# Scenario Generation for Interactive Urban Environments

Paritosh Sharma, Hui-Yin Wu

July 2025

## Project members

- Scientific team: Paritosh Sharma, Hui-Yin Wu

## 1 Context and objectives

The document highlights the work plan for the the WP4 of the ANR Creative 3D [1] project. The expected outcome of this project is to create a generative model that is capable of creating personal

## 2 Context and objectives

The document highlights the work plan for the the WP4 of the ANR Creative 3D [2] project. The expected outcome of this project is to create a generative model that is capable of creating personalized 3D urban traffic scenarios which can be used for conducting simulations in VR.

## 3 Introduction

Virtual Reality (VR) and Augmented Reality (AR) technologies have advanced significantly in recent years, enabling the creation of immersive and interactive environments for various applications. These can be further used to provide personalized training and rehabilitation scenarios. In the context of low vision rehabilitation, these models can be particularly useful to study pedestrian behaviours under normal and simulated vision. However, most simulated environments suffer from perceptual gaps between the designer, the user, and the system. The existing GusT-3D framework [49], developed during the Creative3D project, provides a foundation to address this. However, the current framework is still limited by its reliance on a fixed set of urban environments and interactions innhibiting its ability to scale.

In parallel, recent works on 3D generative models for urban scenario generation has shown promising results in generating diverse and realistic urban environments. Additionally, these models have been able to capture the diverse nature and the complexities of an urban setting, including the interactions between pedestrians, vehicles, and the environment. Driven by these extraordinary capabilities, exploring the potential of generative models will enable us to create more diverse and realistic urban scenarios.

[1] https://project.inria.fr/creattive3d/
[2] https://project.inria.fr/creattive3d/

# 4 Related Work

In this section, we first start by reviewing prior work on scenegraph-guided generation, which is a structured representation of the objects and their relationships within a scene. We then shift our focus existing urban scenario generation models. Finally, we summarize commonly used datasets used for urban scenario generation, and outline validation methods used to assess the quality, diversity, and controllability of the generated scenarios.

## 4.1 Scenegraph-guided Generation

Scenegraph-guided 3D generation is the process of creating 3D environments by leveraging scene graphs, which are structured representations of the objects and their relationships within a scene. A major advantage of scenegraph-guided generation is that it provides a clear relationship between the elements which allows for better generation than other techniques such as simple text prompts or bounding boxes. Scenethesis [26] uses a Vision-Language Model (VLM) to create a scenegraph with parent-child relationships and localizes objects using 3D bounding boxes. Work by Liu et al. [28] defines scenegraphs as graphs where instance nodes represent countable objects with semantic and positional features, a singleton road node encodes global scene structure, and edges capture both physical proximity among instances and connectivity to the road. X-Scene [55] is another work that uses LLMs (Large-Language Models) to create a scenegraph with nodes (objects) and edges (relationships) to facilitate the generation process. Graphdreamer [15] employs scenegraphs structured around the Visual Genome [21] format, where nodes represent objects with associated attributes, and edges encode the relationships between these objects to guide the generation process. Despite the advancements in scenegraph-guided generation, all of these works focus on static scenes and do not consider dynamic scenarios where the state of the objects change over time.

## 4.2 Diffusion-based 4D Generation

Diffusion models have emerged as a powerful approach for generating high-quality 3D content by iteratively refining a random noise input into a coherent output. Dreamfusion [34] initially text-to-3D by optimizing gaussian noise to align with the distribution derived from a text-conditioned diffusion model using Score Distillation Sampling (SDS). This was extended to 4D scenarios in MaV3D [41] and 4d-fy [2]. However, these methods struggle with precise spatial and temporal control of the generated content.

## 4.3 Urban Scenario Generation

This section discusses pre-existing models used for generating scenarios in urban environments. These models can be broadly classified into procedural and deep generative approaches.

### 4.3.1 Procedural Methods

Procedural methods leverage algorithms and rules to generate urban scenarios, often resulting in highly customizable environments. They can be further divided into two categories:

**Classic Procedural Generation** use rules or constraints to generate scenes. The most popular procedural approach used for urban scenarios is Scenic [14], which allows users to define scenarios using a probabilistic programming language. Similarly, MetaUrban [51] is a popular urban scenario generation framework to create urban micromobility scenarios using the Metadrive [23] simulator.

**LLM-based Procedural Generation** can be used to enhance procedural generation with natural language understanding. For example, ScenicNL [11] and ChatScene [59] use LLMs to generate scenic code from prompts and then define the scenarios in Scenic. TTSG[37] is another framework that uses an to plan a traffic scenario using an LLM in JSON and then render it using CARLA [9]. CityX [62] is multi-agent framework that uses LLM to assemble assets using the procedural content generation (PCG) library as well as plan actions for the agents in the scenario.

