Synthesis of Interactive and Expansive Apartment Environments

ChunTeng Chen

National Yang Ming Chiao Tung University

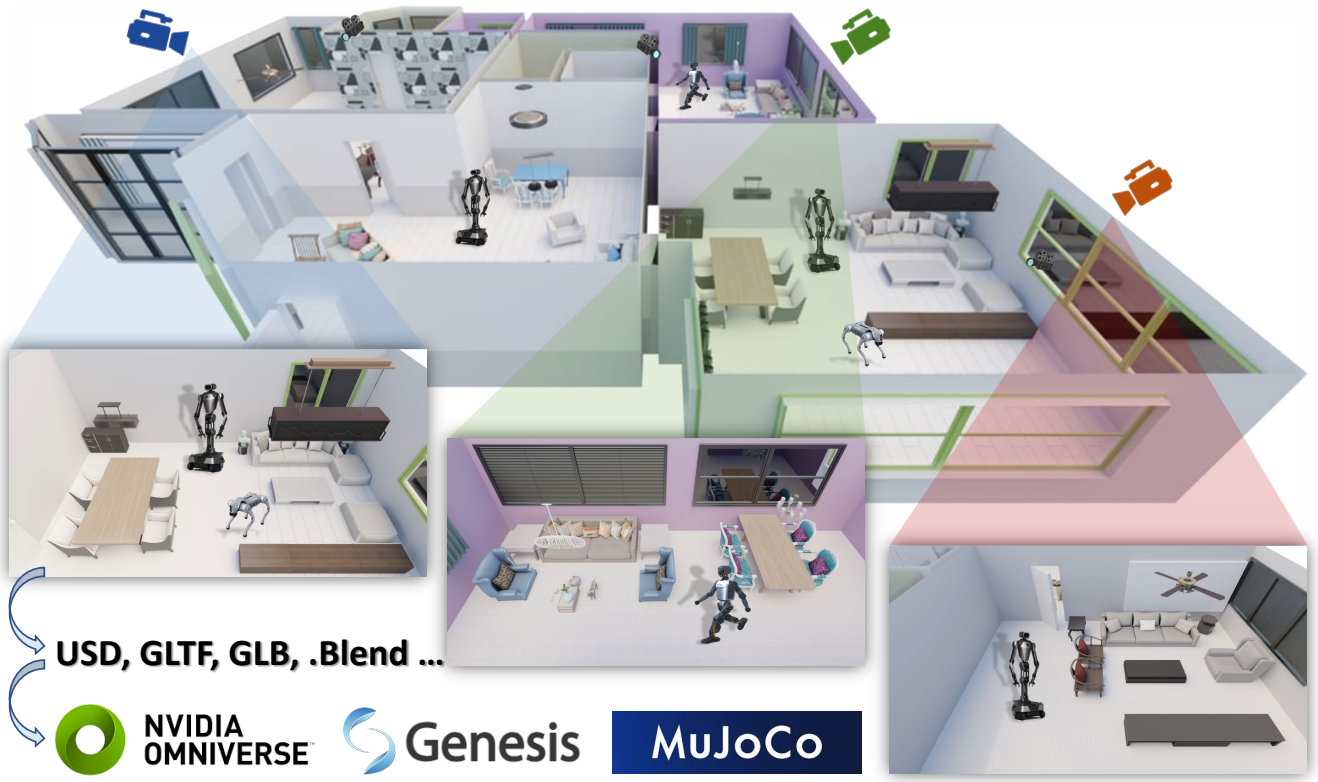


Figure 1. Overview. The framework generates apartment-scale 3D scenes from natural language prompts via LLM-guided floor plan generation, diffusion-based placement, and post-optimization ensuring articulated functionality and robot navigability.

Abstract

Synthesizing interactive environments at the scale of residential apartments provides a necessary foundation for advancing multimodal spatial intelligence. Current literature lacks methodologies for creating such expansive indoor spaces while maintaining the functional complexity and physical realism required for training multimodal large language models. Our framework addresses this void through a generative framework guided by natural language that integrates reasoning capabilities with posterior sampling via diffusion. This methodology utilizes spatial reasoning within language models to determine layout configurations before a floor plan is created in the scene. Integrating differentiable functions ensures that these complex layouts respect physical boundaries and prevent intersections between the moving components of different furniture pieces. Evaluations in various apartment configurations indicate that this approach creates environments with high levels of semantic consistency and operational utility. These generated worlds provide diverse data for training models in tasks requiring spatial reasoning and the manipulation of articulated objects. Achieving synthesis at the scale of a full apartment distinguishes this work from previous attempts and establishes a practical foundation for studies in multimodal understanding and embodied intelligence. Using this framework allows researchers to produce diverse and functionally interactive 3D worlds tailored to the needs of

multimodal agents.

1. Introduction

Development of multimodal spatial intelligence requires diverse 3D environments where models can perform reasoning across visual and linguistic modalities. Discrepancies between synthetic data and the physical world often occur due to a lack of spatial logic and functional interactivity in current modeling approaches. Most research in scene generation focuses on the level of individual rooms or isolated object clusters, leaving the synthesis of environments at the scale of an entire apartment largely unaddressed. Modeling these expansive residential units necessitates maintaining semantic consistency across several interconnected rooms while ensuring that every asset remains functionally interactive. Techniques relying on procedural generation produce large worlds but offer limited control over the specific distribution of assets required for training multimodal agents.

