L/R | F/B | Bi/Sm | Ta/Sl | **Rel↑** | **VolErr↓** |
| In-Distribution | 98 | 98 | 96 | 96 | 97 | 12.15 |
| OOD $\mathcal{V}$ + Orig $\mathcal{G}$ | 87 | 82 | 92 | 90 | 87 | 112.0 |
| OOD $\mathcal{V}$ + OOD $\mathcal{G}$ | 76 | 85 | 77 | 85 | 79 | 82.2 |

- **Baseline (In-distribution):** Standard EchoLayout settings where object representations consists of concatenated CLIP node phrase features and a pre-trained object categorical embedding $v_i = [p_i, o_i]$ as discussed in Sec. 3.1.2, serving as the performance upper bound.
- **OOD $\mathcal{V}$ + Original $\mathcal{G}$:** The learnable categorical embedding portion of the node feature is replaced with a random tensor sampled from a standard Gaussian distribution: $v_i = [E_t(\text{unseen prompt}), \xi], \quad \xi \sim \mathcal{N}(0, I)$. The scene graph input $\mathcal{G}$ remains unchanged from the original training dataset.
- **OOD $\mathcal{V}$ + OOD $\mathcal{G}$:** Both the class embedding and scene graphs are replaced with out-of-distribution (OOD) samples where $\mathcal{G}$ are generated scene graphs do not originate from the original training dataset.

Table 7 reveals a significant degradation in physical realism, as indicated by increased volume estimation error, when node features are out-of-distribution. The magnitude of this degradation matches or even exceeds the scenario where both the node and graph distributions are OOD. However, unlike the latter case, spatial relationships remain largely intact, suggesting that the model still adheres to scene graph edge constraints even under OOD inference. This behavior complements that of the LLM. These findings further motivate our approach of leveraging both scene graphs and LLM-based layout proposals: while scene graphs guide the GDM to establish spatial relationships correctly, layout proposals help ensure physical realism in object placement.

## D. Additional Visuals and Ablations

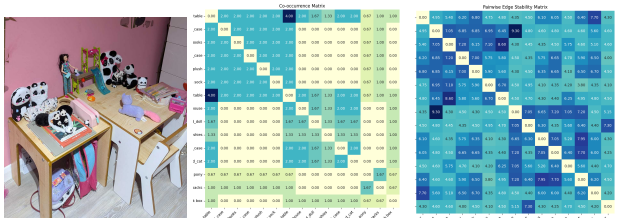



Figure 5. Group co-occurrence (clique stability) $\mathbf{C}$ and edge stability $\mathbf{E}$ of a complex scene with strong functional grouping. The co-occurrence matrix reveals two highly stable cliques, aligning with human perception of the two desk-group arrangements. The edge stability matrix exhibits two strong clusters albeit to a lesser extent, with the most confident (darkest blue) edges capturing left-right relationships between the two desks. This highlights a limitation of our work: while our self-consistency measures are simple, intuitive and motivated by graph structures, they are correlated; hence, we invite the community to seek *more disentangled and statistically rigorous* alternatives to obtain more orthogonal consistency measures.

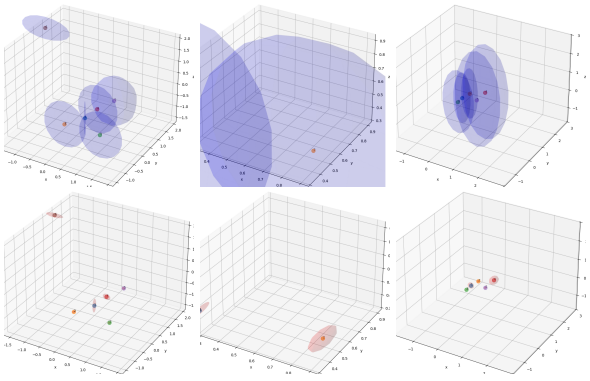


Figure 6. Bounding box parameter consistency $\mathbf{C}_B^{(i)}$ of a few scenes proposed by the LLM. Positional consistency $(x, y, z)$ is shown in blue (top row) and size consistency $(l, h, w)$ in red (bottom row). While object sizes remain relatively stable (small red ellipsoids), spatial positioning exhibits lower consistency (larger blue ellipsoids).

