2025 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) | 979-8-3315-8761-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISMAR67309.2025.00055

DGDiff: Immersive 3D Indoor Scene Synthesis via Dialog-Graph Conditioned Diffusion

Yusen Liu
[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University

Xinyu Zhang
[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University

Qichuan Geng
[2] Information Engineering College, Capital Normal University

Zhong Zhou *
[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[3]Zhongguancun Laboratory

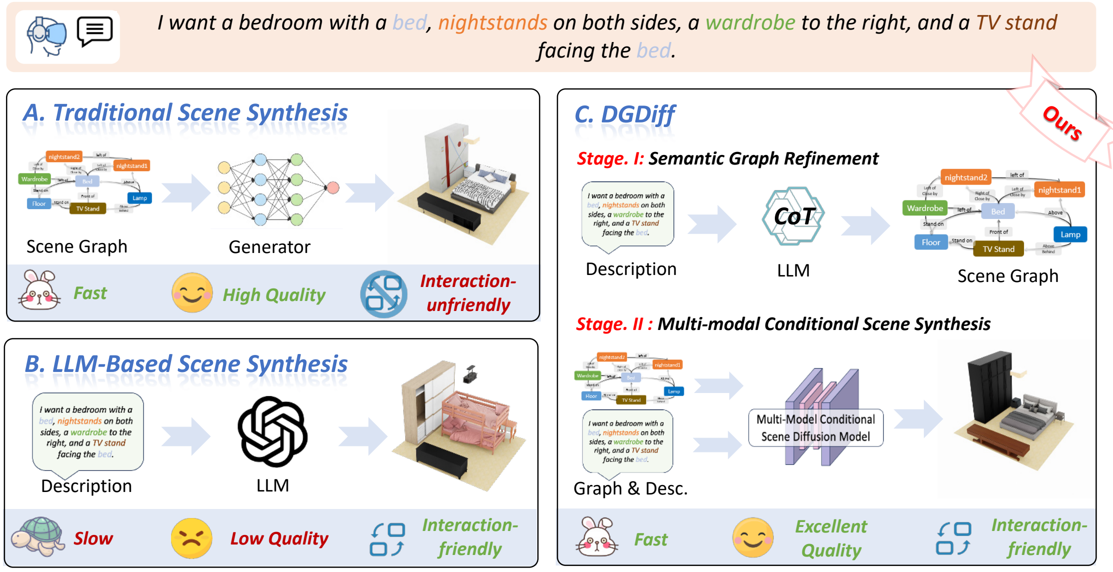

I want a bedroom with a bed, nightstands on both sides, a wardrobe to the right, and a TV stand facing the bed.
A. Traditional Scene Synthesis
Scene Graph
Generator
Fast
High Quality
Interaction-unfriendly
B. LLM-Based Scene Synthesis
I want a bedroom with a bed, nightstands on both sides, a wardrobe to the right, and a TV stand facing the bed.
Description
LLM
Slow
Low Quality
Interaction-friendly
C. DGDiff
Ours
Stage. I: Semantic Graph Refinement
Description
CoT
LLM
nightstand2
nightstand1
Wardrobe
Bed
Lamp
Floor
TV Stand
Scene Graph
Stage. II : Multi-modal Conditional Scene Synthesis
Multi-Model Conditional Scene Diffusion Model
Graph & Desc.
Fast
Excellent Quality
Interaction-friendly

Figure 1: (A) Traditional scene synthesis methods are fast and high-quality but lack user-system interactivity. (B) LLM-based methods are interactive but slow and low-quality. (C) We present DGDiff for immersive 3d scene synthesis, which consists of two stages: Stage I employs a dialog-graph synthesis approach with Chain-of-Thought (CoT) reasoning to generate structured scene graphs from user descriptions, while Stage II employs a multi-modal diffusion model to create immersive, high-quality scenes.

Abstract

Immersive 3D indoor scene synthesis is essential for applications such as AR/VR and 3D content creation. However, existing approaches fail to meet the immersive AR/VR requirements for fidelity, user–system interaction, and production speed simultaneously. Traditional scene synthesis methods are overly rigid, limiting user interactivity, whereas large language model (LLM)-based approaches suffer from slow response times and imprecise spatial structuring. To address these issues, we propose DGDiff, a novel dialog-graph conditioned diffusion framework for immersive, controllable, continuous synthesis and editing of 3D indoor scenes. This framework combines a conversational module powered by LLMs with a multimodal diffusion model. The conversational module translates user dialogue into structured semantic graphs, while the diffusion model integrates textual and graph-based conditions to synthesize realistic, editable indoor scenes. Experimental results demonstrate that DGDiff outperforms single-modality baselines, achieving an improvement of over 10% in FID and a reduction of approximately 30% in response time for dynamic scene interactive editing, offering an immersive and user-friendly synthesis experience. Project page: `https://gitee.com/VR_NAVE/dgdiff.git`

Index Terms: 3D Indoor Scene Synthesis, Dialog-graph, Multi-modal Diffusion Model

1 Introduction

In the rapidly advancing domain of 3D content creation [21, 46, 18, 17, 49, 9, 40], scene synthesis serves as a fundamental capability supporting a variety of critical applications, including augmented and virtual reality (AR/VR), digital generation, and interior design [4, 54, 22, 3, 20, 27, 25]. These applications demand immersive 3D scene synthesis technology, which require not only high-quality and controllable scene synthesis but also efficient editing capabilities to meet diverse user demands. However, achieving high-fidelity and semantically aligned scene synthesis while maintaining interactive usability remains a significant challenge. Two predominant methods have been proposed, each addressing parts of this challenge: (a) the traditional scene synthesis method based on a semantic graph, which focuses on high-fidelity and semantically aligned scene synthesis, and (b) the large language models (LLMs) guided scene synthesis model, which emphasizes maintaining interactive usability.

Traditional scene synthesis methods [41, 34, 45] typically rely on textual descriptions or semantic graphs as inputs. While these

[*] e-mail: zz@buaa.edu.cn

2473-0726/25/$31.00 ©2025 IEEE
DOI 10.1109/ISMAR67309.2025.00055

approaches generally meet basic speed requirements, they often involve complex and unintuitive input formats, limiting user interactivity and editing flexibility. As shown in Fig. 1-A, these methods lack interactive capabilities and frequently generate unrealistic results, including physically implausible layouts and object overlaps, thereby compromising both generation quality and user experience.

Recent methods [11, 13, 50] leverage large language models (LLMs) to directly generate complete indoor scenes from high-level prompts. While these approaches enhance interactivity by enabling natural language control, they often involve a series of complex and cumbersome steps, resulting in slow inference speeds. Moreover, due to limited spatial and structural precision, the generated scenes may satisfy semantic requirements but fail to align with real-world plausibility. As illustrated in Fig. 1-B, LLMs can lead to implausible combinations such as a nightstand being paired with a double bed in an inconsistent layout.

To improve interactivity and editing flexibility, we propose DGDiff, which enables high-quality, controllable, interactive, and fast immersive scene generation. To tackle the challenge of controllable and interpretable indoor scene synthesis, we introduce Semantic Graph Refinement via Dynamic Prompting, guided by a cross-modal semantic decomposition strategy inspired by Chain-of-Thought (CoT) reasoning [48, 44]. This model is a text-driven, progressive scene topology synthesis pipeline comprising three key stages: (1) Object Extraction and Scene Completion, (2) Functional Group Formation, and (3) Group Layout Optimization. This stepwise design facilitates the transformation of sparse textual descriptions into high-fidelity semantic scene graphs. By explicitly modeling functional groupings and spatial constraints, our method significantly enhances the logical consistency and physical plausibility of complex indoor environments, enabling more controllable, structured, and semantically coherent scene synthesis.

