Survey on Compositional 3D Indoor Scene Generation

H. I. I. Tam[1] H. I. D. Pun[1] A. T. Wang[1] X. Sun[1] Q. Wu[1] H. Lee[1] A. X. Chang[1,2] M. Savva[1]

[1]Simon Fraser University [2]Canada CIFAR AI Chair, Amii

3dlg-hcvc.github.io/Comp3DSceneGen

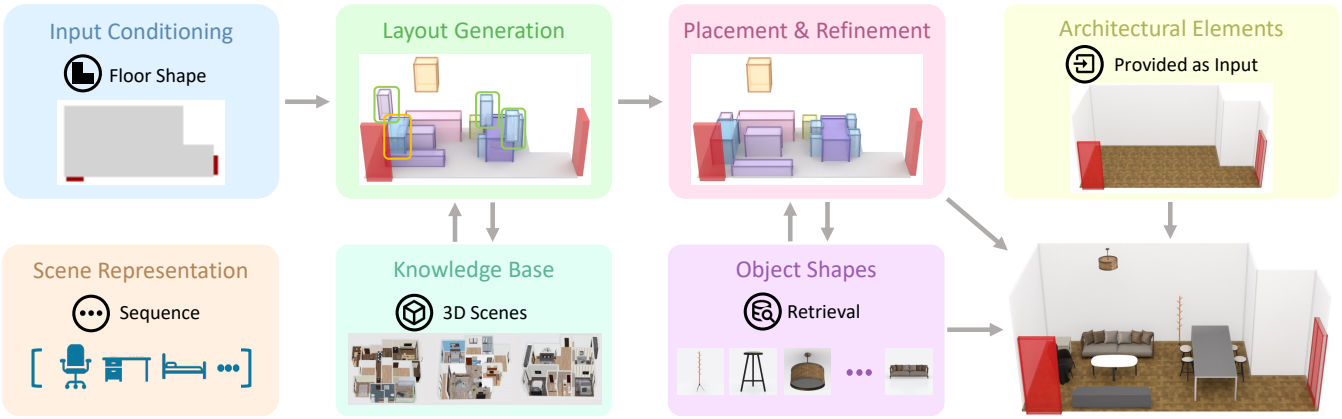


Figure 1: *Overview of the key components of a 3D scene generation method. Given an input condition (Sec. 3.1), and prior knowledge about how objects are arranged (Sec. 3.3), compositional systems typically first generate a coarse layout of the scene (Sec. 3.4), determine and refine object placements (Sec. 3.5), obtain corresponding objects (Sec. 3.6), and combine them with architectural elements (Sec. 3.7) to produce the output 3D scene. As part of this process, an important design choice is the scene representation (Sec. 3.2).*

Abstract

Compositional 3D indoor scene generation is a long-standing problem and a rapidly evolving area of research spanning computer graphics, 3D computer vision, and machine learning. The goal is to model the complex relationships among objects and their spatial and functional arrangements within a scene, enabling the creation of rich, diverse, and useful 3D environments for a wide range of applications. This survey offers a comprehensive overview of the state of the art, formulating a unifying framework for analyzing scene generation systems and systematically categorizing existing methods according to their approaches to key components. We review recent progress, analyze the strengths and limitations of different paradigms, and highlight both major advances and open challenges. Our survey aims to serve as a resource for researchers and practitioners, offering insights into the current landscape and inspiring new ideas for future work in this area.

1 Introduction

3D scenes play a central role in various domains such as video games, virtual and augmented reality, robotic simulation, and digital content creation. Among them, compositional 3D scenes—where each object is represented as individual geometry that can be positioned, oriented, and manipulated—enable rich interactions ith objects, forming the basis for dynamic environments across a wide range of applications. Manual creation of such scenes is labor-intensive, time-consuming, and requires specialized expertise in 3D modeling. Generation of compositional 3D scenes therefore represents an important research direction for enabling efficient, accessible, and interactive scene construction.

However, compositional 3D scene generation remains highly challenging. A generation method must satisfy several often competing requirements. First, generated scenes must be physically plausible, with object placements that are consistent with basic physical laws. Second, the scenes should be functional and coherent, with objects arranged to support intended uses and reflect common real-world patterns. Third, generation methods need to capture user intent, expressed through text, sketches, images, or other modalities. Finally, the scenes should be diverse and varied, encompassing a wide range of configurations within the requirements.

This survey provides a comprehensive overview of compositional 3D scene generation. We introduce a general blueprint for scene generation methods, outlining the core components and their

interactions. Following this framework, we systematically review existing approaches from rule-based and supervised learning methods to those leveraging large language models (LLMs) and vision-language models (VLMs), analyzing their strengths and limitations. We highlight key challenges, and point to promising directions for future work. Our goal is to provide researchers and practitioners with a clear map of the current landscape and to inspire new advances in compositional 3D scene generation.

Scope. Our survey focuses on generation of *compositional 3D indoor scenes*. We thus largely exclude from consideration methods for non-indoor scene generation, as they involve fundamentally different semantics and design principles compared to indoor scenes, and non-compositional (i.e., "monolithic") 3D scene generation. We additionally do not focus on general 3D object generation methods except where integral to the overall scene generation method.

Related surveys. The surveys by Patil et al. [PPL*24] and Ma et al. [MBS*24] cover 3D tasks other than scene generation, including scene understanding and reconstruction. Patil et al. [PPL*24] only include works up to 2023, notably excluding the recent use of LLMs in scene generation, while Ma et al. [MBS*24] focuses on the use of LLMs for 3D and includes just a few works applying LLMs to scene-scale generation. Fime et al. [FMD*25] surveys primarily generation methods but focuses especially on 2D image generation and covers only older works in 3D generation. The recent surveys by Liu et al. [LXN*25] and Wen et al. [WXC*25] tackle 3D scene generation more directly. However, Liu et al. [LXN*25] focuses predominantly on floorplan and layout generation, and Wen et al. [WXC*25] is more focused on 3D representation choices and does not specifically cover methods for compositional 3D scene generation which is our goal. In contrast, we survey compositional 3D scene generation and provide a framework for understanding the key components of recent methods.

Survey organization. We first provide useful background knowledge (Sec. 2). Then we overview the high-level components of compositional scene generation systems (Sec. 3) and systematically categorize prior work (Sec. 4). We summarize how generated scenes are currently evaluated (Sec. 5). Finally, we discuss current challenges and future directions for scene generation (Sec. 6).

2 Background

We begin by defining key terms and concepts used throughout this survey. We also briefly describe statistical and deep learning techniques commonly used in compositional 3D scene generation.

2.1 Definition of Compositional 3D Scenes

Given this survey's focus on *compositional* 3D scene generation, a definition of what constitutes "composition" is critical. A compositional scene provides a structured representation describing distinct objects across spatial scales. For example, furniture items such as coffee tables and couches are placed in living rooms, utensils and plates are placed on dining tables, and living rooms and kitchens are connected in ways that facilitate common human activities such as preparing and having dinner. These examples illustrate composition at three different spatial scales: furniture in a room, tabletop items on a furniture piece, and rooms in a house.

We refer to a scene $S = \{O, L, A\}$ as the set of objects O in the scene, the layout L describing how all the objects are positioned, and the architectural elements A delineating the architecture of all rooms. There are several choices in picking a representation for each of the elements O, L, and A, each with computational tradeoffs. Some common choices for representing each object in O are polygonal meshes, occupancy grids, signed distance functions, and neural radiance fields. Common choices for the layout L are a flat list of transformations applied to each of the objects, or scene graphs defining a transformation hierarchy with objects at the nodes. The architectural elements in A are similar in terms of composition to the objects in O but are usually separated due to the conceptual differences between placement of objects and construction of architecture. Often, these elements are represented by planar primitives defining each wall, floor, and ceiling surface, with additional parameters specifying openings such as doors and windows.

The above compositional scene definition has three desirable properties. First, the discrete nature of objects in O, corresponding placement parameters in L, and architectural elements in A allows for direct and localized control of each aspect of the scene. Second, localized control enables easier human-driven editing operations that change specific objects, their placements, and other properties of the scene. Third, this structured scene representation allows for easy definition and compact storage of animation sequences that modify object state or properties, making it easier to model animated or interactive 3D scenes. These three fundamental advantages explain the increasing popularity for generation methods that produce 3D scenes compositionally.

Despite many advantages, compositional representation of 3D scenes also brings challenges. First, each of the components (O, L, and A) admits several options for representation, making the overall representation space much more complex relative to monolithic representations. Second, the heterogeneity of the components and unordered set nature of elements make it more challenging to apply neural architectures. Third, there is a limited amount of data available to the research community in such structured formats. The last challenge stems from the difficulty of parsing real-world scene data (i.e., 3D reconstruction into structured representations) as well as the wide spectrum of detail levels and conventions in human-authored 3D scenes, making it hard to combine datasets.

2.2 Learning Paradigms

We briefly describe learning paradigms and tools that provide foundations for modeling and inference in compositional 3D scene generation. These include probabilistic models, sampling and optimization methods, neural architectures, and generative models.

Probabilistic modeling tools. A variety of approaches have been used to model the probability distributions characterizing 3D scene arrangements. These include Bayesian networks, Gaussian mixture models (GMMs), Dirichlet process mixture models (DPMMs), Boltzmann distributions, topic models, kernel density estimation, and factor graphs. The fundamentals of these techniques are described in more detail by Koller and Friedman [KF09]. These approaches have been used in 3D scene generation to capture probability distributions describing the existence and spatial relationships of particular objects in a scene.

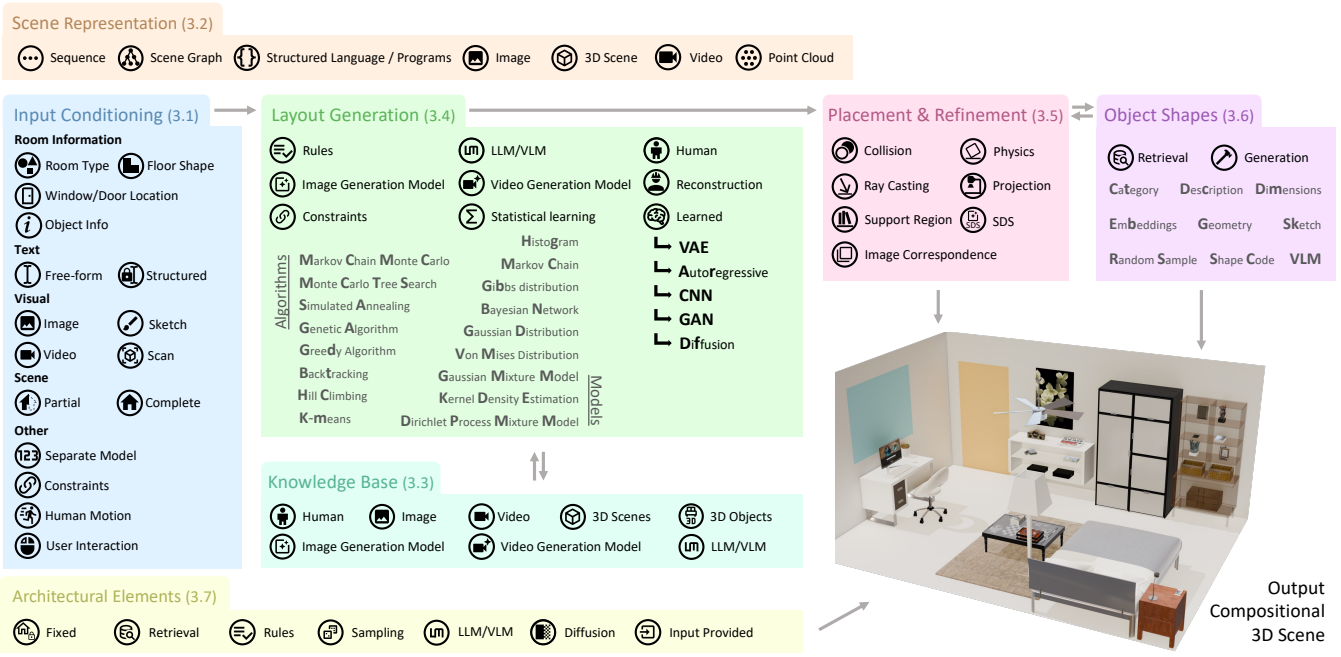


Figure 2: *Blueprint illustrating components of compositional 3D scene generation systems, and implementation choices made by prior work for each component. Parentheses show which section discusses design choices for each component.*

Sampling and optimization tools. Exact inference in the above models is often infeasible, so a range of sampling and optimization techniques are used, including Markov Chain Monte Carlo (MCMC) sampling, Metropolis-Hastings (MH), Gibbs sampling, Simulated annealing (SA), Iterated local search (ILS) and hill climbing, gradient descent, position-based dynamics (PBD), and K-means clustering. Gelman et al. [GCSR95] provide a thorough description of the fundamentals of such sampling strategies. These techniques enable efficient sampling and inference for the fairly complex probabilistic models learned from compositional 3D scene data. A key challenge is the typically transdimensional nature of the probability distributions of realistic indoor scenes, where there is a highly variable number of objects and potential object placements.

Neural architectures. Increasingly, a variety of neural network architectures have been used to learn from data, often replacing traditional probabilistic models. this has been enabled by advances in efficient optimization through backpropagation [RHW86], larger-scale datasets, and hardware enabling faster compute. Popular architectures include multilayer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs) and their gated variants LSTMs and GRUs, and most recently transformers [VSP*17; DBK*21]. Goodfellow et al. [GBC16] provide a more comprehensive overview from fundamentals.

Generative models. Generative models used for 3D scene generation are neural networks designed to learn a data distribution so that they can generate new samples resembling the data. Popular examples include variational autoencoders (VAEs) [KW13], generative adversarial networks (GANs) [GPM*14], autoregressive models [VKK16; VKE*16], and most recently diffusion models [SWMG15; HJA20]. These approaches yield exceptional fidelity and diversity for generation of images, and have since been extended to other modalities including text, audio, and 3D data.

3 A System Blueprint for Scene Generation

In this section, we outline a common set of components that are present in most systems for compositional 3D scene generation. This blueprint serves as a framework for positioning and understanding the design choices made by different scene generation methods, which we will discuss in Sec. 4.

Fig. 1 depicts our proposed blueprint for a compositional 3D scene generation system. The entire process can be conditioned on different user input, such as text, images, or scene graphs (Sec. 3.1). Given the input specifications, a scene generation system models the scene using an internal representation, such as structured language or images (Sec. 3.2), leveraging prior knowledge about valid object arrangements (Sec. 3.3) to sample a layout for the scene as captured in the representation (Sec. 3.4). This layout can be refined to improve physical plausibility, functionality, or other factors (Sec. 3.5). Finally, 3D object assets are retrieved or generated based on the predicted parameters, such as the object class and size, (Sec. 3.6) and rendered in the layout to populate the scene.

Fig. 2 summarizes the different design choices for each component of the blueprint. We use icons throughout Sec. 4 to indicate specific choices made by different methods within the overall blueprint. In the following sections, we describe each component in more detail, discussing the various options and trade-offs involved in their design and implementation.

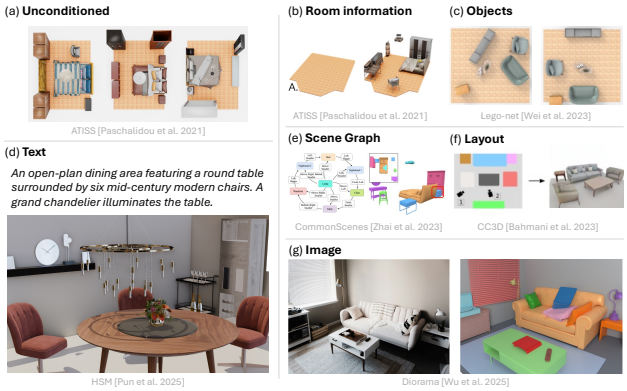


Figure 3: *Examples of different input conditions (Sec. 3.1) and corresponding generated scenes.*

3.1 Conditioning

While scenes can be generated without any conditioning, it is often useful to provide guidance so that the output matches specific requirements. Conditioning can take many forms (see Fig. 3), ranging from minimal signals (e.g., room type) to highly detailed inputs (e.g., partial layouts or images of scenes). An effective generation system should respect the conditioning faithfully, while completing the remaining aspects of the scene in a plausible manner.

Room information. The simplest conditioning is specifying the desired room type, such as "bedroom" or "kitchen." This constrains the set of relevant objects and their typical arrangements. It can also enable the use of models specialized for different room categories [WLW*19; TNM*24]. More detailed room-level inputs include floorplans [PKS*21; HAD*24; ÇHS*24], which define the spatial boundaries for placing objects. A floorplan can be represented as a 2D mask or a polygonal outline. Additional architectural elements, such as door and window locations [WYN21; SGC25], provide further constraints: they must remain unobstructed and often influence functional placements (e.g., an umbrella stand near the door or a sofa facing a window).

Partial scene. Another common form of conditioning is a partially specified scene in which some objects are already placed [SLLG03; FCW*17; MPF*18; BA25]. This provides context for the generation process, which can then add missing objects, refine layouts, or complete the scene. Partial inputs can range from a rough arrangement of a few key items to a nearly finished scene that requires modification. Such conditioning is especially useful for scene completion, editing, and interactive design scenarios.

Text. Natural language conditioning offers a flexible and intuitive way to describe the desired scene [CS01; CMS*15]. Prompts can range from short phrases (e.g., "a modern office") [ZHX*24] to detailed descriptions specifying room type, required objects, spatial relationships, and stylistic attributes [YSW*24; PTW*25]. Text can also express negative constraints by excluding objects or features. However, language is inherently ambiguous, and phrases like "a cozy living room" may be interpreted differently depending on context and cultural bias, posing challenges for generation.

Image. Images can serve as visual cues for guiding scene generation. A single RGB perspective image is the most common form, providing a direct reference for objects and layout that the generated scene should reproduce [ZWWZ25; HZB*25]. However, a single view captures only part of the scene, making it difficult to fully constrain 3D structure. Panoramic images mitigate this limitation by covering a 360° view, offering more complete information on the layout and visible objects [ZCC*21; DFB*24].

Video or multi-view images. Video sequences or multi-view images provide richer conditioning signals than a single image [CCPS25; MPNF22]. They help resolve scale-depth ambiguities and place objects into a global 3D coordinate frame. At the same time, the multi-view setting introduces challenges such as occlusion, objects entering or leaving the field of view, and tracking errors that may fragment or confuse object identities, all of which can harm the quality of the generated scene.

Scan. 3D scans offer direct structural conditioning. They capture room geometry and existing objects explicitly, typically as volumetric grids [ADD*19] or point clouds [FSL*15; LYS*20]. Generation models can then align CAD objects [ADD*19] or add missing elements to complete the scene [KPZ*20]. The challenge lies in effectively interpreting noisy or incomplete scan data and integrating it with learned priors to produce coherent, high-quality scenes.

Human pose. Beyond conditioning directly specifying scene structure or appearance, some works incorporate human-centric signals [YWL*22; YHT*23; NDHN22]. A sequence of human poses captures how people interact with the scene, providing cues about free space, object placements, and functional arrangements. Such conditioning emphasizes affordances and usability, guiding generation toward scenes supporting human activities.

3.2 Representation

Indoor scene synthesis relies fundamentally on how the underlying representation structures the generation process. A variety of different representations are used to model the scene during the process of generation (see Fig. 4). This section reviews the principal paradigms of scene representation, outlining their modeling strategies, advantages, and limitations.

Object sequences. An object sequence representation (Fig. 4 (a)) encodes an indoor scene as a sequence of objects, each represented by a feature vector capturing essential attributes. In early work, the object sequence only contains basic information such as position (x_i, y_i, z_i), size (w_i, l_i, h_i), and orientation θ_i of the object [WYN21; ZYM*20; PKS*21; LGWM22]. More recently, an object shape feature f_i is introduced to better control retrieval or generation of each object [HAD*24; YJZH24; SYW*25]. Formally, a scene S with N objects is represented by $S = \{\mathbf{o}_1, ..., \mathbf{o}_N\}$, where:

$$\mathbf{o}_i = [x_i, y_i, z_i, w_i, l_i, h_i, \theta_i, c_i, f_i]$$

This representation is simple and compact, allowing for easy integration with sequential generative models such as Transformers. However, it inherently lacks explicit spatial relationships, making interactions between objects indirectly modeled.

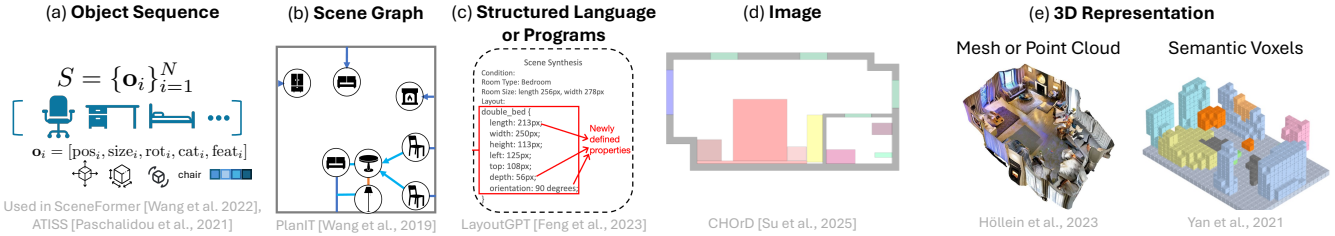


Figure 4: Scene representations *(Sec. 3.2). During scene generation, the scene can be represented as a set or sequence of objects with attributes (e.g., object category, position, size, orientation), a graph including objects and relationships between the objects, or structured language or programs. For better encoding of appearance and spatial relationships between objects, it is possible to represent the scene as images or 3D representations. Monolithic 3D representations such as meshes and point clouds that do not break the scene into objects, need extra steps to extract the objects. In contrast, semantic-aware 3D representations can allow for easier extraction of objects.*

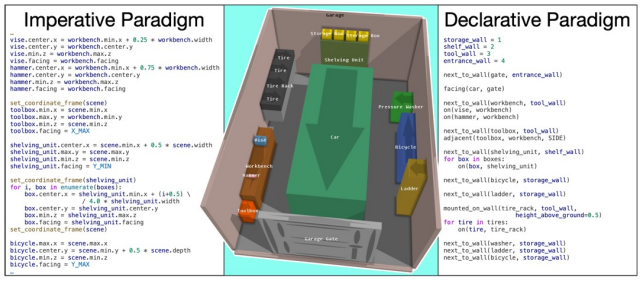


Figure 5: *Example of imperative vs declarative programs. Declarative programs often require an optimization module to preserve constraints. Figure reproduced from Gumin et al. [GHY*25].*

Scene graphs. Scene graph-based representations (Fig. 4 (b)) encode indoor scenes as graphs capturing objects and their relationships. The structure can take various forms, including parse trees [PZR20], directed graphs [LZWT20; WLW*19; LM24; YCC*24], or hierarchies [QZH*18; LPX*19]. Objects are represented as nodes, and edges capture spatial or semantic relationships such as support or adjacency. In hierarchical or tree-based variants, groups of objects are recursively nested to reflect multi-level spatial organization. This explicit modeling of relationships—whether flat or hierarchical—enables complex spatial reasoning and supports structured scene manipulation. However, scene graph representations require careful design of the structure and relation types, introduce computational challenges to determine the relationship between objects, and often face challenges with collision management due to implicit geometric constraints.