Even though procedural approaches can be highly customizable and allow for precise control over the generated scenarios, they typically suffer from a lack of realism, as they rely on predefined rules that may not capture the full complexity of real-world environments. However, they can be useful in creating diverse scenarios that adhere to specific constraints or requirements. For example, a scenario with a crossroad with a specific number of lanes, a certain type of road surface, and a defined layout of buildings. However, no real crossroad would exist in the world that conforms to these specifications.

#### 4.3.2 Deep-Generative Methods

Deep-Generative Approaches have recently become popular for various sorts of 3D generation. To create a complete scenario, some techniques use generative models such as GANs, VAEs or Diffusion models in combination with procedural techniques. These can be broadly classified into the following categories:

**Procedural Environments with Deep-Generative Dynamics** generate the static aspects of the environment using procedural techniques, while the dynamic aspects such as vehicles and pedestrians are generated using generative models. For example, Chatdyn [45] uses CARLA [9] to construct the traffic environment, then populate it with pedestrians and vehicles, each equipped with an LLM agent to generate a high-level scenario plan. This high-level plan is then executed in a low-level PedExecutor using Text2Motion [18] for pedestrians and VehExectutor using a physics-based, history-aware reinforcement learning controller to produce vehicle trajectories.

**Procedural Dynamics with Deep-Generative Environments** generate the dynamic aspects such as vehicles and pedestrians using procedural techniques. UrbanWorld [40] converts 2.5D urban layout data into a structured 3D city with separated assets, applying depth-aware, multi-view diffusion-based texture rendering and UV inpainting to achieve high-fidelity visuals. It also uses an urban MLLM for designing the world and provides dynamic elements which are planned using a random tree path planning algorithm. CityDreamer4D [52] modularly integrates autoregressive token-based generation, neural rendering (e.g., NeRF-style volumetrics), and procedural traffic modeling to synthesize large-scale, time-varying 3D cities.

**Complete Deep-Generative Approaches** recreate the static as well as the dynamic aspects using the learnt representation. Infinicube [31] constructs a large, dynamic 3D voxel world from input HD maps, vehicle positions, and text prompts; then, it generates photorealistic driving videos

and reconstructs the scene into a manipulable 3D environment by fusing voxel- and video-based representations. UniScene [22] UniScene generates driving scenes in three modalities—semantic occupancy, multi-view video, and LiDAR—by first producing a controllable, temporally consistent occupancy sequence from BEV layouts. This occupancy then serves as unified geometric–semantic guidance to synthesize realistic videos and point clouds, ensuring cross-modal consistency and editability.

## 4.4 Datasets

In this section, we review the datasets that are used to train generative models for road crossing scenarios. Table 1 lists some of the datasets commonly used for training and evaluating generative models in urban environments.

Table 1: Overview of selected datasets for foundation model-based scenario generation and analysis.

| **Dataset** | **Year** | **View** | **Source** |
|---|---|---|---|
| SIND [54] | 2022 | BEV | Real |
| Waymo Open [42] | 2020 | FPV | Real |
| Argoverse [5][46] | 2023 | BEV/FPV | Real |
| nuScenes [4] | 2022 | FPV | Real |
| KITTI [17] | 2012 | FPV | Real |
| Cityscapes [6] | 2016 | FPV | .. |
| HoliCity [64] | 2020 | FPV | .. |
| OmniCity [24] | 2023 | FPV | .. |
| GoogleEarth [52] | 2024 | BEV | Real |
| OSM [52] | 2024 | BEV | Real |
| CarlaSC [47] | 2022 | BEV/FPV | Synthetic |
| CityTopia [53] | 2025 | BEV/FPV | .. |

## 4.5 Validation

Table 2 lists common qualitative and quantitative validation methods used to evaluate the performance of generative models for urban road-crossing scenarios. These methods assess the quality, realism, and diversity of generated scenarios to ensure they fulfill their intended applications and research needs.

## 4.6 Code

Currently tested code includes:

### 4.6.1 Scenic

Easy to setup and run on colab since the repository is well maintained.

Table 2: Qualitative and Quantitative Evaluation Methods for Road-Crossing Scenarios