To address the challenge of inference speed and synthesis quality, we propose a multi-modal conditional scene diffusion model that jointly leverages textual descriptions and semantic graphs to guide the synthesis process. The textual modality is designed to encode rich global semantics, including object categories, spatial relationships, and functional intent. In parallel, the semantic graph component is introduced to capture fine-grained spatial structures and local dependencies. By integrating these complementary modalities, the model aims to preserve the semantic coherence of the input while improving spatial accuracy and controllability. Our multi-modal conditional architecture not only supports more precise, semantically aligned, and efficient indoor scene synthesis but also achieves faster generation through the multi-modal conditional scene diffusion model. Experimental results demonstrate that DGDiff outperforms a single-modality baseline, achieving an improvement of over 10% in FID and a reduction of approximately 30% in response time for interactive editing, offering an immersive and user-friendly synthesis experience.

To summarize, our contributions are listed as follows.

- We propose DGDiff, a novel paradigm for immersive 3D indoor scene synthesis based on dialog-graph conditioned diffusion. By integrating large language models with diffusion models, DGDiff introduces a text-driven semantic topology construction process that enables immersive, controllable, continuous synthesis and editing of indoor scenes.

- We design a multi-modal conditional diffusion model that leverages text descriptions for global semantic guidance and incorporates semantic graphs for fine-grained local spatial constraints. By bridging high-level user intent and structured spatial understanding, this multi-modality strategy facilitates the synthesis of semantically coherent, spatially consistent, and detail-rich scenes.

- We conduct comprehensive experiments on the 3D-FRONT dataset to validate the effectiveness of our method. Results show that our approach achieves higher generation fidelity and faster inference in both scene synthesis and editing tasks.

2 Related Work

2.1 Scene Semantic Graph Generation

Early graph-based scene synthesis methods leveraged hierarchical structures [28, 14] to encode object relationships. With the emergence of scene graphs [55, 33], recent works[30, 52] adopted conditional VAE[23] and graph convolutional network[19] or graph attention network[42] to generate object layouts with relational constraints. While these methods achieved fine-grained control over object arrangements, their graph generation modules [43, 52] operated without explicit semantic constraints or were restricted to categorical attributes, such as nodes encoding only object classes. This resulted in semantically impoverished graphs lacking material specifications and texture-aware relationships. Our work advances this paradigm by integrating CLIP-based semantic encoding for comprehensive node attributes.

2.2 Text Guided Scene Synthesis

Early text-based approaches [7, 6, 31, 37] used procedural modeling with handcrafted rules, while transformer-based approaches [45, 34, 41] aligned text and layouts via cross-attention. LayoutGPT[11] employs style sheet language to generate intermediate 2D / 3D layouts using visual planning in context, excelling in numerical spatial reasoning. HOLODECK [50] integrated commonsense reasoning of GPT-4 with physics-aware optimization, formalizing spatial constraints (proximity, alignment) via depth-first search to generate modular 3D scenes. AnyHome[13] pioneers open-vocabulary text-to-3D conversion by combining LLM-guided template prompts with score distillation sampling. Although text guidance allows for interactive input, it often fails to accurately capture 3D spatial constraints, which can result in physically implausible layouts, such as overlapping objects. Directly employing large language models for layout generation typically involves numerous network requests, leading to slow generation speeds. In contrast, by introducing semantic graphs to constrain spatial relationships, our method simultaneously improves visual fidelity and efficiency in scene synthesis, while preserving user-friendly interaction.

2.3 Graph Guided Scene Synthesis

Scene graphs have been shown to be effective for controllable 3D synthesis, as demonstrated in EchoScene [51] and other SG-FRONT-based methods [52]. These approaches encoded graphs through GNNs [24, 42] to predict object poses conditioned on predefined relationships. InstructScene [29] introduced a semantic graph before translating natural language instructions into structured scene graphs via a two-stage diffusion process. However, they required fully specified graphs as input, which constitutes a significant bottleneck for practical applications. While semantic graphs provided structured scene representations, current construction methods required manual template design[51] or task-specific deep learning models[29], suffering from poor user-friendliness and high domain adaptation costs. These methods ensured generation quality and speed to some extent, but failed in user-system interactivity and scene controllability, unable to support continuous, real-time, and user-friendly editing. Our LLM-based approach enables semantic scene graph generation through open-vocabulary language interaction, achieving comparable relationship accuracy.

3 Our Approach

We propose DGDiff, a conversational framework that addresses key limitations of existing 3D scene synthesis methods by synergizing large language model (LLM)-guided interactions with dual-

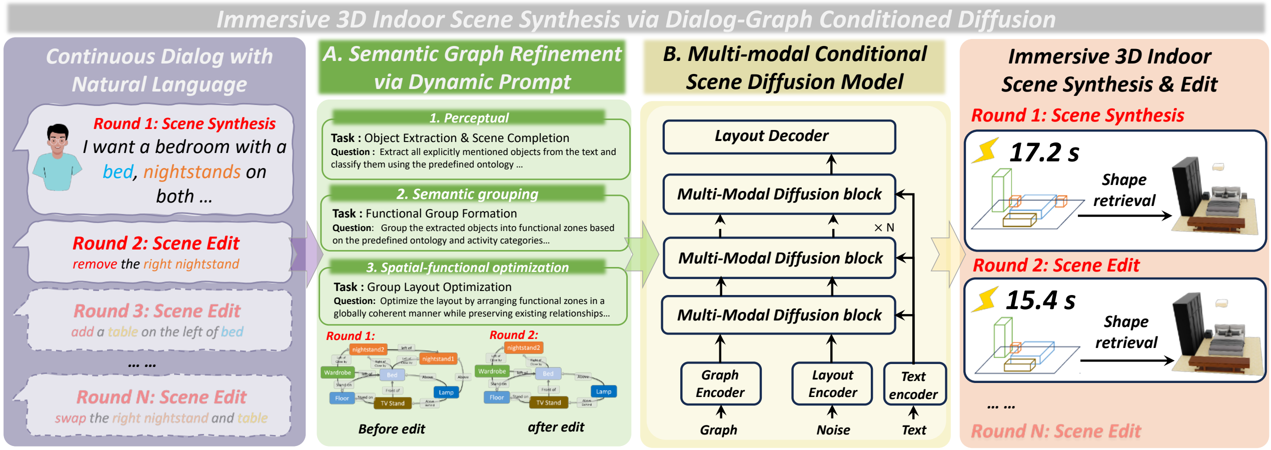

Immersive 3D Indoor Scene Synthesis via Dialog-Graph Conditioned Diffusion
Continuous Dialog with Natural Language
Round 1: Scene Synthesis
I want a bedroom with a bed, nightstands on both ...
Round 2: Scene Edit
remove the right nightstand
Round 3: Scene Edit
add a table on the left of bed
... ...
Round N: Scene Edit
swap the right nightstand and table
A. Semantic Graph Refinement via Dynamic Prompt
1. Perceptual
Task : Object Extraction & Scene Completion
Question : Extract all explicitly mentioned objects from the text and classify them using the predefined ontology ...
2. Semantic grouping
Task : Functional Group Formation
Question: Group the extracted objects into functional zones based on the predefined ontology and activity categories...
3. Spatial-functional optimization
Task : Group Layout Optimization
Question: Optimize the layout by arranging functional zones in a globally coherent manner while preserving existing relationships...
Round 1:
nightstand2
nightstand1
Wardrobe
Bed
Lamp
Floor
TV Stand
Before edit
Round 2:
nightstand2
Wardrobe
Bed
Lamp
Floor
TV Stand
after edit
B. Multi-modal Conditional Scene Diffusion Model
Layout Decoder
Multi-Modal Diffusion block
× N
Multi-Modal Diffusion block
Multi-Modal Diffusion block
Graph Encoder
Layout Encoder
Text encoder
Graph
Noise
Text
Immersive 3D Indoor Scene Synthesis & Edit
Round 1: Scene Synthesis
17.2 s
Shape retrieval
Round 2: Scene Edit
15.4 s
Shape retrieval
... ...
Round N: Scene Edit

Figure 2: **Method overview.** We propose DGDiff, which consists of two key components:(1)Semantic Graph Refinement via Dynamic Prompt, which leverages a chain of thought to convert the input text into a semantically refined scene graph. (2)Multi-modal Conditional Scene Diffusion Model, which integrates the conditions from text and semantic graphs through a dual-conditioned Transformer mechanism to jointly guide scene synthesis. During the editing phase, simple text prompts can be used to refine the semantic graph, thereby altering the generated scene.

conditioned diffusion models, as illustrated in Fig. 2. Specifically, we first briefly revisit the theoretical foundations of diffusion models and semantic scene graphs in Section 3.1. Building upon these foundations, DGDiff utilizes a two-phase approach: (a) Semantic Graph Refinement via Dynamic Prompting (Section 3.2), where sparse textual inputs are transformed into detailed semantic graphs through chain-of-thought (CoT) reasoning; and (b) Multi-modal Conditional Scene Diffusion (Section 3.3), which combines text embeddings for global semantic guidance with graph embeddings for precise object-level constraints, enabling coherent multi-modal fusion for accurate and immersive 3D indoor scene synthesis.