The architecture presented in this work addresses these limitations by synthesizing 3D scenes with high visual diversity and semantic coherence at the scale of a full apartment. This methodology translates natural language into structurally valid layouts that support the evaluation of spatial reasoning in multimodal models. Integration of large language models with differentiable constraints ensures that every generated environment meets specific requirements for interactive simulation. Achieving synthesis at this scale distinguishes this work from previous attempts and establishes a practical methodology for creating datasets for multimodal understanding and embodied intelligence. The contributions of this work are summarized as follows.

- **Interactive 3D Worlds for Multimodal Training.** This framework constructs environments featuring multiple interconnected rooms at the scale of a residential apartment. Unlike prior studies restricted to single rooms, the proposed method ensures semantic and functional coherence across an entire living unit to support complex spatial reasoning.
- **Strategic Control through Language Reasoning.** Integration of large language models allows for the translation of abstract prompts into precise spatial configurations. This mechanism ensures that the generated 3D environments align with the complex requirements of multimodal agents and spatial logic.
- **Guidance for Physical and Operational Usability.** Three distinct control strategies are implemented to guarantee physical realism within the 3D scene. Techniques for regulating object quantity and preventing collisions between moving parts are formulated through posterior sampling. A dedicated step for layout optimization ensures that the generated environments maintain sufficient navigable space for agents to perform multimodal tasks.

2. Related Work

Indoor Scene Layout Generation. Automated 3D indoor scene generation traditionally relied on computationally intensive procedural methods like Infinigen [15]. Recent learning-based approaches often utilize autoregressive models, such as ATISS [14], which frequently suffer from sequential error accumulation. Diffusion models [5, 11, 16] overcome this limitation by providing parallel generation, improved global coherence, and editing flexibility. The proposed framework adopts this diffusion paradigm. Effective training of these models relies on clean CAD-based datasets with well-structured geometry and part-level semantics. We utilize 3D-FRONT [7] and GAPartNet [8] instead of real-world scans like ScanNet [4] and Matterport3D [3], as real-world scans often contain noise and artifacts that hinder generative modeling.

Diffusion Model Guidance. Controllable synthesis requires mechanisms to steer the generative process. Early methods like classifier guidance [10] have evolved into generalized diffusion posterior sampling [1], which enables gradient-based control via any differentiable function. Recent studies apply this principle to enforce physical plausibility and robot reachability [17]. Our approach also leverages this method to apply diverse functional constraints during inference, avoiding the computational overhead of training specialized conditional models.

3. Method

Our framework adopts a multi-stage pipeline to generate controllable, apartment-scale 3D scenes for robot training, as illustrated in Fig. 2. The pipeline begins with an LLM-based parameter space generation (Sec. 3.1) that translates user prompts into low-level parameters for floor plan generation.

A diffusion model employing posterior sampling (Sec. 3.2) then populates these layouts with furniture assets. To ensure functional viability, we integrate three control mechanisms. During sampling, the model is guided by differentiable guidance functions: Object Quantity Control (Sec. 3.3) and the proposed Articulated Object Collision Constraint (Sec. 3.4) to maintain usability of movable parts. A Walkable Area Control post-processing step (Sec. 3.5) further refines spatial density to guarantee navigability. We also introduce a set of evaluation metrics (Sec. 3.6) to quantitatively evaluate the effectiveness of our methods.

3.1. LLM-Guided Parameter Space Generation

While we adopt the Infinigen [15] framework for its ability to generate room layouts through a simulated annealing process governed by twelve reward functions, its high-dimensional parameter space is notoriously non-intuitive for user control. To bridge this usability gap, we design an