Figure 7. Qualitative comparison of selected samples with other methods. **Middle two columns are ours**. The left four columns, including ours, showcase text-to-GS methods; the right four columns, also including ours, represent text-to-immersive scene methods. MiDSummer balances the immersive scene generation capabilities of text-to-scene approaches with the well-defined geometry and appearance of object-centric GS methods while also providing a good trade-off between physical plausibility and fine-grained controllability.

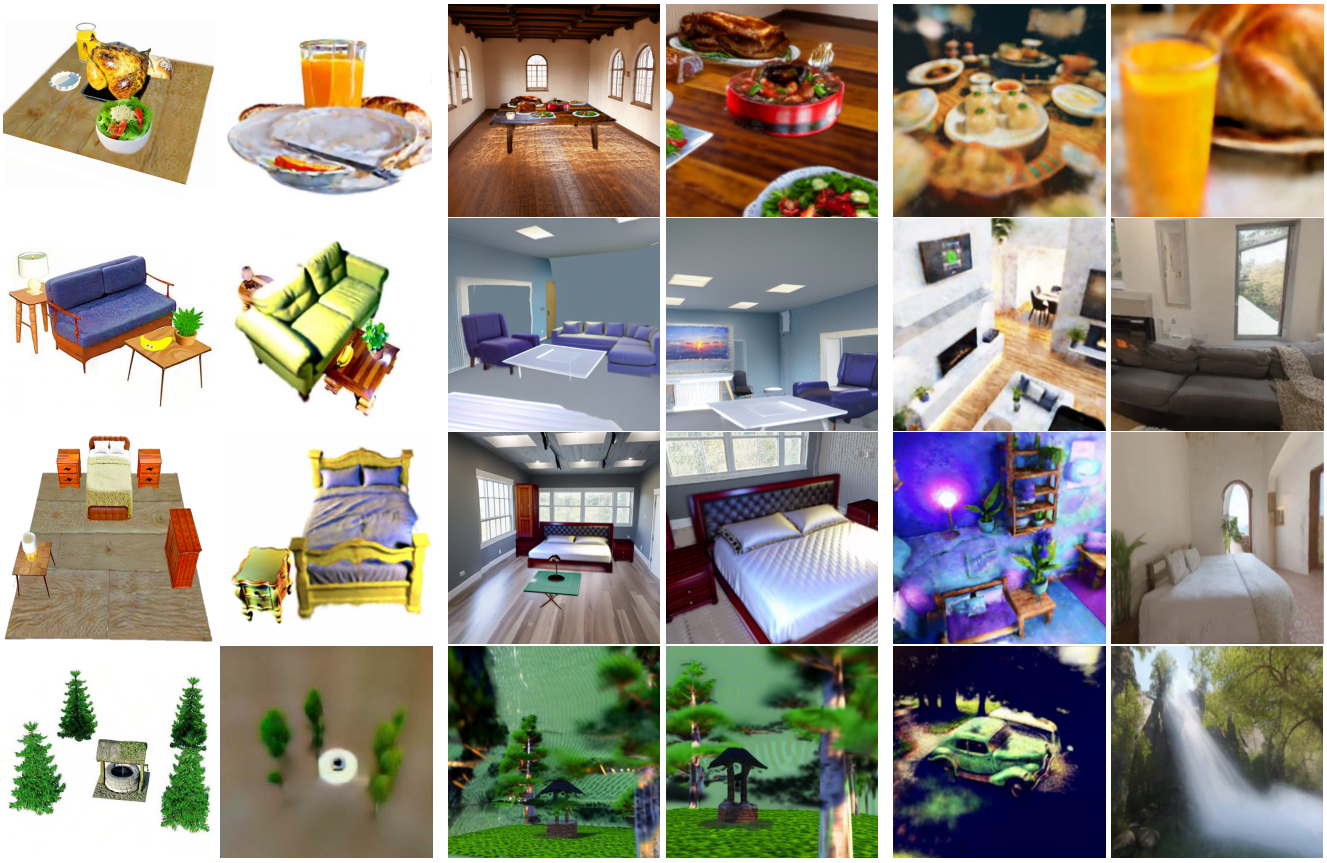


Figure 8. **Pairwise comparison of scenes before (pair left) and after (pair right) layout-guided refinement**. The refinement process improves geometric quality and scene coherence while remaining faithful to overall scene geometry. However, it does introduce a smoothing effect that may occasionally wipe out finer details or high-frequency textures. Laplacian and KID metrics alone may not fully reflect such nuances.