| **Method Type** | **Evaluation Approach** | **Description** | **Use Case** |
|---|---|---|---|
| Qualitative | Human Review | Human experts assess realism, scenario diversity, and layout plausibility. | Validates human-perceived quality and applicability of the scene. |
| | Scenario Visualization | 3D visual inspection or rendered videos showing pedestrian, vehicle, and environment interactions. | Helps detect unrealistic or unnatural behavior/configurations. |
| | Surveys | Collects subjective feedback on realism, perceived difficulty, or stress levels from participants. | Measures human-centric realism or emotional response. |
| | Comparison | Compare scenario features (traffic density, gap opportunities, layout) to real-world statistics or distributions. | Validates realism by matching key scenario characteristics to empirical data. |
| | Failure Cases | Identification and analysis of implausible or unsafe scenarios generated by the model. | Guides iterative model improvement. |
| Quantitative | Scenario Feature Statistics | Statistical analysis of scenario properties (traffic speed, number/duration of gaps, crosswalk presence, etc.). | Ensures generated scenarios span realistic distributions. |
| | Coverage and Diversity Metrics | Measures distributional entropy or spread of key attributes across all generated scenarios. | Assesses generalizability, scenario variety, and model's exploration of edge cases. |
| | Criticality and Opportunity Metrics | Quantifies the frequency and severity of challenging situations (number of safe gaps, minimum feasible gap size, "no-go" cases). | Evaluates risk/challenge spectrum in the scenario catalog. |
| | Sim2Real Gap (Domain Distance) | Computes metrics like KL-divergence, Earth Mover's Distance, t-SNE or FID/KID between generated and real scenario feature distributions. | Evaluates how closely synthetic scenarios match reality. |
| | Controllability Metrics | Measures how well the model can generate scenarios based on prompts (CLIP, BLIP, VQA, etc.) | Assesses controllability and flexibility of the generative model. |

#### 4.6.2 MagicDrive

Had issues running due to model size and GPU memory limitations. Also realised only 2D video generation works for now since the code is not uploaded yet for 3D generation.

#### 4.6.3 MetaUrban/ScenarioNET

Easy to setup and run since docker container is available and working.

#### 4.6.4 Threestudio

Threestudio [29] is a collection of 3D generative techniques. Easy to setup and run since docker container is provided. It is well maintained by the community and has a good documentation. Thus, good to test / refer different generative techniques for 3D.

#### 4.6.5 City Dreamer

CityDreamer4D [53] tried testing the 3D branch with static scenes but encountered issues with the docker container.

#### 4.6.6 Urban Architect

Urban Architect [30] - having some dependency issues with pytorch3d. Models too large ( 100GB).

#### 4.6.7 CLIP

CLIP [35] is a model that can be used to evaluate the quality of generated scenarios. Tested on colab with the pretrained ViT model.

#### 4.6.8 GraphDreamer

GraphDreamer [15] breaks for scenegraph to 3d generation because of GPU memory shortage (surpasses 48 GB). Node level 3d generation works though.

# 5 Problem and Open Question

While recent diffusion-based generative models have demonstrated strong capabilities in synthesizing urban scenes from static scenegraphs, they remain limited in their ability to model dynamic urban scenarios. In particular, they struggle to provide precise spatial and temporal control over objects and their interactions, which is essential for simulating realistic traffic and pedestrian behaviors.

This leads to the following open research question:

"How can diffusion models be grounded using scenegraphs to enable controllable generation of urban scenarios that capture both object relationships and their temporal evolution?"

# 6 Work Plan

## 6.1 Input Scenegraph

We represent a temporal (dynamic) scenegraph as a tuple

$$S = (V, E, \mathcal{T}, \tau) \tag{1}$$

where

- $V = \{v_i\}_{i=1}^{N}$ is a finite set of nodes (objects). Each node $v_i$ is itself a tuple

$$v_i = (\mathrm{id}_i, \mathcal{A}_i, \mathcal{S}_i), \tag{2}$$

  where

  - $\mathrm{id}_i$ is a unique identifier (string),

– $\mathcal{A}_i$ is a (possibly sparse) attribute map:

$$\mathcal{A}_i : \mathcal{K} \rightharpoonup \mathcal{V}$$

(e.g., color, type, material),

– $\mathcal{S}_i = \{(I_{i,k}, \sigma_{i,k})\}_{k=1}^{K_i}$ is an ordered set of temporal states where each

$$I_{i,k} = [t_{i,k}^{(0)}, t_{i,k}^{(1)}] \subseteq \mathcal{T}$$

is a closed time interval and $\sigma_{i,k}$ is a symbolic or structured description of the node's state on that interval (e.g., 'accelerates', 'stops', or a parametric motion/pose).

- $E \subseteq V \times V$ is a set of directed edges. Each edge $e = (v_i, v_j)$ is augmented by

$$e = (v_i, v_j, r_{ij}, \mathcal{I}_{ij}, \delta_{ij}), \tag{3}$$

where

– $r_{ij}$ is the relation type,

– $\mathcal{I}_{ij} = \{J_{ij,\ell}\}_{\ell=1}^{L_{ij}}$ is a set of active time intervals where

$$J_{ij,\ell} = [\tau^{(0)}, \tau^{(1)}] \subseteq \mathcal{T}$$

during which the relation holds,

– $\delta_{ij}$ is an optional descriptive field (string or structured metadata) that further explains the relation (e.g., natural language description or a constraint formula).