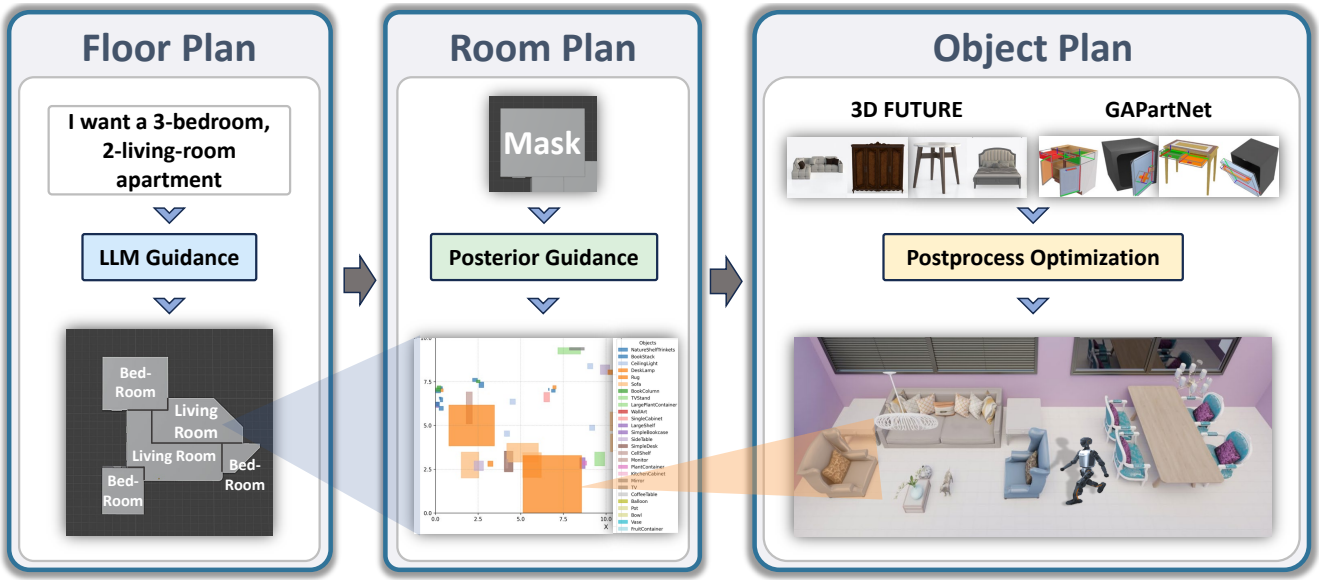


Figure 2. Overview of our apartment-scale generation pipeline. An LLM first guides procedural floor plan generation (Sec. 3.1), diffusion posterior guidance generates plausible room bounding boxes (Sec. 3.2, Sec. 3.3, Sec. 3.4), and 3D assets from 3D-FRONT/GAPartNet are refined via post-optimization to complete the layout (Sec. 3.5).

LLM-based parameterization framework that navigates this complex space to align stochastic generation with semantic intent, as shown in Fig. 3. This semantic-to-parameter mapping converts abstract user descriptions into concrete floor plans while preserving the structural diversity of the procedural kernels.

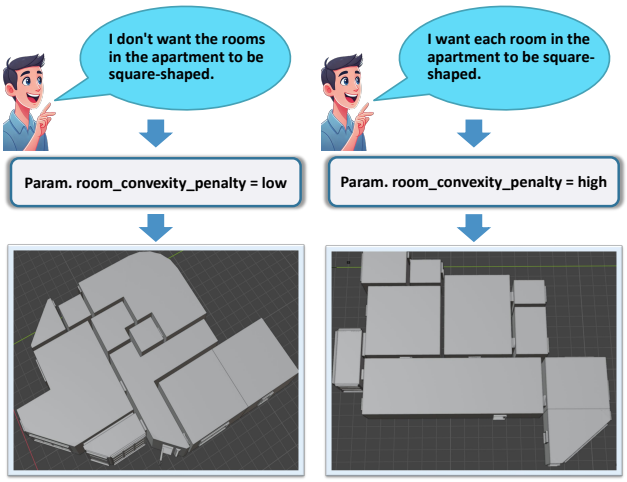


Figure 3. Illustration of our LLM-based Guidance. A low penalty (left) produces diverse, non-rectilinear layouts, whereas a high penalty (right) enforces square-shaped room layouts.

3.2. Diffusion Posterior Sampling

To enforce explicit constraints on the generated 3D scenes, our method steers the reverse diffusion trajectory using a composite guidance function, $\varphi(\cdot)$. This approach adapts the principles of diffusion posterior sampling to ensure structural validity in the generated layouts, as shown in the reverse process part of Fig. 4.

Object Feature. Following [16], we represent a 3D scene $\mathbf{x}$ as an unordered set of N objects, $\{\mathbf{o}_i\}_{i=1}^{N}$. Each object $\mathbf{o}_i$ is a vector $[\mathbf{l}_i, \mathbf{s}_i, \theta_i, \mathbf{c}_i, \mathbf{f}_i]$ encoding its location, size, orientation, semantics, and a latent shape feature. This latent space is derived from a pre-trained VAE, following [16, 17]. The generated latent feature $\mathbf{f}_i$ drives a nearest-neighbor search to retrieve the best-matching asset from either 3D-FRONT or GAPartNet, as shown in the post-processing stage of Fig. 4. This allows us to compose novel scenes that cohesively integrate both static and articulated objects.

Training. We adopt a constraint-guided learning strategy. Rather than using a standard denoising objective, we train the model $\boldsymbol{\epsilon}_\theta$ to predict the noise $\boldsymbol{\epsilon}$ while anticipating the constraint gradient $\mathbf{g}$ derived from the functional constraints φ. This anticipatory mechanism actively aligns the latent manifold with physically plausible configurations early in the pipeline. Optimization is achieved by minimizing a guided $\mathcal{L}_2$ loss:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\|(\boldsymbol{\epsilon} - \lambda\boldsymbol{\Sigma}\mathbf{g}) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathcal{F})\|_2^2\right] \quad (1)$$

This approach embeds knowledge of the constraints directly into the model weights during the training phase.