Structured language and programs. Structured language representations (Fig. 4 (c)) describe scenes using text or programs, specifying objects and their attributes in a human-readable way. Approaches vary from language encoded data-structures (e.g., CSS-like descriptors listing object types and placements) [FZF*23] to declarative programs that provide constraints of how the objects should be placed relative to each other [SLG*25] to imperative programs that explicitly construct or manipulate scenes step-by-step [TPW*25b; PTW*25] (Fig. 5). The choice of language ranges from custom domain-specific languages (DSLs) tailored for scene description to widely used scripting interfaces such as the Blender Python API [HIJ*24] or general Python-based libraries. This allows leveraging the expressive power and compositionality of programming languages for scene synthesis. While such representations benefit from intuitive control and leverage LLMs for open-vocabulary or procedural generation, they heavily depend on parsing and execution accuracy, and often lack explicit geometric grounding, which can lead to unrealistic scene layouts.

Images and videos. Scenes can also be represented as images (Fig. 4 (d)), with different choices for the viewpoint and encoding (Fig. 6). Common viewpoints include top-down (or bird's-eye view) which captures well the dominant objects in a room, perspective which offers a more natural view commonly found in photographs, and more recently isometric and panoramic views which offer a wider field of view. Top-down views can take the form of single-channel semantic maps [SFH*25], RGB images [ZYM*20], or multi-channel maps where each channel encodes a different attribute such as object category, orientation, or size [WSCR18; RWL19]. This format provides explicit pixel-level spatial information in the horizontal plane and allows flexible encoding of scene attributes. However, top-down images lack vertical detail, can miss small objects due to resolution limits, and often require post-processing to recover object instances. Perspective [WQL*24] and isometric [DYY*25; GCL*25] views provide richer visual detail but demand more complex reconstruction pipelines. Recent works extend these approaches by generating short video clips of scenes [HZB*25], leveraging video generation models to capture richer priors about object layouts. Overall, image-based scene representations align naturally with image generation models such as GANs and diffusion models, while video-based representations further benefit from recent progress in generative video modeling.

3D representation. *Monolithic 3D representations* (Fig. 4 (e)) model the entire scene in a single unstructured representation, without decomposing into individual objects or parts. Common approaches include implicit neural functions [LDL*23], dense feature volumes [JSM*20], and neural fields [MST*21], each encoding spatial occupancy [LLJ*24; LRY*24], geometry [MLND25], or appearance [ESL*25] in a unified latent space. These methods capture both global spatial structure and local geometric detail in

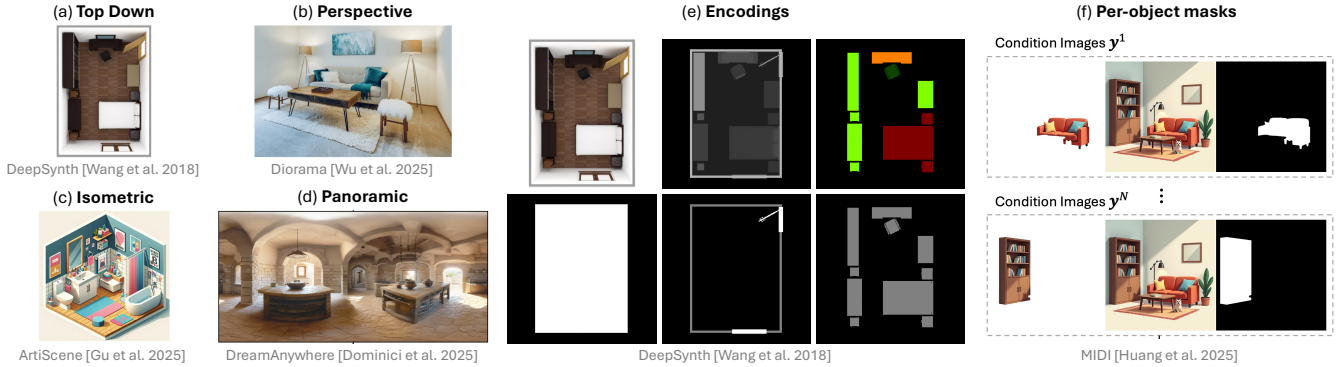


Figure 6: *Different ways to use images to represent the scene including: (a) top-down, (b) perspective, (c) isometric, and (d) panoramic. Each option can encode different information (e), or have separate masked images for each object (f). Images contain rich information about object appearance and relationship to other objects. However, images are a partial specification of the scene (e.g., RGB images do not specify object instances, and a single perspective image can only capture part of the scene). Image representations are often augmented with additional information and metadata to form hybrid representations [WLW*19; ZYM*20]. Overall, images can serve as a source of both input conditioning and knowledge.*

one representation. This is a straightforward way to support generation of high-fidelity, coherent scenes, but also results in high memory and computational requirements. Moreover, the lack of explicit object decomposition limits controllability for editing or semantic manipulation tasks. Some methods include semantic features through *semantic-aware 3D representations* such as semantic voxel grids [YGZT21] or 3D heatmaps encoding functionality [FSL*15]. Depending on the 3D representation used, there may be high computational or memory costs. For instance, dense voxel grids can encode detailed 3D information, allowing for direct operations such as voxel overlap for collisions but requiring tradeoffs between resolution and compute.

3.3 Knowledge

In creating real-world scenes, interior designers incorporate principles about the arrangements of furniture or objects based on practical considerations, geometrical principles, social norms, and cultural practices. For instance, most scenes involve symmetries in arrangements and place objects in alignment with other objects or architecture, such as a bed parallel to the wall. To acquire this knowledge, research in scene generation typically follows one of three approaches (see Fig. 7): 1) procedural generation of scenes based on heuristics (encoding human knowledge of where objects go into procedural rules) [DVH*22; RMK*24]; 2) 3D scene generation by learning priors from data [FRS*12; WSCR18; PKS*21; TNM*24]; and 3) leveraging LLMs to extract world knowledge [FWLS24; YSW*24; AGH*24]. Recently, there has been increasing focus on the third line of work (using LLMs for world knowledge) in order to handle more diverse types of scenes, as it is challenging to scale heuristics (limited by human coding effort) and learning priors from 3D data (limited by available data). For instance, most prior work on 3D scene generation is trained with a single, fairly limited dataset: 3D-FRONT [FCG*21], which has sparsely populated bedrooms, living rooms, and dining rooms. Enticingly, the large amount of common sense and world knowledge encoded in

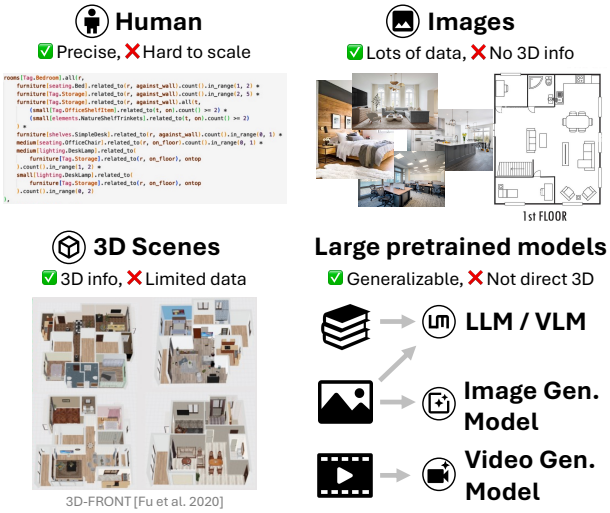


Figure 7: *Sources of* knowledge *(Sec. 3.3): human provided, images, 3D scenes, or large models pretrained on datasets of text / images / videos. Human-provided constraints work well for small numbers of scene and object types, but are hard to scale up. Depending on the source, the knowledge is high-level semantic priors (e.g., from language) or explicit 3D priors on object placement (e.g., from 3D scenes). While there is an abundance of text and image data, there is much less publicly available 3D scene data.*

LLMs would allow for handling arbitrary scene types with arbitrary types of objects populating the scene.

Human-authored heuristics. Procedural generation methods often rely on humans to encode knowledge about design principles, such as from interior design and intuitive human preferences,

into explicit procedures or algorithms for generating scenes or assets [ZLJ*21; DVH*22; RMK*24; MET]. While these methods benefit from interpretability and more guaranteed enforcement of human design principles, they are hard to generalize due to the manual work required to incorporate new knowledge, and limited by the need to balance different design principles.

2D images. Some methods leverage 2D image data to learn priors for 3D generation, especially due to the limited quantity of 3D scene data and larger quantity of images of indoor scenes. [PJBM22; YDH*25; YDH*24; WQL*24; LLL*25; NLL*25; FWLS24; DYY*25]. While 2D images lack depth information for accurate 3D localization, they naturally capture the distribution of real scenes and implicit scene design knowledge. However, 2D views capture a limited and biased view of the scene, suffering from occlusions, a restricted field-of-view, and other limitations that make it challenging to translate from 2D to 3D.

3D scenes. Methods trained on 3D scene datasets attempt to leverage existing 3D scene data to extract design principles [PKS*21; TNM*24; YJZH24; ZÖW*23; LM24]. These scene datasets are typically authored by humans with relevant expertise [KMH*17; GSA*20; FCG*21; KMJ*24]. Compared to 2D images, major advantages of this type of 3D scene data include its compositional nature and more explicitly representation of the 3D space and relationships between objects. However, 3D scene datasets are limited in size as they require significant effort to create, subsequently resulting in limited diversity and complexity in existing datasets. Furthermore, differences in the conventions and geometry of 3D assets as well as intellectual property issues can make it challenging to easily compose new scenes based on existing datasets, despite their prevalent use across various applications.

LLMs and VLMs. LLM- and VLM-based approaches extract design principles encoded in text and visual data in order to apply that knowledge to scene generation [FZF*23; FWLS24; YSW*24; ÇHS*24; SLG*25; HIJ*24; PTW*25]. Explicit sources of knowledge include descriptions of scenes as well as descriptions of design principles found in books or other documentation sources. In addition, VLMs can leverage 2D image data of scenes to learn from the visual representations of said scenes and correspond them more directly to textual descriptions. Thus, pretrained LLMs and VLMs lend themselves to serving as knowledge bases, augmented further by an ability to retrieve information from the web and other data sources. While benefiting from the vast amount of documentation of design principles in text and image data, VLMs have difficulty translating textual representations of this knowledge to 3D space to produce consistent and plausible scenes [TPW*25a].

3.4 Layout Generation

Scene generation models leverage the knowledge base and input conditioning to predict 1) what objects should be present in the scene; and 2) how should the objects be arranged (i.e., the scene layout). The layout should satisfy both the explicit relative spatial constraints between objects in the input as well as implicit human design principles. To accommodate these, it is useful to break down arrangement of the objects into several phases: a) a high-level specification of the semantic relationship between objects; b) a coarse layout of the objects (or subset of the objects); and c) exact placement of the objects based on object geometry and ensuring physical plausibility (i.e., no collisions, proper support). Depending on the input specification and scene generation method, some of these phases are potentially skipped. For instance, deep-learning based models trained on 3D data typically go directly from the input conditions to output layout specification without necessarily explicitly modeling the three phases. In this section we will focus on phase a) and b) and defer discussion of c) to Sec. 3.5.

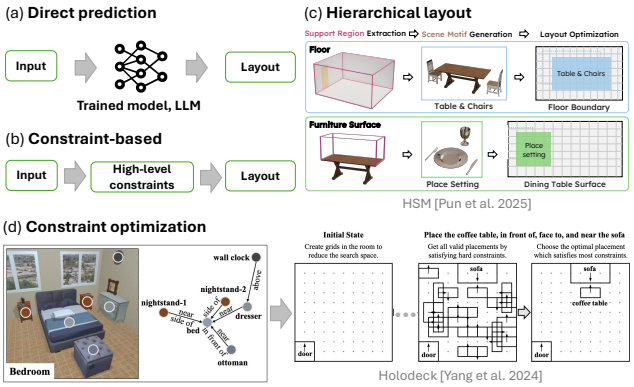


Figure 8: *During* layout *(Sec. 3.4), object categories (or embeddings) are sampled and a coarse layout placement is determined.*

In Fig. 8, we show broad strategies for how layout is achieved: direct prediction, layout via semantic constraints, and hierarchical layout. In direct prediction, a model generates the layout directly from the input as is common for many deep learning models trained on 3D data. Some works such as LayoutGPT [FZF*23] prompt an LLM to generate the layout directly. An alternative approach is to first obtain a set of constraints that specifies the relationships between objects, and then either give the constraints to a learned model to generate the coarse layout, or to solve a constraint optimization problem. It is also possible to perform the layout problem in a hierarchical manner, where objects are arranged at different scales (small objects arranged on a table, furniture arranged in a room, etc.) [SLL*25; PTW*25]. Here, we discuss in more detail how different approaches tackle layout generation.

Human-in-the-loop. Some methods rely on humans-in-the-loop to layout the scene, with some asking users to specify all of the required objects [MSL*11; ZLJ*21], and others coordinating with human artists to spatially arrange the objects [MSL*11; HKAG23] using a graphical interface. This primarily allows methods to automate and improve other aspects of scene generation and most directly incorporates human design principles into generating good scenes, at the cost of manual effort to generate complete scenes.

Procedural rules. To intuitively apply interior design principles for placing objects, some methods construct a set of rules for step-by-step generation of scenes [DVH*22; RMK*24; MET]. These rules can be represented algorithmically as in a decision-based flow [DVH*22], or as compact procedural programs based on domain-specific languages [RMK*24; MET]. Furthermore, these

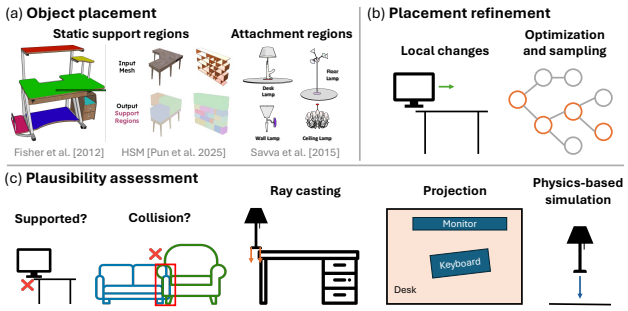


Figure 9: *During* object placement and refinement *(Sec. 3.5), the geometry of objects is taken into account to determine physically plausible placements (properly supported and collision free). To do this, it is necessary to (a) identify support and attachment regions for objects; (b) perform optimization and sampling to ensure plausible placements; and (c) assess physical plausibility by checking whether objects are properly supported, and detecting and scoring collisions and inter-penetrations. Strategies for plausibility assessment include: ray casting, image-based projections and reasoning over images, and physics-based simulation.*

methods often involve stochasticity in design. These methods are interpretable due to their explicit encoding of design principles, but they are hard to generalize either due to the inflexibility of extending complex rule-based flows or due to the manual work required to craft additional procedural routines.

Constraints. Rather than constructing procedural algorithms, other methods convert human design principles into a set of mathematical constraints or density functions which can subsequently be solved to generate object arrangements. The constraints can be represented as a mixture of density functions [MSL*11; JLS12], or in a hierarchical scene graph format with relative relationships [YYT*11; CSM14b; QZH*18; ZLJ*21]. To sample scene layouts under the specified constraints, most methods use a Markov Chain Monte Carlo (MCMC) sampler [MSL*11; YYT*11; YYW*12; QZH*18; ZLJ*21], and Gibbs sampling where direct sampling is difficult [JLS12]. This approach allows direct determination of whether a sampled solution satisfies all of the constraints and thus design principles and scene description. However, building the set of constraints to define a scene cannot easily generalize to new types of constraints without significant manual effort. Further, solving a set of constraints can be complex for challenging scene descriptions.

Statistical learning. Statistical models learn from example scenes to capture which objects tend to co-occur and how they are typically arranged. Some methods represent this knowledge with probabilistic structures such as Bayesian networks and Gaussian mixture models [FRS*12], And-Or graphs with Gibbs distributions [DS14; QZH*18], or scene graphs with priors on object occurrence, support relations, and relative positions [CSM14b; CMS*15]. Others model co-occurrence using factor graphs combined with k-means-based arrangement models [KLTZ16]. Sampling strategies such as Metropolis-Hastings, simulated annealing, Gibbs sampling, or local search are then used to generate layouts consistent with these learned priors, sometimes in conjunction with explicit constraints. While these approaches can reproduce common co-occurrence patterns and coarse spatial statistics, they often struggle to enforce exact counts, fine-grained relationships, or adapt to novel constraints, since they inherit strong biases from the particular statistics they encode. At the same time, they alleviate the need for manually handcrafting rules or constraints by directly extracting priors from data.

Deep learning. Learned methods include models that have been specifically trained on 3D scene datasets to sample object types and placements. Autoregressive methods rely on iteratively generating and placing each object in the scene, including VAEs [LPX*19] and transformers [PKS*21]. Diffusion-based methods, on the other hand, represent the scene layout in matrix form, similar to an image, and leverage a discrete iterative denoising process to map from sampled noise to an internal scene representation of the object classes, sizes, and poses [ZÖW*23; TNM*24; LM24; YJZH24]. While these methods learn directly from 3D scene datasets, they are limited by the relatively low volume and diversity of 3D data, often being limited to only a few common room types.

LLMs and VLMs. LLM- and VLM-based models either leverage encoded knowledge or retrieval to propose object classes and layouts in either a textual or visual representation. LLMs are typically used to generate lists of objects to place in the scene and propose placements of objects either in a structured representation [FWLS24; YSW*24; ÇHS*24; PTW*25] or code [FZF*23; SLG*25; HIJ*24; PTW*25]. VLMs can be used to generate image representations of the scene, which can subsequently be translated to 3D [WQL*24]. Since LLMs are exposed to significant amounts of world knowledge, they can easily generate lists of objects that are semantically relevant for a particular room type or description. However, language models often lack precision in matching specific input conditions, such as honoring counts of objects, and fail to translate scene descriptions into a valid layout due to a lack of robust 3D spatial understanding.

3.5 Object Placement and Refinement

Object placement and refinement aims to address geometric inconsistencies that arise after layout generation, especially with 3D meshes. Many methods, particularly deep learning-based, produce layouts with mesh collisions, floating objects, and objects placed out of bounds, mainly because they do not account for physical constraints. These conflicts compromise the physical plausibility essential for downstream applications. To ensure the plausible placement of objects, there are two common strategies: 1) iterative placement of single objects into a scene while (mostly) ensuring physical plausibility; and 2) optimization and sampling of object placements for plausibility. In both cases, it is also necessary to consider techniques for assessing the plausibility of a placement. This section discusses strategies for object placement and refinement of object placements (Fig. 9).

Object placement. While long a subproblem in compositional scene generation, the placement of objects has recently been studied on its own [HBT*25; AAW*25]. For the placement of single objects, it is important to be able to identify regions or surfaces on which other objects can be placed, and to be able to

determine whether the object is placed free of collision and with proper support. ***Support region detection*** identifies valid surfaces (e.g., floor, walls, ceilings, furniture) where objects can be stably placed. Most works treat this as an annotation or pre-processing step [FRS*12; CSM14b; PTW*25] on individual objects. Several datasets [KMH*17; YRY*23] provide *receptacle* annotations of where other objects can be placed. Other works algorithmically determine the support surface, either by taking the top-most flat surface [DVH*22] or using more advanced techniques. Fisher et al. [FRS*12] segment meshes into planar support surfaces using a region growing algorithm. HSM [PTW*25] extracts planar regions by clustering mesh faces, fitting planes and classify them as horizontal or vertical with clearance constraints. ***Attachment regions.*** Parallel to identifying support surfaces, it is also important to identify the side or point at which the object should be attached. It is common to annotate what side of an object attaches to other objects [DVH*22] or for that to be learned from data [CSM14b].

Placement refinement. Optimization and sampling can be used to refine the placement of objects by incorporating support and collision terms in the objective function [FRS*12]. ***Collision detection & resolution.*** Heuristics can also be used to resolve collisions. RoboGen [WXC*24] resolves collisions by detecting intersecting objects and pushing their centers of mass apart along collision normals. HSM [PTW*25] applies adaptive horizontal displacements based on penetration depth, and preserves each object's support regions (e.g., floor bound, attached on floor, mounted on ceiling).