Figure 9. Effect of **Surface Loss** term during refinement. This term encourages neighboring Gaussian splats to lie on locally coherent surface manifolds by penalizing displacement vectors that deviate from surface tangency. Curved surfaces, such as ceiling and windows, could be straightened as a result. The loss $\mathcal{L}_{\text{surface}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_k(i)} w_{ij}\left(\mathbf{v}_{ij}\cdot\bar{\mathbf{n}}_{ij}\right)^2$ computes the squared dot product between displacement vectors $\mathbf{v}_{ij} = \mathbf{p}_j - \mathbf{p}_i$ and average surface normals $\bar{\mathbf{n}}_{ij} = \frac{\mathbf{n}_i+\mathbf{n}_j}{2}$, weighted by normalized inverse distances $w_{ij} = \frac{1/d_{ij}}{\sum_{l\in\mathcal{N}_k(i)} 1/d_{il}}$ where $d_{ij} = |\mathbf{p}_i - \mathbf{p}_j|_2$. This formulation ensures that splats maintain proper geometric alignment on surfaces, preventing floating artifacts and enforcing manifold constraints. However, while surface loss effectively aligns existing splats, it does not resolve undersampling "stripe" artifacts that arise from insufficient splat density in regions further away from the initial camera position, requiring complementary scale losses and densification strategies to achieve better visual quality.

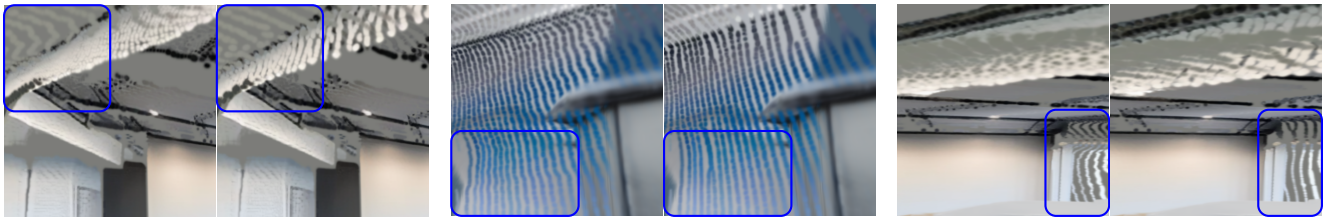


Figure 10. Effect of **Scale Consistency** term during refinement. Nearby splats are encouraged to have similar sizes to avoid sudden scale transitions that create visual artifacts. $\mathcal{L}_{\text{scale}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_k(i)}\sum_{d=1}^{3} w_{ij}\left|\log\left(\frac{s_{j,d}}{s_{i,d}+\epsilon}\right)\right|$ where $s_{i,d}$ is the $d$-th component of the activated scale vector $\mathbf{s}_i$, and $\epsilon = 10^{-6}$ is a small constant to prevent division by zero. The logarithm ensures scale ratios are treated symmetrically This term encourages gradual scaling up of splats in sparse regions to fill the gaps. Rather than having isolated large splats (which would incur high penalties due to scale mismatch with smaller neighbors), the loss function promotes cooperative scaling behavior where neighboring splats gradually increase in size together. This creates smoother coverage transitions and helps fill sparse regions while maintaining geometric coherence.