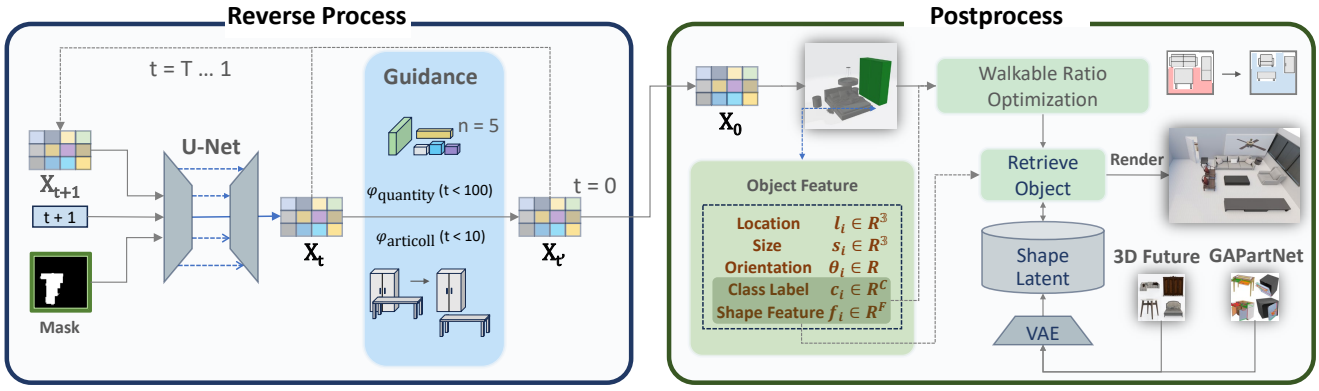


Figure 4. Guidance scheduling during the reverse diffusion process. Object quantity control is applied at $t < 100$ and articulated collision constraint at $t < 10$, followed by a final walkable-ratio optimization at $t = 0$ to generate a realistic scene.

Sampling. During inference, we execute an iterative reverse process starting from pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At each diffusion step t, the architecture initially predicts the statistical parameters $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$, which characterize the unguided posterior $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. The exact gradient of the composite guidance function $\nabla_{\mathbf{x}_t}\varphi(\mathbf{x}_t, \mathcal{F})$ is then computed. This gradient signal is scaled by a dedicated guidance weight λ to perturb the predicted mean, actively steering the generation trajectory towards functionally valid spatial regions:

$$\mathbf{x}_{t-1} \sim \mathcal{N}\Big(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathcal{F}) + \lambda\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathcal{F})\nabla_{\mathbf{x}_t}\varphi(\mathbf{x}_t, \mathcal{F}), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathcal{F})\Big) \tag{2}$$

Iterating this process yields the final sample $\mathbf{x}_0$. The complete procedure is shown in Algorithm 1.

3.3. Object Quantity Constraint

To control object quantity, we introduce a differentiable guidance function, $\varphi_{\text{quantity}}$, which operates on the predicted class logits during the reverse diffusion process. Our scene representation uses N_{max} potential object slots, where each slot's logits include a channel for an "empty" class, denoted c_i. To enforce a target count N_{target}, we define a binary target vector $\mathbf{T} \in \mathbb{R}^{N_{max}}$ specifying which of the N_{max} slots should be non-empty.

The guidance function is formulated as the Binary Cross-Entropy (BCEWithLogits) loss between the predicted "empty" logits and this target vector:

$$\varphi_{\text{quantity}}(\mathbf{x}) = \text{BCEWithLogits}(\{c_i\}_{i=1}^{N_{max}}, \mathbf{T}) \tag{3}$$

The gradient of this function, $\nabla_{\mathbf{x}_t}\varphi_{\text{quantity}}$, provides a direct signal during sampling, steering the model to populate the scene with the N_{target} desired objects.

3.4. Articulated Collision Constraint

Standard collision losses are insufficient as they strictly evaluate static geometry, ignoring the functional plausibility of movable parts. We introduce a differentiable guidance function, φ_{acoll}, to explicitly penalize these "functional collisions." During diffusion sampling, we identify if each object b_i in the scene $\mathcal{B}$ is articulated via semantic metadata. If so, we compute its functionally extended state, b_i', by expanding its bounding box along its primary axis of motion to encapsulate its operational sweep volume. For static

Algorithm 1: Guidance Sampling in Model

Modules: Model $p_\theta(\cdot|\mathcal{F})$, guidance functions $\varphi(\cdot) = \{\varphi_{\text{quantity}}(\cdot), \varphi_{\text{articoll}}(\cdot)\}$.

// constraint-guided learning

Input: 3D scene layout $\mathbf{x} = \{\mathbf{o}_1, \ldots, \mathbf{o}_N\}$ and floor plan $\mathcal{F}$, where N is a fixed number of objects.

repeat
- $\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathcal{F})$
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(\{1, \ldots, T\})$
- $\mathbf{x}_t = \sqrt{\hat{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\hat{\alpha}_t}\boldsymbol{\epsilon}$, $\tilde{\mathbf{x}}_0^t \sim p_\theta(\cdot)$
- $\theta = \theta - \eta\nabla_\theta\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \lambda\boldsymbol{\Sigma}\mathbf{g}\|_2^2$

until *converged*;