3.1 Preliminary

Diffusion Model (DM) [16] is generative framework based on Markov chains, which progressively denoise data. The forward process gradually adds Gaussian noise to data $\mathbf{x}_0$ according to a variance schedule β:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s), \tag{1}$$

where $\mathbf{x}_0 \in \mathbb{R}^d$ denotes the original data sample; $\mathbf{x}_t \in \mathbb{R}^d$ is the noised data at timestep t; $\{\beta_s\} \in (0,1)^T$ is the predefined variance schedule; and $\varepsilon \sim \mathcal{N}(0,\mathbf{I})$ is the standard Gaussian noise.

The reverse process learns to recover data by training a noise predictor ε_θ with the objective:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{t,\mathbf{x}_0,\varepsilon}\left[\|\varepsilon - \varepsilon_\theta(\mathbf{x}_t,t)\|_2^2\right], \tag{2}$$

where $t \sim \mathcal{U}(1,T)$ is the uniformly sampled timestep, $\varepsilon_\theta : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^d$ the denoising network parameterized by θ, and $\|\cdot\|_2$ the Euclidean norm measuring prediction errors.

Semantic Scene Graph. A scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ formally represents a 3D scene through nodes $\mathcal{V}$ (objects) and directed edges $\mathcal{E}$ (spatial/semantic relationships). In our work, we employ a triplet-GCN [19, 30, 10] to process the nodes and edges of a scene graph, incrementally updating their features to capture structured information within the graph. This process is repeated across multiple layers, enriching the features of nodes and edges to better represent complex relationships within the scene graph. In addition, we follow the approach in [52, 51], using CLIP to encode features for edges and nodes, which are proven to bring strong inter-object information to the scene graph [52]. Thus, each node $v_i \in \mathcal{V}$ contains features $\mathbf{h}_i \in \mathbb{R}^{d_v}$ encoding object attributes (category, size, position), while an edge $e_{ij} = (v_i, v_j, r_{ij}) \in \mathcal{E}$ contains relation embedding $r_{ij} \in \mathbb{R}^{d_e}$.

3.2 Semantic Graph Refinement via Dynamic Prompt

To convert natural language instructions into structured scene synthesis, we formulate the semantic scene graph construction as a progressive disambiguation process guided by dynamic prompting. Let $\mathcal{S} = \{s_1, ..., s_n\}$ denote the tokenized user input, and $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the target semantic scene graph where:

- $\mathcal{V} = \{v_i | v_i = (\mathbf{c}_i, \mathbf{s}_i, \mathbf{p}_i, \theta_i)\}$: Nodes encoding object category $\mathbf{c}_i \in \mathcal{C}$, where $\mathcal{C}$ contains 15 predefined types, spatial parameters (size $\mathbf{s}_i \in \mathbb{R}^3$, position $\mathbf{p}_i \in \mathbb{R}^3$, orientation $\theta_i \in [0, 360°)$)
- $\mathcal{E} = \{e_{ij} | e_{ij} = (v_i, r_{ij}, v_j)\}$: Directed edges connecting nodes v_i and v_j through spatial relations $r_{ij} \in \mathcal{R}$, where $\mathcal{R}$ contains 12 predefined types.

The translation process employs a **three-stage chain-of-thought prompting architecture** that progressively constructs semantic graphs through functional composition reasoning. We provide detailed illustrations of the architecture and prompt design in the appendix for further reference.

Object Extraction & Scene Completion. The initial phase bridges the gap between sparse user descriptions and comprehensive scene understanding by jointly extracting explicit objects and inferring implicit structural elements. Real-world indoor spaces inherently contain foundational components (e.g., floors, ceilings, lighting) that users often omit in textual inputs. Our dynamic prompting strategy addresses this through two complementary operations:

$$\mathcal{O} = \underbrace{\text{LLM}(p_{\text{exp}}, \mathcal{S})}_{\text{Explicit Extraction}} \cup \underbrace{\text{LLM}(p_{\text{imp}}, \mathcal{S})}_{\text{Implicit Completion}}, \tag{3}$$

where p_{exp} is a constrained decoding prompt to parse primary furniture from text for Explicit Object Extraction: p_{exp} = *"List all furniture in* $\{S\}$ *using ONTOLOGY: [bed, sofa, table, cabinet...]"*,

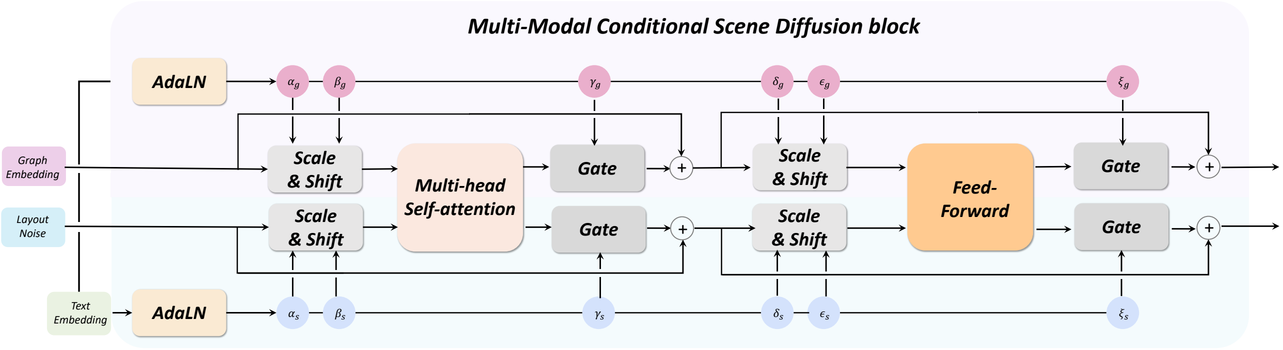


Figure 3: **Our model architecture.** We present a multi-modal conditional scene diffusion block, where distinct modalities are processed separately via dual-stream AdaLN and integrated through attention mechanisms. Note that time-step information is incorporated into the text embedding.

and p_{imp} is the spatial reasoning rules to supplement necessary elements for Implicit Scene Completion: p_{imp} = *"Add room structures for* $\{S\}$*: Floor if indoor scene, Ceiling light if no windows and etc"*. Additionally, at this stage, if the user inputs objects that are not supported by the system (e.g., non-furniture items), the LLM utilizes dynamic prompting to substitute them with commonsense alternatives according to the predefined ontology.

Functional Group Formation. Building upon the extracted objects $\mathscr{O}$, this phase establishes semantically meaningful clusters where co-located objects form functional units. Professional scene design principles emphasize grouping furniture by activity zones (e.g., sleeping, working), which our method operationalizes through domain-aware prompting.

$$\mathscr{F} = \bigcup_k \mathscr{F}_k, \quad \mathscr{F}_k = \text{LLM}(p_{\text{grp}}, \mathscr{O}), \tag{4}$$

where p_{grp} is a taxonomy-guided prompt to group objects into activity zones using categories. $\mathscr{O}$ denotes the explicit extraction and implicit completion results from the previous stage, where $\mathscr{F}_k$ represents the k-th functional subgraph inferred by the LLM through domain-specific prompting.

