Controllable 3D Outdoor Scene Generation via Scene Graphs

Yuheng Liu[1,2], Xinke Li[3†], Yuning Zhang[4], Lu Qi[5], Xin Li[1], Wenping Wang[1],
Chongshou Li[4], Xueting Li[6*], Ming-Hsuan Yang[2*]

[1]Texas A&M University, [2]UC Merced, [3]City University of Hong Kong
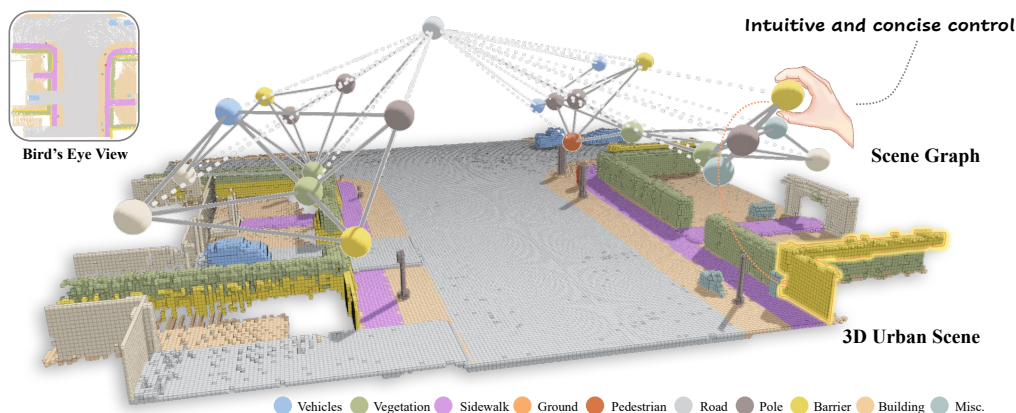[4]Southwest Jiaotong University, [5]Insta360 Research, [6]NVIDIA

Figure 1. **Scene Graph Guided 3D Outdoor Scene Generation.** Compared to text descriptions and BEV maps, scene graphs offer a more intuitive and user-friendly format for controlling 3D scene generation. We also develop an interactive system that allows users to generate/edit dense 3D scenes through scene graph interaction.

Abstract

Three-dimensional scene generation is crucial in computer vision, with applications spanning autonomous driving, gaming and the metaverse. Current methods either lack user control or rely on imprecise, non-intuitive conditions. In this work, we propose a method that uses scene graphs—an accessible, user-friendly control format—to generate outdoor 3D scenes. We develop an interactive system that transforms a sparse scene graph into a dense BEV (Bird's Eye View) Embedding Map, which guides a conditional diffusion model to generate 3D scenes that match the scene graph description. During inference, users can easily create or modify scene graphs to generate large-scale outdoor scenes. We create a large-scale dataset with paired scene graphs and 3D semantic scenes to train the BEV embedding and diffusion models. Experimental results show that our approach consistently produces high-quality 3D urban scenes closely aligned with the input scene graphs. To the best of our knowledge, this is the first approach to generate 3D outdoor scenes conditioned on scene graphs.

---

† Corresponding author.
* Equal contribution.

1. Introduction

3D scene generation has garnered wide attention due to its potential for creating realistic, physically coherent 3D scenes. These models offer a powerful approach to understanding and simulating the complexities of our 3D world. Among the various methods for 3D scene generation, probabilistic generative models have shown great promise in recent advancements. However, the stochastic nature of these models makes the generation process difficult to control precisely, emphasizing the need for an editable and controllable generation process.

To enable controllable scene generation, many methods leverage recent advances in 2D conditional generation, such as DALL-E [3] and Stable Diffusion [40], where high-quality images are generated based on natural language. Inspired by these models, some approaches [28, 31, 32] use 2D views to guide 3D content generation. However, these methods are primarily object-centric and do not scale to complex outdoor scenes due to their large scales and interconnected structures. Other approaches apply text-based conditions to directly control 3D scene generation, such as Text2LiDAR [48]. Yet, text-to-3D provides insufficient control over both physical constraints and spatial de-

tails: it cannot effectively enforce real-world physical rules or precisely control scene elements (e.g., number of objects), leading to outputs that fail to meet specified requirements [63].

One possible solution is to extend existing 3D indoor scene generation methods [15, 46, 58, 60, 61] to outdoor environments. However, this adaptation is highly challenging. Indoor scene generation typically relies on multi-view images to synthesize bounded, textured surfaces, focusing on object appearance and spatial relationships. In contrast, outdoor scenes are unbounded and predominantly captured using textureless LiDAR point clouds, which aims to model large-scale spatial layouts and background continuity.

Recent outdoor scene generation research attempts to explore unique controls for 3D outdoor scene generation. For example, [9, 62] rely on BEV layouts or semantic maps, which require users to provide pixel-level control signals, posing a challenge in terms of interaction, especially for large-scale, complex 3D outdoor scenes. Therefore, selecting an appropriate medium for controllable 3D outdoor scene generation is crucial. In this context, the scene graph emerges as an ideal candidate due to its structured, regularized, and sparse representation, which makes it particularly well-suited for 3D outdoor scene generation and allows efficient control over complex layouts. Additionally, scene graphs are intuitive, enabling users to interact with and edit them easily. Motivated by these benefits, we propose a new framework that leverages the scene graph as a sparse-to-dense pipeline for 3D outdoor scene generation.

However, it is non-trivial to utilize scene graph as condition, due to its sparse and abstract nature. To resolve this, we begin by employing a Graph Neural Network (GNN) that aggregates information from the scene graph through message passing. Next, a novel Allocation Module is developed to assign spatial positions to produce a Bird's Eye View Embedding Map (BEM). Finally, the BEM is used to condition a 3D Pyramid Discrete Diffusion Model [34] to generate the complete 3D scene. We jointly train the GNN and the diffusion model for seamless integration. To enhance scene understanding within the GNN, two auxiliary tasks are introduced: edge reconstruction and node classification, which further improve the model's ability to interpret and represent the scene graph effectively. Additionally, we develop an interactive system to enable intuitive scene graph creation and editing, allowing users to control scene content through both manual editing and text-based scene graph generation, bridging the gap between text input and 3D outdoor scene generation. To support our approach, we construct a scene graph dataset for each 3D scene in the CarlaSC dataset [47], defining node attributes and establishing edges based on spatial relationships. The primary contributions of our work are as follows:

• To the best of our knowledge, this is the first work that

generates a large-scale 3D outdoor scene conditioning on a scene graph input.
• We propose a GNN equipped with a novel Allocation Module that converts a sparse scene graph to a compact scene embedding, which then conditions a diffusion model for 3D scene generation.
• We curate a large-scale dataset including paired 3D scenes and scene-graphs for model training. Extensive experiments demonstrate that our approach generates 3D outdoor scenes that closely align with the definitions in the scene graphs.
• We develop and provide a user-friendly system for constructing scene graphs, enabling users to flexibly create custom scene graphs to guide 3D outdoor scene generation according to their needs.