// one-step guided sampling

function sample $(\boldsymbol{\tau}^t, \varphi)$**:**
- $\boldsymbol{\mu} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathcal{F})$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathcal{F})$
- $\varphi(\mathbf{x}_t) = \gamma_1\varphi_{\text{quantity}}(\mathbf{x}_t) + \gamma_2\varphi_{\text{articoll}}(\mathbf{x}_t)$
- $\mathbf{x}_{t-1} = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu} + \lambda\boldsymbol{\Sigma}\nabla_{\mathbf{x}_t}\varphi(\mathbf{x}_t, \mathcal{F})|_{\mathbf{x}_t=\boldsymbol{\mu}}, \boldsymbol{\Sigma})$
- **return** $\mathbf{x}_{t-1}$

// constraint-guided generation

Input: initial scene layout $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = T, \ldots, 1$ **do**
- // sampling with optimization
- $\mathbf{x}_{t-1} =$ **sample**$(\mathbf{x}_t, \varphi)$

end

return $\mathbf{x}_0$

Algorithm 2: Walkable Area Optimization

Input: Generated scene layout S, ratio threshold τ, max iterations M, search depth k, object database $\mathcal{D}$

$r \leftarrow \text{AreaRatio}(S)$;
$iter \leftarrow 0$;
while $r < \tau$ ***and*** $iter < M$ **do**
  $\mathcal{O}_k \leftarrow$ Top k valid objects in S sorted by footprint area;
  replaced $\leftarrow$ False;
  for $o \in \mathcal{O}_k$ **do**
    $o^* \leftarrow \text{RetrieveSmallerCandidate}(o, \mathcal{D})$;
    if $o^* \neq \emptyset$ **then**
      $S \leftarrow \text{ReplaceObject}(S, o, o^*)$;
      replaced $\leftarrow$ True;
    end
  end
  if ***not*** *replaced* **then**
    break;
  end
  $r \leftarrow \text{AreaRatio}(S)$;
  $iter \leftarrow iter + 1$;
end
return S

objects, $b'_i = b_i$. The total penalty is the sum of pairwise 3D Intersection over Union (IoU) between each object's extended state b'_i and all other objects b_j:

$$\varphi_{\text{articoll}}(\mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \text{IoU}_{3D}(b'_i, b_j) \tag{4}$$

Because the guidance function is differentiable with respect to scene parameters, its gradient $\nabla_{\mathbf{x}_t} \varphi_{\text{articoll}}$ actively steers the reverse diffusion process away from obstructed configurations, ensuring the generated scenes remain functionally viable for embodied agents.

3.5. Walkable Area Control

Ensuring a specific walkable space ratio is critical for robotic navigation. Enforcing such a constraint directly within the diffusion sampling loop would be computationally prohibitive, likely requiring expensive spatial queries at every step, and could potentially destabilize the generative process. We therefore introduce an efficient post-processing optimization as shown in Algorithm 2, which decouples semantic layout generation from spatial density tuning, as shown in Fig. 4. Our algorithm iteratively refines the scene to meet a target ratio τ by modifying only object *sizes* while preserving their *placements*. This strategy retains the core semantic structure while guaranteeing navigability.

3.6. Proposed Task-Specific Evaluation Metrics

To quantify the efficacy of the control mechanisms detailed in Secs. 3.1 and 3.3 to 3.5, four task-specific evaluation metrics are established. These include LLM Controllability, Object Quantity Controllability, Articulated Object Collision Ratio, and Walkable Area Controllability.

3.6.1. LLM-Guided Layout Metric

To evaluate the structural and semantic fidelity of a generated graph $G_{\text{gen}} = (V_{\text{gen}}, E_{\text{gen}})$ against the ground-truth $G_{\text{gt}} = (V_{\text{gt}}, E_{\text{gt}})$, node similarity, constraint satisfaction, and edge similarity are measured.

Node Similarity. A maximum cardinality matching $M : V_{\text{gen}} \to V_{\text{gt}}$ constrained by node type ($T(v_{\text{gen}}) = T(v_{\text{gt}})$) is computed. The score is defined as the match size normalized by the larger graph size to penalize extraneous nodes:

$$S_{\text{node}}(G_{\text{gen}}, G_{\text{gt}}) = \frac{|M|}{\max(|V_{\text{gen}}|, |V_{\text{gt}}|)} \tag{5}$$

Constraint Satisfaction Score. This metric evaluates the area ratio distribution per room category ($R(G, c)$). The L1 distance between the generated and ground truth distributions is initially measured:

$$D_{L1} = \sum_{c \in \mathcal{C}} |R(G_{\text{gen}}, c) - R(G_{\text{gt}}, c)| \tag{6}$$

The normalized constraint satisfaction score $S_{\text{constraint}} \in [0, 1]$ is defined as:

$$S_{\text{constraint}}(G_{\text{gen}}, G_{\text{gt}}) = 1 - \frac{1}{2} D_{L1} \tag{7}$$

This formulation quantifies how accurately the synthesized spatial proportions adhere to the intended architectural specifications.