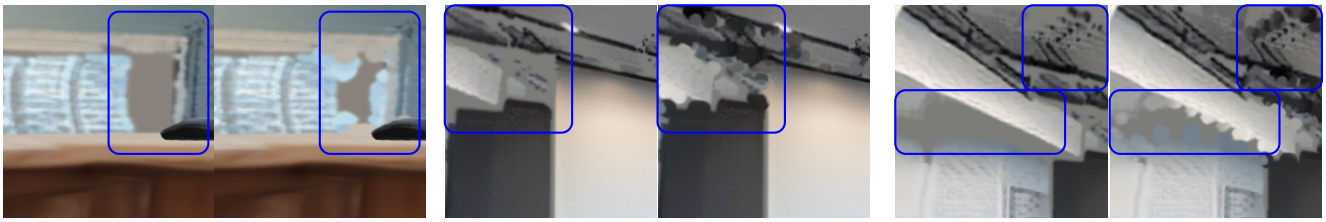


Figure 11. Effect of **Depth Consistency** term during refinement. This term encourages spatial coherence in rendered depth maps across multiple camera viewpoints to reduce stripe artifacts and rendering holes. As a result, splats near under-sampled areas are displaced to fill in the gaps. However, the Surface Loss term is needed to prevent them from deviating from existing surfaces. For a set of camera views $\mathcal{V}$, the loss is defined as $\mathcal{L}_{\text{depth}} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left(\mathcal{L}_{\text{depth}}^{(v,x)} + \mathcal{L}_{\text{depth}}^{(v,y)}\right)$, where $\mathcal{L}_{\text{depth}}^{(v,x)} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W-1}\max(0, |D_v[i,j+1] - D_v[i,j]| - \tau)^2$ and $\mathcal{L}_{\text{depth}}^{(v,y)} = \frac{1}{HW}\sum_{i=1}^{H-1}\sum_{j=1}^{W}\max(0, |D_v[i+1,j] - D_v[i,j]| - \tau)^2$ measure horizontal and vertical depth discontinuities respectively. Here, $D_v[i,j]$ represents the rendered depth at pixel $(i,j)$ in view $v$, $\tau = 0.2$ is a threshold parameter that preserves legitimate geometric edges while penalizing artificial depth jumps, and the $\max(0,\cdot)$ function ensures that only depth gradients exceeding the threshold contribute to the loss. This formulation maintains depth smoothness across multiple viewpoints while preserving surface integrity, effectively reducing visual artifacts.

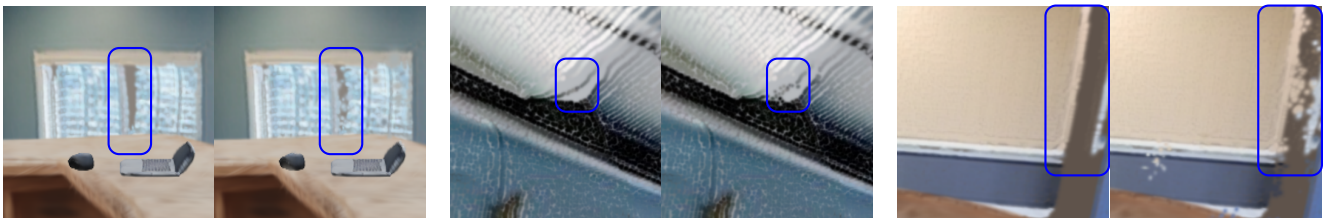
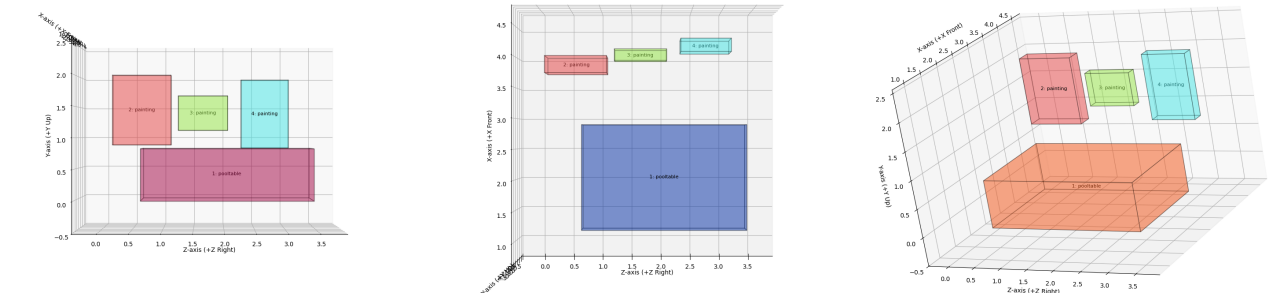
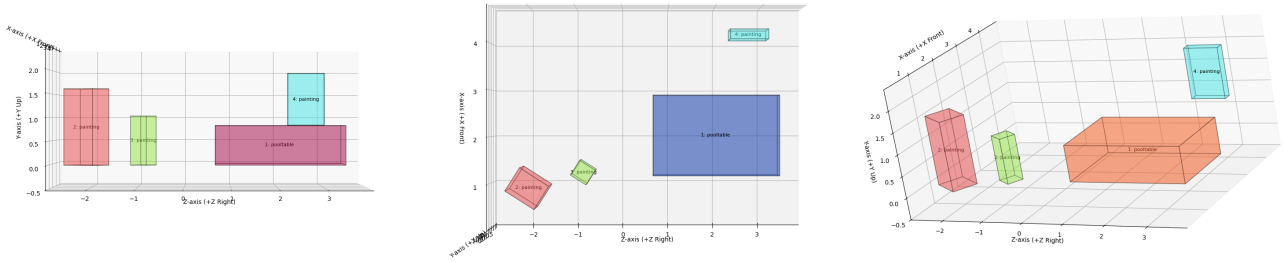