2. Related Work

3D Generation via Diffusion Models. Diffusion models have expanded their applications from 2D image synthesis to complex 3D data modeling [8]. Compared to conventional GANs [13] and VAEs [19], they offer improved performance through a progressive denoising mechanism [14], enhancing training stability and the ability to model complex distributions. This makes them particularly effective for 3D data generation. However, existing research primarily focuses on object-level generation [33, 38, 39, 49, 52, 54] or indoor environments [4, 11, 41, 57]. Meanwhile, the few works [21, 22, 29, 34, 50] designed for outdoor scene generation prioritize visual fidelity over controllability. In this work, we aim to develop a framework for controllable 3D outdoor scene generation, emphasizing easy and precise user control.

Scene Graph Application. A scene graph is a structured representation of a scene, encoding objects, their attributes, and the relationships between them [17]. Unlike dense representations such as point clouds [23, 24, 37] or meshes [5], a scene graph provides a concise yet comprehensive view of a scene's structure, making it a powerful conditional signal for generation tasks. By explicitly modeling inter-object relationships, scene graphs offer a structured way to control scenes, facilitating both human-driven and AI-driven content generation, and this capability has been widely explored especially in 2D tasks [6, 12, 18, 30, 53, 55, 56, 64]. Extending this concept to 3D environments, Armeni et al. [1] introduce the 3D indoor scene graph. This framework integrates semantic, spatial, and geometric information into a hierarchical graph, where nodes represent objects, rooms, and spaces, and edges encode their spatial and semantic relationships. Although this well-defined representation has proven effective for indoor scenes, a comparable formulation for outdoor environments remains largely unexplored.

Controllable 3D Scene Generation. To date, controllable 3D outdoor scene generation has received limited attention.
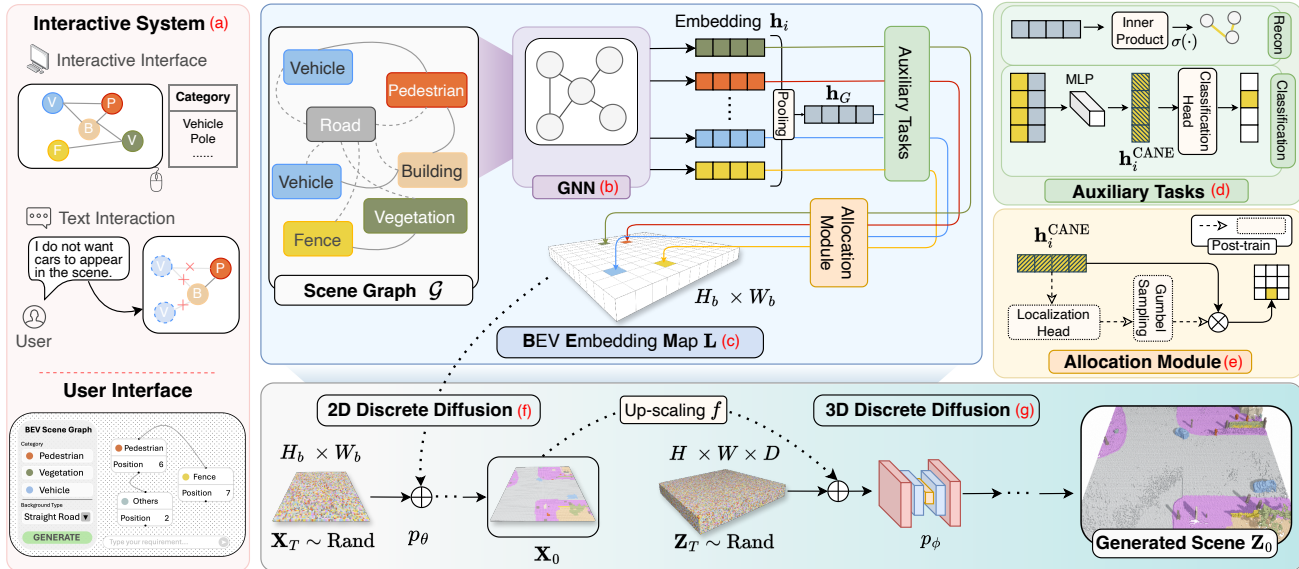
Figure 2. **Overview of Scene Graph Guided 3D Scene Generation.** The Scene Graph Guided 3D Generation structure consists of three main components: the interactive system (red), BEM processing (blue), and diffusion generation (bottom). Through the interactive system, users can construct their own Scene Graphs using either an interactive interface or text interaction. The constructed scene graph is processed by a GNN, which is jointly trained with the diffusion model using auxiliary tasks to enhance control. Each node in the Scene Graph is then positioned by the Allocation Module to form the BEM. This BEM serves as a conditioning input to the 3D Pyramid Discrete Diffusion Model [34], which generates the final 3D outdoor scene. Note that "Recon", "Classification", and "CANE" denote "Edge Reconstruction", "Node Classification", and "Context-aware Node Embedding", respectively.

One of the closest efforts in this area is Text2LiDAR [48], which generates LiDAR points from text inputs. While text-based control is an interesting attempt, it lacks explicit spatial structure, making it unsuitable for precise scene composition [63]. Instead, scene graphs offer a more interpretable and structured approach to controlling scene generation, a concept that has been well established in indoor scene generation [25, 27, 41, 45, 66]. However, adapting them to outdoor scenes is non-trivial, as outdoor scenes are large-scale, unbounded, and contain a diverse set of objects [51]. Unlike indoor scenes, which often use a compositional approach by placing objects within predefined or generated bounding boxes [10, 36, 44, 60, 61], outdoor scenes feature complex backgrounds and incomplete structures, such as roads and buildings. To address these challenges, we propose a novel pipeline and scene graph representation tailored for controllable 3D outdoor scene generation.

3. Method

We first discuss the formulation of a scene graph in Sec. 3.1. We then introduce how to generate a 3D outdoor scene conditioned on a scene graph in Sec. 3.2. In Sec. 3.3, we describe how the proposed method facilitates the convenient creation of 3D scenes.