Edge Similarity. Utilizing the node matching M, the set of matched edges E_{match} is identified, where edges in E_{gen} possess corresponding nodes under M that are simultaneously connected in E_{gt}:

$$E_{\text{match}} = \{(u, v) \in E_{\text{gen}} \mid (M(u), M(v)) \in E_{\text{gt}}\} \tag{8}$$

The score is normalized by the larger edge set to penalize spurious connections. This topological validation secures structural navigability by explicitly discouraging physically impossible spatial intersections:

$$S_{\text{edge}}(G_{\text{gen}}, G_{\text{gt}}) = \frac{|E_{\text{match}}|}{\max(|E_{\text{gen}}|, |E_{\text{gt}}|)} \tag{9}$$

3.6.2. Object Quantity Control Metric

To assess the capability to dictate the number of objects within a synthesized scene, a targeted quantitative evaluation is conducted. The architecture is prompted to generate rooms containing a specific target number of objects,

N_{target}. A scene S_i is considered a match if its generated object count, denoted $N(S_i)$, exactly equals the target. The quantity ratio (Q_{ratio}) for a designated target quantity over a set of M scenes is formally defined as:

$$Q_{\text{ratio}}(N_{\text{target}}) = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}(N(S_i) = N_{\text{target}}) \tag{10}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. For each target quantity, this ratio is calculated over $M = 100$ generated scenes.

3.6.3. Articulation Collision Metric

To quantitatively validate the effectiveness of the proposed articulated object collision constraint φ_{acoll}, we evaluate the incremental contribution of this functional module against a stochastic diffusion baseline [17]. For each setup, a test set of 100 scenes is generated and subsequently post-processed by transforming all articulated objects into their functionally extended states to simulate real-world robotic manipulation.

The metric R_{acoll} is introduced as the primary indicator, defined as the proportion of articulated objects involved in functional collisions. For a given scene S, it is calculated as:

$$R_{\text{acoll}}(S) = \frac{1}{N_A}\sum_{j\in\mathcal{A}}\mathbb{I}\left(\max_{i\neq j}\left(\text{IoU}_{3D}(b'_i, b'_j)\right) > 0\right) \tag{11}$$

where $\mathcal{A}$ is the set of articulated objects in the scene ($N_A = |\mathcal{A}|$), and b'_i, b'_j represent the functional bounding boxes. For an articulated object, this constitutes its volume in the extended state; for a static object, it remains its original bounding box. A lower R_{acoll} score signifies superior functional plausibility.

3.6.4. Walkable Area Controllability Metric

To quantitatively gauge the navigability and spaciousness of a generated scene, the ratio R_{walkable} is defined. This metric is computed as the ratio of the total unobstructed floor area relative to the overall area of the room. Let A_{room} denote the total floor plan area. For a scene containing N objects with individual footprints A_i, the total walkable area A_{walkable} equals the room area minus the sum of all object footprints. The ratio is formally defined as:

$$R_{\text{walkable}} = \frac{A_{\text{room}} - \sum_{i=1}^{N} A_i}{A_{\text{room}}} \tag{12}$$

The SR over a set of M generated scenes for a given threshold τ_{walkable} is then formulated as:

$$SR(\tau_{\text{walkable}}) = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}(R_{\text{walkable}}(S_i) \geq \tau_{\text{walkable}}) \tag{13}$$

4. Experiment

4.1. Implementation Detail

Datasets. Our pipeline uses three specialized datasets. We train the layout generator on 14,629 indoor scenes from 3D-FRONT [7] and populate the generated layouts with textured assets from 3D-FUTURE [6], which contains 16,563 furniture models. To enable interactivity, we use GAPartNet [8], which provides part-level semantics and pose data for 8,489 parts across 1,166 objects.

Baselines. We compare with three baselines to demonstrate methodological progression. ATISS [14] is an autoregressive Transformer for sequential object generation, DiffuScene [16] employs diffusion to improve global consistency, and PhyScene [17] adds physics-based guidance for physically plausible scene synthesis.

Evaluation Metrics. We assess layout quality using Fréchet Inception Distance (FID) [9], Kernel Inception Distance (KID) [2], Scene Classification Accuracy (SCA), and Category KL divergence (CKL) following [16]. Constraint-specific metrics are defined in Sec. 3.6 and evaluated in Secs. 4.3 to 4.6.

4.2. Conditioned Scene Synthesis Evaluation

As shown in Tab. 1, Our proposed method matches or exceeds baselines in single-room generation. We also extend our framework to apartment-scale synthesis, as shown in Fig. 5. Since existing baselines are limited to single rooms, precluding direct quantitative comparisons on larger scales, our multi-room results showcase a unique and robust capability unaddressed by prior methods.

Table 1. **Floor-conditioned Scene Synthesis.** We compare our proposed method with baselines.

Method	FID ↓	KID ↓	SCA	CKL ↓
ATISS	30.19	0.0010	49.14	0.0028
DiffuScene	**25.00**	**0.0004**	51.78	0.0031
PhyScene	25.52	0.0006	**50.10**	0.0025
Ours(Single)	29.02	**0.0004**	49.11	**0.0024**
Ours(Multi.)	29.14	0.0006	49.04	0.0025

4.3. LLM-Guided Layout Generation Evaluation

As shown in Fig. 5, our framework utilizes LLMs to translate textual prompts into complex floor plans. The resulting layouts are quantitatively evaluated against ground-truth graphs that perfectly satisfy the given high-level constraints. We use S_{node}, $S_{\text{constraint}}$, and S_{edge} defined in Sec. 3.6.1 to measure the semantic and structural fidelity of the generated floor plans. Our method achieves extremely high scores across all three metrics, as shown in Tab. 2.