Intra-group spatialization then applies role-specific placement rules through dynamically composed prompts, thereby constructing functional subgraphs $\mathscr{F}_k$ that encapsulate both objects and their spatial relations. It is noteworthy that an error-checking mechanism is also established, which leverages the spatial reasoning capabilities of LLM to detect implausible spatial relations (e.g., "sofa on top of computer"), flagging and adjusting such inconsistencies during the functional grouping process.

Group Layout Optimization. The final phase coordinates functional groups $\{\mathscr{F}_k\}$ into a globally coherent layout through context-aware chain-of-thought prompting, which synergizes few-shot examples with the LLM's inherent spatial commonsense. A commonsense verification step is further incorporated to check for physical plausibility, such as proper furniture, thereby ensuring the overall consistency and plausibility of the layout.

Our core methodology integrates three key components: exemplar-guided initialization, commonsense verification, and contextual adaptation. First, we adopt few-shot approach to generate seed layouts based on prototypical arrangements, establishing a robust starting point for scene synthesis. Second, commonsense verification automatically identifies violations of design principles, such as improper furniture-wall adjacencies, ensuring logical and functional coherence within the scene. Finally, contextual adaptation enables dynamic and context-aware scene edits by making local adjustments that preserve functional group relations while optimizing spatial flow, thereby maintaining the integrity of the overall design. Through this three-stage refinement process, text descriptions are converted into semantic graphs, with both serving as conditions for the subsequent multi-modal diffusion model. Notably, error-checking mechanism is also incorporated at each stage to handle errors and prevent their further propagation.

3.3 Multi-modal Conditional Scene Diffusion Model

Previous works [41, 52, 51] that rely on denoising architectures conditioned on single-modal inputs suffer from limited conditional guidance and inefficient cross-modal alignment, which may compromise spatial-semantic consistency. To address this issue, we design a multi-modal conditional scene diffusion model aiming at learning the distribution of 3D indoor scenes, which includes semantic classes and placements of multiple objects. We assume a 3D scene $\mathscr{S}$ is represented as an object set $\{o_i\}_{i=1}^N$, where each object o_i comprises eight geometric degrees-of-freedom (8DoF) and semantic attributes. These components include Cartesian coordinates (x,y,z) define object position, bounding box dimensions (l,w,h) specify physical extents, and trigonometric components $(\sin\theta, \cos\theta)$ encode orientation to circumvent angular discontinuity. Semantic context is captured through a C-dimensional one-hot class vector $\mathbf{c}_i$. This explicit parameterization enables differentiable optimization of spatial-semantic relationships. The sinusoidal angular representation ensures stable gradient propagation during pose refinement while maintaining rotation equivariance.

Transformer Architecture for Scene Synthesis. Our denoising backbone adopts the Diffusion Transformer (DiT) [35], replacing conventional U-Net with a transformer-based architecture. Unlike convolutional networks that rely on local receptive fields, the self-attention mechanism in DiT inherently models global dependencies among all objects in a scene. This is critical for indoor scene synthesis, where spatial-semantic coherence requires understanding long-range object relations (e.g., nightstands typically flanking a bed, or a dining table centrally surrounded by chairs). This allows each object to dynamically attend to others based on their spatial-semantic compatibility, avoiding implausible arrangements caused by local-only processing in CNNs. Moreover, DiT's multi-head attention partitions feature into h subspaces, enabling joint modeling of diverse relations (e.g., directional proximity, functional grouping) within a single layer.

Multi-modal Conditional Scene Diffusion Model. To derive structured multimodal representations, we employ a two-stage encoding process. First, the textual scene description (e.g., "a bedroom with twin nightstands besides the bed") is encoded into a semantic vector $\mathbf{c}_{\text{text}} \in \mathbb{R}^{d_t}$ using the pretrained CLIP [38] text

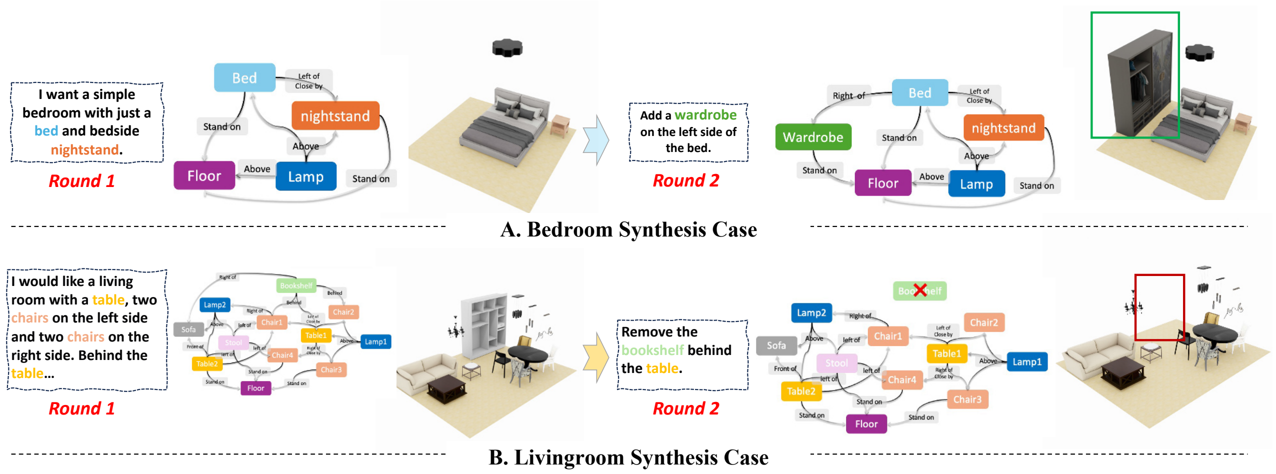

I want a simple bedroom with just a bed and bedside nightstand.
Round 1
Bed
nightstand
Floor
Lamp
Left of
Close by
Stand on
Above
Add a wardrobe on the left side of the bed.
Round 2
Wardrobe
Right of
A. Bedroom Synthesis Case
I would like a living room with a table, two chairs on the left side and two chairs on the right side. Behind the table...
Round 1
Bookshelf
Lamp2
Chair1
Chair2
Sofa
Stool
Table1
Lamp1
Table2
Chair4
Chair3
Floor
Right of
Behind
left of
Front of
Left of
Close by
Right of
Close by
Remove the bookshelf behind the table.
Round 2
B. Livingroom Synthesis Case

Figure 4: **Edit results.** We present two case studies of scene synthesis via text, showcasing both generation (Round 1) and editing (Round 2). We only display a subset of the edges in the semantic graphs. Added objects are highlighted with green boxes, while removed objects are marked with red boxes.

encoder, which preserves fine-grained object relationships ("besides"). Subsequently, this text embedding is transformed into a semantic scene graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ through a GCN network, where nodes $\mathscr{V}$ represent objects with CLIP-encoded category features $\{\mathbf{h}_i\}_{i=1}^{N}$, and edges $\mathscr{E}$ encode spatial relations (e.g., left/right, behind).

For a conditional sampling of 3D layout parameters, our model take multiple modalities into account, including text, scene graphs, and 3D layout parameters. The diffusion model learns to gradually denoise the latent scene representation $\mathbf{x}_t \in \mathbb{R}^{N\times D}$ at timestep t, conditioned on three complementary modalities: textual scene descriptions, semantic scene graphs, and noisy object parameters $\mathbf{x}_t$ itself.

Given the fundamental differences between scene graph and 3D layout parameters embeddings in terms of modality, we employ two distinct sets of weights for each modality. As shown in Figure 3, our approach first involves using two separate transformers for each type of input, and then merges the sequences from both modalities during the attention mechanism. This allows each representation to operate within its own branch while also enabling it to consider the other's information.