3.1. Scene Graph Formulation

Formally, a scene graph is characterized by its nodes and edges as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of nodes $\mathcal{V}$ consists of two types (i.e., $\mathcal{V} = \mathcal{V}_I \cup \mathcal{V}_R$): a) **Instance Nodes** ($\mathcal{V}_I$) that represent countable objects with standard labels defined in [47], such as vehicles and pedestrians. Each node $v_i \in \mathcal{V}_I$ is associated with a feature vector $[\mathbf{c}_i; \mathbf{p}_i]$, where $\mathbf{c}_i \in \mathbb{R}^d$ denotes the node attributes with dimension d, and $\mathbf{p}_i \in \mathbb{R}^2$ represents a coordinate specifying its center position in the BEV map. b) **Scene Road Nodes** ($\mathcal{V}_R$) that define the structure of the road and other global background information of the scene in one node, i.e., $\mathcal{V}_R = \{v_r\}$. We further construct the graph structure by defining two types of edges $\mathcal{E}$ to capture essential relationships: a) **Physical Proximity**: For any two instance nodes $v_i, v_j \in \mathcal{V}_I$, we define an edge $e_{ij} \in \mathcal{E}$ if the Euclidean distance $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ is smaller than a threshold δ_d. b) **Road Connectivity**: For any instance node $v_i \in \mathcal{V}_I$ and the singleton road node $v_r \in \mathcal{V}_R$, an edge $e_{ir} \in \mathcal{E}$ is created for the connects to the road structure.

In practice, we use the simplified graph as control signals to ease user interaction, where each instance node contains only its semantic label $\mathbf{c}_i$ and an approximate 2D position $\mathbf{p}_i$ represented as a patch index. Also, the scene road node is represented by the road type.
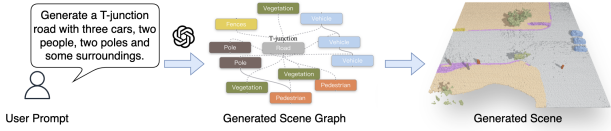
Figure 3. **Scene Graph Generation.** LLMs convert the user's prompt into a scene graph, which guides 3D scene generation.

3.2. Scene-graph-guided Diffusion

Given a 2D scene graph, our method aims at generating a 3D semantic scene that aligns with the structure defined by the scene graph. In the following, we first convert the scene graph into a dense 2D embedding using a Graph Neural Network (GNN). Next, we synthesize a plausible 2D scene map by training a 2D diffusion model conditioned on the scene graph embedding. Finally, we use a conditional 3D diffusion model to generate the final 3D outdoor scene given the 2D scene map.

Scene Graph Neural Network. The scene Graph Neural Network (Fig. 2(b)) aims to generate node embeddings for scene graphs that capture both local structural and global context information. Particularly, we utilize Graph Attention Network (GAT) [43] for GNN implementation. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ that encodes the connectivity between nodes, the node embeddings $\mathbf{h}_i$ are computed by a two-layer GAT for node v_i in the graph. To incorporate global context into the node embeddings, we concatenate each node's embedding with a pooled global embedding $\mathbf{h}_G$ as the final embedding, i.e.,

$$\mathbf{h}_i^{\text{CANE}} = \text{MLP}([\mathbf{h}_i; \mathbf{h}_G]), \mathbf{h}_G = \text{Pooling}(\{\mathbf{h}_i \mid v_i \in \mathcal{V}\}), \quad (1)$$

where $[\cdot; \cdot]$ denotes concatenation, $\text{Pooling}(\cdot)$ is a graph global mean pooling operation [35], $\mathbf{h}_G \in \mathbb{R}^{64}$ and $\text{MLP}(\cdot)$ is a multi-layer perceptron. We name the output embedding $\mathbf{h}_i^{\text{CANE}}$ as Context-aware node embedding (CANE).

We train the GNN with two objectives: the auxiliary tasks (Fig. 2(d)) and the downstream task (Fig. 2(e)). For auxiliary tasks, we apply edge reconstruction loss as in Graph Auto-encoder (GAE) [20] and node classification loss, given by

$$\mathcal{L}_a = \text{BCE}(\hat{\mathbf{A}}, \mathbf{A}) + \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \text{CE}(y_i, \hat{y}_i), \quad (2)$$

where BCE is binary cross-entropy, CE is node-wise cross-entropy loss, and,

$$\hat{\mathbf{A}} = \sigma(\mathbf{h}_G \mathbf{h}_G^\top), \hat{y}_i = \text{Softmax}(\text{MLP}(\mathbf{h}_i^{\text{CANE}})), \quad (3)$$

for $\sigma(\cdot)$ being sigmoid function. The first term of (2) is to reconstruct the global edge structure of the scene, specifically the adjacency matrix $\mathbf{A}$. The second term of (2) is to classify each node v_i into the original category. The auxiliary tasks ensure that the network learns both structural relationships and node-specific features, effectively capturing both local and global information in the CANE.

For downstream task, embeddings $\mathbf{h}_i^{\text{CANE}}$ are used as inputs to compute BEV Embedding Map (BEM, Fig. 2(c)) which serves as condition for the sequential diffusion model in the next phase of diffusion. These embeddings are passed to an **allocation module**, defined by

$$\mathbf{L} = \sum_{i=1}^{|\mathcal{V}|} \mathcal{M}(\hat{p}_i) \odot \mathbf{h}_i^{\text{CANE}}, \quad (4)$$

where $\mathbf{L} \in \mathbb{R}^{H_b \times W_b \times C}$ is the output BEM with height H_b, width W_b, and channel dimension C. The binary map $\mathcal{M}(\hat{p}_i) \in \{0, 1\}^{H_b \times W_b}$ is expanded along the channel dimension to $\mathbb{R}^{H_b \times W_b \times C}$ to enable element-wise multiplication with the node embedding $\mathbf{h}_i^{\text{CANE}} \in \mathbb{R}^C$. In the implementation, we perform the inference by sampling position from an MLP-based localization head, i.e.,

$$\hat{p}_i \sim \text{GumbelSoftmax}_\tau(\text{Head}(\mathbf{h}_i^{\text{CANE}})), \quad (5)$$

where τ is the temperature for Gumbel softmax [16]. While in the training process of the diffusion model, we replace the $\hat{p}_i$ in (4) by the ground truth position p_i. The localization head is trained after the diffusion model training. As a result, the allocation module effectively converts irregular and sparse graph representation into dense 2D map, i.e., BEM, which offers better compatibility with the subsequent 2D diffusion process.