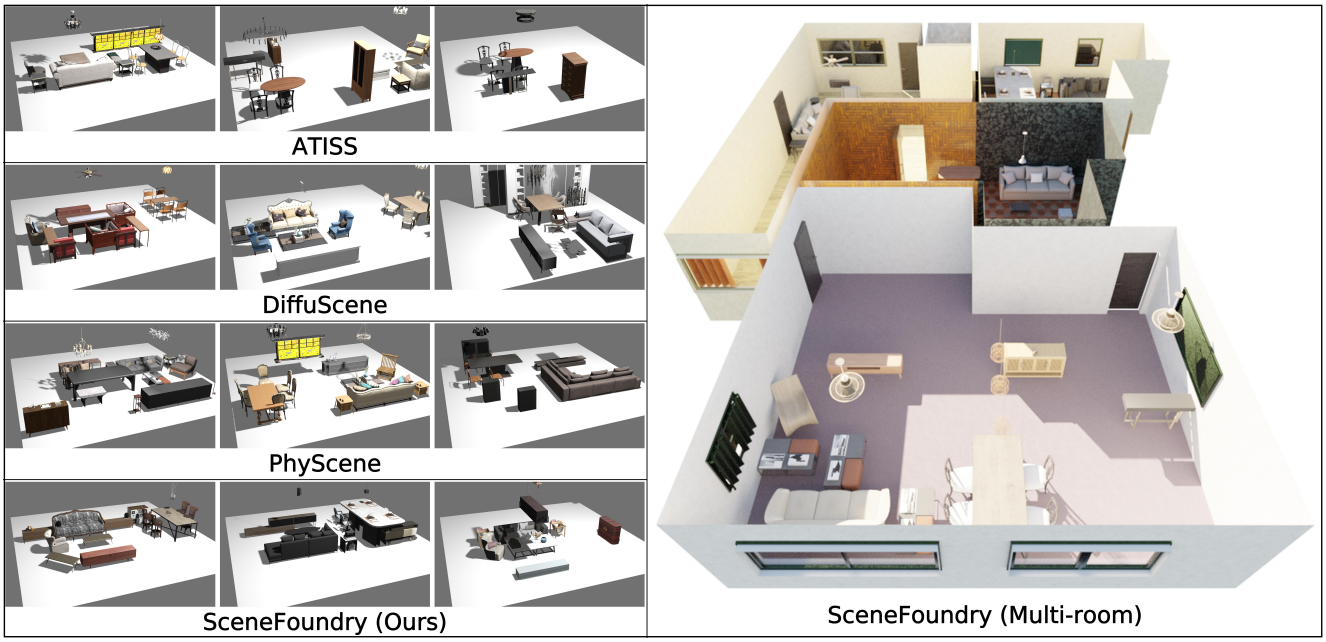


Figure 5. Qualitative comparison of conditioned scene synthesis results among PhyScene, ATISS, DiffuScene, and Ours.

Table 2. Results for our LLM Controllability experiment.

Method	$S_{\text{node}} \uparrow$	$S_{\text{constraint}} \uparrow$	$S_{\text{edge}} \uparrow$
Ours (LLM control)	**0.989**	**0.923**	**0.954**

4.4. Object Quantity Controllability Evaluation

We evaluate our control mechanism using Q_{ratio}, as proposed in Sec. 3.6.2. As Tab. 3 shows, our method consistently achieves a high Q_{ratio} (≥ 0.95). This demonstrates that our constraint effectively reshapes the baseline's Gaussian output into a stable, uniform distribution of targeted quantities.

Table 3. Q_{ratio} **of generating scenes.**

N_{target}	Q_{ratio}	N_{target}	Q_{ratio}	N_{target}	Q_{ratio}
5	0.95	9	0.96	13	0.96
6	0.95	10	0.97	14	0.95
7	0.96	11	0.96	15	0.95
8	0.95	12	0.96	16	0.95

4.5. Articulated Collision Evaluation

Our Articulated Collision Constraint ensures functional clearance for movable assets. As shown in Fig. 6, while unconstrained baselines often generate obstructed parts, our constraint effectively eliminates these functional collisions. Our proposed method achieves a significantly lower R_{acoll} and higher R_{reach} (Sec. 3.6.3) compared to the baseline (Tab. 4), demonstrating superior accessibility and task-readiness in the synthesized environments.

Table 4. Comparison of R_{acoll} and R_{reach}. Our method drastically reduces functional collisions ($R_{\text{acoll}} \downarrow$) and improves object accessibility ($R_{\text{reach}} \uparrow$).

Method	$R_{\text{acoll}} \downarrow$	$R_{\text{reach}} \uparrow$
Baseline (w/o $\varphi_{\text{articoll}}$)	0.191	0.742
Ours (w/ $\varphi_{\text{articoll}}$)	**0.109**	**0.808**

Figure 6. Visualization of the Articulated Object Collision Constraint. Synthesized scenes without the constraint (top) show obstructed articulated furniture, while applying the constraint (bottom) enables proper motion and functional layouts.