- $\mathcal{T}$ is the global time domain. For normalized scenarios, we can take $\mathcal{T} = [0, 1]$; for absolute-time scenarios take $\mathcal{T} = [0, T]$ for some duration $T > 0$.

- $\tau : \bigcup_i \mathcal{S}_i \cup \bigcup_{i,j} \mathcal{I}_{ij} \to \mathcal{P}$ is an optional grounding map from temporal states and relations to parameterized representations (e.g., trajectories, kinematic parameters, animation clips, or low-level simulator actions). $\mathcal{P}$ denotes the space of such parameters.

**Semantics** A dynamic scenegraph $S$ encodes both (i) the *static layout* through attribute maps $\mathcal{A}_i$ and relations $r_{ij}$ that are continuously active (or active on $\mathcal{T}$), and (ii) the *dynamic behaviour* through temporal states $\mathcal{S}_i$ and relation active intervals $\mathcal{I}_{ij}$. The grounding $\tau$ maps symbolic descriptions to concrete, simulator-ready parameters.

**Example**

Figure 1 illustrates the example scenegraph.

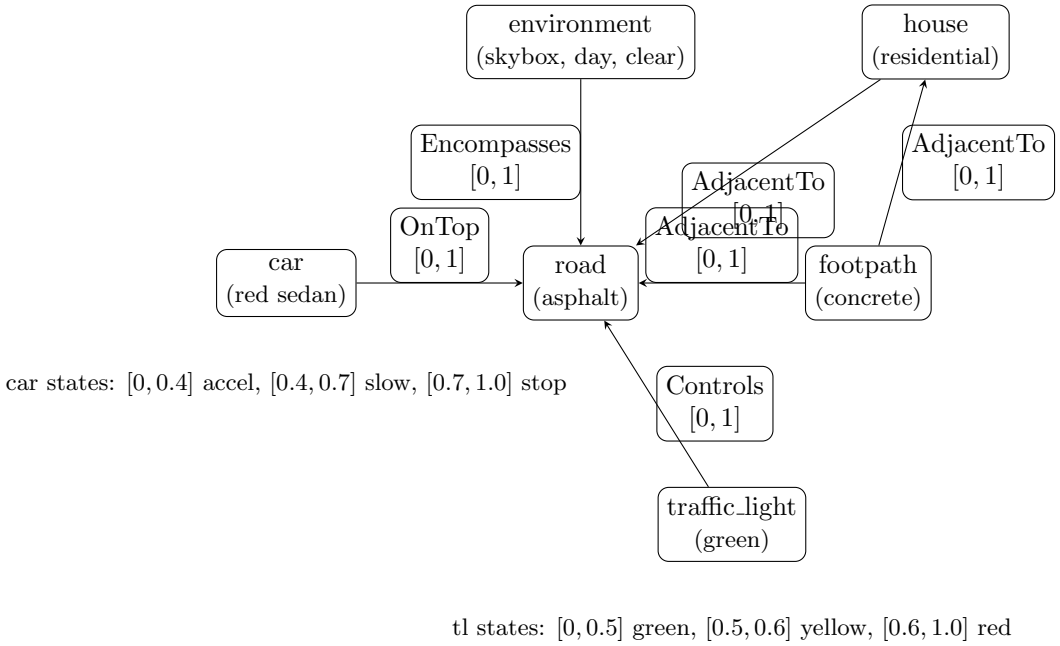


Figure 1: Example dynamic scenegraph (nodes, relations with active intervals, and temporal states).

# Appendix

# 7 Misc

## 7.1 Meeting Notes

### 7.1.1 01/08/2025

- **Clarify Image Usage:** Specify what kinds of images are being referred to in the input for the models (e.g., diagrams, real-world photos, layouts).
- **Add details on scenegraph:** Provide more information about the scenegraphs (e.g., format, usage, etc.) in the context of the models.
- **Avoid Overly High-Level Descriptions:** Some explanations are too abstract.
- **Improve Focus on Vocabulary:** Review and refine terminology. Ensure technical or domain-specific terms are clearly defined and used consistently.
- **Identify and Specify Missing Interaction Types:** Clearly outline which user/system interactions are missing or underexplored.
- **Sharpen Research Question (RQ):** Make the RQ more concrete.
- **Better Categorization of Literature:** Reorganize cited papers using clear categories such as research themes, methodologies, etc.