2D Map Discrete Diffusion (Fig. 2(f)). Given a scene graph, there may be multiple plausible 2D maps that align with its structure. To model this variability, we use a diffusion model that converts the sparse scene graph embedding into a dense 2D map representation. Formally, the 2D Map Diffusion refines the sparse BEM $\mathbf{L}$ into a dense 2D map, $\mathbf{X} \in \{0, 1\}^{H_b \times W_b \times c}$, where c is the semantic class number. We apply the standard discrete diffusion [2] for 2D map generation. In the forward process, the 2D map $\mathbf{X}_0$ is gradually corrupted in T timesteps using a transition matrix $\mathbf{Q}_t$, which adds noise as $\mathbf{X}_t = \mathbf{X}_{t-1} \mathbf{Q}_t$. This process can also be represented using a cumulative matrix $\bar{\mathbf{Q}}_t$, allowing us to sample $\mathbf{X}_t$ directly from the original map $\mathbf{X}_0$,

$$q(\mathbf{X}_t \mid \mathbf{X}_0) = \text{Cat}(\mathbf{X}_t; \mathbf{P} = \mathbf{X}_0 \bar{\mathbf{Q}}_t), \quad (6)$$

where Cat represents a categorical distribution.

In the reverse diffusion stage, a model p_θ learns to reverse the noise process, predicting the less-noised map $\mathbf{X}_{t-1}$ from the noisy map $\mathbf{X}_t$, using $\mathbf{L}$ as guidance:

$$p_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \mathbf{L}) = \mathbb{E}_{\tilde{p}_\theta(\tilde{\mathbf{X}}_0 \mid \mathbf{X}_t, \mathbf{L})} q(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \tilde{\mathbf{X}}_0). \quad (7)$$

The model is trained by minimizing the KL divergence between the forward process and the learned reverse process. The loss function $\mathcal{L}_\theta$ is defined as:

$$\begin{aligned} \mathcal{L}_\theta = &d_{\text{KL}}\left(q(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \mathbf{X}_0) \| p_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \mathbf{L})\right) \\ &+ \lambda d_{\text{KL}}\left(q(\mathbf{X}_0) \| \tilde{p}_\theta(\tilde{\mathbf{X}}_0 \mid \mathbf{X}_t, \mathbf{L})\right), \end{aligned} \quad (8)$$

where λ controls the weight of the auxiliary term for better reconstruction. During inference, we start from random
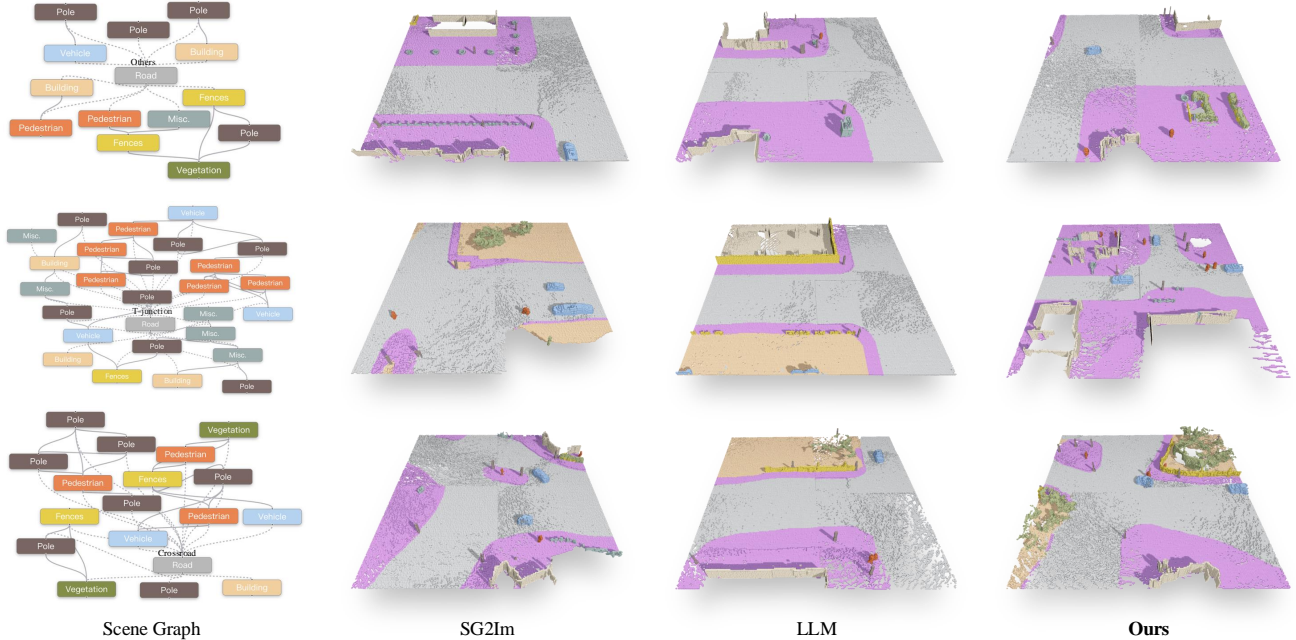
Figure 4. **Controlling 3D Outdoor Scene Generation with Scene Graphs.** We compare baseline methods. Results show that our method generates scenes consistent with the provided scene graph, whereas the SG2Im and LLM approaches exhibit inconsistencies in object quantities and road types.

noise and use the learned reverse diffusion to generate a dense 2D map $\mathbf{X}_0$, guided by the BEM $\mathbf{L}$. This refined map provides a more complete spatial layout for further 3D scene generation. We train the GNN and the 2D map diffusion model using a loss $\mathcal{L}_a + \mathcal{L}_\theta$.

3D Scene Discrete Diffusion (Fig. 2(g)). We convert the generated 2D map into a dense 3D scene using a discrete diffusion process similar to the 2D Map Diffusion. The initial 2D map $\mathbf{X}_0 \in \{0,1\}^{H_d \times W_d \times c}$, generated from the previous diffusion step, is used as a condition to guide the generation of the 3D scene. We define the 3D scene as $\mathbf{Z} \in \{0,1\}^{H \times W \times D \times c}$, where H, W, and D are the dimensions of the 3D scenes and c represents semantic categories.