The workflow of our model begins with three inputs: graph embedding, layout noise, and text embedding. Each input is processed through separate pathways to preserve modality-specific characteristics. The inputs are converted into vector representations via embedding layers and passed through adaptive layer normalization (AdaLN) layers, which dynamically adjust normalization parameters based on input distribution. The AdaLN layers generate 12 parameters (α_g, β_g, γ_g, δ_g, ε_g, ξ_g for the graph modality, and α_s, β_s, γ_s, δ_s, ε_s, ξ_s for the scene modality) . These parameters scale and shift embeddings to prepare them for attention mechanisms. The adjusted embeddings are processed through multi-head self-attention to capture intra- and inter-modality relationships. Outputs pass through gating layers that control information flow based on relevance, ensuring only pertinent information is propagated. Gated outputs are combined and refined through a feed-forward network. After another gating step, modality-specific representations are integrated into the denoised latent scene representation $\mathbf{x}_{t-1}$. This structured workflow allows independent processing of modalities while enabling cross-modal information exchange, leading to accurate 3D layout reconstruction.

Training Objective. Following the DDPM [16], we optimize the denoising model ε_θ to predict the noise added to the scene layout at each diffusion timestep $t \in \{1,\ldots,1000\}$. Given a scene $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ with semantic graph $\mathbf{g}$ and text description $\mathbf{s}$, the forward process, which gradually corrupts $\mathbf{x}_0$ to $\mathbf{x}_t$, follows a cosine schedule [32]. The training objective minimizes the weighted ℓ_2 distance between predicted and actual noise, conditioned on both $\mathbf{g}$ and $\mathbf{s}$:

$$\mathscr{L}_{\text{layout}} = \mathbb{E}_{\mathbf{x}_0,t,\varepsilon,\mathbf{g},\mathbf{s}}\left[\|\varepsilon - \varepsilon_\theta(\mathbf{x}_t,t,\mathbf{g},\mathbf{s})\|_2^2\right], \tag{5}$$

here, $\mathbf{g} = \text{GCN}(\mathscr{G})$ encodes the scene graph topology, and $\mathbf{s} = \text{CLIP}_{\text{text}}(\textit{prompt})$ represents text semantics. The dual conditioning allows ε_θ to simultaneously resolve geometric uncertainty via $\mathbf{g}$ and align with linguistic constraints via $\mathbf{s}$. For stable training, we scale the gradients from $\mathbf{g}$ and $\mathbf{s}$ pathways by factors λ_{graph} and λ_{text}, respectively.

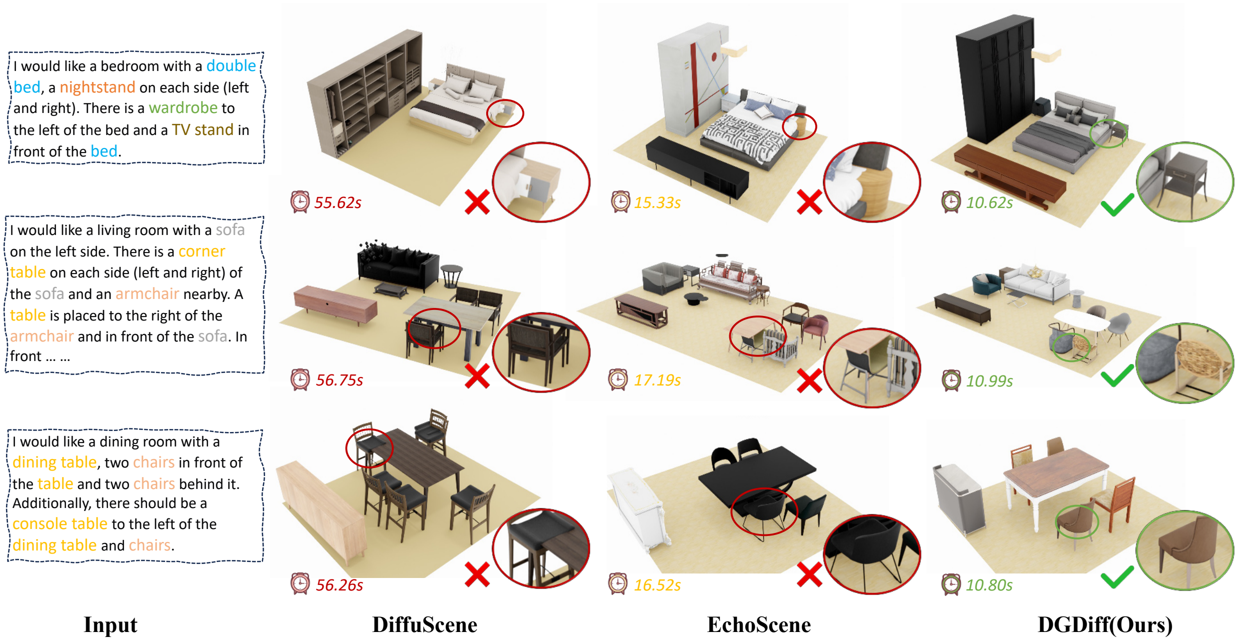
4 Experiments

4.1 Implementation Details.

Datasets. Our experiments are conducted on the SG-FRONT dataset [52], which integrates the rich 3D scene data from 3D-FRONT [12] while augmenting it with structured semantic annotations. The base 3D-FRONT contains 6,813 professionally designed indoor scenes with 14,629 high-quality furnished rooms, covering diverse categories including 4,041 bedrooms, 900 dining rooms, and 813 living rooms. Each room is populated with 3D-FUTURE [12] objects annotated with precise geometric attributes (position, size, orientation) and material properties. Built upon this foundation, SG-FRONT introduces fine-grained scene graph annotations to bridge low-level geometry and high-level semantics. Specifically, it defines 15 types of spatial-semantic relationships.

Metrics. We adopt Frechet Inception Distance (FID) [15], FID-CLIP [26], and Kernel Inception Distance (KID×1000) [5] to holistically evaluate layout fidelity, semantic-textual alignment, and distribution matching between generated and real scenes. All metrics are computed on 256×256 orthographic projections of scenes, where objects are rendered with fixed color codes according to their semantic categories under standardized camera parameters.

Baselines. We introduces an innovative framework for multi-modal 3D scene synthesis and evaluates its performance against



I would like a bedroom with a double bed, a nightstand on each side (left and right). There is a wardrobe to the left of the bed and a TV stand in front of the bed.

I would like a living room with a sofa on the left side. There is a corner table on each side (left and right) of the sofa and an armchair nearby. A table is placed to the right of the armchair and in front of the sofa. In front ... ...

I would like a dining room with a dining table, two chairs in front of the table and two chairs behind it. Additionally, there should be a console table to the left of the dining table and chairs.

Figure 5: **Qualitative results.** In contrast to methods that use semantic graphs as input, we translate the input into text for easier understanding. The red boxes highlight areas with unreasonable placements, while the green boxes show the improvements made by our method.

eight contemporary techniques. These methods are categorized into two groups: the first group consists of methods based on variational auto-encoders [23, 39], including 3D-SLN [30], Progressive [10], and Graph-to-Box [10]; the second group comprises methods based on diffusion models, such as CommonScene [52], DiffuScene [41], InstructScene [29], and EchoScene [51]. For a fair comparison, all experiments involving bedroom, living room, and dining room scenarios adhere to the same experimental setup, including identical neural architectures and parameter configurations.

LLMs. In Section 3.2, we introduce the implementation of the three-stage CoT architecture using large language models (LLMs). We experimented with several mainstream LLMs, including DeepSeek v3 [8], GPT 4o [1], and Qwen 2 [2]. Our results indicate that semantic graphs can be successfully generated by different LLMs, and the final outputs show minimal variation across these models. Based on these findings, we ultimately utilized DeepSeek v3 for our experiments, based on its superior performance and efficiency.

4.2 Scene Fidelity

In this part, we focus on fidelity, the critical aspect of 3D scene synthesis. We present our experiments through qualitative results, quantitative results, and inference speed.