#### 7.1.2 14/08/2025

- **RQ still too high level:** Provide more information about the scenegraphs (e.g., format, usage, etc.) in the context of the models.
- **Clarify the missing pieces in the SOTA:**
- **How to address:** diversity -¿ generative models, realism -¿ ontology, dynamic -¿ tasks
- **Start with WorkPlan:**
- **Remove relation between GusT-3D and scenegraphs**

### 7.2 Scenegraph-Controlled Diffusion

Despite the advancements in urban scenario generation, very few methods have explored the use of scenegraphs to control the generation process. In recent years, the achievements in text-to-image generation has enabled the advancement of text-to-3D generation using Score Distillation Sampling (SDS) [34] which optimizes a 3D model by aligning 2D images rendered at arbitrary viewpoints with the distribution derived from a text-conditioned diffusion model. Subsequent works including ProlificDreamer [44] introduced Variational Score Distillation (VSD) which addresses the over-regularization and mode collapse issues of SDS by introducing a variational formulation that jointly optimizes the 3D representation and a learnable Gaussian noise distribution, enabling more faithful geometry and richer texture generation. GraphDreamer [15] uses the the SDS process with scenegraphs to enable more structured and controllable scene generation. However, it uses a simplified static scenegraph representation as shown in Visual genome [21] in which, nodes represent the objects in the scene, edges represent the relationship between these nodes and attriubtes represent the properties of the node. Example, Elderly (attribute) man (Node) wearing (edge) a hat(node). This limits the ability to generate dynamic scenarios where the relationships between objects change over time.

### 7.3 Pedestrian-in-the-Loop

VR Simulations have been widely used to study pedestrian behavior and interactions in urban scenarios [48, 43, 32, 39]. This human-in-the-loop approach (also referred as pedestrian-in-the-loop[19]) allows researchers to collect data on how pedestrians interact with their environment, including their decision-making processes, movement patterns, and responses to various stimuli. These simulations can be used to study a wide range of scenarios, from simple road crossings to complex urban environments with multiple interacting agents. More recently, JaywalkerVR [32], a VR human-in-the-loop simulator, used CARLA [9] to create four different scenarios: jaywalking, parked cars, four-way stops, and parking lot entrances. The authors created the CARLA-VR dataset by collecting data from 80 participants for these scenarios and used AgentFormer [57], a trajectory forecasting model for evaluation. Despite the obvious advantages, developing such simulators is still challenging because of the perceptual gaps between the designer, the user and the system as identified by Dourish [10].

The GUsT-3D framework [49] addresses this by first defining the scene using a scenegraph, which captures the relationships between different elements in the scenario (ontology) and then defining the task to be carried out during the course of the scenario using a GUTasks (intersubjectivity).

Lastly, it uses a query component for logging and post-scenario analysis of the experience (intentionality). This framework was also applied by creating a dataset of 6 road-crossing scenarios to study pedestrian behavior under normal and low vision [50]. Even though GusT-3D framework addresses the perceptual gaps which were identified by Dourish, it still relies on a fixed set of scenarios and interactions.

## 7.4 Grounding: Input for Scene Generation

Generative models have already been applied to urban scenario generation, where models synthesize plausible urban environments, pedestrian layouts, or vehicle positions from various types of input. An input in the context of a generative model is any data—such as a prompt, image, or scenario graph—provided to influence the output. Grounding, on the other hand, is the process of linking elements of that input to specific, coherent representations in the generated scenario, such as ensuring the "footpath" appears visually plausible, is correctly positioned on the "road", and respects spatial relationships or physical constraints. Grounding ensures the generated scenario isn't just randomly composed but meaningfully aligned with the intended semantics of the input. The common ways to do grounding are,

### 7.4.1 Rules and Constraints

**Rules** can be used to give a prescriptive logic that defines how to generate or modify a scenario. Rule-based systems are necessarily procedural, meaning they follow a set of predefined steps or algorithms to create scenarios. Table 3 lists some of the recent rule-based models for scenario generation. Although these systems allow users to define rules for generating scenarios—such as the placement of buildings, roads, and other elements—they often require an intermediate representation which captures the real-world environments. Creating such representations can be challenging, as it requires a deep understanding of the underlying rules and relationships between different elements. Additionally, rule-based systems can be limited in their ability to generate diverse and realistic scenarios, as they rely on predefined rules that may not capture the full complexity of real-world environments.traffic

| Model | Technique | Output |
|---|---|---|
| CityEngine [33] | Procedural Modeling with Rules | 3D scene |
| Infinigen [36] | Procedural Modeling with Rules | 3D scene |
| MetaUrban [51] | Description Scripts for Scene Layout | 3D Scenario |

Table 3: Rule-based Models for Urban Scenario Generation

**Constraints** define conditions that must be satisfied to get a valid scenario. Scenic [14] is a probabilistic programming language that allows users to define constraints for generating scenarios. It uses a declarative approach to specify the properties of the scenario, such as the layout, objects, and their relationships. These can then be rendered using a frontend such as CARLA [9]. However, it suffers from the same limitations of rule-based systems, as it can only generate scenarios that fit within the defined constraints, potentially missing out on the richness and variability of real-world environments.