4.6. Walkable Area Controllability Evaluation

We test thresholds ranging from 0.60 to 0.95 in the $M = 100$ generated scenes. For each threshold, we compare the $SR(\tau_{\text{walkable}})$ (Sec. 3.6.4) across thresholds contrasts unconstrained baselines with our active constraints. Our method significantly increases the SR on all tested thresholds, as shown in Fig. 7. Qualitative examples confirm that the constraint maintains sufficient free space for navigation while preserving realistic scene density, as shown in Fig. 8.

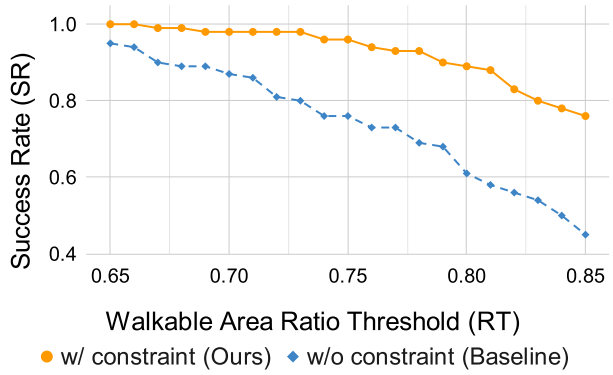


Figure 7. SR versus Walkable Area Ratio Threshold (R_T). Walkable Area Control (orange) consistently outperforms the baseline (blue), ensuring navigable layouts.

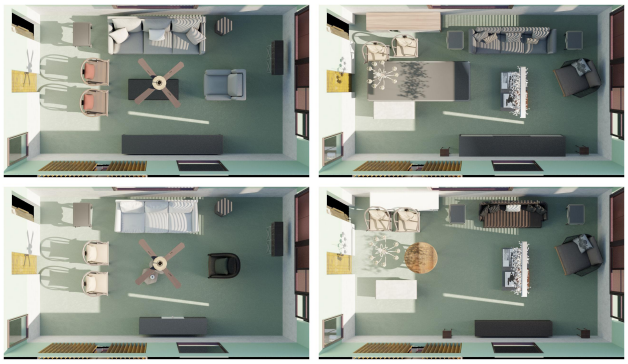


Figure 8. Visualization of Walkable Area Control.

Table 5. Ablation study on the use of guidance functions.

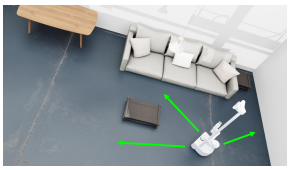
ArtiCollision	Walkable Ratio	$\text{Col}_{\text{obj}} \downarrow$	$\text{R}_{\text{reach}} \uparrow$
-	-	0.279	0.742
✓	-	0.267	0.808
-	✓	0.250	0.782
✓	✓	**0.249**	**0.830**

4.7. Embodied Agent Evaluation

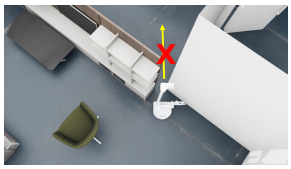
We deploy a NAV2 navigation stack [13] for navigation and a CogACT [12] for manipulation across 1000 synthesized scenes to evaluate task readiness (Tab. 6). Our proposed method achieves **99.8%** SR_{nav}, outperforming the baseline through Walkable Area Control that ensures collision-free pathways (Figs. 9a and 9b). For manipulation, our model attains **99.2%** SR_{interact} (Figs. 9c and 9d). While baselines often produce obstructed articulated parts, our Articulated Object Collision Constraint preserves functional clearance and improves interaction success.

Table 6. **Embodied Navigation and Manipulation Performance.** Navigation (SR_{nav}) and interaction (SR_{interact}) success rates across 1000 scenes in Isaac Sim.

Method	SR_{nav} (%) ↑	SR_{interact} (%) ↑
Ours (w/o φ)	95.2	94.7
Ours (w/ φ)	**99.8**	**99.2**



(a) Navigation Success



(b) Navigation Failure

(c) Interact Success

(d) Interact Failure

Figure 9. Qualitative Evaluation of Robot Tasks.

4.8. Integrated Functional Analysis

We evaluate our guidance mechanisms following [17] using: collision ($\text{Col}_{\text{obj}} \downarrow$) and reachability ($R_{\text{reach}} \uparrow$), as shown in Tab. 5. The full implementation achieves a minimum collision value of 0.249 and a maximum reachability of 0.830, validating the effectiveness of using differentiable functions to supervise the synthesis of expansive 3D worlds.

5. Conclusion

Synthesis of 3D environments at the scale of an apartment provides a foundation for multimodal spatial intelligence. Integrating language reasoning with diffusion ensures spatial coherence and topological connectivity across functional zones. These interactive worlds facilitate spatial reasoning and training for multimodal agents.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 843–852, 2023. 2

[2] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. 6

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments, 2017. 2

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794. Curran Associates, Inc., 2021. 2

[6] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture, 2020. 6

[7] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10933–10942, 2021. 2, 6

[8] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7081–7091, 2023. 2, 6

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 6

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 2

[12] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. 8

[13] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2718–2725, 2020. 8

[14] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems*, pages 12013–12026. Curran Associates, Inc., 2021. 2, 6

[15] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21783–21794, 2024. 2

[16] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20507–20518, 2024. 2, 3, 6

[17] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16262–16272, 2024. 2, 3, 6, 8