Placement plausibility assessment. There are several different strategies for identifying proper support and collision detection. ***Ray casting*** ensures objects are properly supported from their corresponding surfaces such as floors, walls, ceilings, or objects surfaces. Holodeck [YSW*24] uses ray casting to initialize small objects on larger surfaces to ensure they are properly supported. HSM [PTW*25] refines object placement by ray casting according to each object's support region (e.g., floor, wall, ceiling, furniture). ***Projection*** methods [WZC*24] project (or render) the profile of the object to be placed onto the support surface. Then image-based techniques can be used to determine whether there are collisions and sufficient support. Instead of casting many different rays, this approach leverages faster raster-based rendering, which can work better than ray-casting for more complex shapes. ***Physics-based simulation*** is an alternative to projection or ray-tracing. For instance, objects can be placed on a support surface by dropping an object from above, and letting the physics simulation to ensure support contact is tight and free of collisions [SF95; XSF02]. Another approach is to integrate a physics-aware optimization module to enforce stability constraints, which ensures objects rest stably on their parent surfaces to simulate gravity. One example is Scenethesis [LLL*25] which integrates such constraints with signed distance fields. Physics-based simulation can be expensive, and often requires the use of shape approximations such as convex decompositions [LA07; MLP16; WLLS22] for faster computation. Improperly initialized placements can cause simulations to explode (due to initial intersections), and require additional information for each asset (object masses, which objects are fixed). An advantage of physics-based simulation is that the resulting scenes are readier for use in simulation frameworks for embodied AI and robotics.

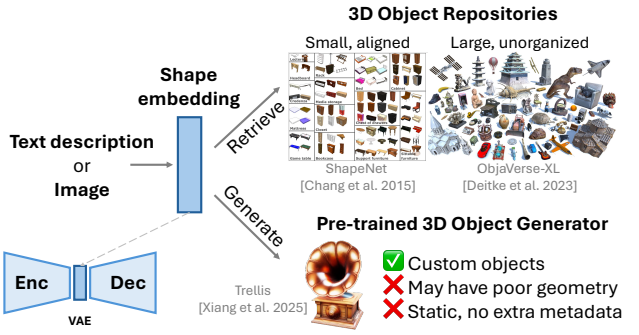


Figure 10: *Objects can be retrieved from a repository of 3D shapes, or generated using an object shape generator (Sec. 3.6). Both retrieval and generation can be done with shape embeddings extracted from a text description (can be as simple as object category) or from an image. It is common to train a specialized VAE for shape embeddings and use that for retrieval or generation.*

3.6 Object Shapes

To turn a scene layout into a complete 3D scene, the layout must be populated with concrete object shapes. Broadly, there are two ways to obtain these shapes: retrieving them from an existing dataset or generating them from scratch (see Fig. 10).

Retrieval. Retrieval-based approaches assume access to a database of 3D object shapes, from which suitable candidates can be selected to populate a scene [FRS*12; WSCR18; PKS*21]. Several cues can guide retrieval, including object category, object size, and similarity in a learned embedding space. Category-based retrieval requires both the generated layout and the database shapes to be annotated with object categories, which is common in layout generation pipelines. Size-based retrieval, typically using bounding box dimensions, is often combined with category information to further narrow down candidate shapes. More recently, embedding-based retrieval has become popular, using approaches such as CLIP [RKH*21] and DuoDuoCLIP [LZC25] that align text, images, and sometimes shapes in a shared space. In this setting, embeddings of database shapes (from text descriptions, rendered images, or directly from 3D data) are precomputed, and suitable objects are retrieved by comparing these to the embedding of a query, which can be text or image-based. For example, Diorama [WIR*25] adopts a hierarchical retrieval strategy: first narrowing candidates with text embeddings, then ranking them using image similarity. The retrieved shapes are then placed into the scene according to the layout specification.

Generation. Object shapes can also be generated directly [HGA*25]. This eliminates the need for retrieval from a database of object shapes, and allows for more flexibility in generating unique shapes that may not exist in a database. Similar to retrieval, object shapes are typically generated based on object category, text description, image, or embedding. There are also procedural generation approaches that represent objects as visual programs and generate the object shapes by executing the program [RMK*24]. These approaches typically only work

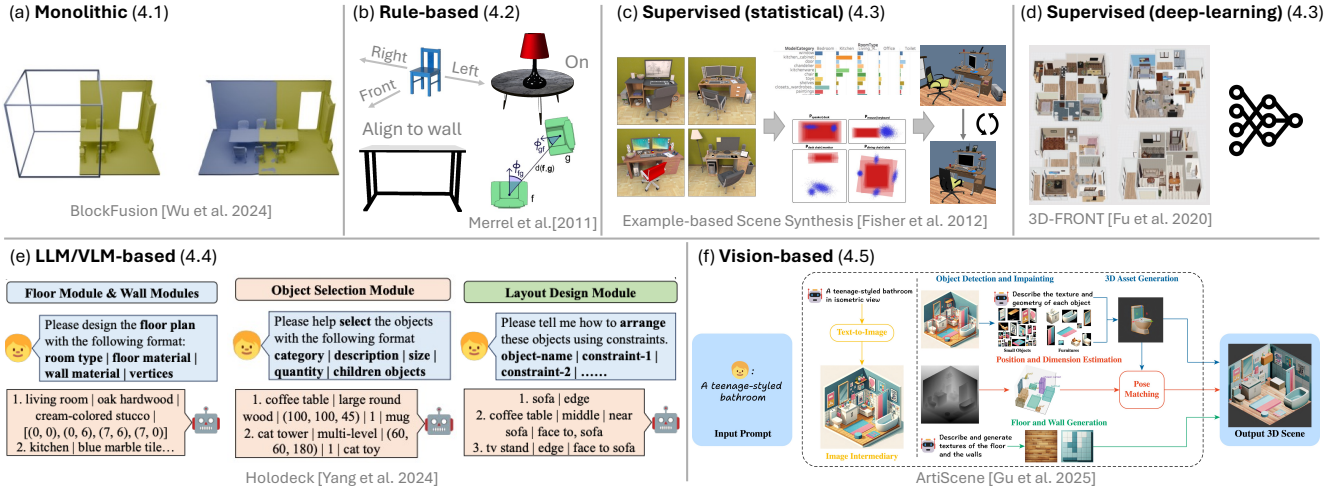


Figure 11: *Comparison of method families for scene generation. Monolithic methods generate 3D scenes without consideration to individual objects, while our survey focuses on compositional 3D generation. For completeness, we discuss how monolithic methods (a, Sec. 4.1) typically do 2D or 3D infilling to generate the scene geometry. Compositional scene generation started with rule-based methods (b, Sec. 4.2) that use heuristics to map between common spatial relations and geometric placements, and rules for encoding common object arrangements. Supervised methods (Sec. 4.3) learned distributions of objects and their placements based on 3D scene datasets. Early data-driven methods used statistical models such as Bayes Nets (c, Sec. 4.3.1) to learn object co-occurrences and distributions over relative placements, and then iteratively sample and optimize object placements. With the development of larger datasets and neural network architectures, deep learning models (d, Sec. 4.3.2) used larger amounts of training data to learn the object arrangement distributions and sample from them. More recent methods leverage pretrained models for open-vocabulary scene generation, by prompting LLMs (e, Sec. 4.4) or using image generation models to obtain an initial image and then decomposing the image into individual objects (f, Sec. 4.5).*

for objects composed of simple geometric primitives, such as boxes and cylinders, and struggle to model complex shapes. For a broader review of text-to-shape methods, see Lee et al. [LSC24].

Challenges. Both retrieval and generation face important challenges. Retrieval relies on databases that are diverse, well-annotated, and normalized in terms of scale and orientation. Otherwise, additional modules are required to standardize the retrieved shapes. Generation methods must ensure that outputs are realistic and artifact-free, which remains difficult for complex geometries. Beyond obtaining objects, laying out into a scene poses its own difficulties. While placing objects on the floor is straightforward, supporting relations (e.g., books on shelves) require reasoning about 3D geometry and identifying spatial regions on supporting objects that can accommodate placement of the supported object.

3.7 Architecture

Apart from the objects, the architecture of a scene, including elements such as walls, floors, ceilings, doors, and windows, plays a crucial role in defining its overall structure and functionality. The floor establishes the valid region for object placement, while openings such as doors and windows influence the layout by constraining movement and sight lines. Some methods [SCH*16; ZÖW*23] omit architectural components entirely, focusing solely on arranging objects within an empty volume.

Fixed template. A simple strategy is to assume a fixed architectural layout for all scenes. For example, DiffuScene [TNM*24] uses a 6m $\times$ 6m square room as a constant boundary, applied post-hoc to all generated scenes. Another option is to sample from a set of predefined candidates [ZLJ*21]. This approach simplifies the generation process but limits the diversity of the resulting scenes.

User input. Many approaches instead rely on architectural information provided as part of the user input [WYN21; PKS*21; PGMW23; YHT*23; MSDO24]. This information can take several forms, such as room dimensions (e.g., length and width) for rectangular spaces [FZF*23], or a 2D floor plan that specifies the room boundaries and shape [PKS*21; YHT*23; GSM*23]. Elements like doors and windows may also be included to constrain object placement [SGC25]. This way, the architectural layout serves as a strong geometric prior that anchors the generation process.

Automatic generation. Architectural elements can also be generated automatically. The simplest approach is to create a rectangular room after the objects have been placed, enclosing them within a fitted floor plan [LM24]. More sophisticated methods generate detailed architectural structures, including walls, doors, and windows, either using procedural generation techniques [DVH*22], or by prompting LLMs to design plausible layouts [YSW*24; FWLS24; PTW*25]. There is also a line of work that focuses specifically on room layout generation [NHC*21; SHF23], producing multi-room floor plans that can then be used for scene generation.

4 Scene Generation Methods

Scene generation methods can be broadly categorized based on how they handle the generation process, the type of knowledge they leverage, and the representation of the generated scenes (Fig. 11). In this section, we review the most common approaches to compositional 3D scene generation. We start with monolithic methods in Sec. 4.1, which generate a scene as a single fused geometry. While these methods are by nature not compositional and not the focus of this survey, we include them here for completeness and to highlight their limitations. Next, we discuss compositional approaches, starting with rule-based methods in Sec. 4.2, supervised methods in Sec. 4.3, and LLM/VLM-based methods in Sec. 4.4. Finally, we discuss vision-based methods in Sec. 4.5.

4.1 Monolithic

Monolithic approaches output a fused geometry representation that does not explicitly separate object instances. We categorize these methods into two groups: methods that rely on image-based priors and methods that use explicit 3D supervision or 3D priors.

Image-based outpainting. These methods typically follow two stages: text-to-image diffusion and monocular depth estimation. First, an initial image is either generated from text or taken as input. A depth estimation model then predicts depth for the initial image. Using color from the image and the estimated depth, the scene is lifted to 3D. The camera then shifts to a new viewpoint, from which the lifted 3D scene is rendered. To complete missing regions, the text-to-image model is used to inpaint between existing regions and outpaint new areas. Depth is estimated for new areas, with various strategies to align depth to existing geometry for better consistency. Finally, the newly predicted color and depth information are then fused into the 3D representation. This process repeats to continuously expand the 3D scene. Earlier works used explicit 3D representation like meshes [FAKD24; HCO*23] or point clouds [YDH*24] for more straightforward back-projection. Recent methods use gaussian splatting-based optimization [CLN*23; STLR25; YDH*25] for higher quality. A major limitation is the lack of consistency in long-range generation as well as distortion or artifacts in the geometry due to inaccuracies in depth prediction. Recent work like WonderFree [NLL*25] attempts to remedy this by fine-tuning video diffusion models on a combined dataset of real and synthetic 3D datasets. However, modality gaps between 2D and 3D still exist. Moreover, the process is computationally expensive, requiring repeated runs of text-to-image and depth estimation models, as well as potential optimization of the 3D representation.

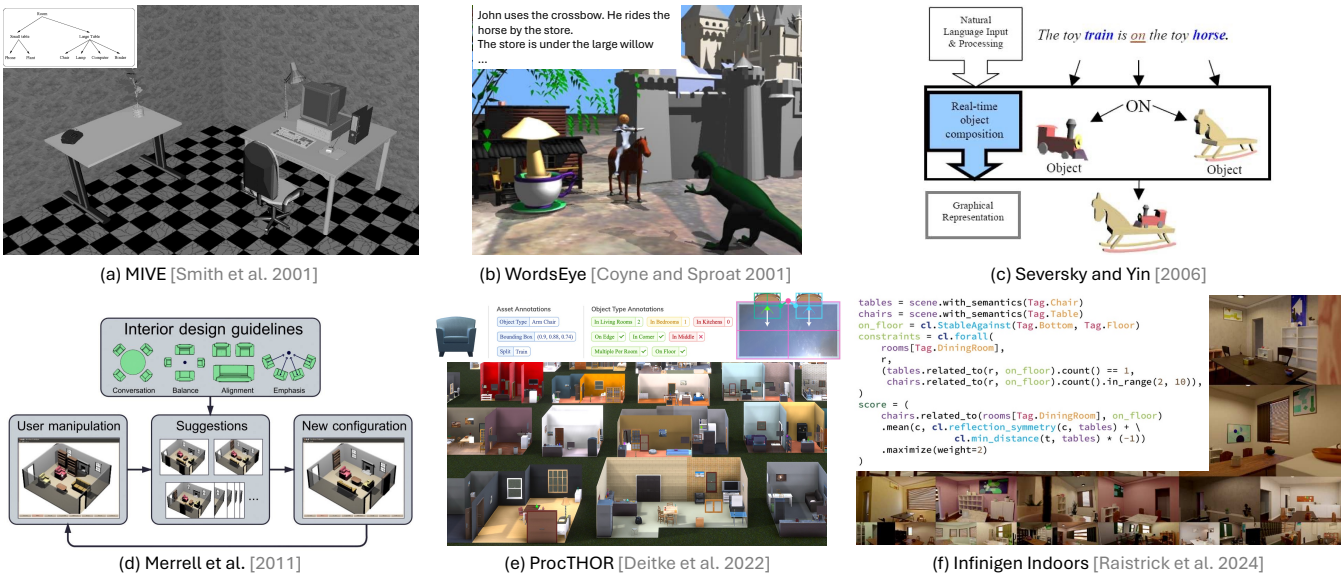
3D-based outpainting. In contrast to image-based approaches, 3D-based outpainting methods operate directly on a 3D representation. They typically encode spatial chunks into a compact latent space, enabling diffusion models to generate new regions conditioned on existing geometry and progressively extend large-scale scenes. Most methods adopt a latent diffusion architecture (LDM) [RBL*22], first learning to compress smaller scene chunks in representations such as triplanes [LLJ*24; WLY*24], sparse voxels [RHZ*24; LRY*24], or feature grids [MLND25] into low-dimensional representations. Generation proceeds by resampling and denoising neighboring chunks, analogous to 2D repainting [LDR*22], allowing continuous expansion of the 3D scene while maintaining coherence between regions. This framework has been applied to driving scenes [LLJ*24; LRY*24; LLL*24], city-scale environments [XCHL24; LLM*23; XCHL25; HJCZ25], outdoor scenes [LHC25; ESL*25; ZZG*25], and indoor scenes [WLY*24; MLND25]. Each method adapts the general framework to the characteristics of the target scene type, using strategies such as repaint-like [LDR*22] resampling, top-down or overlap conditioning. Of particular interest to us are methods that generate indoor scenes: BlockFusion [WLY*24] and LT3SD [MLND25]. The former uses a triplane and the latter a feature grid representation for LDM compression. Both methods use repaint-like resampling for scene outpainting with BlockFusion training a conditional model based on semantic layout and some outdoor scenes, and LT3SD incorporating MultiDiffusion [BYLD23] for their multi-level generation.

Limitations and challenges. A major limitation is the *monolithic* output representation of the 3D scene, where geometry, semantics, and appearance are fused together. This makes it difficult to use such scenes in interactive settings where objects need to be manipulated or articulated. While segmentation could be applied post-hoc to obtain object instances, how to do this effectively to obtain high quality output remains an active research problem. InfiniCube [LRY*24] demonstrates this by segmenting out cars and reconstructing dynamic driving scenes. However, the geometry is quite coarse and low quality, and is not suitable for indoor scenarios where fine detail is more important.

4.2 Rule-based

To generate compositional 3D scenes—where each object is an entity that can be arranged independently—it is crucial to capture the principles that govern plausible arrangements. Just as humans do not place objects in everyday environments at random but instead follow implicit rules and conventions, rule-based approaches to scene generation seek to formalize these principles into explicit rules or constraints. One of the earliest such systems is the Design Problem Solver [Pfe75], which frames object arrangement as a constraint satisfaction problem with 2D polygons specifying object footprints and access regions that must remain unobstructed. A depth-first search algorithm is employed to find an arrangement that satisfies the constraints. Subsequent work extended this formulation from 2D to full 3D scene generation, enabling richer spatial reasoning and object interactions. We categorize rule-based approaches into four main groups, based on how their rules are specified and applied: interactive interfaces, handcrafted rules, interior design guidelines, and robotics applications. See Fig. 12 for representatives and Tab. 1 for a categorized summary.

Interactive interfaces. Early work on 3D scene generation focused on developing interactive interfaces that assist users in manually placing objects within a 3D environment. Object Associations [BS95] assigns rules to objects (e.g., “obey pseudo-gravity”, “fit against walls”) and uses them to automatically adjust placements as users manipulate them in the scene. Shinya and Forgue [SF95] extends this idea with collision detection and simple physics simulation, while Smith et al. [SSS*01] represent constraints in a



(a) MIVE [Smith et al. 2001] (b) WordsEye [Coyne and Sproat 2001] (c) Seversky and Yin [2006]

(d) Merrell et al. [2011] (e) ProcTHOR [Deitke et al. 2022] (f) Infinigen Indoors [Raistrick et al. 2024]

Figure 12: *Representative rule-based methods for 3D scene generation (Sec. 4.2). Initial scene generation systems were rule-based and allowed for interaction using mouse clicks (a,d), or language (b). Objects were placed based on constraints mapping to geometric rules (a-c) with some systems incorporating interior design rules (d). More recently, rule-based systems have been used to generate scenes for embodied AI and vision learning (e,f). Figures adapted from those in the original papers.*

directed acyclic graph to preserve spatial relationships between objects during manipulation. These systems constrain the degrees of freedom in object placement, making it easier for users to construct plausible scenes, but they still rely heavily on manual input. As such, they function as design aids rather than fully automated scene generators, laying the groundwork for later methods that seek to automate the placement process.

Handcrafted rules. The first attempts to automate scene generation define constraints for object placement manually, encoding spatial relationships that guide how objects can be arranged. Put [CW96], FurnIt [Kjø00], and WordsEye [CS01] are early examples. Put provides a text-based interactive interface with a predefined set of spatial operators for placing objects in a 3D scene. For instance, it can move an object to a specified location given input such as "put *cup* on *table*". FurnIt populates scenes using handcrafted arrangement templates and resolves conflicts recursively. WordsEye generates 3D scenes from simple natural language descriptions by semantically analyzing the input text. Objects in its database are annotated with spatial tags that denote semantically meaningful regions. The analyzed text is then used to retrieve objects and arrange them in the scene using the annotations. Subsequent work follows a similar vein, with variations in the types of constraints and optimization strategies [XSF02; CCD03; SLLG03; SY06; TBSD09; YYW*12; WLD*18; ZLL*23]. These methods depend on users to devise the rules, yet specifying what makes a scene desirable is not a straightforward task. Without expertise, it is difficult to articulate higher-level principles of scene quality needed to generate coherent and functional scenes.

Interior design guidelines. An alternative to handcrafted rules is to derive constraints from established interior design guidelines. Merrell et al. [MSL*11] first followed this approach, incorporating functional criteria (e.g., circulation space in a room, seating proximity for conversation) and visual criteria (e.g., visual balance, furniture alignment, presence of a focal point) into an optimization framework. Optimization used a parallelized Metropolis-Hastings algorithm to search for layouts that satisfy the criteria. Kán and Kaufmann [KK17; KK18] similarly adapt interior design guidelines as constraints, using a genetic algorithm and greedy cost minimization. By grounding constraints on professional design principles, these methods cover a broader range of considerations and better align with human preferences.

Robotics applications. Rule-based methods have recently seen a resurgence in the robotics community, where there is a need to generate diverse and plausible 3D scenes at scale for training and evaluating embodied AI agents. LUMINOUS [ZLJ*21] takes as input a user-defined room specification with required objects and relationships, and augments it with heuristic rules (e.g., "a bed is usually placed against a wall"). It first samples an architectural layout from predefined templates, and then sequentially places objects in the order of large furniture, small items, and decorations. ProcTHOR [DVH*22] adopts a procedural generation approach, using handcrafted rules and object annotations to iteratively create rooms and populate them with objects, demonstrating the ability to generate 10,000 diverse indoor scenes that yield state-of-the-art performance on various embodied AI tasks. Wang et al. [WZJ*23] propose a rearrangement framework that balances human preference rules with robot preference rules, optimizing layouts for human–robot co-activity using adaptive simulated annealing. Most recently, Infinigen Indoor [RMK*24] provides a unified

	Input	Representation	Knowledge	Layout	Placement	Object	
Object Associations [BS95]							
Put [CW96]							
CAPS [XSF02]							
Merrell et al. [MSL*11]							
MIVE [SSS*01]							
Shinya and Forgue [SF95]							
Seversky and Yin [SY06]							
WordsEye [CS01]							CT
ProcTHOR [DVH*22]							RS
DPS [Pfe75]							
Sanchez et al. [SLLG03]							
LARJ-MCMC [YYW*12]							
Infinigen Indoors [RMK*24]							
Weiss et al. [WLD*18]							
Wang et al. [WZJ*23]							
Zhao et al. [ZHG*16]							GT

Complete Scene · Partial Scene · Image · Human · Physics Simulation · GT Geometry
Floor Shape · Room Type · Scene Graph · Constraint · Ray Casting · RS Random Sample
Free-form Text · Structured Text · Sequence · Human · Generation
Functional Zone · User Interaction · Structured Language · Rule · Retrieval
Object Information · 3D Scene · 3D Scene · Collision Solving · CT Category

Table 1: *Table comparing rule-based methods for compositional 3D scene generation (see Sec. 4.2).* ***Input*** *denotes the type of conditioning the method takes.* ***Representation*** *denotes how the methods structure the scene information.* ***Knowledge*** *denotes the data source leveraged for learning scene priors, while* ***Layout*** *indicates how the layout is specified.* ***Placement*** *indicates the approach used for placing objects in the scene, and* ***Object*** *indicates how the object shapes are obtained. See legend at bottom for details. Empty cells indicate components that are not applicable or insufficiently described in the original paper. All methods rely on human knowledge as the source of rules or constraints that govern layout generation, and many are implemented as interactive systems accepting user input.*

procedural generation system that produces both scene layouts and the objects within them, with a constraint specification API that allows users to define desirable constraints as Python expressions.