### 7.4.2 Prompts

Prompts are textual cues provided to the generative model to guide the scenario creation process. Since the popularity of LLMs like GPT-3, prompts have become a common way to interact with generative models. They can be used to specify the desired characteristics of the scenario, such as the type of environment, objects, and their relationships. Table 4 lists some of the recent prompt-based models for scenario generation.

| Model | Technique | Output |
|---|---|---|
| ScenicNL [11] | Converts LLM Prompts to Scenic Constraints | Scenic Scenario |
| ChatScene [59] | Conversational Agent for Scenario Definition using Scenic | Scenic Scenario |
| LayoutGPT [12] | Prompts converted to CSS-like Layout Formatting by LLMs | Layout Representation |
| ChatDyn [45] | LLM-based planning and low-level trajectory generation for Pedestrian and Vehicle | 3D Scenario |
| Work by Feng et al. [13] | JSON to describe layout and 3D models | 3D Scene |
| TTSG [37] | LLM-based planning and retrieval | 3D Scenario |
| 3D-SceneDreamer [61] | Prompt to point-of-view image followed by image to 3D | 3D Scene |
| GraphCanvas3D [27] | Uses LLM to create a scenegraph and then a 3D scene | 3D Scene |
| Scenethesis [26] | Uses LLM to create a scenegraph and then a 3D scene | 3D Scene |
| SceneX [63] | LLM to plan PCG (Procedural Controllable Generation) | 3D Scene |
| Surreal Drivers [20] | Chain-of-thought prompts | 3D Scene |
| Text2nerf [58] | Text prompts | 3D Scene |
| X-Scene [55] | Text prompts | 3D Scene |

Table 4: Prompt-based Models for Urban Scenario Generation

### 7.4.3 Layouts

Layouts are structured representations of scenarios that capture the spatial relationships between different elements, such as buildings, roads, and pedestrians. These may include bird's-eye views, top-down maps, or other forms of spatial representations that provide a high-level overview of the scenario. Table 5 lists some of the recent layout-based models for scenario generation.

| Model | Technique | Output |
|---|---|---|
| CC3D [1] | 2D Layout-based 3D Scene Generation | 3D Scene |
| CityDreamer4D [53] | Uses a Layout Generator and a traffic scenario generator | 3D Scenario |
| Infinicube [31] | Text prompts, HD Maps and Bounding Boxes | 3D Scenario |
| Work by Zhang et al. [60] | BEV map | 3D Scene |
| UniScene [22] | BEV map | Multi-view video |
| Savkin et al. [38] | Scenegraph | Scenario Image |

Table 5: Layout-based Models for Urban Scenario Generation

### 7.4.4 Multimodal

Multimodal inputs combine different types of data, such as text, images, and structured representations, to provide a richer context for scenario generation. Table 6 lists some of the recent multimodal models for scenario generation.

| Model | Input Type | Output |
|---|---|---|
| CityX [62] | Prompt, OSM file or Semantic Map | 3D Scenario |
| CityCraft [8] | Layout data and text prompts | 3D Scene |
| Work by Liu et al. [28] | Scenegraph assisted using text prompts | 3D Scene |
| GAUDI [3] | Conditioning using prompts or point-of-view images | 3D Scene |
| MagicDrive3D [16] | Text prompts, Bird Eye View (BEV) map and 3D Bounding Boxes | Reconstructed 3D video |
| Scene123 [56] | Text prompt, point-of-view image or Text Description | 3D Scene |
| StreetScapes [7] | BEV and height map with support for prompts | Video |
| Urban Architect [30] | 3D Layout and Text Prompts | 3D Scene |
| Urban World [40] | Layout (generation) and prompts (refinement) | 3D Scenario |
| Wonderplay [25] | point-of-view image and action (physics) | 3D Video |

Table 6: Multimodal Models for Urban Scenario Generation

# References


[1] S. Bahmani, J. J. Park, D. Paschalidou, X. Yan, G. Wetzstein, L. Guibas, and A. Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7171–7181, 2023.

[2] S. Bahmani, I. Skorokhodov, V. Rong, G. Wetzstein, L. Guibas, P. Wonka, S. Tulyakov, J. J. Park, A. Tagliasacchi, and D. B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7996–8006, 2024.

[3] 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.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] B. Deng, R. Tucker, Z. Li, L. Guibas, N. Snavely, and G. Wetzstein. Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[8] J. Deng, W. Chai, J. Huang, Z. Zhao, Q. Huang, M. Gao, J. Guo, S. Hao, W. Hu, J.-N. Hwang, et al. Citycraft: A real crafter for 3d city generation. *arXiv preprint arXiv:2406.04983*, 2024.

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[10] P. Dourish. *Where the action is: the foundations of embodied interaction*. MIT press, 2001.