Quantitative results. We present the quantitative evaluations for 3D scenes systhesis and provide FID / FID-CLIP / KID in Table 1. As demonstrated, our framework shows superior performance against seven comparative approaches across three distinct scenarios. The experimental results reveal that our method achieves state-of-the-art (SOTA) metrics in most evaluation criteria, consistently the first place across the majority of benchmarks. FID is a primary metric for our work, which directly measures visual fidelity. DGDiff achieves an improvement of more than 10% in FID scores compared to previous SOTA methods, which demonstrates its superiority in generating realistic visuals. In particular, DGDiff excels in handling more complex scenes, such as dining rooms, which typically contain more than a dozen objects, significantly more than the average of eight objects in bedrooms. We achieved a significant improvement by reducing the FID from the SOTA 59.66 to 53.89 in this challenging scenario. In terms of the two secondary metrics, FID-CLIP and KID, our method consistently ranks the first or second among all compared approaches. These results demonstrate that our approach achieves SOTA performance in both semantic consistency and detail distribution, which benefits from its superior visual fidelity. In summary, DGDiff delivers the best overall performance. Notably, owing to its excellent interactivity, our method is able to provide novel scene generation solutions for AR/VR applications.

Qualitative results. To verify the effectiveness of our method, we conducted experiments in two aspects. First, we performed a direct comparison with previous methods. Second, we compared scene synthesis results between two groups: those generated from semantic graphs constructed by LLMs and those generated from semantic graphs in the dataset. We used paired text and semantic graph combinations for our experiments. We selected two SOTA methods: DiffuScene [41], which is conditioned on text, and EchoScene [51], which is conditioned on semantic graphs. We provide a comparison of these methods for different room type in Fig. 5. DiffuScene exhibited issues in layout rationality, indicating that text-guided methods may struggle to learn angle and position information effectively. EchoScene, which uses semantic graphs as a guide, improved layout accuracy but still encountered problems such as object overlap. In contrast, our method achieved the best results overall.

4.3 Zero-shot Application

By harnessing the contextual reasoning power of LLMs, we generate semantic scene graphs via CoT prompting from user text. The interactive nature of LLMs further enables intuitive editing through textual input. Given the refined text and semantic graphs, our model rapidly synthesizes high-fidelity 3D scenes.

For zero-shot scene synthesis, our method can handle both detailed user designs and rough prompts. It demonstrates excellent

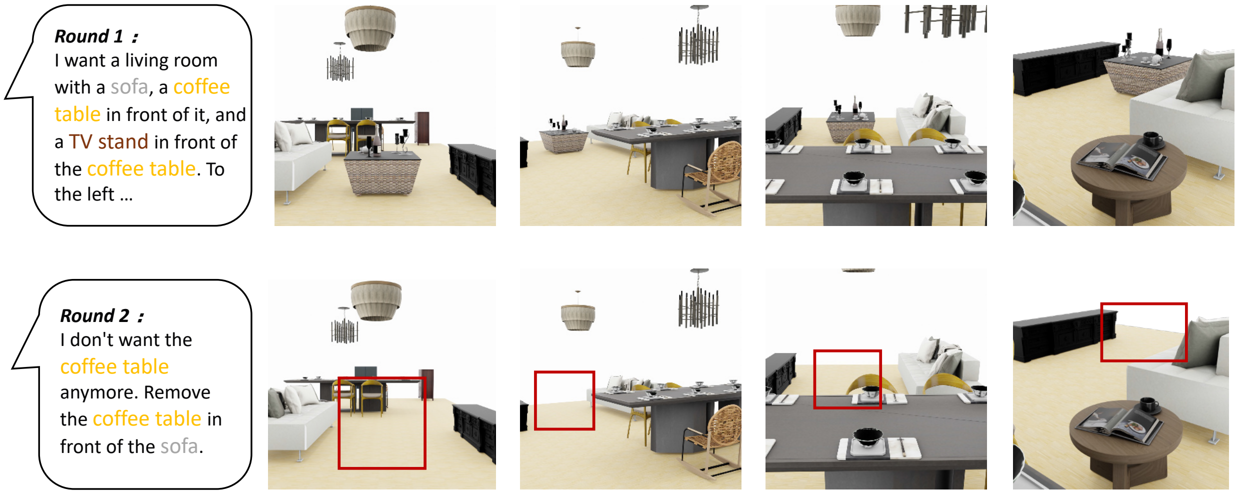

Round 1 :
I want a living room with a sofa, a coffee table in front of it, and a TV stand in front of the coffee table. To the left ...
Round 2 :
I don't want the coffee table anymore. Remove the coffee table in front of the sofa.

Figure 6: **Immersive wandering in VR.** We show the instructions during scene synthesis and editing on the left, and three immersive perspectives on the right. The red box highlights the object deletion result.

instruction-following capability for detailed inputs and reasonable expansion for rough inputs while maintaining high-quality generation. As shown in Fig. 4, we tested our method on different room categories with both ”simple prompts” and ”complex prompts”. Our method can generate reasonable layouts for all cases.

For scene editing, we support operations such as adding, removing, and changing object positions with simple natural language interactions. The results highlight the strong instruction-following capability and high-quality generation of our method.

Moreover, our approach facilitates effortlessly generated and interactively edited 3D scenes, enabling rapid and precise generation for immersive wandering within VR environments. As shown in Fig. 6, we present a case of immersive 3D scene synthesis and editing. Thanks to the efficiency of our method, users can experience immersive exploration of generated or edited scenes in VR within tens of seconds. We demonstrate a living room scene, showcasing the immersive wandering from multiple perspectives before and after editing.

4.4 Ablation study

Our ablation studies are designed in three parts. First, we ablate $\pi(\mathbf{t})$ to see the impact of removing temporal information. Second, we ablate the method of incorporating conditional information by replacing adaLN-zero with cross-attention. Third, we conduct separate ablations for each condition: using only semantic graphs and only text information. We ablate these four parts of the full version and report the performance in Tab 2.

Experiments show that ablating the time-step information in the conditioning leads to a performance drop of over 5%, particularly for KID, which decreases to one-fifth of its original value. For conditional guidance, we confirm DIT’s findings: injecting conditional information via adaLN-Zero reduces the model’s GFLOPs compared to cross-attention and improves results by 5%. Finally, our ablation studies show that removing the graph condition significantly degrades performance, indicating that text-only guidance is limited for scene synthesis. In contrast, removing the text modality has a minor impact, highlighting the strong guidance capability of graph. This underscores the importance of generating semantic graphs from text. Using both conditions together yields the best results.

Additionally, as shown in Table 3, we conduct experiments across three scenarios in the dataset and analyzed the inference speed under different conditions. We compare our method’s Multimodal Diffusion Model module with other diffusion model approaches [41, 29, 51]. Using the same semantic graph or text as input, we test the inference time for generating a scene. Thanks to transformers’ superior computational efficiency in diffusion model denoising networks, due to their global self-attention mechanisms and massive parallelism, which enable faster inference speeds, our method surpasses all current diffusion model-based approaches. We also compare the inference speed of our method with current state-of-the-art open-source methods based on LLM [50] for scene synthesis. Methods based on LLMs need to reason about relationships and assess scene rationality, leading to much longer inference times than generative model-based methods. Our method, which only infers semantic graphs and uses pre-trained generative models for scene synthesis, shows significant speed advantages.

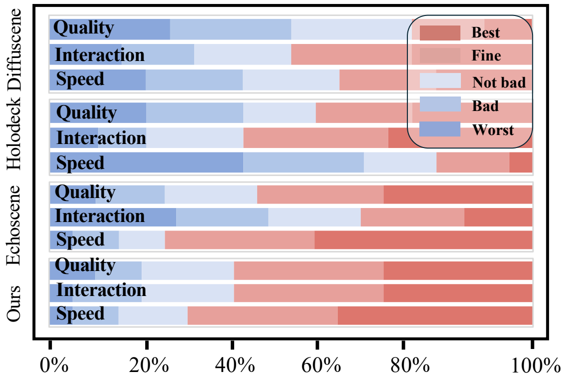


Figure 7: **User study.** The color bars in the figure indicate the percentage of the scores. The X-axis represents the percentage of participants.