Limitations and challenges. Rule-based methods provide fine-grained control over generated scenes and allow explicit specification of user preferences, but they face several limitations. First, rules are difficult to specify, since object placements are often context-dependent and hard to formalize. Second, the reliance on handcrafted or predefined rules makes it difficult to capture the diversity of real-world environments. Third, although some approaches incorporate higher-level design guidelines, most rules remain focused on low-level spatial relationships, leaving global scene coherence largely unaddressed. Finally, optimization with large numbers of objects and complex constraints can be computationally intensive, limiting scalability.

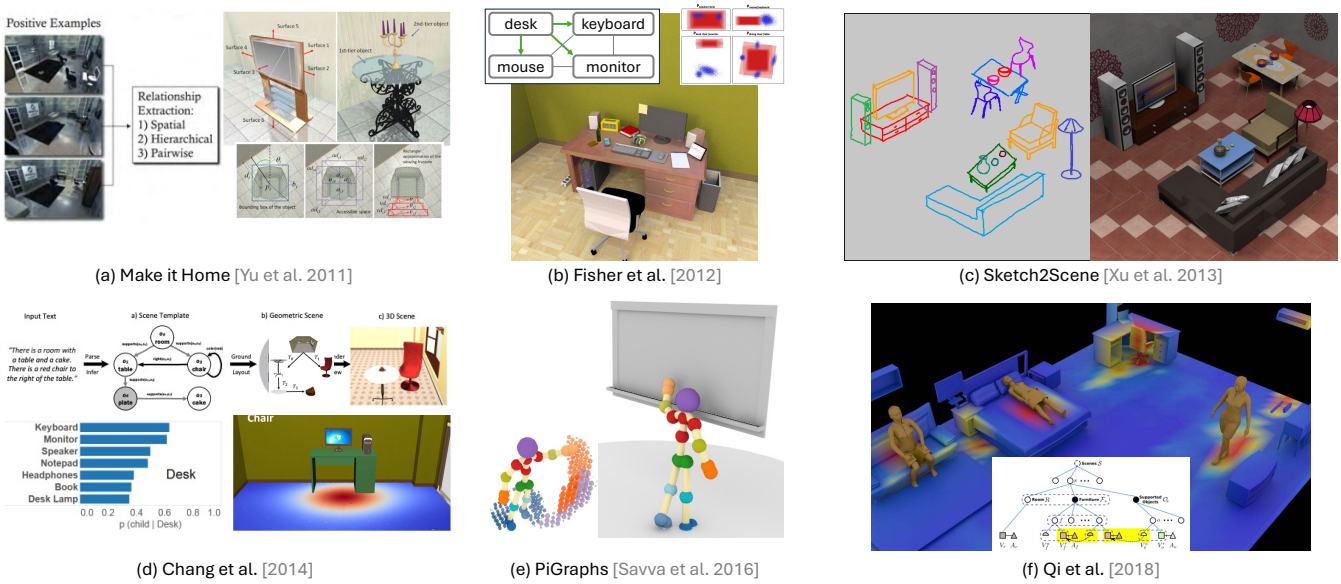
4.3 Supervised Methods

Supervised methods alleviate the need for manually designing rules or constraints by directly learning how to generate scenes from data. Given a collection of example scenes, ranging from small curated sets to large-scale datasets, these methods aim to synthesize new scenes that are similar to the examples while adhering to input conditions, such as room type or object relationships. The development of supervised approaches for indoor scene synthesis progressed in two main stages. Early work used statistical learning to capture patterns of object co-occurrence and spatial relationships from the examples, which can then be sampled to generate new scenes. With the advent of large-scale datasets such as SUNCG [SYZ*17] and 3D-FRONT [FCG*21], deep learning methods became feasible, training neural networks to learn complex distributions of objects and their placements. Recent methods span a wide range of paradigms, including variational autoencoders (VAEs), generative adversarial networks (GANs), autoregressive models, and diffusion models, each offering distinct ways of learning distributions and synthesizing scenes. The following subsections review these two categories of supervised methods.

4.3.1 Statistical Learning Methods

Statistical learning methods model indoor scenes by learning priors from existing layouts and reusing them for synthesis. These priors describe how objects tend to co-occur, how they are spatially arranged relative to one another and to the room, and how they interact with human activities. Such methods rely on explicit probabilistic or energy-based formulations, often coupled with sampling or optimization algorithms such as Gibbs sampling, Metropolis-Hastings, simulated annealing, or hill climbing. Representative examples are summarized in Fig. 13 and Tab. 2.

Probabilistic graphical and statistical models. Many approaches formulate scene generation as inference in probabilistic graphical models or statistical mixtures, learning distributions from data to capture object dependencies and spatial arrangements. Bayesian networks can be used to capture dependencies between object categories and their spatial attributes [FRS*12; CSM14b; FSL*15;



(a) Make it Home [Yu et al. 2011] (b) Fisher et al. [2012] (c) Sketch2Scene [Xu et al. 2013]

(d) Chang et al. [2014] (e) PiGraphs [Savva et al. 2016] (f) Qi et al. [2018]

Figure 13: *Representative statistical methods for 3D scene generation (Sec. 4.3.1). With a small number of example scenes, statistical-based scene generation was able to learn co-occurrence and support patterns of objects, and probability distributions for object placements. These methods typically work by starting with an initial placement of objects and then iteratively sampling local updates to arrive at the final optimized scene. We show representative methods that can generate scenes similar to a given set of inputs (a,b), or by conditioning on sketch (c) or text (d). As indoor scenes are designed for human use, some works took an human-centric approach (e,f).*

CMS*15; YYT15]. Gaussian mixture models (GMMs) are also widely used to represent spatial relations such as distances, orientations, and heights among objects [FRS*12; XCF*13; SCA17; MPF*18]. To incorporate human context, PiGraphs [SCH*16] models human poses as Gaussian and von Mises distributions and links them to object placements, while Jiang et al. [JLS12] incorporate a Dirichlet process mixture model (DPMM) to jointly model human poses and object arrangements. Henderson et al. [HSF17] further show that DPMM can be used to automatically discover recurring spatial motifs from data, which can then serve as compositional units during synthesis. These probabilistic formulations make it possible to explicitly model object arrangements with uncertainty, encode interpretable dependencies, and sample new plausible configurations by drawing from the learned distributions.

Energy-based and optimization models. Another formulation relies on energy-based models, where an explicit energy function is defined over scene configurations and the goal is to find low-energy (high-probability) arrangements. The energy typically combines learned statistical priors with geometric and physical constraints such as accessibility, visibility, or object intersection avoidance. Make it Home [YYT*11] is a classic example that learns distributions of object distances and orientations relative to walls and optimizes layouts with simulated annealing using Metropolis-Hastings proposals. Many other methods follow a similar approach [FRS*12; HPSC16; LZM19; ZHL*19; ZZX*20; ZZX*21], optimizing placements with learned statistical priors under various constraints using techniques such as Metropolis-Hastings, simulated annealing, hill climbing, or position-based dynamics.

Grammar- and relation-based compositional models. Indoor scenes can also be modeled using grammar- or relation-based models, which represent scenes as hierarchical or relational structures. Scene graphs encode spatial and semantic dependencies as potentials on nodes and edges, supporting inference through MCMC or local search. Fisher et al. [FSH11] design graph kernels to measure scene similarity and guide object suggestion. And-Or graphs [DS14; QZH*18; JQZ*18] provide a hierarchical representation that encodes both alternative and mandatory relationships among objects, while factor-graph formulations [KLTZ16] capture multi-way dependencies among object groups. These representations emphasize relational reasoning and enable generation preserving the compositional hierarchy of scenes.

Interactive and data-driven systems. Statistical priors have also been embedded into interactive scene design tools that assist users during layout creation or editing. SceneSeer [CESM17], SceneSuggest [SCA17], and MageAdd [ZLH*21] integrate learned co-occurrence and spatial priors to propose plausible objects in response to language or context-based queries. Ma et al. [MPF*18] extend this idea by mapping linguistic relations to spatial constraints using Gaussian mixture models, enabling text-guided augmentation of 3D layouts. Other systems, such as Clutterpalette [YYT15] and SceneDirector [ZTL*23], use Bayesian or clustering models to provide interactive object suggestion and group editing. These systems show how statistical priors can support practical, data-driven design assistance.

Limitations and challenges. Statistical learning methods were the first to leverage data for indoor scene synthesis, significantly reduc-

	Input	Representation	Knowledge	Layout	Placement	Object
Fu et al. [FCW*17]				HS		CT, DM
Make it Home [YYT*11]				SA		
Qi et al. [QZH*18]				SA, GB		
Clutterpalette [YYT15]				BN		CT, RS
Fisher et al. [FSL*15]				BN, GD		CT, RS
Ma et al. [MPF*18]				GMM, HC		
Sketch2Scene [XCF*13]				GMM, BS, GO		SK
Jiang et al. [JLS12]				DPMM, GS, VM		
Henderson et al. [HSF17]				DPMM, SP, MC, RJS		CT
Kermani et al. [KLTZ16]				KM, MCMC		
Chang et al. [CSM14a]				SP, HC		CT
Chang et al. [CSM14b]				SP, HC		CT
SceneGen [KPZ*20]				SP, KDE		
PiGraphs [SCH*16]				SP, GS, VM, HG		CT
Fisher et al. [FRS*12]				SP, HC, BN, GMM, KDE		CT, RS

Complete Scene, Floor Shape, Free-form Text, Object Information, Partial Scene, Room Type, Scan, Sketch, Structured Text, User Interaction, Window/Door Location, 3D Scene, Image, Scene Graph, Sequence, 2D Image, 3D Scene, Human, Constraint, Statistics, BN Bayesian Network, BS Beam Search, DPMM Dirichlet Process Mixture Model, GS Gaussian Distribution, GMM Gaussian Mixture Model, GB Gibbs Distribution, GO Gradient Optimization, GD Greedy Algorithm, HS Heuristic Search, HC Hill Climbing, HG Histogram, KM K-Means Clustering, KDE Kernel Density Estimation, MC Markov Chain, MCMC Markov Chain Monte Carlo, RJS Rejection Sampling, SP Sampling, SA Simulated Annealing, VM Von Mises Distribution, Collision Solving, Image Correspondence, Support Region Extraction, Retrieval, CT Category, DM Dimensions, RS Random Sample, SK Sketch

Table 2: *Table comparing statistical learning methods for compositional 3D scene generation (Sec. 4.3.1). These methods learn statistical priors from 3D scenes or 2D images, and generate new scenes by sampling from these priors. Human-authored constraints are often incorporated to enforce explicit conditions such as preventing object collisions. Empty cells indicate components that are not applicable or insufficiently described in the original paper.*

ing the manual effort required to design rules and constraints. However, to model the complex distributions of real-world scenes, these methods often rely on strong assumptions and simplifications, such as limiting relationships to pairwise statistics or predefined spatial relations. The distributions they used to model object placements are often simple, such as Gaussian or von Mises distributions, or kernel density estimates, which may not capture the full complexity of real-world scenes. It is also challenging to learn all necessary priors from limited data, especially for rare objects or arrangements. For this reason, many methods add hand-crafted constraints to ensure physical plausibility, such as avoiding object intersection or ensuring accessibility. Exact sampling from these distributions can be intractable, requiring approximate methods such as MCMC or greedy search, which in addition requires careful design of score functions to balance between different objectives.

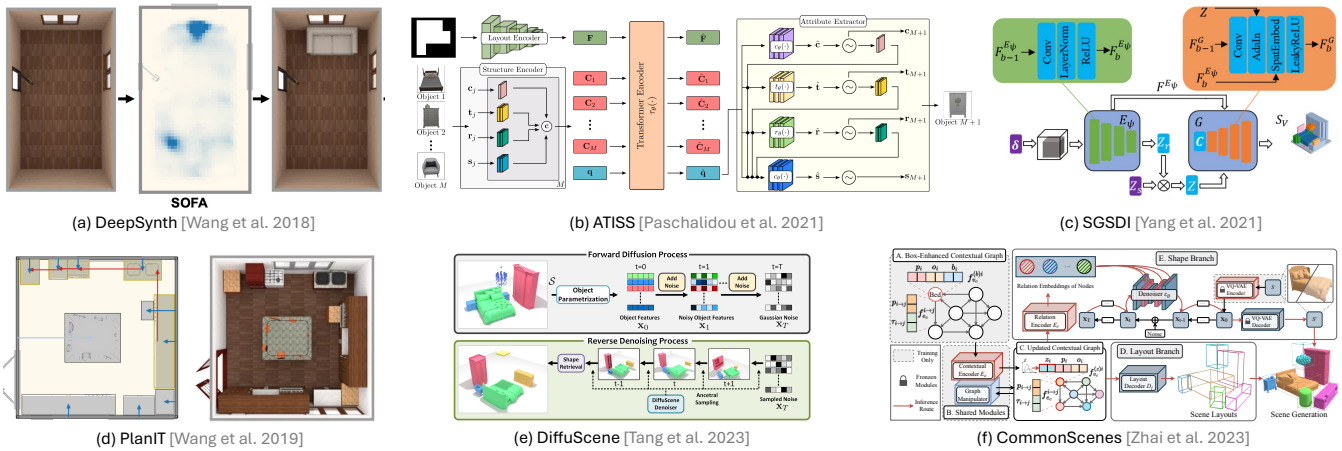
4.3.2 Deep Learning Methods

With the advent of deep learning, researchers investigated how generative models with neural networks can be used for 3D indoor scene generation. One of the first works, Wang et al. [WSCR18] used an autoregressive approach that iteratively fed a top-down image-based representation of the scene to a CNN to predict what is the next object, and where it should be placed. Following this work, researchers investigated approaches with different types of generative models and representations (see Fig. 14 and Tab. 3). Here, we group the works by the generative model used to learn the distribution of objects and their placements.

Autoregressive methods generate indoor scenes object by object, with each placement conditioned on the current state of the scene, making the assumption that the joint distribution of all objects can be factorized into a product of conditional distributions. Early approaches used image-based representations ([WSCR18; RWL19]), predicting each object's attribute channels (location, category, size, etc.) sequentially for one object at a time in a top-down layout. Later, PlanIT [WLW*19] explored graph representations, adding new nodes and edges one by one based on the current graph state. Recent work [WYN21; PKS*21; LGWM22] has shifted to using transformers architectures. Transformers predict the next token given previous ones. A natural representation that fits this is an object sequence, with each object represented as a vector of attributes (e.g., category, size, position, orientation). ATISS [PKS*21] introduced a training strategy that randomly permutes the order of objects during training, alleviating the issue of order-dependence in autoregressive models. By flattening hierarchical structures into sequences, autoregressive methods can also handle generation of tree-structured scenes, a notable example being Forest2Seq [SZZ*24].

VAE-based methods encode scenes as scene graphs [LZWT20], trees [PZR20], or hierarchical structures [LPX*19], where objects and their relationships are represented explicitly. The variational autoencoder framework learns a compact latent space from these structured scene representations and decodes samples from this space back into complete layouts. This approach enables interpolation and random sampling of plausible scenes, while the structured encoding helps maintain spatial relationships and hierarchy. CommonScenes [ZÖW*23] extends this idea by incorporating shape features into the latent space and generating both the layout and geometry simultaneously using two decoders. However, modeling dependencies between objects remains challenging, and fine-grained spatial details may be lost in the compressed latent representation.

GAN-based methods for indoor scene synthesis train a generator and a discriminator to produce realistic scenes. HybridGAN [ZYM*20] uses hybrid representations, combining object sequences with rendered images to leverage both structural and visual information during training. SGSDI [YGZT21] adopts voxel-based representations, generating scenes as volumetric grids to capture



(a) DeepSynth [Wang et al. 2018] (b) ATISS [Paschalidou et al. 2021] (c) SGSDI [Yang et al. 2021]

(d) PlanIT [Wang et al. 2019] (e) DiffuScene [Tang et al. 2023] (f) CommonScenes [Zhai et al. 2023]

Figure 14: *Representative deep learning-based methods for 3D scene generation (Sec. 4.3.2). Wang et al. [WSCR18] (a) was the first to use deep learning for scene generation, adopting a autoregressive approach that predicted objects and their placement on top-down images. ATISS (b) used transformers and modeled scene generation as an autoregressive token-based sequence generation task, and SGSDI (c) used GANs to predict semantic voxels. More recently, graph-based representations are popular, with both autoregressive (d) and diffusion-based (f) variants, and diffusion models operating on unordered sets of objects have also been explored (e). Recent methods also learn to encode object shapes using a VAE, which can then use embeddings to retrieve (e) or generate the corresponding object shape (f). Figures reproduced from original papers.*

	Input	Representation	Knowledge	Layout	Placement	Object
Ritchie et al. [RWL19]				AR		CT, DM
Pose2Room [NDHN22]				AR		
ATISS [PKS*21]				AR		CT, DM
MIME [YHT*23]				AR		CT, DM
LayoutEnhancer [LGWM22]				AR		CT, DM
SceneFormer [WYN21]				AR		CT, DM
Wang et al. [WSCR18]				AR, CNN		CT
GRAINS [LPX*19]				AR, VAE		CT, DM
PlanIT [WLW*19]				AR, VAE, CNN, BT		CT, DM
3D-SLN [LZWT20]				VAE		CT, DM
SceneHGN [GSM*23]				VAE		
Yang et al. [YGZT21]				GAN		CT, DM
Zhang et al. [ZYM*20]				GAN		CT, DM, SC
LEGO-Net [WDP*23]				DF		
DiffuScene [TNM*24]				DF		CT, DM, SC
RelTriple [SYW*25]				DF		CT, DM
PhyScene [YJZH24]				DF		CT, DM, SC
Fang et al. [FYMH25]				DF		CT, DM, SC
SemLayoutDiff [SGC25]				DF		CT, DM
CommonScenes [ZÖW*23]				DF, VAE		
InstructScene [LM24]				DF, VAE		CT, SC
Pfaff et al. [PDZ*25]				DF, RL, MCTS		
Haisor [SYM*24]				RL, MCTS		

Complete Scene, Floor Shape, Free-form Text, Human Motion, Partial Scene, Room Type, Scene Graph, Separate Model, Window/Door Location, 3D Scene, Image, Scene Graph, Sequence, 3D Scene, Human, LLM/VLM, Learned, AR Autoregressive, BT Backtracking, CNN Convolutional Neural Network, DF Diffusion, GAN Generative Adversarial Network, MCTS Monte Carlo Tree Search, RL Reinforcement Learning, VAE Variational Autoencoder, Collision Solving, Physics Simulation, Generation, Retrieval, CT Category, DM Dimensions, SC Shape Code

Table 3: *Comparison of deep learning-based methods (Sec. 4.3.2). All methods use a collection of 3D scenes as their knowledge base. Sequence and scene graph representations are most common due to their compatibility with deep learning architectures. Empty cells indicate components that are not applicable or insufficiently described in the original paper.*

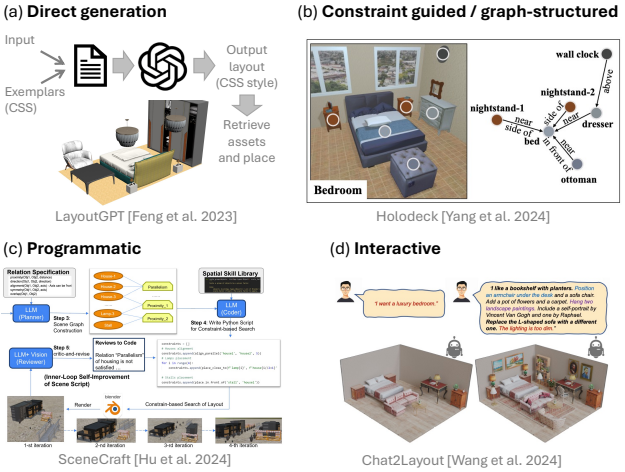


Figure 15: *Representative LLM/VLM-based methods for 3D scene generation (Sec. 4.4). We show examples that use direct generation (a), constraint-guided and graph-structured intermediates (b), programs (c), and interactive systems (d).*

3D geometry directly. While GANs can produce visually realistic layouts, training remains challenging and often struggles with diversity and stability, especially for complex 3D scenes.

Diffusion-based methods use an iterative denoising process, gradually transforming random noise into structured scene layouts. DiffuScene [TNM*24] first applied this approach by encoding scenes as object sets, each object represented by a vector of attributes, and modeling the scene as a tensor of fixed dimensions, which fits well with the diffusion process, establishing a foundation for subsequent diffusion-based methods [HAD*24; YJZH24; SYW*25]. InstructScene [LM24] adopted diffusion with a graph representation, enabling controllable generation from text instructions. Other recent work leveraged diffusion models for generating top-down semantic images of scenes [SFH*25].

Limitations and challenges. Deep learning methods alleviate the need for manual rule design and statistical assumptions, learning complex distributions directly from data. In theory, they can model arbitrarily complex relationships and distributions given sufficient data and model capacity. However, they often require large-scale datasets to train effectively — a significant challenge in this domain, as 3D indoor scene datasets are still relatively small compared to other domains such as 2D image datasets, and their quality and diversity can vary widely. Some methods [NDHN23] attempt to learn from 2D images, but this introduces other challenges such as accurate object detection and pose estimation to extract 3D information from 2D images. A lot of works train a separate model for each room type, which further limits the amount of data available for training each model. The cost of training can also be high. Lastly, similar to statistical methods, the scenes they can generate are limited by what is seen in the training data.