[11] K. Elmaaroufi, D. Shanker, A. Cismaru, M. Vazquez-Chanlatte, A. Sangiovanni-Vincentelli, M. Zaharia, and S. A. Seshia. Scenicnl: generating probabilistic scenario programs from natural language. *arXiv preprint arXiv:2405.03709*, 2024.

[12] 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.

[13] Y. Feng, J. Jiang, J. Ren, W. Li, R. Li, and X. Fan. Text-guided editable 3d city scene generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[14] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pages 63–78, 2019.

[15] G. Gao, W. Liu, A. Chen, A. Geiger, and B. Schölkopf. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21295–21304, 2024.

[16] R. Gao, K. Chen, Z. Li, L. Hong, Z. Li, and Q. Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[18] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024.

[19] M. Hartmann, M. Viehweger, W. Desmet, M. Stolz, and D. Watzenig. "pedestrian in the loop": An approach using virtual reality. In *2017 XXVI International Conference on Information, Communication and Automation Technologies (ICAT)*, pages 1–8. IEEE, 2017.

[20] Y. Jin, R. Yang, Z. Yi, X. Shen, H. Peng, X. Liu, J. Qin, J. Li, J. Xie, P. Gao, et al. Surrealdriver: Designing llm-powered generative driver agent framework based on human drivers' driving-thinking data. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 966–971. IEEE, 2024.

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[22] B. Li, J. Guo, H. Liu, Y. Zou, Y. Ding, X. Chen, H. Zhu, F. Tan, C. Zhang, T. Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11971–11981, 2025.

[23] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

[24] W. Li, Y. Lai, L. Xu, Y. Xiangli, J. Yu, C. He, G.-S. Xia, and D. Lin. Omnicity: Omnipotent city understanding with multi-level and multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17397–17407, 2023.

[25] Z. Li, H.-X. Yu, W. Liu, Y. Yang, C. Herrmann, G. Wetzstein, and J. Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. *arXiv preprint arXiv:2505.18151*, 2025.

[26] L. Ling, C.-H. Lin, T.-Y. Lin, Y. Ding, Y. Zeng, Y. Sheng, Y. Ge, M.-Y. Liu, A. Bera, and Z. Li. Scenethesis: A language and vision agentic framework for 3d scene generation. *arXiv preprint arXiv:2505.02836*, 2025.

[27] L. Liu, S. Chen, S. Jia, J. Shi, Z. Jiang, C. Jin, W. Zongkai, J.-N. Hwang, and L. Li. Graph canvas for controllable 3d scene generation. *arXiv preprint arXiv:2412.00091*, 2024.

[28] Y. Liu, X. Li, Y. Zhang, L. Qi, X. Li, W. Wang, C. Li, X. Li, and M.-H. Yang. Controllable 3d outdoor scene generation via scene graphs. *arXiv preprint arXiv:2503.07152*, 2025.

[29] Y.-T. Liu, Y.-C. Guo, V. Voleti, R. Shao, C.-H. Chen, G. Luo, Z. Zou, C. Wang, C. Laforte, Y.-P. Cao, et al. Threestudio: A modular framework for diffusion-guided 3d generation. *cg. cs. tsinghua. edu. cn*, 2023.

[30] F. Lu, K.-Y. Lin, Y. Xu, H. Li, G. Chen, and C. Jiang. Urban architect: Steerable 3d urban scene generation with layout prior. *arXiv preprint arXiv:2404.06780*, 2024.

[31] Y. Lu, X. Ren, J. Yang, T. Shen, Z. Wu, J. Gao, Y. Wang, S. Chen, M. Chen, S. Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024.

[32] K. Mukoya, E. Weng, R. Choudhury, and K. Kitani. Jaywalkervr: A vr system for collecting safety-critical pedestrian-vehicle interactions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9600–9607. IEEE, 2024.

[33] Y. I. Parish and P. Müller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308, 2001.

[34] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[35] 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*, pages 8748–8763. PmLR, 2021.

[36] A. Raistrick, L. Lipson, Z. Ma, L. Mei, M. Wang, Y. Zuo, K. Kayan, H. Wen, B. Han, Y. Wang, et al. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12630–12641, 2023.

[37] B.-K. Ruan, H.-T. Tsui, Y.-H. Li, and H.-H. Shuai. Traffic scene generation from natural language description for autonomous vehicles with large language model. *arXiv preprint arXiv:2409.09575*, 2024.

[38] A. Savkin, R. Ellouze, N. Navab, and F. Tombari. Unsupervised traffic scene generation with synthetic 3d scene graphs. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1229–1235. IEEE, 2021.

[39] S. Schneider and K. Bengler. Virtually the same? analysing pedestrian behaviour by means of virtual reality. *Transportation research part F: traffic psychology and behaviour*, 68:231–256, 2020.

[40] Y. Shang, Y. Lin, Y. Zheng, H. Fan, J. Ding, J. Feng, J. Chen, L. Tian, and Y. Li. Urbanworld: An urban world model for 3d city generation. *arXiv preprint arXiv:2407.11965*, 2024.