The 3D diffusion follows the same forward and reverse diffusion steps as in the 2D case but operates on the 3D scene grid. Specifically, a learnable model p_ϕ predicts each denoised state $\mathbf{Z}_{t-1}$ from $\mathbf{Z}_t$, conditioned on the input 2D map $\mathbf{X}_0$, given by,

$$p_\phi(\mathbf{Z}_{t-1} \mid \mathbf{Z}_t, \mathbf{X}_0) = \mathbb{E}_{\tilde{p}_\theta(\tilde{\mathbf{z}}_0 \mid \mathbf{Z}_t, f(\mathbf{X}_0))} q(\mathbf{Z}_{t-1} \mid \mathbf{Z}_t, \tilde{\mathbf{Z}}_0), \quad (9)$$

where $f : \mathbb{R}^{H_d \times W_d \times c} \to \mathbb{R}^{H \times W \times c}$ is an up-scaling function. Furthermore, the training loss $\mathcal{L}_\phi$ follows the same form as (8), given by,

$$\mathcal{L}_\phi = d_{\mathrm{KL}}\left(q(\mathbf{Z}_{t-1} \mid \mathbf{Z}_t, \mathbf{Z}_0) \| p_\phi(\mathbf{Z}_{t-1} \mid \mathbf{Z}_t, \mathbf{X}_0)\right) \quad (10)$$
$$+ \lambda d_{\mathrm{KL}}\left(q(\mathbf{Z}_0) \| \tilde{p}_\phi(\tilde{\mathbf{Z}}_0 \mid \mathbf{Z}_t, \mathbf{X}_0)\right).$$

During inference, the network generates the final 3D scene $\mathbf{Z}_0$ by starting from a noisy 3D state and applying the reverse diffusion process conditioned on $\mathbf{X}_0$. Specifically,

each step of the reverse diffusion process is performed by sampling as (9). This produces a fully detailed 3D scene aligned with the spatial layout of the 2D map.

3.3. Interactive System

We develop an interactive control system (Fig. 2(a)) that prioritizes user-directed scene graph manipulation. The core component is a graphical interface where users can precisely construct and modify scene graphs through intuitive operations such as node addition, deletion, and position adjustment. This direct manipulation ensures fine-grained control over the scene generation. Additionally, users can provide text prompts, which are processed by large language models to generate corresponding scene graphs (Fig. 3). These scene graphs are then used as input to our method to generate the final 3D scenes. Details on the design of system-level prompts can be found in the supplementary materials.

4. Experimental Results

4.1. Data Preparation

Due to the lack of paired scene graph data for existing 3D outdoor LiDAR scenes, we generate scene graph data from each scene in the CarlaSC [47] dataset, creating a new dataset termed *CarlaSG*. Based on the scene graph formulation discussed in Sec 3.1, we extract a 3D scene graph from each 3D semantic map in CarlaSC and project it onto

Table 1. **Comparison of Different Conditioning Methods on 3D Outdoor Scene Generation.** Uncon-Gen, SG2Im, and LLM represent Unconditional Generation, Scene Graph to Image, and Large Language Model, while M-Pole, M-Pede, and M-Vech represent the MAE calculated individually for *Pole*, *Pedestrian*, and *Vehicle* categories. In the Scene Quality Evaluation, higher mIoU and MA scores indicate better semantic consistency, while a lower F3D score [34] signifies closer feature alignment with the original dataset. In the Control Capacity Evaluation, a lower MAE reflects a smaller discrepancy between the generated scene and the object quantities defined in the scene graph for conditioning. A higher Jaccard Index indicates greater alignment in the object categories between the generated scenes and the specified scene graph.

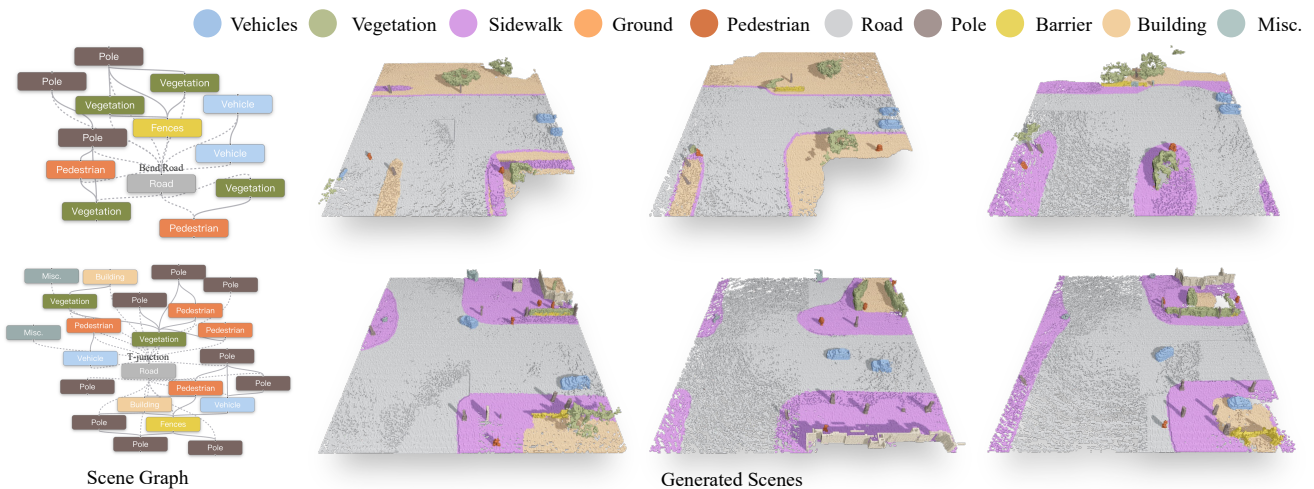| Method	Condition	Scene Quality			Control Capacity				
		mIoU	MA	F3D ($\downarrow$)	MAE ($\downarrow$)	Jaccard	M-Pole ($\downarrow$)	M-Pede ($\downarrow$)	M-Vech ($\downarrow$)
Uncon-Gen [34]	-	68.21	**85.69**	**0.338**	2.07	0.68	4.78	4.71	3.59
SG2Im [18]	Scene Graph	65.43	81.72	0.486	0.97	0.81	2.25	2.79	2.64
LLM [59, 65]	Text-Embedding	68.19	85.62	0.386	1.44	0.70	3.41	3.57	3.51
Ours	Scene Graph	**68.69**	85.01	0.393	**0.63**	**0.93**	**1.39**	**1.81**	**1.35**

Figure 5. **Diversity in Scene Generation.** Comparison of three scenes generated by our method under the same scene graph. This demonstrates our method's ability to produce varied yet consistent scenes based on identical input.