4.4 LLM/VLM-based Scene Generation

Here, we discuss methods that use large language models (LLMs) and vision language models (VLMs) for compositional 3D scene generation. These approaches interpret natural language instructions to produce structured outputs, such as object layouts with fine-grained instance information. This focus on a structured representation extends early rule-based methods. Unlike traditional data-driven approaches that rely on limited datasets and costly 3D scene collection, LLMs and VLMs can parse complex user instructions. They infer 3D layouts from language using common-sense spatial reasoning about how objects typically relate in indoor spaces. We categorize these methods based on the role the LLMs/VLMs play: as generators that directly output layouts, as planners that produce intermediate structures such as constraints, graphs or programs for downstream solvers, or as modules for human-in-the-loop systems. LLMs and VLMs are typically used as modules in larger systems, with guardrails to post-process and iteratively check and correct their outputs. When multiple LLM/VLM modules are used, often the system is called a multi-agent one. See Fig. 15 for representative methods and Tab. 4 for categorization.

Direct generation. This line of work [FZF*23; ÖTKG24; RLX*25; YLZ*24; BA25] explored whether semantic knowledge and common sense reasoning from LLMs is sufficient to produce complete layouts without additional processing. LayoutGPT [FZF*23] initiated this direction by generating CSS-like coordinates to position objects for indoor scenes with in-context examples from 3D-FRONT [FCG*21]. However, scenes often suffered from overlapping objects or furniture placed outside room boundaries. These methods motivated future approaches that split responsibilities across different modules, where LLM/VLM modules provide semantic structure and specialized solvers or optimizers enforce geometric validity.

Constraint-guided systems such as FlairGPT [LDM25] and Fireplace [HBT*25] use an LLM to generate spatial rules or relationships. Holodeck [YSW*24] leverages LLM common-sense knowledge to decompose input descriptions into object lists and infer missing elements based on typical room functionality. Then, it produces spatial relation constraints between objects, which are optimized with a DFS solver into layouts. FirePlace [HBT*25] combines multimodal reasoning with fine-grained 3D geometric constraints to propose precise object placements given a scene. Most systems still rely on some form of constraints, since these are necessary to produce plausible layouts.

Graph structured systems use LLMs to create a scene graph or hierarchal tree as a blueprint to guide layout generation [FWLS24; BBC*25; PTW*25; ÇHS*24; LTT25a; WMV*25; LLL*25; SLL*25; DQM25; LZZ*]. AnyHome [FWLS24] converts natural language descriptions into structured scene graphs capturing object semantics and spatial relationships. Similarly, I-Design [ÇHS*24] uses a multi-agent system to generate a relative scene graph, which a backtracking algorithm uses to solve for the layout.

Programmatic systems build on constraint-based and graph-based approaches [TPW*25b; NLNN25; PTW*25; SLG*25; AGH*24; WXC*24; HIJ*24; XZL*24; LTT25b; LLJ25; ZWWZ25; KPKK25; WZC*24]. Program-based methods offer a more

	Input	Representation	Knowledge	Layout	Placement	Object	
Scenethesis [LLL*25]							CT, SC, DM
AnyHome [FWLS24]							SC, DC
Holodeck [YSW*24]							CT, SC, DM
Aguina-Kang et al. [AGH*24]							CT, SC, DM
SceneCraft [HIJ*24]							SC, DC
FlairGPT [LDM25]							CT, DC
SceneLCM [LLJ25]							
LayoutGPT [FZF*23]							CT, DM
OptiScene [YLD*25]							CT, DM
Chat2Layout [WZC*24]							
RoboGen [WXC*24]							SC, DC, VLM
EchoLadder [HTL*25]							CT, SC, DC
SceneMotifCoder [TPW*25b]							CT, DM
ReSpace [BA25]							DM, DC, EB
FreeScene [BBC*25]							CT, SC
I-Design [ÇHS*24]							SC, DM, DC
FirePlace [HBT*25]							
LayoutVLM [SLG*25]							CT, SC
HSM [PTW*25]							CT, SC, DM, DC

Complete Scene · Floor Shape · Free-form Text · Image · Object Information · Partial Scene · Sketch · User Interaction · Window/Door Location · Image · Scene Graph · Sequence · Structured Language · 3D Scene · Human · Image Generation Model · LLM/VLM · Constraint · Learned · LLM/VLM · Reconstruction · Collision Solving · Image Correspondence · Physics Simulation · Projection · Ray Casting · Score Distillation Sampling · Support Region Extraction · Generation · Retrieval · CT Category · DC Description · DM Dimensions · EB Embeddings · SC Shape Code · VLM Vision-Language Model

Table 4: *Table comparing LLM/VLM-based methods for compositional 3D scene generation (Sec. 4.4). All methods rely on an LLM or VLM as a central component and knowledge base, accepting free-form text as input and typically representing scenes using structured language or scene graphs. Empty cells indicate components that are not applicable or insufficiently described in the original paper.*

systematic alternative by leveraging the code generation ability of LLMs. These methods synthesize domain-specific programs that encode spatial relationships or object arrangements as executable code. In this sense, they resemble constraint-guided systems, which rely on constraint solvers or optimizers to generate layouts. LayoutVLM [SLG*25] combines a VLM and differentiable optimization to produce 3D object layouts from textual descriptions. SceneMotifCoder [TPW*25b] introduced a framework that uses examples to learn object arrangements (motifs) as reusable visual programs using a LLM. Recently, HSM [PTW*25] extended this idea to the scene level using a VLM and extracted support regions from meshes to ensure physically plausible object placements.

Human-in-the-loop systems focus on an iterative and collaborative process, rather than a single, one-shot input [FWLS24; WZC*24; HTL*25; LTT25a; YLZ*24; KPKK25; BA25; ZWWZ25; LZZ*; YLD*25]. These methods usually prioritize user control by allowing dynamic scene modification and refinement of scenes through natural language. For example, Chat2Layout [WZC*24] uses multiple LLM agents with visual context and instructions to allow the user to create, rearrange, and refine an indoor scene.

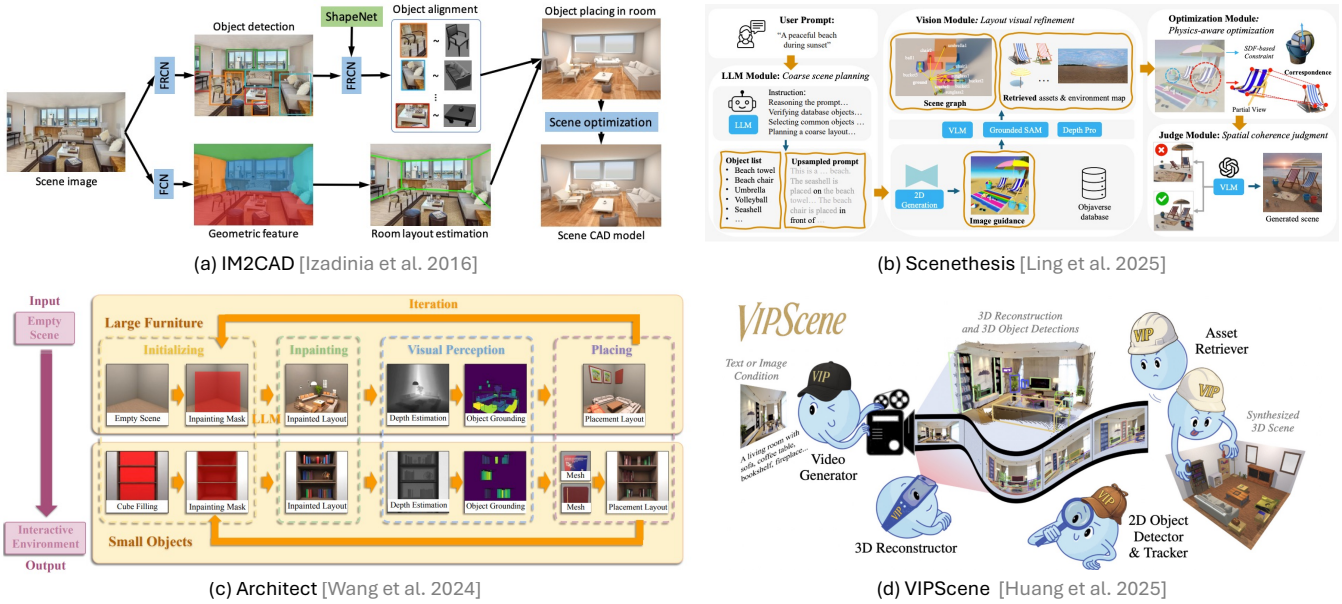
Limitations and challenges. Although LLM/VLM-based methods allow open vocabulary generation, these systems typically retrieve object meshes from an asset database. This significantly limits the diversity and controllability of the assets. SceneLCM [LLJ25] has attempted end-to-end generation of both layouts and meshes, but the resulting meshes exhibit noticeable artifacts or collisions between objects. The open-ended nature of LLM/VLM-based methods is also a challenge in evaluation, as there are no standardized benchmarks, leaving the definition of a "good" scene ambiguous. Another concern is speed, as iterative LLM calls lead to longer generation time than learning-based or rule-based methods.

4.5 Vision-based Scene Generation

Here, we discuss methods that leverage images and videos for compositional 3D scene generation (see Fig. 16 and Tab. 5). In this setting, the visual observation dictates the spatial arrangement of objects, and the focus shifts to recovering the underlying 3D scene—estimating object geometries, appearances, and placements consistent with the depicted view. The following subsections outline the typical pipeline of this approach, covering how visual inputs serve as layout cues, how objects are detected and reconstructed, how their appearances are modeled, and how the resulting objects are finally positioned to form a coherent 3D scene.

Visual as layout cue. Visuals provide information about the spatial arrangement of a 3D scene—what objects are present, where they are located, and how they appear. This stands in contrast to text-based descriptions, which are sparse and ambiguous. Visual inputs can take various forms, including single images, panoramic images, multi-view images, or videos. They may come from *user-provided* sources, where users supply photographs or videos of environments they wish to recreate in 3D, or be *model-generated* during the scene synthesis process. In the latter case, an auxiliary modality such as text is first used to condition an image or video generation model (e.g., SDXL [PEL*24], Cosmos [AAB*25]). In both cases, the visual cue serves as a pseudo-ground truth for the 3D scene to be reconstructed, defining a layout that guides subsequent geometry, appearance, and placement estimation.

Visual input understanding. The first step in using visual inputs



(a) IM2CAD [Izadinia et al. 2016]

(b) Scenethesis [Ling et al. 2025]

(c) Architect [Wang et al. 2024]

(d) VIPScene [Huang et al. 2025]

Figure 16: *Representative vision-based methods for 3D scene generation (Sec. 4.5). Visuals provide rich information about objects in the scene and how they are arranged. Early work (a) illustrated the key components that go into a vision-based scene generation system with models to detect and identify objects, estimate their poses, and placement of objects in the room and optimization of the object placements. More recent work leverage advances in LLMs and VLMs, enabling the generation of images from text (b) as well as the use of VLMs to judge the generated scene. It is also possible to iteratively populate a scene by using inpainting to imagine what other objects are in the scene (c), but the key steps of identifying objects in the image and placing them remain. More recent methods also consider generating a video from a text or image condition, and construct a scene based on the video (d). For vision-based scene generation, there is a choice between selecting / reconstructing objects that faithfully match the original or just taking the visual appearance as an inspiration (e.g., allowing for different looking objects that semantically match). Figures reproduced from original papers.*

	Input	Representation	Knowledge	Layout	Placement	Object	
VIPScene [HZB*25]							CT, DM, SC
Gen3DSR [DÖE25]							
DeepPriorAssembly [ZLH24]							
ACDC [DWJ*24]							EB, CT
Diorama [WIR*25]							EB, CT, DC
ArtiScene [GCL*25]							
Chabal et al. [CCPS25]							EB, CT
FastCAD [LJD*24]							EB
Mask2CAD [KALD20]							EB
LiteReality [HWZ*25]							EB, CT, DM
CAST [YZY*25]							
Vid2CAD [MPNF22]							EB
SceneGen [MWZX25]							
Architect [WQL*24]							DM, DC, SC

Floor Shape, Free-form Text, Image, Partial Scene, Scan, Video

Image, Point Cloud, Scene Graph, Sequence, Video, 2D Image

3D Object, 3D Scene, Human, Image Generation Model, LLM/VLM, Video

Video Generation Model, Constraint, Image Generation Model, Learned, LLM/VLM, Reconstruction

Collision Solving, Physics Simulation, Ray Casting, Support Region Extraction, Generation, Retrieval

CT Category, DC Description, DM Dimensions, EB Embeddings, SC Shape Code

Table 5: *Comparison of vision-based compositional scene generation methods (see Sec. 4.5). These methods reconstruct scenes directly from visual information—such as images or videos provided as input, or outputs from image or video generation models—and employ predominantly visual representations (e.g., images, videos, or point clouds). Empty cells indicate components that are not applicable or insufficiently described in the original paper.*

for 3D scene generation is to detect and extract the constituent objects depicted in the visual. This involves identifying the objects present in the scene, their locations, and their extents. Many works build upon advances in 2D perception networks, using off-the-shelf or fine-tuned architectures for object detection and depth estimation [ISS17; NHG*20; ZCZ*21; LZC*22; ZCC*21]. The detected objects are then cropped from the visual to serve as input for subsequent stages of the pipeline. Earlier approaches typically classified detected objects into a fixed set of categories, limiting their ability to generalize beyond the set of categories. More recent works [ZLH24; DWJ*24; WIR*25] have expanded this step to an open-vocabulary setting by incorporating vision-language foundation models such as GPT-4o [HLG*24], enabling recognition of novel object categories beyond the training data.

Object-centric geometry modeling. With the objects detected, the next task is to model the geometry of each object. Early works on single-view scene reconstruction generate compositional 3D scenes from perspective or panoramic images by applying object-level reconstruction to each detected instance [NHG*20; ZCZ*21; LZC*22; CNJ*24; ZCC*21; DFB*24]. Due to the limited availability of real-world data, these methods are typically trained on synthetic 3D assets in an end-to-end manner, leading to limitations such as over-smoothed and incomplete meshes, loss of fine-grained scene structure, and real to synthetic domain gaps.

Another line of works [ISS17; KALD20; KALD21; GDN22; GRLD24; DWJ*24; WIR*25; HWZ*25; HZB*25] formulates geometry modeling as a retrieval problem, selecting suitable CAD assets from a curated 3D library based on visual similarity. Compared to reconstruction-based approaches, retrieval-based methods yield complete and compact 3D object representations that are interaction friendly and directly editable in modern graphics pipeline software. Early efforts typically require curated data of image-CAD pair annotations to train the model in an end-to-end manner, limiting generalization. Recently, several zero-shot methods [ZSS*25; DWJ*24; WIR*25] mitigate the problem of scene data scarcity by leveraging LLMs and VLMs. For example, ACDC [DWJ*24] ranks 3D object candidates using DinoV2 [ODM*23] features on multi-view renderings and asks GPT-4o to pick the top-K best matching ones, while Diorama [WIR*25] proposes a hierarchical retrieval strategy where both text and visual modalities are used for better retrieval results using DuoDuoCLIP [LZC25].

A key limitation of retrieval-based methods is their limited control over the geometry for the matched objects. Using a vision-conditioned generative model allows the object geometry to be generated, without requiring 3D training supervision. A representative line of work generates each object individually in a normalized space and aligned to the view space using the scene layout guides from the depth map [DÖE25; ZLH24; WQL*24; YZY*25; GCL*25; MWZX25]. In particular, these methods usually incorporate a diffusion-based image generation model for inpainting occluded parts of objects. Beyond off-the-shelf 3D object generative models, CAST [YZY*25] designs a latent diffusion-based generative model to produce high-fidelity object meshes conditioned on partial image segments and optional point clouds.

Appearance recovery. Most research has focused on geometric reconstruction, and few works address the complementary problem of modeling surface appearance. IM2CAD [ISS17] simply finds the median value of each color channel separately, and assigns the closest color appearing within the mask to the retrieved object. SSR [CNJ*24] reconstructs both shape and texture from single-view images by employing neural implicit functions and radiance fields. PSDR-Room [YLH*23] instead chooses either a homogeneous material or a procedural node graph from a database for each material part, and uses differentiable rendering to jointly optimize the materials and lighting. Following the trend of using generative models for geometry modeling, CAST [YZY*25] leverages a texture generation model to predict UV mappings for each object.

Object placement. After individual objects are reconstructed or retrieved, they must be positioned in the scene according to the spatial relationships depicted in the visual input. A common approach is to fit each object into its predicted 3D bounding box [ADD*19; NHG*20; ZCZ*21; LZC*22; ZCC*21; DFB*24] or to estimate its pose [KALD20; KALD21; GDN22; GRLD24]. These learning-based pose estimation methods are usually category-specific, limiting their generalization to unseen objects. To overcome such limitations, recent works have introduced more flexible and data-efficient strategies. ACDC [DWJ*24] queries VLMs about the orientation of objects using rendered views. DPA [ZLH24] proposes an iterative method to optimize the location, orientation and size for each 3D object by matching it with the estimated segmentation mask and the estimated depth. Diorama [WIR*25] instead builds open-world 2D-3D correspondence between observed objects and CAD renderings using foundation models (DINOv2 [ODM*23]). CAST [YZY*25] uses an alignment generative model that produces a transformed partial point cloud, aligning with the complete geometry implicitly represented in the latent space.

Limitations and challenges. A key challenge for vision-based scene generation lies in achieving robustness and consistency under unconstrained real-world settings. Most existing methods are designed for monocular inputs, leading to scale ambiguity and view inconsistency when applied to videos or sparse multi-view data. Videos bring more challenges such as objects moving in and out of view, occlusions, motion blur, and in the case of video generation models the need to select camera trajectories that provide sufficient scene coverage. Some approaches reformulate video-based reconstruction as a single-view or scan-based problem, but these conversions are typically inefficient and error-prone. At the same time, current 3D localization and pose estimation techniques, whether based on object detection or CAD-conditioned fitting, have limited generalization to novel object categories not seen during training. Both issues stem from data scarcity and the difficulty of capturing diverse real-world configurations, demanding research into developing more generalizable models that can reconstruct view-consistent scenes from multi-view or video inputs, while remaining robust to occlusions, clutter, and noise through the combined use of real and synthetic data during training.

5 Evaluation

Robust evaluation of generated 3D scenes is crucial for advancing research in scene generation. Without consistent evaluation protocols, it is difficult to compare methods fairly, identify strengths and weaknesses, or understand where progress is being made. In this

Eval. Axis	Approach	Metric	Adoption
Realism (Sec. 5.2.1)	Comparison to reference scene	Match object category Match position Match orientation Match size	*SceneGraphNet [ZWK19], Fang et al. [FYMH25], *Jiang et al. [JLS12], SG-VAE [PZR20], SceneGen [KPZ*20], Fang et al. [FYMH25] *SG-VAE [PZR20], SceneGen [KPZ*20], Fang et al. [FYMH25] *SceneGraphNet [ZWK19]
	Distributional distance	Fréchet Inception Distance (FID) [HRU*17] CLIP-based FID (CLIP-FID) [KKA*23] Kernel Inception Distance (KID) [BSAG18]	*ATISS [PKS*21], DiffuScene [TNM*24], LayoutGPT [FZF*23], SemLayoutDiff [SGC25] *CLIP-Layout [LXJ*23], InstructScene [LM24], EchoScene [ZÖC*24], FreeScene [BBC*25] *DiffuScene [TNM*24], CommonScenes [ZÖW*23], InstructScene [LM24], MiDiffusion [HAD*24]
	Real vs fake	Classification Accuracy (SCA)	*Qi et al. [QZH*18], Ritchie et al. [RWL19] ATISS [PKS*21], PhyScene [YJZH24]
Fidelity (Sec. 5.2.2)	Human motion alignment	Collision with human mesh Ground free space obstruction Contact region alignment	*Ye et al. [YWL*22] *MIME [YHT*23], Hong et al. [HYHC24] *MIME [YHT*23], Hong et al. [HYHC24]
	Text-image alignment	CLIPScore [HHF*21] BLIPScore [LLSH23] VQAScore [LPL*24]	*AnyHome [FWLS24], Holodeck [YSW*24], HSM [PTW*25], VIPScene [HZB*25] *RoboGen [WXC*24], Architect [WQL*24], HSM [PTW*25], Scenethesis [LLL*25] *Architect [WQL*24], HSM [PTW*25], Scenethesis [LLL*25], VIPScene [HZB*25]
	Scene graph and relation-based	Scene graph consistency iRecall [LM24] SceneEval [TPW*25a]	*Luo et al. [LZWT20], Graph-to-3D [DMNT21], CommonScenes [ZÖW*23], EchoScene [ZÖC*24] *InstructScene [LM24], Fang et al. [FYMH25], FreeScene [BBC*25] *HSM [PTW*25]
Plausibility & Functionality (Sec. 5.2.3)	Low-level measures	Object collision rate Valid support Out-of-bound rate (OOB)	*RoomDesigner [ZZL*24], AnyHome [FWLS24], PhyScene [YJZH24], LayoutVLM [SLG*25] *Scenethesis [LLL*25] *LayoutGPT [FZF*23], AnyHome [FWLS24], I-Design [ÇHS*24], SemLayoutDiff [SGC25]
	Scene-level measures	Navigability Object accessibility Robotic task success rate	*HAISOR [SYM*24], PhyScene [YJZH24], Scenethesis [LLL*25], SemLayoutDiff [SGC25] *RoomCraft [ZWWZ25] *LUMINOUS [ZLJ*21], ProcTHOR [DVH*22], ClutterGen [JC24], HAISOR [SYM*24]
Aesthetics (Sec. 5.2.4)	Relative measure	CLIP-based relative score	*InstructScene [LM24]
	Predictors	LAION aesthetic predictor [SB22] ImageReward [XLW*23]	*Zhang et al. [ZHX*24] *HiScene [DYY*25]
System-level Properties (Sec. 5.2.5)	Efficiency	Generation time Resource usage Model size Convergence rate	Ritchie et al. [RWL19], SceneFormer [WYN21], ATISS [PKS*21], FuncScene [MWZZ24] Sanchez et al. [SLLG03], [SY06], [MSL*11], SceneFormer [WYN21] ATISS [PKS*21], Yao et al. [YCC*24], DeBaRa [MSDO24], CasaGPT [FZL*25] Weiss et al. [WLD*18]
	Diversity	Standard deviation of object pose and dimensions Distance to nearest neighbor	*Luo et al. [LZWT20], Graph-to-3D [DMNT21], Xu et al. [XHH*23], MiDiffusion [HAD*24] *Wang et al. [WSCR18], Ritchie et al. [RWL19], PlanIT [WLW*19], Yang et al. [YGZT21]
	Usability	NASA TLX [HS88] Task completion time	*Chat2Layout [WZC*24], EchoLadder [HTL*25] Smith et al. [SSS*01], SceneSuggest [SCA17], Zhang et al. [ZHL*19]

Table 6: *Summary of commonly used evaluation metrics, organized by evaluation axis and general approach. The * symbol denotes work that introduced or first adopted the metric for scene generation. Note that the list of works is not intended to be exhaustive.*

section, we outline key axes for evaluation of 3D scene generation (Sec. 5.1). We then review commonly used approaches and metrics along with their limitations (Sec. 5.2).