[41] U. Singer, S. Sheynin, A. Polyak, O. Ashual, I. Makarov, F. Kokkinos, N. Goyal, A. Vedaldi, D. Parikh, J. Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023.

[42] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[43] T. T. M. Tran, C. Parker, and M. Tomitsch. A review of virtual reality studies on autonomous vehicle–pedestrian interaction. *IEEE Transactions on Human-Machine Systems*, 51(6):641–652, 2021.

[44] 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.

[45] Y. Wei, J. Wang, Y. Du, D. Wang, L. Pan, C. Xu, Y. Feng, B. Dai, and S. Chen. Chatdyn: Language-driven multi-actor dynamics generation in street scenes. *arXiv preprint arXiv:2412.08685*, 2024.

[46] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

[47] J. Wilson, J. Song, Y. Fu, A. Zhang, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari. Motionsc: Data set and network for real-time semantic mapping in dynamic environments. *IEEE Robotics and Automation Letters*, 7(3):8439–8446, 2022.

[48] H. Wu, D. H. Ashmead, H. Adams, and B. Bodenheimer. Using virtual reality to assess the street crossing behavior of pedestrians with simulated macular degeneration at a roundabout. *Frontiers in ICT*, 5:27, 2018.

[49] H.-Y. Wu, F. Robert, T. Fafet, B. Graulier, B. Passin-Cauneau, L. Sassatelli, and M. Winckler. Designing guided user tasks in vr embodied experiences. *Proceedings of the ACM on Human-Computer Interaction*, 6(EICS):1–24, 2022.

[50] H.-Y. Wu, F. A. S. Robert, F. F. Gallo, K. Pirkovets, C. Quere, J. Delachambre, S. Ramanoël, A. Gros, M. Winckler, L. Sassatelli, et al. Exploring, walking, and interacting in virtual reality with simulated low vision: a living contextual dataset (2023), 2023.

[51] W. Wu, H. He, Y. Wang, C. Duan, J. He, Z. Liu, Q. Li, and B. Zhou. Metaurban: A simulation platform for embodied ai in urban spaces. *arXiv e-prints*, pages arXiv–2407, 2024.

[52] H. Xie, Z. Chen, F. Hong, and Z. Liu. Citydreamer: Compositional generative model of unbounded 3d cities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9666–9675, 2024.

[53] H. Xie, Z. Chen, F. Hong, and Z. Liu. Citydreamer4d: Compositional generative model of unbounded 4d cities. *arXiv e-prints*, pages arXiv–2501, 2025.

[54] Y. Xu, W. Shao, J. Li, K. Yang, W. Wang, H. Huang, C. Lv, and H. Wang. Sind: A drone dataset at signalized intersection in china. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2471–2478. IEEE, 2022.

[55] Y. Yang, A. Liang, J. Mei, Y. Ma, Y. Liu, and G. H. Lee. X-scene: Large-scale driving scene generation with high fidelity and flexible controllability. *arXiv preprint arXiv:2506.13558*, 2025.

[56] Y. Yang, F. Yin, J. Fan, X. Chen, W. Li, and G. Yu. Scene123: One prompt to 3d scene generation via video-assisted and consistency-enhanced mae. *arXiv preprint arXiv:2408.05477*, 2024.

[57] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9813–9823, 2021.

[58] J. Zhang, X. Li, Z. Wan, C. Wang, and J. Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7749–7762, 2024.

[59] J. Zhang, C. Xu, and B. Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024.

[60] J. Zhang, Q. Zhang, L. Zhang, R. R. Kompella, G. Liu, J. Li, and B. Zhou. Urban scene diffusion through semantic occupancy map. *arXiv preprint arXiv:2403.11697*, 2024.

[61] S. Zhang, Y. Zhang, Q. Zheng, R. Ma, W. Hua, H. Bao, W. Xu, and C. Zou. 3d-scenedreamer: Text-driven 3d-consistent scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10170–10180, 2024.

[62] S. Zhang, M. Zhou, Y. Wang, C. Luo, R. Wang, Y. Li, Z. Zhang, and J. Peng. Cityx: Controllable procedural content generation for unbounded 3d cities. *arXiv preprint arXiv:2407.17572*, 2024.

[63] M. Zhou, Y. Wang, J. Hou, S. Zhang, Y. Li, C. Luo, J. Peng, and Z. Zhang. Scenex: Procedural controllable large-scale scene generation. *arXiv preprint arXiv:2403.15698*, 2024.

[64] Y. Zhou, J. Huang, X. Dai, S. Liu, L. Luo, Z. Chen, and Y. Ma. Holicity: A city-scale data platform for learning holistic 3d structures. *arXiv preprint arXiv:2008.03286*, 2020.