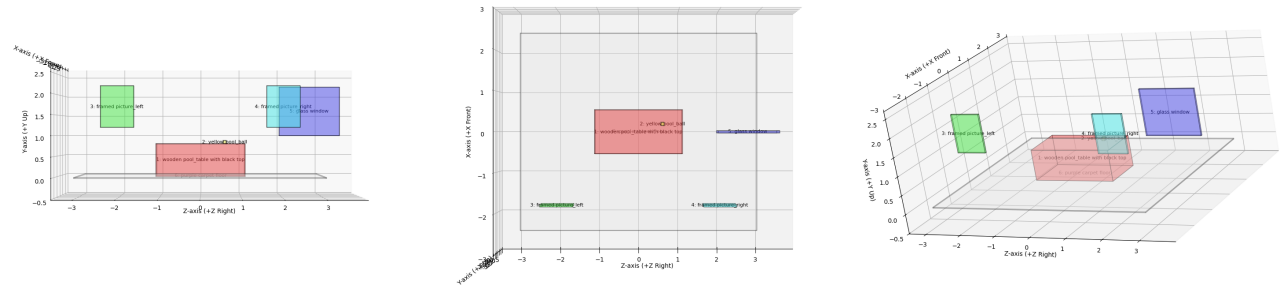
Figure 12. Qualitative results for scene layout generation evaluated on randomly sampled scenes from the SUN RGB-D NYU subset. LLM-based methods frequently identify objects beyond ground truth annotations, often detecting additional or more semantically descriptive entities. Spatial arrangement of objects varies between methods, and semantically equivalent objects are treated interchangeably.