5.1 Axes of Evaluation

Evaluating the quality of generated 3D scenes requires first clarifying what makes a scene "good". While there is no universal definition, below we describe several desiderata (also see Fig. 17).

Physical plausibility. Objects in a scene should respect physical constraints: no interpenetration, and stable static support (e.g., not floating in mid-air).

Functionality. Objects positioned so they can be used for their intended purposes, with functional sides accessible and the layout supporting natural movement and interaction. Doors and windows are unobstructed.

Semantic coherence. Objects should appear in semantically meaningful relationships: chairs typically surround tables, and sofas often face televisions.

Aesthetics and style consistency. Harmonious object styles, materials, and color schemes contribute to overall quality, while poor aesthetics can make a scene feel messy and unappealing.

The above contribute to the perceived scene realism. In addition, scenes should faithfully reflect the user's input conditions.

Fidelity. The generated scene should closely match the user's specifications. This includes ensuring the presence of requested objects, their specified attributes (e.g., size, color), their spatial relationships with one another and with architectural elements, or higher-level constraints such as room type or style.

Beyond scene-level qualities, system-level properties also affect the practicality of scene generation methods.

Efficiency. The time taken to generate a scene and the computational resources required are important practical considerations.

Diversity. A system should be able to generate a wide variety of scenes while still adhering to the desired quality criteria.

Usability. What input modalities are supported (e.g., text, sketches, example images), how intuitive the interface is, and how much user effort is required to achieve desired results all impact practicality.

The above scene-level qualities and system-level properties provide a multifaceted basis for evaluating 3D scene generation methods. Next, we discuss commonly used evaluation approaches and metrics along with their limitations.

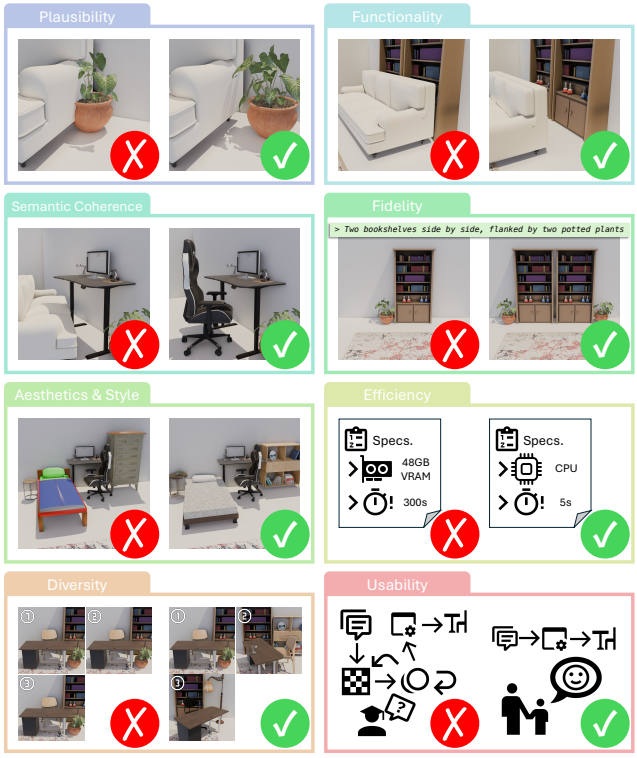


Figure 17: *Compositional 3D scene generation can be evaluated along several complementary axes. The examples shows a 'good' and 'bad' pair to illustrate each axis.*

5.2 Evaluation Approaches and Metrics

Various approaches and metrics have been proposed for evaluating generated 3D scenes. We summarize the approaches, and categorize metrics in the following sections and in Tab. 6.

Demonstration and qualitative comparison. Early works on 3D scene generation rarely included quantitative evaluation, relying instead on demonstrations and qualitative comparisons to show system capabilities [Pfe75; CW96; CS01; SLLG03; CSM14b].

Human as judge. Humans are often treated as the gold standard for judging physical plausibility, functionality, semantic coherence, aesthetics, and fidelity to user intent. As such, a large body of work relies on human judgment to evaluate scene quality [ZHG*16; WZC*24; TPW*25b; XZL*24; YYT*11; XCF*13; CMS*15; FSL*15; SCH*16; KLTZ16; WSCR18; LPX*19; WLW*19; WYN21; PKS*21; ZÖW*23; AGH*24; RMK*24; PTW*25; SGC25]. The simplest form is *manual verification*, where authors or other verifiers visually inspect generated scenes for plausibility, collisions, or adherence to constraints [ZHG*16; WZC*24; TPW*25b; XZL*24]. While easy to conduct, this approach is limited in scope and introduces potential bias. A more systematic variant is the *user study*, where participants are recruited to provide subjective assessments of scene quality. Common designs include: 1) rating scenes on Likert scales [FRS*12; CMS*15; FSL*15; LZM19; LDM25]; 2) ranking outputs from different methods [GSM*23; SLG*25; SYM*24; BBC*25]; and 3) pairwise comparisons via two-alternative forced choice (2AFC) [RWL19; WLW*19; ZÖW*23; VVN*24]. Less common approaches involve matching tasks (e.g., aligning scenes to input conditions) [MSSH13] or testing interactive systems, where measures such as task time [SSS*01; SCA17] or subjective workload (e.g., NASA TLX [HS88; WZC*24]) are collected. Some studies also involve domain experts such as interior designers to provide more informed assessments [MSL*11; KK18]. Human-in-the-loop evaluation provides direct judgments of quality, but it is costly, slow, and difficult to standardize. Small sample sizes, participant variation, and limited 3D visualization can bias results, and neither manual verification nor user studies scale well to large evaluations. Nonetheless, human judgment is widely used, especially for subjective qualities that are difficult to capture with automated metrics.

LLM/VLM as judge. Recent works have begun using LLMs and VLMs as automated judges. They can be prompted to describe a scene, identify inconsistencies, or rate quality along multiple dimensions. Reported criteria include layout realism [FWLS24; ÇHS*24; YLZ*24; LLL*25; GCL*25; NLNN25], functionality [ÇHS*24; YLZ*24; LTT25b; LTT25a], aesthetics [ÇHS*24; YLZ*24; LTT25b], plausibility and attributes of objects [FWLS24; TPW*25b; HBT*25], color and style consistency [ÇHS*24; NLNN25], positional and rotational coherence [SLG*25; LLL*25; RLX*25], and overall quality scores [FWLS24; WQL*24; LLL*25; GCL*25]. Compared to human studies, LLM/VLM-based evaluations are faster, and scale to large numbers of scenes. In this sense, they serve as an automated analogue of user studies. However, results are sensitive to the choice of model, prompts, rendering setup, and evaluation parameters. Models can also hallucinate or misinterpret content, and they remain weak at fine-grained spatial reasoning or physical plausibility [SMT25; REB*25]. Thus, while promising and increasingly popular, LLM/VLM judges are best seen as complementary to human evaluation and other automated metrics, rather than a replacement.

Below, we review commonly used metrics that aim to automate evaluation, often complementing or replacing human judgment or LLM/VLM-based evaluation. We categorize them into several groups based on their evaluation focus.

5.2.1 Realism

Realism metrics aim to provide a high-level measure of how "realistic" or "natural" a generated scene appears, often by comparing it to a reference dataset of real scenes.

Comparison to reference scenes. Some works introduce metrics that compare generated scenes against reference scenes. A common setup is to remove one or more objects from a scene, generate replacements, and then measure how closely the generated objects match the originals in category [ZWK19; FYMH25], position [JLS12; PZR20; KPZ*20; KRS*21; FYMH25], orientation [PZR20; KPZ*20; FYMH25], or size [ZWK19]. Early works reported simple statistics such as the percentage of correctly predicted object categories or mean position/orientation errors. Later studies adopted metrics like precision, recall, and F1 score [LXJ*23; ÖTKG24; CWL*25]. For methods that generate objects from scratch, metrics that measure point cloud dis-

tance (e.g., Chamfer Distance [ZZL*24; VVN*24; LTT25a], Earth Mover's Distance [VVN*24]) are also used to measure geometric similarity. While these provide objective measures, they depend heavily on the reference dataset and capture realism only in aggregate, without considering the constituent aspects that make a scene plausible, functional, or coherent. They also overlook the fact that multiple valid configurations may exist in indoor environments.

Distributional distance. Distances between distributions of generated and reference scenes can be used to assess realism. Borrowed from image generation, metrics such as Fréchet Inception Distance (FID) [HRU*17], its CLIP-based variant (CLIP-FID) [KKA*23], and Kernel Inception Distance (KID) [BSAG18] have been adapted for 3D scenes [PKS*21; PGMW23; TNM*24; FZF*23; ZÖW*23; LM24; HAD*24; YJZH24; SZZ*24; YZLP24; ZÖC*24; PDZ*25; BA25; SGC25]. Other measures include Categorical Kullback-Leibler (CKL) divergence, which compares the distribution of object categories, and object co-occurrence statistics, which check whether pairs of objects appear with similar frequencies as in real scenes [LPX*19; RWL19; YGZT21; PKS*21; PGMW23; YHT*23; TNM*24; FZF*23; HAD*24; SYW*25; FZL*25; SGC25]. These metrics either operate directly on object-level distributions (CKL, co-occurrence) or on image features extracted from renderings with pretrained encoders (e.g., InceptionV3 [SVI*16], CLIP [RKH*21]). They provide a single summary score for realism, convenient for comparing methods. However, they have notable limitations: 1) reliance on 2D renderings makes results sensitive to rendering choices; 2) they miss inherently 3D or functional aspects such as plausibility and usability; 3) they do not capture alignment with user inputs, limiting their use for conditional generation; and 4) their dependence on a reference dataset restricts applicability to open-ended scenarios. As such, while useful, they provide only a partial picture of scene quality, and are often used in conjunction with other metrics.

'Real vs fake' classifiers. Given a set of generated scenes and a reference dataset, a binary classifier can be trained on scene renderings to distinguish real from generated examples [QZH*18; RWL19; WLW*19; PZR20; PKS*21; PGMW23; TNM*24; LM24; HAD*24; SZZ*24; YJZH24; SYW*25; PDZ*25; SGC25]. The classifier's accuracy on a held-out test set is then interpreted as a measure of realism, with low accuracy indicating that generated scenes are difficult to distinguish from real ones. This approach shares many limitations with distributional metrics, including reliance on 2D renderings, lack of sensitivity to specific scene qualities, and dependence on a reference dataset. In addition, results may vary with the choice of classifier, the training setup, and the rendering style, all of which can affect consistency across studies.

5.2.2 Fidelity

Fidelity metrics assess how well a scene matches user input conditions, which may include text descriptions, sketches, example images, scene graphs, or human motion data. Overall, they aim to quantify controllability.

Human motion alignment measures. When human motion is provided as input, fidelity requires that the generated scene affords the intended activities. One approach checks whether the human mesh intersects any surrounding objects during the motion [YWL*22]. Another approach treats the trajectory as a representation of required free space on the ground plane and measures how much of it is blocked by objects in the generated scene [YHT*23; HYHC24]. Other methods evaluate whether the layout supports specific activities inferred from the motion, such as sitting, lying down, or reaching, by verifying that expected contact regions align with actual object surfaces [YHT*23; HYHC24].

Text-image alignment measures. To evaluate how closely a generated scene reflects a text description, metrics adapted from text-to-image generation are often used, most commonly CLIPScore [HHF*21], BLIPScore [LLSH23], and VQAScore [LPL*24]. These approaches compare renderings of the generated scene with the input text using pretrained models. CLIPScore measures similarity between CLIP image and text embeddings, serving as a proxy for text-image alignment for scene renderings [FWLS24; ZHX*24; HIJ*24; YSW*24; WZC*24; WQL*24; LTT25b; PTW*25; DYY*25; GCL*25; HZB*25; ZWWZ25]. BLIPScore applies the same principle but leverages BLIP's embedding space instead of CLIP's [WXC*24; WQL*24; PTW*25; LLL*25; HZB*25]. VQAScore frames the task as visual question answering: the input text is reformulated into a yes/no question about whether the rendered scene contains the described content, and the probability of a "yes" response is taken as the alignment score [WQL*24; PTW*25; LLL*25; HZB*25]. These metrics are straightforward to compute and scale efficiently, but they also have limitations: they capture alignment only at a coarse semantic level, miss fine-grained spatial or relational details, provide scores that are difficult to interpret, and are sensitive to the rendering setup.

Scene graph and relation-based measures. To more explicitly assess spatial and relational fidelity, many methods extract pairwise relationships between objects in the generated scene using geometric heuristics and compare them to the input specifications. For methods conditioned on scene graphs, a corresponding graph can be reconstructed from the generated scene using a trained relation prediction model and compared against the input graph. This measure, often termed scene graph consistency [LZWT20; DMNT21; XHH*23; ZÖW*23; ZÖC*24; WMV*25], evaluates whether both the objects and their relationships are faithfully preserved. For text-conditioned generation, evaluation typically relies on annotated object relationships derived from textual descriptions. A representative metric is iRecall, introduced in InstructScene [LM24] and later adopted by others [FYMH25; BBC*25], which measures whether pairwise spatial relations are correctly realized in the generated scene. A key limitation of these approaches is their reliance on knowing the target set of relationships for a given input, which is hard for free-form or loosely specified inputs. Moreover, inconsistencies in input formats and annotation standards across studies make direct comparison difficult. To address this, SceneEval [TPW*25a] recently introduced a dataset of 500 text descriptions of scenes annotated with expected spatial and relational properties, along with a suite of metrics for standardized evaluation.

5.2.3 Plausibility and Functionality

Plausibility and functionality metrics assess whether a generated scene is physically valid and supports natural use of the space. A

scene can have high fidelity to input conditions yet still be implausible or non-functional if objects intersect, float, or block access to one another. We categorize these metrics into low-level geometric measures and higher-level scene-level measures.

Low-level geometric measures. A straightforward way to assess plausibility is to check for geometric violations in the generated scene. Common metrics include counting the number of object collisions, using either bounding box intersection tests or more precise mesh-based methods, and measuring the ratio of interpenetrating objects [ZZL*24; FWLS24; YJZH24; YLW*24; SLG*25; FZL*25; LLL*25; PDZ*25; BA25; SGC25]. Another basic check is whether objects are properly supported by valid surfaces rather than floating in mid-air [LLL*25]. Similarly, objects should remain within the boundaries of the room; placing them partially or entirely outside indicates an implausible layout [FZF*23; FWLS24; ÇHS*24; YJZH24; WZC*24; SLL*25; SYW*25; RLX*25; ZWWZ25; BA25; SGC25]. These metrics are easy to compute automatically and directly capture violations of physical plausibility. However, they remain low-level: they ensure that objects do not intersect, float, or extend beyond the room, but they do not capture whether the scene is functionally usable or semantically meaningful as a whole.

Scene-level measures. Beyond geometric validity, plausibility and functionality also depend on whether the layout supports natural use of the space. *Navigability* metrics assess whether all free space in a scene is connected and accessible. This is often tested by simulating a virtual agent navigating through the environment or by performing 2D connected component analysis on the object occupancy map [SYM*24; YJZH24; LLL*25; SGC25]. *Object accessibility* measures whether objects can be reached and serve their intended functions, for example by checking that functional sides are unobstructed [ZWWZ25]. A more task-oriented proxy is the *success rate of downstream robotic tasks*, where scenes are assessed based on whether embodied agents can complete activities such as navigation or object interaction [ZLJ*21; DVH*22; JC24; SYM*24]. Together, these measures provide a higher-level view of plausibility and functionality, complementing low-level geometric checks.

5.2.4 Aesthetics

Aesthetics is an important dimension of scene quality but is also subjective and difficult to quantify. There are few established automated metrics for aesthetics in 3D scenes. Some works adapt CLIP-based approaches, for example by subtracting the similarity of a scene rendering to a description without style information from the similarity to a style description, thereby providing a relative measure of style consistency [LM24]. Others use pretrained aesthetic predictors developed for 2D images, such as the LAION aesthetic predictor [SB22] or ImageReward [XLW*23], applying them to renderings of the generated scenes to obtain a score for overall aesthetics [ZHX*24; DYY*25]. However, these metrics are often entangled with general realism and other aspects of the images, making it difficult to isolate aesthetics specifically.

5.2.5 System-level Properties

In addition to scene-level quality, practical evaluation also considers system-level properties that affect usability and deployment. *Efficiency* measures the cost of training and inference. It can be quantified in various ways, for example by measuring generation time [RWL19; WYN21; PKS*21; MWZZ24; KPKK25], computational resource usage (e.g., memory), model parameter size, or convergence rate during training. *Diversity* captures whether a system can generate varied outputs rather than repeating similar layouts. This is often quantified by computing the standard deviation of object categories, locations, orientations, and dimensions [LZWT20; DMNT21; XHH*23; HAD*24], or by measuring the distance to the nearest neighbor in a reference dataset [ZYM*20; WSCR18; RWL19; WLW*19; YGZT21]. *Usability* refers to a system's ease of use and the effort required from users. It is difficult to quantify, and few works report on it directly. Reported metrics include subjective workload scores from user studies (e.g., NASA TLX [HS88; WZC*24; HTL*25]) or the time taken to complete tasks as a proxy for user effort [SSS*01; SCA17; ZHL*19]. These properties do not measure scene quality directly, but they are crucial for assessing the practicality of scene generation systems in real-world applications.

6 Discussion

While previous sections reviewed existing methods and evaluations, the broader question is: where do we go from here? Here, we identify major challenges that limit current methods (Sec. 6.1), and sketch out directions for future research (Sec. 6.2).

6.1 Main Challenges

Depending on the application, different attributes of generated scenes are important: realism, diversity, functionality, or controllability. However, current approaches face limitations that prevent scenes from fully meeting these requirements.

Dense, realistic clutter. Many methods struggle to generate scenes with dense and realistically distributed small objects. This gap limits its applications such as gaming and embodied AI, where clutter is crucial both for visual realism and for training robust agents that must navigate and interact in complex environments. Recent work on small object placement [HBT*25; AAW*25; PTW*25] has made progress, but scaling to diverse object categories while maintaining plausible arrangements remains a significant challenge.

Scenes with articulated objects. Most current methods rely on static 3D assets. This fails to capture the articulated nature of common household items such as cabinet doors, drawers, or chairs. Ignoring articulation limits both realism and functionality, especially for robotics and embodied AI, where interaction with movable parts is essential. For instance, an agent cannot learn to open a cabinet if the door is modeled as fixed. Recent efforts such as PhyScene [YJZH24] take initial steps toward incorporating articulated objects, but progress is constrained by the limited availability of articulated models and the added complexity of ensuring plausible clearance and interactions. Addressing this challenge is crucial for enabling richer physical interactions and making generated scenes more faithful to everyday environments.

Functional, physically-based scenes. Beyond articulation, generated scenes are often not functional. Here, functionality refers to the ability of environments to support realistic human activities——for

example, flipping a light switch to turn on a lamp or using a stove to heat a pot. This is particularly important for immersive applications and embodied AI, where environments should respond to user actions in a believable way. Achieving this also requires physically plausible properties such as weight, friction, and material characteristics. Most current methods ignore these aspects, limiting the utility of generated scenes for applications that demand dynamic environments. Closing this gap will be essential for bridging the divide between static scene geometry and interactive virtual worlds.

Evaluation. While generation methods continue to advance, evaluation approaches have lagged behind. As discussed in Sec. 5, existing metrics often emphasize narrow aspects such as fidelity to input prompts or low-level physical plausibility. Yet, realistic scenes also depend on higher-level qualities: common-sense arrangements (e.g., bookshelves typically placed against walls but sometimes used as room dividers), stylistic consistency (e.g., furniture in a room matching or complementing in design), and functional requirements (e.g., ensuring sufficient clearance for wheelchair accessibility). These dimensions are difficult to capture with current metrics and still require costly human evaluation. Developing comprehensive and scalable metrics for such semantic, stylistic, and functional qualities can enable better assessment of the true performance of current methods, but this remains an open challenge.

6.2 Future Outlook

Several promising directions can address the challenges outlined above and advance the field of compositional 3D scene generation.

Incorporating human knowledge. Current generative models often lack the common-sense priors that humans use when arranging objects and designing spaces. Incorporating human knowledge—whether through curated design rules, constraints learned from human demonstrations, or integration with large language and vision-language models—can guide generation toward more semantically plausible and realistic scenes. Interior design principles such as focal points and balance have been explored in earlier work [MSL*11], but recent methods have largely overlooked these insights, relying instead on learning directly from datasets. This shift highlights an opportunity to revisit and expand knowledge-driven constraints. Whereas domains such as text and 2D image generation benefit from massive corpora of training data, the scale and diversity of 3D datasets remain limited and cannot fully capture the breadth of human environments. Incorporating explicit human knowledge offers a promising way to bridge this gap and move toward more functional and human-centered scene generation.