2D. Figure 1 provides an example of a 3D outdoor scene alongside its scene graph. Notably, as the spatial distribution of sidewalks and ground closely aligns with the road layout, we merge the *Ground* and *Sidewalk* classes in the original CarlaSC dataset with the *Road* class and mark them as *Road*. We further categorize the roads into five types: *Straight Road*, *T-Junction*, *Crossroad*, *Bend Road*, and *Others*. Additional details of processing techniques are available in the supplementary materials.

4.2. Evaluation Protocols

We evaluate our approach from two aspects: assessing the quality of generated scenes and measuring the alignment between the generated scenes and their corresponding scene graphs. Additionally, we conduct a user study to perceptually evaluate the scene graphs' alignment of the generated scenes. All experiments are performed on a testing set with *1k* randomly selected scene graphs. Details on the evaluation metrics and examples of the user study can be found in

the supplementary materials.

Scene Quality Evaluation. We follow the evaluation protocols from [34] to assess scene quality. We use mean Intersection over Union (mIoU) and mean Accuracy (MA) to evaluate semantic plausibility. Additionally, we measure feature similarity with Fréchet 3D Distance (F3D) [34], which computes the Fréchet distance between generated and real scenes in a pre-trained 3D CNN-based autoencoder's feature space.

Control Capacity. We evaluate the alignment between generated object counts and scene graph node counts using Mean Absolute Error (MAE) and the Jaccard index. MAE quantifies numerical discrepancies, while the Jaccard index measures the overlap in object types, reflecting how closely the generated scenes match the scene graph structure.

User Study. We use the Differential Mean Opinion Score (DMOS) [42], a subjective rating method, to evaluate the alignment of generated scenes with input scene graphs, considering object quantity, positioning, and road type.

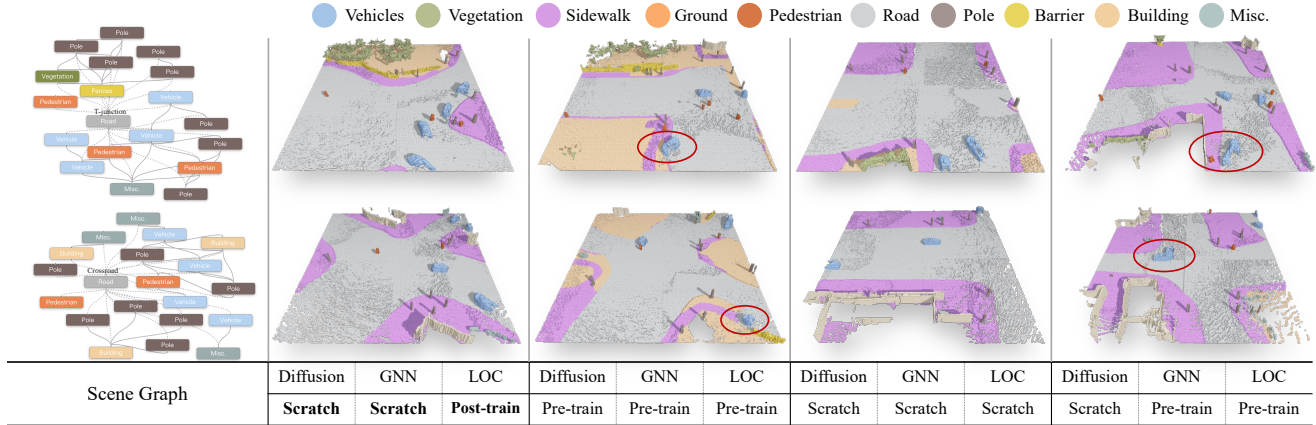| Scene Graph	Diffusion	GNN	LOC	Diffusion	GNN	LOC	Diffusion	GNN	LOC	Diffusion	GNN	LOC
	Scratch	**Scratch**	**Post-train**	Pre-train	Pre-train	Pre-train	Scratch	Scratch	Scratch	Scratch	Pre-train	Pre-train

Figure 6. **Impact of Different Training Strategies.** Models trained with the second and last strategies exhibit issues such as vehicles positioned on sidewalks, overlapping objects, and inconsistencies in capturing object quantities. The third strategy generates semantically reasonable scenes but struggles with accurately matching object quantities and road types to the scene graph. In contrast, the first strategy produces high-quality scenes with good alignment to the input scene graph, thus we choose the first strategy to train our networks.

4.3. Experimental Settings

Training and Inference Settings. During joint training of the 2D diffusion model and the GNN, we apply data augmentation and include 10% unconditional data for the diffusion model, along with a 30% feature mask on the GNN input to simulate scenarios where some users may not provide positional information for certain nodes. During inference, we set the Gumbel temperature τ in the allocation module to 2.0 to introduce randomness in generated scenes. Further details on learning rates, batch size, and other training parameters are in the supplementary material.

Network Architecture. We employ a diffusion model and GNN in a joint training setup. The 2D/3D diffusion models use 3D-UNet [7] as the backbone which is often used in outdoor understanding [26], while the GNN consists of a two-layer GAT encoder [43].

Comparison Baselines. As discussed in Sec. 2 and the supplementary material, adapting indoor scene generation methods to outdoor environments is non-trivial due to fundamental differences. Moreover, such adaptations would significantly alter their original pipeline, making direct comparisons less meaningful. Therefore, we consider three baselines: (1) A large language model (LLM) [59, 65] extracts embeddings from the graph's textual description, followed by a 2D deconvolution to align with the downstream 2D diffusion model. Details of the operations are provided in the supplementary materials; (2) Scene Graph to Image (SG2Im) [18], a GAN-based method for generating images from scene graphs, which we adapt to produce the BEM from scene graphs; and (3) an unconditional generation (Uncon-Gen) [34] model without scene graph conditioning.

4.4. Main Results

Qualitative Results. Figure 4 shows the 3D outdoor scenes generated separately using our method and baseline methods [59, 65], based on three different scene graphs. The results demonstrate that our method effectively captures the object quantities specified in the scene graph and the road type information. In contrast, the scenes generated by the LLM and SG2Im methods show significant discrepancies in object counts across most categories, and the generated road types differ substantially from the intended configurations.

Quantitative Results. Table 1 compares our method with baselines. In Scene Quality, Uncon-Gen, LLM, and our method perform comparably, while SG2Im lags behind. Meanwhile, in Control Capacity, our method outperforms all baselines across metrics, achieving low MAE values below 1.0, demonstrating precise control over object quantities. In contrast, SG2Im has a higher MAE (0.97), and the LLM baseline yields 1.44, over twice our method's 0.63, indicating a significant accuracy gap. Additionally, our method achieves a higher Jaccard Index, reflecting its effectiveness in capturing object categories from scene graphs across diverse scenes.