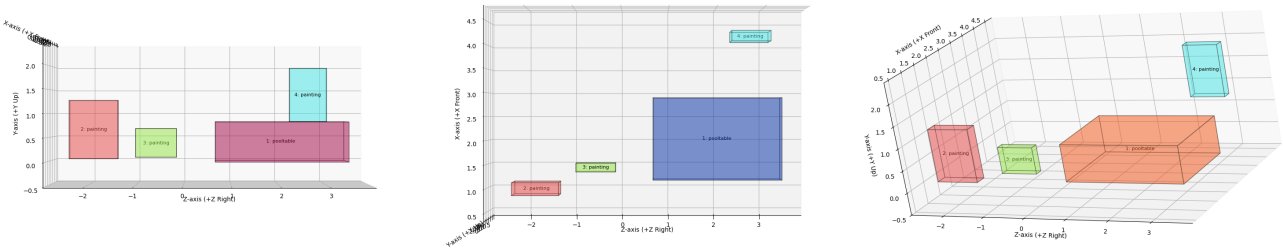
(a) Ground Truth



(b) GDM

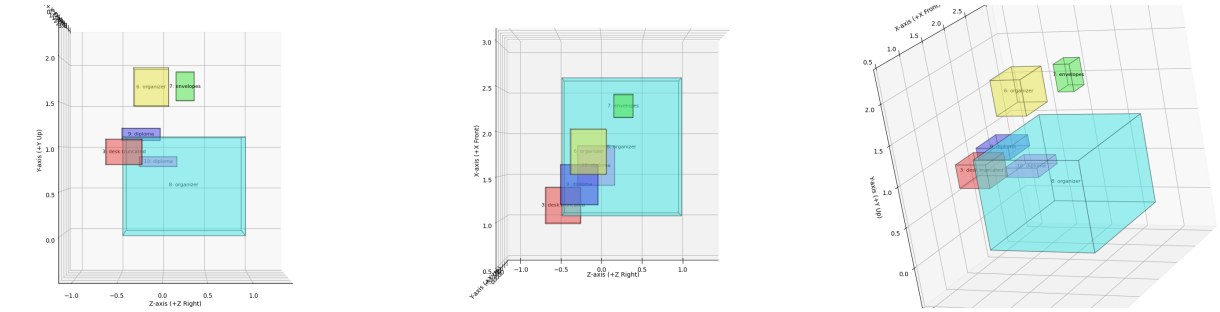


(c) LLM

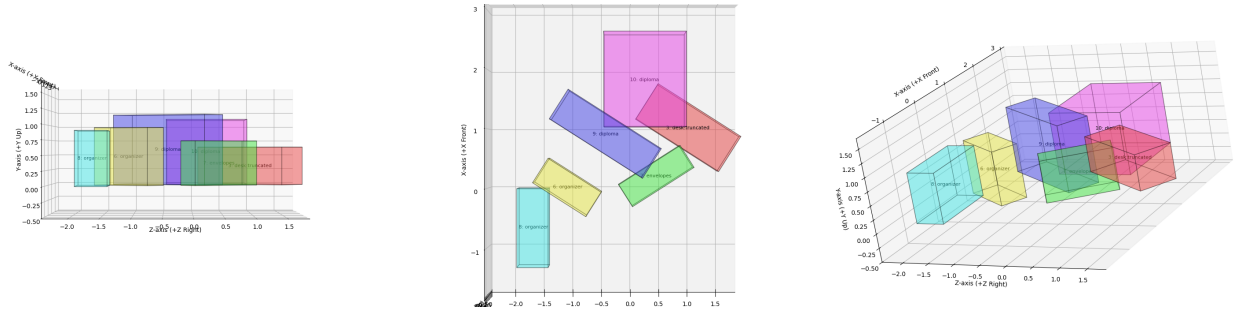


(d) Consistency

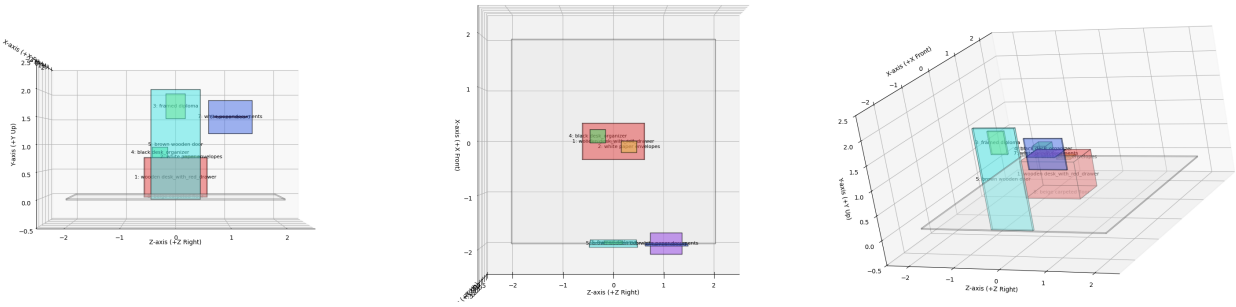
Figure 13. Qualitative results for scene layout generation evaluated on randomly sampled scenes from the SUN RGB-D NYU subset. LLM-based methods frequently identify objects beyond ground truth annotations, often detecting additional or more semantically descriptive entities. Spatial arrangement of objects varies between methods, and semantically equivalent objects are treated interchangeably.



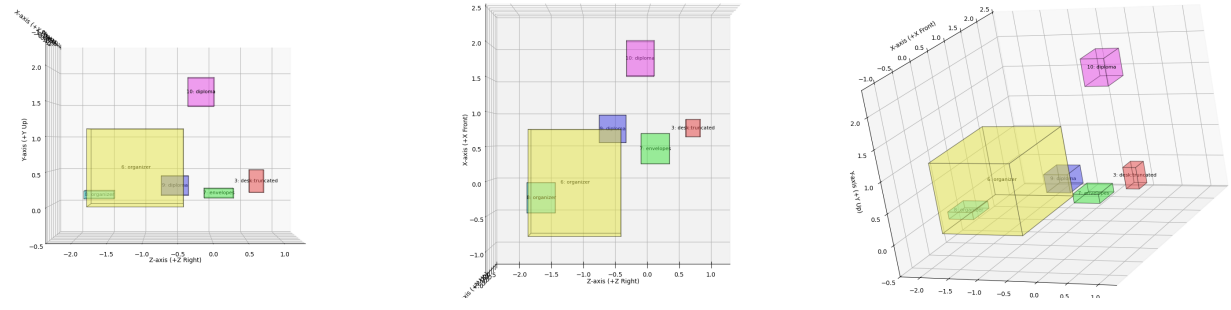
(a) Ground Truth



(b) GDM



(c) LLM



(d) Consistency