Interactive generation. Most current systems generate entire scenes in a single forward pass, offering little opportunity for user control or iterative refinement. In contrast, human artists and designers rarely create a complete scene all at once—they iteratively sketch, adjust, and refine until the result satisfies aesthetic and functional goals. Similarly, rather than fixing an incorrect scene only after generation, intermediate user input during the process can guide layouts, object placements, or stylistic choices as the scene evolves. Such co-creative systems could make scene generation more practical for interior design, education, or entertainment, while also producing outputs that better reflect human preferences and creativity. Developing methods that balance automation with interactive steering remains an open and promising direction.

Dynamic scenes. Most existing methods focus on generating static scenes, yet real-world environments are inherently dynamic, with objects and agents constantly moving and interacting. Generating dynamic scenes that capture temporal changes, object interactions, and human activities is crucial for applications such as gaming, virtual reality, and embodied AI. This requires not only modeling the geometry, appearance, and articulation of objects but also their behaviors, affordances, and interactions over time. Physics-based simulation offers one avenue, enabling scenes where objects respond realistically to forces and collisions. Knowledge distilled from video generation models could also inform plausible object motions and interactions. Another promising direction is generating scenes populated with virtual agents that interact with objects, making environments feel alive and enabling embodied agents to train in socially and physically realistic settings. Pursuing this direction will enable richer and more immersive environments that better reflect the complexity of the real world.

Efficient generation. Recent methods that incorporate large language and vision-language models have shown promise for guiding compositional scene generation, but they also introduce new efficiency challenges. Unlike trained generative models, which can produce scenes relatively quickly at inference time, LLM- or VLM-based pipelines often require repeated model calls, leading to significant computational cost and latency. This becomes especially problematic in interactive settings, where users expect rapid feedback when refining or exploring scene variations. Many of the models used are prohibitively large and resource-intensive, making them difficult to run locally. Since these models store vast amounts of general information, an alternative direction is to distill them into smaller, more specialized models focused on scene-related knowledge. In parallel, reducing reliance on repeated calls—through strategies such as caching, reuse of intermediate outputs, or retrieval-augmented generation (RAG) that incorporates external knowledge bases—can further mitigate both time and cost. Addressing these issues can enable practical, real-time, and cost-effective scene generation workflows.

7 Conclusion

Compositional 3D scene generation continues to be a vibrant and rapidly evolving area of research, with different paradigms offering unique strengths and challenges. This survey presented a systematic overview of progress in the field, outlining the key components of a scene generation system and categorizing existing methods by their approaches to each. By analyzing design choices, trade-offs, and paradigm shifts over time, we highlighted both major advances and persistent open challenges. We also discussed promising directions for future research, emphasizing opportunities to push the boundaries of what is possible in generating rich, diverse, and functional 3D environments. We hope this survey serves as a valuable resource for researchers and practitioners, inspiring new ideas toward compositional 3D scene generation that is increasingly realistic, interactive, and impactful across a wide range of applications.

Acknowledgments. This work was funded in part by a CIFAR AI Chair, a Canada Research Chair, and NSERC Discovery Grant.

References

[AAB*25] AGARWAL, NIKET, ALI, ARSLAN, BALA, MACIEJ, et al. "Cosmos world foundation model platform for physical AI". *arXiv preprint arXiv:2501.03575* (2025) 18.

[AAW*25] ABDELREHEEM, AHMED, ALEOTTI, FILIPPO, WATSON, JAMIE, et al. "PlaceIt3D: Language-Guided Object Placement in Real 3D Scenes". *Proc. of International Conference on Computer Vision (ICCV)*. 2025. URL: https://arxiv.org/abs/2505.05288 8, 24.

[ADD*19] AVETISYAN, ARMEN, DAHNERT, MANUEL, DAI, ANGELA, et al. "Scan2CAD: Learning CAD model alignment in RGB-D scans". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 4, 20.

[AGH*24] AGUINA-KANG, RIO, GUMIN, MAXIM, HAN, DO HEON, et al. "Open-Universe Indoor Scene Generation using LLM Program Synthesis and Uncurated Object Databases". *arXiv:2403.09675* (2024). DOI: 10.48550/arXiv.2403.09675 6, 17, 18, 22.

[BA25] BUCHER, MARTIN JJ and ARMENI, IRO. "ReSpace: Text-Driven 3D Scene Synthesis and Editing with Preference Alignment". *arXiv preprint arXiv:2506.02459* (2025) 4, 17, 18, 23, 24.

[BBC*25] BAI, TONGYUAN, BAI, WANGYUANFAN, CHEN, DONG, et al. "FreeScene: Mixed Graph Diffusion for 3D Scene Synthesis from Free Prompts". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 5893–5903 17, 18, 21–23.

[BS95] BUKOWSKI, RICHARD W and SÉQUIN, CARLO H. "Object Associations: A simple and practical approach to virtual 3D manipulation". *Proceedings of the 1995 symposium on Interactive 3D graphics*. 1995, 131–ff 11, 13.

[BSAG18] BIŃKOWSKI, MIKOŁAJ, SUTHERLAND, DOUGAL J., ARBEL, MICHAEL, and GRETTON, ARTHUR. "Demystifying MMD GANs". *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=r1lUOzWCW 21, 23.

[BYLD23] BAR-TAL, OMER, YARIV, LIOR, LIPMAN, YARON, and DEKEL, TALI. "MultiDiffusion: Fusing diffusion paths for controlled image generation". (2023) 11.

[CCD03] CALDERON, CARLOS, CAVAZZA, MARC, and DIAZ, DANIEL. "A new approach to the interactive resolution of configuration problems in virtual environments". *International Symposium on Smart Graphics*. Springer. 2003, 112–122 12.

[CCPS25] CHABAL, THOMAS, CHEN, SHIZHE, PONCE, JEAN, and SCHMID, CORDELIA. "Online 3D Scene Reconstruction Using Neural Object Priors". *Proc. of International Conference on 3D Vision (3DV)*. 2025, 723–734 4, 19.

[CESM17] CHANG, ANGEL X, ERIC, MIHAIL, SAVVA, MANOLIS, and MANNING, CHRISTOPHER D. "SceneSeer: 3D scene design with natural language". *arXiv:1703.00050* (2017) 14.

[ÇHS*24] ÇELEN, ATA, HAN, GUO, SCHINDLER, KONRAD, et al. "I-design: Personalized LLM interior designer". *European Conference on Computer Vision*. Springer. 2024, 217–234 4, 7, 8, 17, 18, 21, 22, 24.

[CLN*23] CHUNG, JAEYOUNG, LEE, SUYOUNG, NAM, HYEONGJIN, et al. "LucidDreamer: Domain-free Generation of 3D Gaussian Splatting Scenes". *arXiv preprint arXiv:2311.13384* (2023) 11.

[CMS*15] CHANG, ANGEL, MONROE, WILL, SAVVA, MANOLIS, et al. "Text to 3D scene generation with rich lexical grounding". *Proc. of the Conference of the Association for Computational Linguistics (ACL)*. 2015 4, 8, 14, 22.

[CNJ*24] CHEN, YIXIN, NI, JUNFENG, JIANG, NAN, et al. "Single-view 3D scene reconstruction with high-fidelity shape and texture". *2024 International Conference on 3D Vision (3DV)*. IEEE. 2024, 1456–1467 20.

[CS01] COYNE, BOB and SPROAT, RICHARD. "WordsEye: An automatic text-to-scene conversion system". *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, 487–496 4, 12, 13, 22.

[CSM14a] CHANG, ANGEL, SAVVA, MANOLIS, and MANNING, CHRISTOPHER D. "Interactive learning of spatial knowledge for text to 3D scene generation". *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014, 14–21 15.

[CSM14b] CHANG, ANGEL, SAVVA, MANOLIS, and MANNING, CHRISTOPHER D. "Learning spatial knowledge for text to 3D scene generation". *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. 2014, 2028–2038. DOI: 10.3115/v1/D14-1217 8, 9, 13, 15, 22.

[CW96] CLAY, SHARON ROSE and WILHELMS, JANE. "Put: Language-based interactive manipulation of objects". *IEEE Computer Graphics and applications* 16.2 (1996), 31–39 12, 13, 22.

[CWL*25] CHANG, ADRIAN, WANG, KAI, LI, YUANBO, et al. "Learning object placement programs for indoor scene synthesis with iterative self training". *arXiv preprint arXiv:2503.04496* (2025) 22.

[DBK*21] DOSOVITSKIY, ALEXEY, BEYER, LUCAS, KOLESNIKOV, ALEXANDER, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". *Proc. of International Conference on Learning Representations (ICLR)*. 2021 3.

[DFB*24] DONG, YUAN, FANG, CHUAN, BO, LIEFENG, et al. "PanoContext-Former: Panoramic total scene understanding with a transformer". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 28087–28097 4, 20.

[DMNT21] DHAMO, HELISA, MANHARDT, FABIAN, NAVAB, NASSIR, and TOMBARI, FEDERICO. "Graph-to-3D: End-to-end generation and manipulation of 3D scenes using scene graphs". *Proc. of International Conference on Computer Vision (ICCV)*. 2021, 16352–16361 21, 23, 24.

[DÖE25] DOGARU, ANDREEA, ÖZER, MERT, and EGGER, BERNHARD. "Gen3DSR: Generalizable 3D Scene Reconstruction via Divide and Conquer from a Single View". *International Conference on 3D Vision 2025*. 2025 19, 20.

[DQM25] DENG, WEI, QI, MENGSHI, and MA, HUADONG. "Global-local tree search in vlms for 3D indoor scene generation". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 8975–8984 17.

[DS14] DEMA, MESFIN and SARI-SARRAF, HAMED. "A relevancy, hierarchical and contextual maximum entropy framework for a data-driven 3D scene generation". *Entropy* 16.5 (2014), 2568–2591 8, 14.

[DVH*22] DEITKE, MATT, VANDERBILT, ELI, HERRASTI, ALVARO, et al. "ProcTHOR: Large-Scale Embodied AI Using Procedural Generation". *Advances in neural information processing systems* 35 (2022), 5982–5994 6, 7, 9, 10, 12, 13, 21, 24.

[DWJ*24] DAI, TIANYUAN, WONG, JOSIAH, JIANG, YUNFAN, et al. "ACDC: Automated creation of digital cousins for robust policy learning". *arXiv e-prints* (2024), arXiv–2410 19, 20.

[DYY*25] DONG, WENQI, YANG, BANGBANG, YANG, ZESONG, et al. "HiScene: Creating Hierarchical 3D Scenes with Isometric View Generation". *arXiv:2504.13072* (2025) 5, 7, 21, 23, 24.

[ESL*25] ENGSTLER, PAUL, SHTEDRITSKI, ALEKSANDAR, LAINA, IRO, et al. "SynCity: Training-free generation of 3D worlds". *arXiv preprint arXiv:2503.16420* (2025) 5, 11.

[FAKD24] FRIDMAN, RAFAIL, ABECASIS, AMIT, KASTEN, YONI, and DEKEL, TALI. "Scenescape: Text-driven consistent scene generation". *Advances in Neural Information Processing Systems* 36 (2024) 11.

[FCG*21] FU, HUAN, CAI, BOWEN, GAO, LIN, et al. "3D-FRONT: 3D furnished rooms with layouts and semantics". *Proc. of International Conference on Computer Vision (ICCV)*. 2021, 10913–10922. DOI: 10.1109/iccv48922.2021.01075. URL: http://dx.doi.org/10.1109/iccv48922.2021.01075 6, 7, 13, 17.

[FCW*17] FU, QIANG, CHEN, XIAOWU, WANG, XIAOTIAN, et al. "Adaptive synthesis of indoor scenes via activity-associated object relation graphs". *ACM Transactions on Graphics (TOG)* 36.6 (2017), 1–13 4, 15.

[FMD*25] FIME, AWAL AHMED, MAHMUD, SAIFUDDIN, DAS, ARPITA, et al. "Automatic Scene Generation: State-of-the-Art Techniques, Models, Datasets, Challenges, and Future Prospects". *IEEE Access* (2025) 2.

[FRS*12] FISHER, MATTHEW, RITCHIE, DANIEL, SAVVA, MANOLIS, et al. "Example-based synthesis of 3D object arrangements". *ACM Transactions on Graphics (TOG)* 31.6 (2012), 1–11 6, 8, 9, 13–15, 22.

[FSH11] FISHER, MATTHEW, SAVVA, MANOLIS, and HANRAHAN, PAT. "Characterizing structural relationships in scenes using graph kernels". *ACM SIGGRAPH 2011 papers*. 2011, 1–12 14.

[FSL*15] FISHER, MATTHEW, SAVVA, MANOLIS, LI, YANGYAN, et al. "Activity-centric scene synthesis for functional 3D scene modeling". *ACM Transactions on Graphics (TOG)* 34.6 (2015), 1–13 4, 6, 13, 15, 22.

[FWLS24] FU, RAO, WEN, ZEHAO, LIU, ZICHEN, and SRIDHAR, SRINATH. "AnyHome: Open-vocabulary generation of structured and textured 3D homes". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2024, 52–70 6–8, 10, 17, 18, 21–24.

[FYMH25] FANG, SHAOHENG, YANG, HAITAO, MOONEY, RAYMOND, and HUANG, QIXING. "Text-Guided Interactive Scene Synthesis with Scene Prior Guidance". *Computer Graphics Forum*. Wiley Online Library. 2025, e70039 16, 21–23.

[FZF*23] FENG, WEIXI, ZHU, WANRONG, FU, TSU-JUI, et al. "LayoutGPT: Compositional visual planning and generation with large language models". *Advances in neural information processing systems* 36 (2023), 18225–18250 5, 7, 8, 10, 17, 18, 21, 23, 24.

[FZL*25] FENG, WEITAO, ZHOU, HANG, LIAO, JING, et al. "CasaGPT: cuboid arrangement and scene assembly for interior design". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025, 29173–29182 21, 23, 24.

[GBC16] GOODFELLOW, IAN, BENGIO, YOSHUA, and COURVILLE, AARON. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 3.

[GCL*25] GU, ZEQI, CUI, YIN, LI, ZHAOSHUO, et al. "ArtiScene: Language-Driven Artistic 3D Scene Generation Through Image Intermediary". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 2891–2901 5, 19, 20, 22, 23.

[GCSR95] GELMAN, ANDREW, CARLIN, JOHN B, STERN, HAL S, and RUBIN, DONALD B. *Bayesian data analysis*. Chapman and Hall/CRC, 1995 3.

[GDN22] GÜMELI, CAN, DAI, ANGELA, and NIESSNER, MATTHIAS. "ROCA: Robust CAD Model Retrieval and Alignment from a Single Image". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 20.

[GHY*25] GUMIN, MAXIM, HAN, DO HEON, YOO, SEUNG JEAN, et al. "Imperative vs. Declarative Programming Paradigms for Open-Universe Scene Generation". *arXiv:2504.05482* (2025) 5.

[GPM*14] GOODFELLOW, IAN J, POUGET-ABADIE, JEAN, MIRZA, MEHDI, et al. "Generative adversarial nets". *Advances in neural information processing systems* 27 (2014) 3.

[GRLD24] GAO, DAOYI, ROZENBERSZKI, DAVID, LEUTENEGGER, STEFAN, and DAI, ANGELA. "DiffCAD: Weakly-Supervised Probabilistic CAD Model Retrieval and Alignment from an RGB Image". *ACM Transactions on Graphics (TOG)* 43.4 (2024), 1–15 20.

[GSA*20] GAN, CHUANG, SCHWARTZ, JEREMY, ALTER, SETH, et al. "ThreeDWorld: A platform for interactive multi-modal physical simulation". *arXiv preprint arXiv:2007.04954* (2020) 7.

[GSM*23] GAO, LIN, SUN, JIA-MU, MO, KAICHUN, et al. "SceneHGN: Hierarchical graph networks for 3D indoor scene generation with fine-grained geometry". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.7 (2023), 8902–8919 10, 16, 22.

[HAD*24] HU, SIYI, ARROYO, DIEGO MARTIN, DEBATS, STEPHANIE, et al. "Mixed Diffusion for 3D Indoor Scene Synthesis". *arXiv:2405.21066* (2024) 4, 17, 21, 23, 24.

[HBT*25] HUANG, IAN, BAO, YANAN, TRUONG, KAREN, et al. "Fireplace: Geometric refinements of llm common sense reasoning for 3D object placement". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 13466–13476 8, 17, 18, 22, 24.

[HCO*23] HÖLLEIN, LUKAS, CAO, ANG, OWENS, ANDREW, et al. "Text2room: Extracting textured 3D meshes from 2D text-to-image models". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 7909–7920 11.

[HGA*25] HUANG, ZEHUAN, GUO, YUAN-CHEN, AN, XINGQIAO, et al. "MIDI: Multi-Instance Diffusion for Single Image to 3D Scene Generation". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 23646–23657 9.

[HHF*21] HESSEL, JACK, HOLTZMAN, ARI, FORBES, MAXWELL, et al. "CLIPScore: A Reference-free Evaluation Metric for Image Captioning". *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. 2021, 7514–7528 21, 23.

[HIJ*24] HU, ZINIU, ISCEN, AHMET, JAIN, AASHI, et al. "SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code". *International Conference on Machine Learning (ICML)*. 2024, 19252–19282 5, 7, 8, 17, 18, 23.

[HJA20] HO, JONATHAN, JAIN, AJAY, and ABBEEL, PIETER. "Denoising diffusion probabilistic models". *Advances in neural information processing systems* 33 (2020), 6840–6851 3.

[HJCZ25] HUA, TONGYAN, JIANG, LUTAO, CHEN, YING-CONG, and ZHAO, WUFAN. "Sat2City: 3D City Generation from A Single Satellite Image with Cascaded Latent Diffusion". *arXiv preprint arXiv:2507.04403* (2025) 11.

[HKAG23] HUANG, IAN, KRISHNA, VRISHAB, ATEKHA, OMORUYI, and GUIBAS, LEONIDAS. "Aladdin: Zero-Shot Hallucination of Stylized 3D Assets from Abstract Scene Descriptions". *arXiv:2306.06212* (2023) 7.

[HLG*24] HURST, AARON, LERER, ADAM, GOUCHER, ADAM P, et al. "GPT-4o system card". *arXiv preprint arXiv:2410.21276* (2024) 20.

[HPSC16] HANDA, ANKUR, PĂTRĂUCEAN, VIORICA, STENT, SIMON, and CIPOLLA, ROBERTO. "SceneNet: An annotated model generator for indoor scene understanding". *International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, 5737–5743 14.

[HRU*17] HEUSEL, MARTIN, RAMSAUER, HUBERT, UNTERTHINER, THOMAS, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". *Advances in neural information processing systems* 30 (2017) 21, 23.

[HS88] HART, SANDRA G and STAVELAND, LOWELL E. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". *Advances in psychology*. Vol. 52. Elsevier, 1988, 139–183 21, 22, 24.

[HSF17] HENDERSON, PAUL, SUBR, KARTIC, and FERRARI, VITTORIO. "Automatic generation of constrained furniture layouts". *arXiv:1711.10939* (2017) 14, 15.

[HTL*25] HOU, ZHUANGZE, TIAN, JINGZE, LI, NIANLONG, et al. "EchoLadder: Progressive AI-Assisted Design of Immersive VR Scenes". *arXiv preprint arXiv:2508.02173* (2025) 18, 21, 24.

[HWZ*25] HUANG, ZHENING, WU, XIAOYANG, ZHONG, FANGCHENG, et al. *LiteReality: Graphics-Ready 3D Scene Reconstruction from RGB-D Scans*. 2025. eprint: 2507.02861. URL: https://arxiv.org/abs/2507.02861 19, 20.

[HYHC24] HONG, XIAOLIN, YI, HONGWEI, HE, FAZHI, and CAO, QIONG. "Human-Aware 3D Scene Generation with Spatially-constrained Diffusion Models". *arXiv:2406.18159* (2024) 21, 23.

[HZB*25] HUANG, RUI, ZHAI, GUANGYAO, BAUER, ZURIA, et al. "Video Perception Models for 3D Scene Synthesis". *arXiv preprint arXiv:2506.20601* (2025) 4, 5, 19–21, 23.

[ISS17] IZADINIA, HAMID, SHAN, QI, and SEITZ, STEVEN M. "IM2CAD". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 5134–5143 20.

[JC24] JIA, YINSEN and CHEN, BOYUAN. "Cluttergen: A cluttered scene generator for robot learning". *Proc. of Conference on Robot Learning (CoRL)*. 2024 21, 24.

[JLS12] JIANG, YUN, LIM, MARCUS, and SAXENA, ASHUTOSH. "Learning object arrangements in 3D scenes using human context". *arXiv:1206.6462* (2012) 8, 14, 15, 21, 22.

[JQZ*18] JIANG, CHENFANFU, QI, SIYUAN, ZHU, YIXIN, et al. "Configurable 3D scene synthesis and 2D image rendering with per-pixel ground truth using stochastic grammars". *International Journal of Computer Vision (IJCV)* 126.9 (2018), 920–941 14.

[JSM*20] JIANG, CHIYU, SUD, AVNEESH, MAKADIA, AMEESH, et al. "Local implicit grid representations for 3D scenes". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 6001–6010 5.

[KALD20] KUO, WEICHENG, ANGELOVA, ANELIA, LIN, TSUNG-YI, and DAI, ANGELA. "Mask2CAD: 3D shape prediction by learning to segment and retrieve". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2020, 260–277 19, 20.

[KALD21] KUO, WEICHENG, ANGELOVA, ANELIA, LIN, TSUNG-YI, and DAI, ANGELA. "Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 12589–12599 20.

[KF09] KOLLER, DAPHNE and FRIEDMAN, NIR. *Probabilistic graphical models: principles and techniques*. MIT press, 2009 2.

[Kjø00] KJØLAAS, KARI ANNE HØIER. "Automatic furniture population of large architectural models". PhD thesis. Massachusetts Institute of Technology, 2000 12.

[KK17] KÁN, PETER and KAUFMANN, HANNES. "Automated interior design using a genetic algorithm". *Proceedings of the 23rd ACM symposium on virtual reality software and technology*. 2017, 1–10 12.

[KK18] KÁN, PETER and KAUFMANN, HANNES. "Automatic furniture arrangement using greedy cost minimization". *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2018, 491–498 12, 22.