Generation Diversity. To validate that our method produces diverse outputs rather than strictly memorizing scenes based on the scene graph, we generate scenes three times using the same scene graph. The results are shown in Figure 5. The outcomes demonstrate that our method can generate varied scenes even when conditioned on the same scene graph, yet each generated scene remains consistent with the structural and categorical information provided in the scene graph. This confirms that our method introduces randomness in the generation process while maintaining alignment with the input scene graph.

4.5. Ablation Experiments and User Studies

Unconditional Proportion. We examine the effect of the unconditional proportion in diffusion training, as shown in Figure 7. Results indicate that scene quality (mIoU and
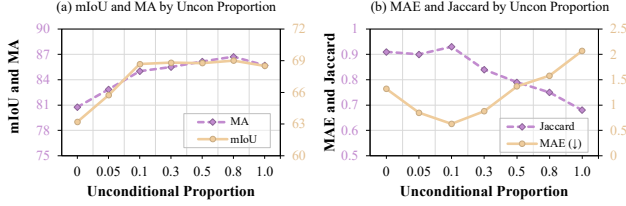
Figure 7. **Unconditional Proportion *v.s.* Generation Quality and Control.** Evaluation mIoU, MA, Jaccard Index, and MAE as the unconditional proportion varies during diffusion training. Considering the trade-off between scene quality and control, we choose 0.1 as the balance point.
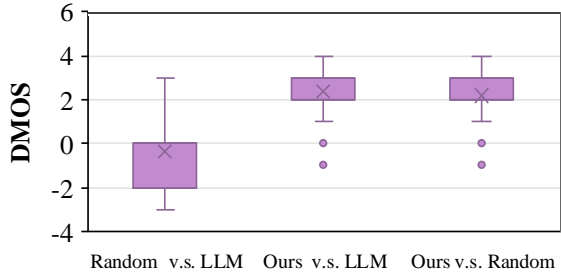

Figure 8. **User Study: DMOS Comparison of Scene Generation Methods.** Our method aligns well with scene graph specifications.

MA) improves as the unconditional proportion increases, with a noticeable bottleneck at 0.1. While further increases lead to marginal improvements in scene quality, they come at the cost of reduced control capacity, as reflected by worsening Jaccard Index and MAE. To balance scene quality and control capacity, we set the unconditional proportion to 0.1 in our experiments.

Effect of Auxiliary Tasks. We evaluate the impact of adding edge reconstruction and node classification as auxiliary tasks to the GNN during joint training with the diffusion model. As shown in Table 2, both tasks yield the best performance, with a low MAE of 0.63 and a high Jaccard Index of 0.93. Removing either task leads to notable drops in performance, particularly in the Jaccard Index. Omitting both results in further declines. This shows that both tasks contribute to improved alignment in scene generation.

Different Training Strategies. We explore alternative training strategies for our method: (a) pre-train the diffusion model, GNN, and localization head (LOC), then freeze GNN and LOC while fine-tuning the diffusion model; (b) end-to-end training of all components from scratch; (c) pre-train GNN and LOC, freeze their parameters, and train the diffusion model from scratch; and (d) jointly train the diffusion model and GNN from scratch, then freeze GNN and post-train LOC. As shown in Table 3 and Figure 6, strategy (d) achieves the best performance. Strategies (a) and (c) show semantic inconsistencies, while (b) generates scenes of reasonable quality but struggles with object quantity and road type alignment. Joint training of the diffusion

Table 2. **Impact of Auxiliary Tasks on Generation Performance.** Comparison of MAE and Jaccard Index w/ and w/o edge reconstruction and node classification tasks in the GNN. Including both tasks yields the best alignment with the scene graph.

Reconstruction	Classification	MAE ($\downarrow$)	Jaccard
✓	✓	**0.63**	**0.93**
✓	✗	0.89	0.84
✗	✓	0.80	0.83
✗	✗	0.79	0.81

Table 3. **Comparison of Training Strategies for Our Method.** The bolded row is our adopted strategy.

Diffusion	GNN	LOC	MAE ($\downarrow$)	Jaccard
Pre-train	Pre-train	Pre-train	0.79	0.88
Scratch	Scratch	Scratch	1.01	0.82
Scratch	Pre-train	Pre-train	0.95	0.90
Scratch	**Scratch**	**Post-train**	**0.63**	**0.93**

model and GNN (d) allows the diffusion model to learn scene structure in sync with encoded features, while post-training LOC assigns precise object positions without disrupting learned structural relationships, achieving a balance between semantic coherence and quantity control.

User Study. We generate 100 pairs of scenes and conduct user studies with 20 subjects. Each user scores paired scenes based on object quantity, positioning, and road type accuracy relative to their scene graphs. The resulting Differential Mean Opinion Score (DMOS), shown in Figure 8, indicates that our method outperforms the baselines. Additionally, we conduct a one-tailed paired t-test on the MOS score difference among three methods. In this test, the null hypothesis is that our generation method does not possess a higher score than baseline methods. The results support the rejection of the null hypothesis at a significance level of $p < 10^{-3}$, indicating that our method statistically performs better than both baselines with high confidence.

5. Conclusion

In this work, we propose a solution that integrates an interactive system, BEV Embedding Map, and diffusion generation to enable controllable 3D outdoor scene generation. The challenges stem from complex outdoor landscapes with rich information and structural diversity. Our approach utilizes scene graphs to transition information from sparse to dense representations. Coupled with the interactive system, it enables users to intuitively and concisely generate their desired 3D outdoor scenes. Comparative experiments demonstrate that our method achieves more accurate object quantities and alignment with the input scene graph. These results indicate that our approach is a robust and effective solution for controllable 3D outdoor scene generation.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *ICCV*, 2019. 2

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021. 4

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2023. 1

[4] Alexey Bokhovkin, Quan Meng, Shubham Tulsiani, and Angela Dai. Scenefactor: Factored latent 3d diffusion for controllable 3d scene generation. *arXiv preprint arXiv:2412.01801*, 2024. 2

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[6] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE TPAMI*, 2023. 2

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, 2016. 7

[8] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE TPAMI*, 2023. 2

[9] Jie Deng, Wenhao Chai, Jianshu Guo, Qixuan Huang, Wenhao Hu, Jenq-Neng Hwang, and Gaoang Wang. Citygen: Infinite and controllable 3d city layout generation. *arXiv preprint arXiv:2312.01508*, 2023. 2