4.5 Semantic-graphs Accuracy

To verify the accuracy of the semantic graphs, we conduct a detailed comparison between DGDiff and InstructScene with the SG-

Table 1: **Quantitative evaluation.** We compare methods on FID, FID_{CLIP}, and KID ($\times 0.001$) scores at 256^2 pixels between the top-down rendering of generated and real scenes, following previous works [51]. We color the best and the second-best results. ↓ indicates that the values are better if the metric is smaller.

Methods	Network Structure	Bedroom			Living Room			Dining Room		
		FID ↓	FID_{CLIP} ↓	KID ↓	FID ↓	FID_{CLIP} ↓	KID ↓	FID ↓	FID_{CLIP} ↓	KID ↓
3D-SLN [30]	Autoregressive	57.90	5.45	3.85	77.82	7.02	3.65	69.13	7.99	6.23
Progressive [10]		58.01	5.67	7.36	79.84	7.41	4.24	71.35	8.28	6.21
Graph-to-Box [10]		54.61	5.26	2.93	78.53	6.88	3.32	67.80	7.75	6.30
CommonScene [52]		52.69	5.22	2.82	76.52	6.58	2.08	65.10	7.55	6.11
DiffuScene [41]	Diffusion	52.02	5.01	2.52	81.61	7.52	2.52	65.90	7.39	0.09
InstructScene [29]		45.40	3.87	1.06	75.83	6.98	1.23	61.56	6.49	4.90
EchoScene [51]		46.53	4.24	0.33	75.54	6.35	1.60	59.66	6.24	2.63
DGDiff(Ours)		44.84	3.93	0.04	73.27	6.37	0.96	53.89	4.53	2.23

Table 2: **Ablations study**. We show FID, FID_{CLIP}, KID.

Ablation	FID↓	FID_{CLIP} ↓	KID↓
Ours w/o $\pi(\mathbf{t})$	47.02	4.07	0.23
Ours with cross-attn	46.46	4.01	1.09
Ours w/o text	45.92	4.46	0.11
Ours w/o graph	51.546	5.22	2.04
Ours	**44.84**	**3.93**	**0.04**

Table 3: **Average inference time** (in seconds). We show the inference time of different methods across various scene types, with the average inference time of 10 scene generations as our result. Note that DGDiff * represents the DGDiff version with only the diffusion model, while DGDiff $^{+}$ indicates the DGDiff with semantic graph refinement.

Method	Bedroom	Livingroom	Diningroom
Diffuscene [41]	55.72	56.81	56.45
Instructscene[29]	21.34	20.15	22.43
Echoscene[51]	15.52	16.38	16.21
DGDiff *	**10.62**	**11.87**	**11.23**
Holodeck[50]	578.67	585.33	583.12
DGDiff $^{+}$	**17.20**	**18.87**	**18.23**

Table 4: Scene-graph accuracy comparison on Node_{acc}, Edge_{acc}, GPT-Score. Our DGDiff achieves better results.

Ablation	Node_{acc}↑	Edge_{acc}↑	GPT-Score↑
InstructScene [29]	96%	85.2%	4.1
Ours	**98%**	**91.7%**	**4.5**

FRONT dataset as a benchmark. We leverage GPT-4o to convert the semantic graphs from the dataset into textual descriptions, which serves as a reference standard for our evaluation, as shown in Tab 4.

In quantitative terms, we focus on the accuracy of node and edge generation. For node extraction, DGDiff achieves a recognition accuracy of 98% for explicit objects mentioned in the text, which is on par with InstructScene's 96%. However, DGDiff has the additional capability of inferring and completing implicit structural elements such as floors and ceilings. Regarding edge generation, which captures spatial relationships, DGDiff demonstrates an accuracy of 91.7% across 12 predefined relationship types, such as "to the left of" and "facing," compared to InstructScene's 85.2%. In qualitative term, we assess the semantic consistency of the generated scene graphs with the reference texts and graphs using GPT-4o. DGDiff achieves a score of 4.5/5 for layout rationality and semantic consistency, which is higher than the 4.1/5 achieved by InstructScene.

It is worth highlighting that the interactivity of DGDiff allows users to construct and edit semantic graphs in real time through text instructions. This feature enables users to adjust and refine the semantic graphs according to their specific requirements, ensuring that the generated graphs better meet user expectations and scene demands.

4.6 User Study

To evaluate the practical usability and user experience of DGDiff, we conduct a user study involving 42 participants. Among them, 30 are students from diverse academic backgrounds, including computer science, software engineering, foreign languages, and law. The remaining 12 are working professionals, with occupations such as teachers, public servants, and freelancers. The study aims to assess the quality of generated scenes, interaction fluency, and overall satisfaction. Participants are tasked with generating and editing 3D indoor scenes using DGDiff, DiffuScene, EchoScene, and HOLODECK. They provide scores for generation quality, speed, and interaction friendliness on a scale of 1 to 5, where 1 denotes the poorest performance and 5 the best. The tasks include creating scenes from natural language descriptions (e.g., "a bedroom with a double bed and two nightstands") and editing scenes via instructions (e.g., "move the nightstand to the other side of the bed"), with the order of the model randomized across different scenarios to mitigate potential bias. Participants also provide qualitative feedback on their experience with each method. We provide detailed illustrations of the specifics of the user study in the appendix for further reference. The results demonstrate DGDiff's superiority in both generation and editing tasks. In terms of generation quality, DGDiff receives an average score of 3.6/5, outperforming DiffuScene [41] (2.6/5), EchoScene [51] (3.5/5), and HOLODECK [50] (3.1/5). Participants highlight DGDiff's ability to produce logically consistent and visually appealing scenes, even in complex settings like dining rooms. For speed, DGDiff completes tasks in an average of 18.4 seconds, significantly faster than DiffuScene (55.7 seconds) and HOLODECK (578.3 seconds). Notably, when using the same semantic graph input as EchoScene, DGDiff achieves this in just 11.05 seconds, outperforming EchoScene (16.3 seconds). Users also rate DGDiff highest for interaction friendliness (3.65/5), praising its intuitive natural language interface and rapid response times. However, participants find EchoScene's semantic graph input confusing and difficult to provide appropriate semantic graphs, often requiring assistance to complete the input. They also express dis-

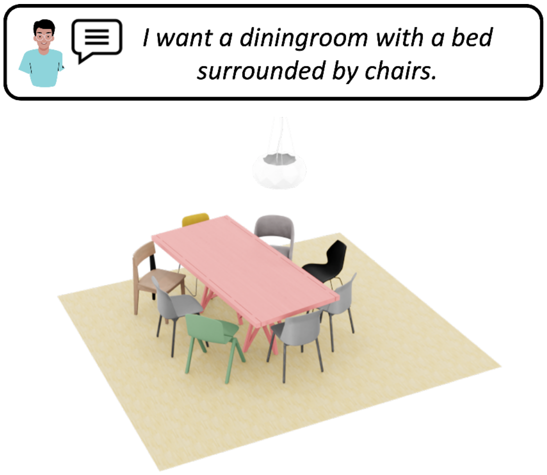


Figure 8: **Failure case.** Our method replaces the bed with a dining table since there's no bed in the object database of the dining room.

satisfaction with HOLODECK's lengthy output time (up to 600 seconds), which hinders immersive experiences. Overall, 73.3% of participants prefer DGDiff over other methods, citing its superiority in quality, speed, and ease of use. These results validate DGDiff's effectiveness in providing a user-friendly and efficient solution for 3D scene synthesis.

5 Limitation

As a retrieval-based scene generation paradigm, we use 3D-FRONT as the object database. When encountering such unconventional inputs, our method resorts to replacing them with semantically similar objects from the existing collection, which may not fully align with the user's intent for highly specialized or unique items, as shown in Fig. 8. To address this, we are committed to integrating 3D generation techniques [47, 53, 36] in the future, shifting from a retrieval-based approach to a generation-based method for complete scene construction.