[10] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *ICCV*, 2021. 3

[11] Chuan Fang, Xiaotao Hu, Kunming Luo, and Ping Tan. Ctrlroom: Controllable text-to-3d room meshes generation with layout constraints. *arXiv preprint arXiv:2310.03602*, 2023. 2

[12] Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *ICCV*, 2023. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2

[15] Siyi Hu, Diego Martin Arroyo, Stephanie Debats, Fabian Manhardt, Luca Carlone, and Federico Tombari. Mixed diffusion for 3d indoor scene synthesis. *arXiv preprint arXiv:2405.21066*, 2024. 2

[16] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2016. 4

[17] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. *CVPR*, 2015. 2

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 2, 6, 7

[19] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2013. 2

[20] Thomas N Kipf and Max Welling. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308*, 2016. 4

[21] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*, 2023. 2

[22] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *CVPR*, 2024. 2

[23] Chongshou Li, Pin Tang, Xinke Li, and Tianrui Li. Enhancing sampling protocol for robust point cloud classification. *arXiv preprint arXiv:2408.12062*, 2024. 2

[24] Chongshou Li, Yuheng Liu, Xinke Li, Yuning Zhang, Tianrui Li, and Junsong Yuan. Deep hierarchical learning for 3d semantic segmentation. *IJCV*, 2025. 2

[25] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM TOG*, pages 1–16, 2019. 3

[26] Xinke Li, Chongshou Li, Zekun Tong, Andrew Lim, Junsong Yuan, Yuwei Wu, Jing Tang, and Raymond Huang. Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In *ACM MM*, 2020. 7

[27] Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. *arXiv preprint arXiv:2402.04717*, 2024. 3

[28] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *CVPR*, 2023. 1

[29] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. Infinicity: Infinite-scale city synthesis. *ICCV*, 2023. 2

[30] Jinxiu Liu and Qi Liu. R3cd: Scene graph to image generation with relation-aware compositional contrastive control diffusion. *AAAI*, 2024. 2

[31] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *CVPR*, 2024. 1

[32] Pengkun Liu, Yikai Wang, Fuchun Sun, Jiafang Li, Hang Xiao, Hongxiang Xue, and Xinzhou Wang. Isotropic3d: Image-to-3d generation based on a single clip embedding. *arXiv preprint arXiv:2403.10395*, 2024. 1

[33] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tok-makov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *ICCV*, 2023. 2

[34] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. *ECCV*, 2024. 2, 3, 6, 7

[35] Diego Mesquita, Amauri Souza, and Samuel Kaski. Rethinking pooling in graph neural networks. *NeurIPS*, 2020. 4

[36] Wamiq Para, Paul Guerrero, Tom Kelly, Leonidas J Guibas, and Peter Wonka. Generative layout modeling using constraint graphs. In *ICCV*, 2021. 3

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 2

[38] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *ICLR*, 2024. 2

[39] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. *CVPR*, 2024. 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[41] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *CVPR*, 2024. 2, 3

[42] L.A. Thorpe and B.R. Shelton. Subjective test methodology: Mos vs. dmos in evaluation of speech coding algorithms. In *Proceedings., IEEE Workshop on Speech Coding for Telecommunications,*, 1993. 6

[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2017. 4, 7

[44] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM TOG*, 2019. 3

[45] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021. 3

[46] Yao Wei, Martin Renqiang Min, George Vosselman, Li Erran Li, and Michael Ying Yang. Planner3d: Llm-enhanced graph prior meets 3d indoor scene explicit regularization. *arXiv preprint arXiv:2403.12848*, 2024. 2

[47] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Motionsc: Data set and network for real-time semantic mapping in dynamic environments. *IEEE Robotics and Automation Letters*, 2022. 2, 3, 5

[48] Yang Wu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Text2lidar: Text-guided lidar point cloud generation via equirectangular transformer. *ECCV*, 2024. 1, 3

[49] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 2

[50] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer: Compositional generative model of unbounded 3d cities. *CVPR*, 2024. 2

[51] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer4d: Compositional generative model of unbounded 4d cities, 2025. 3

[52] Chao Xu, Ang Li, Linghao Chen, Yulin Liu, Ruoxi Shi, Hao Su, and Minghua Liu. Sparp: Fast 3d object reconstruction and pose estimation from sparse views. *ECCV*, 2024. 2

[53] Ning Xu, An-An Liu, Jing Liu, Weizhi Nie, and Yuting Su. Scene graph captioner: Image captioning based on structural visual representation. *Journal of Visual Communication and Image Representation*, 2019. 2

[54] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 2

[55] Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. Diffusion-based scene graph to image generation with masked contrastive pre-training. *arXiv preprint arXiv:2211.11138*, 2022. 2

[56] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. *CVPR*, 2019. 2

[57] Xiuyu Yang, Yunze Man, Jun-Kun Chen, and Yu-Xiong Wang. Scenecraft: Layout-guided 3d scene generation. *arXiv preprint arXiv:2410.09049*, 2024. 2

[58] Zhifei Yang, Keyang Lu, Chao Zhang, Jiaxing Qi, Hanqi Jiang, Ruifei Ma, Shenglin Yin, Yifan Xu, Mingzhe Xing, Zhen Xiao, et al. Mmgdreamer: Mixed-modality graph for geometry-controllable 3d indoor scene generation. *arXiv preprint arXiv:2502.05874*, 2025. 2

[59] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In *Findings of the Association for Computational Linguistics: EACL 2024*, 2024. 6, 7

[60] Guangyao Zhai, Evin Pınar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graph diffusion. *NeurIPS*, 2023. 2, 3

[61] Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. Echoscene: Indoor scene generation via information echo over scene graph diffusion. *ECCV*, 2024. 2, 3

[62] Junge Zhang, Qihang Zhang, Li Zhang, Ramana Rao Kompella, Gaowen Liu, and Bolei Zhou. Urban scene diffusion through semantic occupancy map. *arXiv preprint arXiv:2403.11697*, 2024. 2

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *ICCV*, 2023. 2, 3

[64] Zhiwang Zhang, Dong Xu, Wanli Ouyang, and Luping Zhou. Dense video captioning using graph-based sentence summarization. *IEEE TMM*, 2021. 2

[65] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023. 6, 7

[66] Yang Zhou, Zachary While, and Evangelos Kalogerakis. Scenegraphnet: Neural message passing for 3d indoor scene augmentation. In *ICCV*, 2019. 3