6 Conclusion

In this paper, we propose DGDiff, a novel multi-modal diffusion framework, which combines semantic graph refinement with dynamic prompting to generate high-quality, meaningful, and interactive 3D indoor scenes. Our approach bridges the gap between high-level user intent and structured spatial understanding, enabling fast and controllable scene synthesis through a progressive text-driven semantic topology construction pipeline. By leveraging complementary modalities, our method draws on text for global semantics and semantic graphs for fine-grained spatial constraints. This integration enables DGDiff to achieve significant improvements in generation fidelity and speed, as demonstrated on the 3D-FRONT dataset. Thanks to its interactivity and low-latency generation, DGDiff is well-suited for immersive VR applications. In future work, we could explore further optimizations for large-scale collaborative VR environments and support more dynamic interactions within the 3D environment.

7 Acknowledgments

This paper is supported by Zhongguancun Laboratory and the National Natural Science Foundation of China (Grant No. 62272018 and No. 62206184).

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[2] J. Bai, S. Bai, Y. Chu, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 6

[3] D. Bauer, C. Zheng, O.-H. Kwon, and K.-L. Ma. A multi-layout design for immersive visualization of hierarchical network data. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1038–1047. IEEE, 2024. 1

[4] M. A. Bautista, P. Guo, S. Abnar, W. Talbott, A. Toshev, Z. Chen, L. Dinh, S. Zhai, H. Goh, D. Ulbricht, et al. Gaudi: A neural architect for immersive 3d scene generation. *Advances in Neural Information Processing Systems*, 35:25102–25116, 2022. 1

[5] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. 5

[6] A. Chang, W. Monroe, M. Savva, C. Potts, and C. D. Manning. Text to 3d scene generation with rich lexical grounding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 53–62, 2015. 2

[7] A. Chang, M. Savva, and C. D. Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 2028–2038, 2014. 2

[8] DeepSeek-AI. Deepseek-v3 technical report, 2024. 6

[9] J. Deng, W. Chai, J. Guo, Q. Huang, J. Huang, W. Hu, S. Hao, J.-N. Hwang, and G. Wang. Citygen: Infinite and controllable city layout generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1995–2005, 2025. 1

[10] H. Dhamo, F. Manhardt, N. Navab, and F. Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16352–16361, 2021. 3, 6, 8

[11] W. Feng, W. Zhu, T.-j. Fu, V. Jampani, A. Akula, X. He, S. Basu, X. E. Wang, and W. Y. Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023. 2

[12] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10933–10942, 2021. 5

[13] R. Fu, Z. Wen, Z. Liu, and S. Sridhar. Anyhome: Open-vocabulary generation of structured and textured 3d homes. In *European Conference on Computer Vision*, pp. 52–70. Springer, 2024. 2

[14] L. Gao, J.-M. Sun, K. Mo, Y.-K. Lai, L. J. Guibas, and J. Yang. Scenehgn: Hierarchical graph networks for 3d indoor scene generation with fine-grained geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8902–8919, 2023. 2

[15] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[16] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3, 5

[17] Y. Hu, W. Hu, and A. Quigley. Towards using generative ai for facilitating image creation in spatial augmented reality. In *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 441–443. IEEE, 2023. 1

[18] C. Jeon, S. Park, S. Kim, and C. Woo. A framework for automatic generation of augmented reality objects. In *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 283–288. IEEE, 2023. 1

[19] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1219–1228, 2018. 2, 3

[20] J. Kaeder, M. Vergari, V. Biener, T. Kojić, J. Grubert, S. Möller, and

J.-N. V. Antons. Working with mixed reality in public: Effects of virtual display layouts on productivity, feeling of safety, and social acceptability. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 740–748. IEEE, 2024. 1

[21] S. G. Kaya, B. Zhou, R. R. Arora, N. Zheutlin, G. Vanloo, and E. K. Eyigoz. Dynamic content generation for augmented technical support. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 441–446. IEEE, 2021. 1

[22] S. J. Kim, D. D. Cao, F. Spinola, S. J. Lee, and K. S. Cho. Roomrecon: High-quality textured room layout reconstruction on mobile devices. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 544–553. IEEE, 2024. 1

[23] D. P. Kingma, M. Welling, et al. Auto-encoding variational bayes, 2013. 2, 6

[24] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 2

[25] P. Kulshreshtha, N. Lianos, B. Pugh, and S. Jiddi. Layout aware inpainting for automated furniture removal in indoor scenes. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 839–844. IEEE, 2022. 1

[26] T. Kynkäänniemi, T. Karras, M. Aittala, T. Aila, and J. Lehtinen. The role of imagenet classes in fréchet inception distance. In *International Conference on Learning Representations*. OpenReview. net, 2023. 5

[27] J. Li, L. Zhao, H.-N. Liang, and L. Yu. Immerview: Adaptive multiview layout for immersive situated visualizations. In *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 108–112. IEEE, 2023. 1

[28] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, C. Tu, B. Chen, D. Cohen-Or, and H. Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 2

[29] C. Lin and Y. Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 6, 7, 8

[30] A. Luo, Z. Zhang, J. Wu, and J. B. Tenenbaum. End-to-end optimization of scene layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3754–3763, 2020. 2, 3, 6, 8

[31] R. Ma, A. G. Patil, M. Fisher, M. Li, S. Pirk, B.-S. Hua, S.-K. Yeung, X. Tong, L. Guibas, and H. Zhang. Language-driven synthesis of 3d scenes from scene databases. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 2

[32] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021. 5

[33] W. Para, P. Guerrero, T. Kelly, L. J. Guibas, and P. Wonka. Generative layout modeling using constraint graphs. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6690–6700, 2021. 2

[34] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021. 1, 2

[35] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023. 4

[36] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*. 9

[37] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5899–5908, 2018. 2

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 4

[39] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 6

[40] W. Song, X. Zhang, S. Li, Y. Gao, A. Hao, X. Hou, C. Chen, N. Li, and H. Qin. Hoianimator: Generating text-prompt human-object animations using novel perceptive diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 811–820, 2024. 1

[41] J. Tang, Y. Nie, L. Markhasin, A. Dai, J. Thies, and M. Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20507–20518, 2024. 1, 2, 4, 6, 7, 8

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 2

[43] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 2

[44] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. 2

[45] X. Wang, C. Yeshwanth, and M. Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pp. 106–115. IEEE, 2021. 1, 2

[46] Y. Wang, J. Ma, R. Shao, Q. Feng, Y.-K. Lai, and K. Li. Humancoser: Layered 3d human generation via semantic-aware diffusion model. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 436–445. IEEE, 2024. 1

[47] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023. 9

[48] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2

[49] G. Xu, S. Wang, Y. Hu, and X. Shen. A multilayer component pane 3d layout frame design for responsive websites. In *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 474–478. IEEE, 2023. 1

[50] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 2024. 2, 7, 8

[51] G. Zhai, E. P. Örnek, D. Z. Chen, R. Liao, Y. Di, N. Navab, F. Tombari, and B. Busam. Echoscene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision*, pp. 167–184. Springer, 2024. 2, 3, 4, 6, 7, 8

[52] G. Zhai, E. P. Örnek, S.-C. Wu, Y. Di, F. Tombari, N. Navab, and B. Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graph diffusion. *Advances in Neural Information Processing Systems*, 36:30026–30038, 2023. 2, 3, 4, 5, 6, 8

[53] Z. Zhao, Z. Lai, Q. Lin, Y. Zhao, H. Liu, S. Yang, Y. Feng, M. Yang, S. Zhang, X. Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 9

[54] S. Zhou, Z. Fan, D. Xu, H. Chang, P. Chari, T. Bharadwaj, S. You, Z. Wang, and A. Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*, pp. 324–342. Springer, 2024. 1

[55] Y. Zhou, Z. While, and E. Kalogerakis. Scenegraphnet: Neural message passing for 3d indoor scene augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7384–7392, 2019. 2