[KKA*23] KYNKÄÄNNIEMI, TUOMAS, KARRAS, TERO, AITTALA, MIIKA, et al. "The Role of ImageNet Classes in Fréchet Inception Distance". *ICLR*. 2023. URL: https://openreview.net/forum?id=4oXTQ6m_ws8 21, 23.

[KLTZ16] KERMANI, Z SADEGHIPOUR, LIAO, ZICHENG, TAN, PING, and ZHANG, HAO. "Learning 3D Scene Synthesis from Annotated RGB-D Images". *Computer Graphics Forum*. Vol. 35. Wiley Online Library. 2016, 197–206 8, 14, 15, 22.

[KMH*17] KOLVE, ERIC, MOTTAGHI, ROOZBEH, HAN, WINSON, et al. "AI2-THOR: An interactive 3D environment for visual AI". *arXiv preprint arXiv:1712.05474* (2017) 7, 9.

[KMJ*24] KHANNA, MUKUL, MAO, YONGSEN, JIANG, HANXIAO, et al. "Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 7.

[KPKK25] KIM, JIHYUN, PARK, JUNHO, KONG, KYEONGBO, and KANG, SUK-JU. "Programmable-Room: Interactive Textured 3D Room Meshes Generation Empowered by Large Language Models". *IEEE Transactions on Multimedia* 27 (2025), 6358–6368 17, 18, 24.

[KPZ*20] KESHAVARZI, MOHAMMAD, PARIKH, AAKASH, ZHAI, XIYU, et al. "SceneGen: Generative contextual scene augmentation using scene graph priors". *arXiv:2009.12395* (2020) 4, 15, 21, 22.

[KRS*21] KESHAVARZI, MOHAMMAD, REYES, FLAVIANO CHRISTIAN, SHRIVASTAVA, RITIKA, et al. "Contextual scene augmentation and synthesis via gsacnet". *arXiv preprint arXiv:2103.15369* (2021) 22.

[KW13] KINGMA, DIEDERIK P and WELLING, MAX. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114* (2013) 3.

[LA07] LIEN, JYH-MING and AMATO, NANCY M. "Approximate convex decomposition of polyhedra". *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 2007, 121–131 9.

[LDL*23] LYU, XIAOYANG, DAI, PENG, LI, ZIZHANG, et al. "Learning a room with the occ-sdf hybrid: Signed distance function mingled with occupancy aids scene representation". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 8940–8950 5.

[LDM25] LITTLEFAIR, GABRIELLE, DUTT, NILADRI SHEKHAR, and MITRA, NILOY J. "FlairGPT: Repurposing LLMs for interior designs". *Computer Graphics Forum*. Wiley Online Library. 2025, e70036 17, 18, 22.

[LDR*22] LUGMAYR, ANDREAS, DANELLJAN, MARTIN, ROMERO, ANDRES, et al. "Repaint: Inpainting using denoising diffusion probabilistic models". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, 11461–11471 11.

[LGWM22] LEIMER, KURT, GUERRERO, PAUL, WEISS, TOMER, and MUSIALSKI, PRZEMYSLAW. "LayoutEnhancer: Generating good indoor layouts from imperfect data". *ACM SIGGRAPH Asia Conference Proceedings*. 2022, 1–8 4, 15, 16.

[LHC25] LEE, HAN-HUNG, HAN, QINGHONG, and CHANG, ANGEL X. "NuiScene: Exploring Efficient Generation of Unbounded Outdoor Scenes". *Proc. of International Conference on Computer Vision (ICCV)*. 2025. DOI: 10.48550/arXiv.2503.16375 11.

[LJD*24] LANGER, FLORIAN, JU, JIHONG, DIKOV, GEORGI, et al. "FastCAD: Real-Time CAD Retrieval and Alignment from Scans and Videos". *European Conference on Computer Vision*. Springer. 2024, 60–77 19.

[LLJ*24] LEE, JUMIN, LEE, SEBIN, JO, CHANGHO, et al. "SemCity: Semantic scene generation with triplane diffusion". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, 28337–28347 5, 11.

[LLJ25] LIN, YANGKAI, LEI, JIABAO, and JIA, KUI. "SceneLCM: End-to-End Layout-Guided Interactive Indoor Scene Generation with Latent Consistency Model". *arXiv preprint arXiv:2506.07091* (2025) 17, 18.

[LLL*24] LIU, YUHENG, LI, XINKE, LI, XUETING, et al. "Pyramid diffusion for fine 3D large scene generation". *European Conference on Computer Vision*. Springer. 2024, 71–87 11.

[LLL*25] LING, LU, LIN, CHEN-HSUAN, LIN, TSUNG-YI, et al. "Scenethesis: A Language and Vision Agentic Framework for 3D Scene Generation". *arXiv:2505.02836* (2025) 7, 9, 17, 18, 21–24.

[LLM*23] LIN, CHIEH HUBERT, LEE, HSIN-YING, MENAPACE, WILLI, et al. "InfiniCity: Infinite-scale city synthesis". *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, 22808–22818 11.

[LLSH23] LI, JUNNAN, LI, DONGXU, SAVARESE, SILVIO, and HOI, STEVEN. "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". *International Conference on Machine Learning (ICML)*. PMLR. 2023, 19730–19742 21, 23.

[LM24] LIN, CHENGUO and MU, YADONG. "InstructScene: Instruction-Driven 3D Indoor Scene Synthesis with Semantic Graph Prior". *Proc. of International Conference on Learning Representations (ICLR)*. 2024 5, 7, 8, 10, 16, 17, 21, 23, 24.

[LPL*24] LIN, ZHIQIU, PATHAK, DEEPAK, LI, BAIQI, et al. "Evaluating text-to-visual generation with image-to-text generation". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2024, 366–384 21, 23.

[LPX*19] LI, MANYI, PATIL, AKSHAY GADI, XU, KAI, et al. "GRAINS: Generative recursive autoencoders for indoor scenes". *ACM Transactions on Graphics (TOG)* 38.2 (2019), 1–16 5, 8, 15, 16, 22, 23.

[LRY*24] LU, YIFAN, REN, XUANCHI, YANG, JIAWEI, et al. "InfiniCube: Unbounded and controllable dynamic 3D driving scene generation with world-guided video models". *arXiv preprint arXiv:2412.03934* (2024) 5, 11.

[LSC24] LEE, HANHUNG, SAVVA, MANOLIS, and CHANG, ANGEL X. "Text-to-3D Shape Generation". *Computer Graphics Forum*. Wiley Online Library. 2024, e15061. DOI: 10.1111/cgf.15061 10.

[LTT25a] LIU, XINHANG, TAI, YU-WING, and TANG, CHI-KEUNG. "Agentic 3D Scene Generation with Spatially Contextualized VLMs". *arXiv preprint arXiv:2505.20129* (2025) 17, 18, 22, 23.

[LTT25b] LIU, XINHANG, TANG, CHI-KEUNG, and TAI, YU-WING. "WorldCraft: Photo-Realistic 3D World Creation and Customization via LLM Agents". *arXiv:2502.15601* (2025) 17, 22, 23.

[LXJ*23] LIU, JINGYU, XIONG, WENHAN, JONES, IAN, et al. "CLIP-Layout: Style-consistent indoor scene synthesis with semantic furniture embedding". *arXiv:2303.03565* (2023) 21, 22.

[LXN*25] LIU, JIACHEN, XUE, YUAN, NI, HAOMIAO, et al. "Computer-aided layout generation for building design: A review". *Computational Visual Media* 11.4 (2025), 677–707. DOI: 10.26599/CVM.2025.9450484 2.

[LYS*20] LI, ZHENGQIN, YU, TING-WEI, SANG, SHEN, et al. "Openrooms: An end-to-end open framework for photorealistic indoor scene datasets". *arXiv preprint arXiv:2007.12868* (2020) 4.

[LZC*22] LIU, HAOLIN, ZHENG, YUJIAN, CHEN, GUANYING, et al. "Towards high-fidelity single-view holistic reconstruction of indoor scenes". *European Conference on Computer Vision*. Springer. 2022, 429–446 20.

[LZC25] LEE, HAN-HUNG, ZHANG, YIMING, and CHANG, ANGEL X. "Duoduo CLIP: Efficient 3D Understanding with Multi-View Images". *Proc. of International Conference on Learning Representations (ICLR)*. 2025. DOI: 10.48550/arXiv.2406.11579 9, 20.

[LZM19] LIANG, YUAN, ZHANG, SONG-HAI, and MARTIN, RALPH ROBERT. "Learning guidelines for automatic indoor scene design". *Multimedia Tools and Applications* 78 (2019), 5003–5023 14, 22.

[LZWT20] LUO, ANDREW, ZHANG, ZHOUTONG, WU, JIAJUN, and TENENBAUM, JOSHUA B. "End-to-end optimization of scene layout". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, 3754–3763 5, 15, 16, 21, 23, 24.

[LZZ*] LIU, JIA-HONG, ZHANG, SHAO-KUI, ZHANG, TIANQI, et al. "SceneFunctioner: Tailoring Large Language Model for Function-Oriented Interactive Scene Synthesis". () 17, 18.

[MBS*24] MA, XIANZHENG, BHALGAT, YASH, SMART, BRANDON, et al. "When LLMs step into the 3D World: A Survey and Meta-Analysis of 3D Tasks via Multi-modal Large Language Models". *arXiv:2405.10255* (2024) 2.

[MET] META. *Project Aria*. https://www.projectaria.com/datasets/ase/ 7.

[MLND25] MENG, QUAN, LI, LEI, NIESSNER, MATTHIAS, and DAI, ANGELA. "LT3SD: Latent trees for 3D scene diffusion". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 650–660 5, 11.

[MLP16] MAMOU, KHALED, LENGYEL, E, and PETERS, A. "Volumetric hierarchical approximate convex decomposition". *Game engine gems* 3 (2016), 141–158 9.

[MPF*18] MA, RUI, PATIL, AKSHAY GADI, FISHER, MATTHEW, et al. "Language-driven synthesis of 3D scenes from scene databases". *ACM Transactions on Graphics (TOG)* 37.6 (2018), 1–16 4, 14, 15.

[MPNF22] MANINIS, KEVIS-KOKITSI, POPOV, STEFAN, NIESSNER, MATTHIAS, and FERRARI, VITTORIO. "Vid2Cad: CAD model alignment using multi-view constraints from videos". *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022), 1320–1327 4, 19.

[MSDO24] MAILLARD, LÉOPOLD, SEREYJOL-GARROS, NICOLAS, DURAND, TOM, and OVSJANIKOV, MAKS. "Debara: Denoising-based 3D room arrangement generation". *Advances in neural information processing systems* 37 (2024), 109202–109232 10, 21.

[MSL*11] MERRELL, PAUL, SCHKUFZA, ERIC, LI, ZEYANG, et al. "Interactive furniture layout using interior design guidelines". *ACM Transactions on Graphics (TOG)* 30.4 (2011), 1–10 7, 8, 12, 13, 21, 22, 25.

[MSSH13] MAJEROWICZ, LUCAS, SHAMIR, ARIEL, SHEFFER, ALLA, and HOOS, HOLGER H. "Filling your shelves: Synthesizing diverse style-preserving artifact arrangements". *IEEE transactions on visualization and computer graphics* 20.11 (2013), 1507–1518 22.

[MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P, TANCIK, MATTHEW, et al. "NeRF: Representing scenes as neural radiance fields for view synthesis". *Communications of the ACM* 65.1 (2021), 99–106 5.

[MWZX25] MENG, YANXU, WU, HAONING, ZHANG, YA, and XIE, WEIDI. "SceneGen: Single-Image 3D Scene Generation in One Feed-forward Pass". *arXiv preprint arXiv:2508.15769* (2025) 19, 20.

[MWZZ24] MIN, WENJIE, WU, WENMING, ZHANG, GAOFENG, and ZHENG, LIPING. "FuncScene: Function-centric indoor scene synthesis via a variational autoencoder framework". *Computer Aided Geometric Design* 111 (2024), 102319 21, 24.

[NDHN22] NIE, YINYU, DAI, ANGELA, HAN, XIAOGUANG, and NIESSNER, MATTHIAS. "Pose2room: understanding 3D scenes from human activities". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2022, 425–443 4, 16.

[NDHN23] NIE, YINYU, DAI, ANGELA, HAN, XIAOGUANG, and NIESSNER, MATTHIAS. "Learning 3D scene priors with 2D supervision". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, 792–802 17.

[NHC*21] NAUATA, NELSON, HOSSEINI, SEPIDEHSADAT, CHANG, KAI-HUNG, et al. "House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, 13632–13641 10.

[NHG*20] NIE, YINYU, HAN, XIAOGUANG, GUO, SHIHUI, et al. "Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 55–64 20.

[NLL*25] NI, CHAOJUN, LI, JIE, LI, HAOYUN, et al. "WonderFree: Enhancing Novel View Quality and Cross-View Consistency for 3D Scene Exploration". *arXiv preprint arXiv:2506.20590* (2025) 7, 11.

[NLNN25] NGUYEN, TOAN, LE, TRI, NGUYEN, QUANG, and NGUYEN, ANH. "FurniMAS: Language-Guided Furniture Decoration using Multi-Agent System". *arXiv preprint arXiv:2507.04770* (2025) 17, 22.

[ODM*23] OQUAB, MAXIME, DARCET, TIMOTHÉE, MOUTAKANNI, THÉO, et al. "DINOv2: Learning robust visual features without supervision". *arXiv preprint arXiv:2304.07193* (2023) 20.

[ÖTKG24] ÖCAL, BAŞAK MELIS, TATARCHENKO, MAXIM, KARAOĞLU, SEZER, and GEVERS, THEO. "SceneTeller: Language-to-3D Scene Generation". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2024, 362–378 17, 22.

[PDZ*25] PFAFF, NICHOLAS, DAI, HONGKAI, ZAKHAROV, SERGEY, et al. "Steerable Scene Generation with Post Training and Inference-Time Search". *arXiv preprint arXiv:2505.04831* (2025) 16, 23, 24.

[PEL*24] PODELL, DUSTIN, ENGLISH, ZION, LACEY, KYLE, et al. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis". *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=di52zR8xgf 18.

[Pfe75] PFEFFERKORN, CHARLES E. "A heuristic problem solving design system for equipment or furniture layouts". *Communications of the ACM* 18.5 (1975), 286–297 11, 13, 22.

[PGMW23] PARA, WAMIQ REYAZ, GUERRERO, PAUL, MITRA, NILOY, and WONKA, PETER. "COFS: Controllable furniture layout synthesis". *ACM SIGGRAPH Conference Proceedings*. 2023, 1–11 10, 23.

[PJBM22] POOLE, BEN, JAIN, AJAY, BARRON, JONATHAN T, and MILDENHALL, BEN. "DreamFusion: Text-to-3D using 2D diffusion". *arXiv preprint arXiv:2209.14988* (2022) 7.

[PKS*21] PASCHALIDOU, DESPOINA, KAR, AMLAN, SHUGRINA, MARIA, et al. "ATISS: Autoregressive transformers for indoor scene synthesis". *Advances in neural information processing systems* 34 (2021), 12013–12026 4, 6–10, 15, 16, 21–24.

[PPL*24] PATIL, AKSHAY GADI, PATIL, SUPRIYA GADI, LI, MANYI, et al. "Advances in Data-Driven Analysis and Synthesis of 3D Indoor Scenes". *Computer Graphics Forum* 43.1 (2024), e14927. DOI: 10.1111/cgf.14927 2.

[PTW*25] PUN, HOU IN DEREK, TAM, HOU IN IVAN, WANG, AUSTIN T, et al. "HSM: Hierarchical Scene Motifs for Multi-Scale Indoor Scene Generation". *arXiv:2503.16848* (2025) 4, 5, 7–10, 17, 18, 21–24.

[PZR20] PURKAIT, PULAK, ZACH, CHRISTOPHER, and REID, IAN. "SG-VAE: Scene grammar variational autoencoder to generate new indoor scenes". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2020, 155–171 5, 15, 21–23.

[QZH*18] QI, SIYUAN, ZHU, YIXIN, HUANG, SIYUAN, et al. "Human-centric indoor scene synthesis using stochastic grammar". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, 5899–5908 5, 8, 14, 15, 21, 23.

[RBL*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. "High-resolution image synthesis with latent diffusion models". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, 10684–10695 11.

[REB*25] RODIONOV, FEDOR, ELDESOKEY, ABDELRAHMAN, BIRSAK, MICHAEL, et al. "PlanQA: A Benchmark for Spatial Reasoning in LLMs using Structured Representations". *arXiv preprint arXiv:2507.07644* (2025) 22.

[RHW86] RUMELHART, DAVID E, HINTON, GEOFFREY E, and WILLIAMS, RONALD J. "Learning representations by back-propagating errors". *nature* 323.6088 (1986), 533–536 3.

[RHZ*24] REN, XUANCHI, HUANG, JIAHUI, ZENG, XIAOHUI, et al. "XCube: Large-Scale 3D Generative Modeling using Sparse Voxel Hierarchies". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024 11.

[RKH*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. "Learning transferable visual models from natural language supervision". *International Conference on Machine Learning (ICML)*. Ed. by MEILA, MARINA and ZHANG, TONG. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, 8748–8763 9, 23.

[RLX*25] RAN, XINGJIAN, LI, YIXUAN, XU, LINNING, et al. "Direct Numerical Layout Generation for 3D Indoor Scene Synthesis via Spatial Reasoning". *arXiv preprint arXiv:2506.05341* (2025) 17, 22, 24.

[RMK*24] RAISTRICK, ALEXANDER, MEI, LINGJIE, KAYAN, KARHAN, et al. "Infinigen Indoors: Photorealisltic Indoor Scenes using Procedural Generation". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, 21783–21794 6, 7, 9, 12, 13, 22.

[RWL19] RITCHIE, DANIEL, WANG, KAI, and LIN, YU-AN. "Fast and flexible indoor scene synthesis via deep convolutional generative models". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, 6182–6190. DOI: 10.1109/CVPR.2019.00634 5, 15, 16, 21–24.

[SB22] SCHUHMANN, CHRISTOPH and BEAUMONT, ROMAIN. "LAION-aesthetics". *LAION. AI* (2022) 21, 24.

[SCA17] SAVVA, MANOLIS, CHANG, ANGEL X, and AGRAWALA, MANEESH. "SceneSuggest: Context-driven 3D scene design". *arXiv:1703.00061* (2017) 14, 21, 22, 24.

[SCH*16] SAVVA, MANOLIS, CHANG, ANGEL X, HANRAHAN, PAT, et al. "PiGraphs: Learning interaction snapshots from observations". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 1–12 10, 14, 15, 22.

[SF95] SHINYA, MIKIO and FORGUE, MARIE-CLAIRE. "Laying out objects with geometric and physical constraints". *The Visual Computer* 11 (1995), 188–201 9, 11, 13.

[SFH*25] SU, CHONG, FU, YINGBIN, HU, ZHEYUAN, et al. "CHOrD: Generation of collision-free, house-scale, and organized digital twins for 3D indoor scenes with controllable floor plans and optimal layouts". *arXiv:2503.11958* (2025) 5, 17.

[SGC25] SUN, XIAOHAO, GOEL, DIVYAM, and CHANG, ANGEL X. "SemLayoutDiff: Semantic Layout Generation with Diffusion Model for Indoor Scene Synthesis". *arXiv preprint arXiv:2508.18597* (2025) 4, 10, 16, 21–24.

[SHF23] SHABANI, MOHAMMAD AMIN, HOSSEINI, SEPIDEHSADAT, and FURUKAWA, YASUTAKA. "HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, 5466–5475 10.

[SLG*25] SUN, FAN-YUN, LIU, WEIYU, GU, SIYI, et al. "LayoutVLM: Differentiable optimization of 3D layout via vision-language models". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 29469–29478 5, 7, 8, 17, 18, 21, 22, 24.

[SLL*25] SUN, WEILIN, LI, XINRAN, LI, MANYI, et al. "Hierarchically-Structured Open-Vocabulary Indoor Scene Synthesis with Pre-trained Large Language Model". *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 7. 2025, 7122–7130 7, 17, 24.

[SLLG03] SANCHEZ, STEPHANE, LE ROUX, OLIVIER, LUGA, HERVÉ, and GAILDRAT, VÉRONIQUE. "Constraint-Based 3D-Object Layout using a Genetic Algorithm". *Proc. Conf. on Computer Graphics and Artificial Intelligence*. Vol. 28. 2003 4, 12, 13, 21, 22.

[SMT25] STOGIANNIDIS, ILIAS, MCDONAGH, STEVEN, and TSAFTARIS, SOTIRIOS A. "Mind the gap: Benchmarking spatial reasoning in vision-language models". *arXiv preprint arXiv:2503.19707* (2025) 22.

[SSS*01] SMITH, GRAHAM, STUERZLINGER, WOLFGANG, SALZMAN, TIM, et al. "3D scene manipulation with 2D devices and constraints". *Graphics interface*. Vol. 1. 2001, 135–142 11, 13, 21, 22, 24.

[STLR25] SHRIRAM, JAIDEV, TREVITHICK, ALEX, LIU, LINGJIE, and RAMAMOORTHI, RAVI. "RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion". 2025 11.

[SVI*16] SZEGEDY, CHRISTIAN, VANHOUCKE, VINCENT, IOFFE, SERGEY, et al. "Rethinking the inception architecture for computer vision". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, 2818–2826 23.

[SWMG15] SOHL-DICKSTEIN, JASCHA, WEISS, ERIC, MAHESWARANATHAN, NIRU, and GANGULI, SURYA. "Deep unsupervised learning using nonequilibrium thermodynamics". *International Conference on Machine Learning (ICML)*. pmlr. 2015, 2256–2265 3.

[SY06] SEVERSKY, LEE M and YIN, LIJUN. "Real-time automatic 3D scene generation from natural language voice and text descriptions". *Proceedings of the 14th ACM international conference on Multimedia*. 2006, 61–64 12, 13, 21.

[SYM*24] SUN, JIA-MU, YANG, JIE, MO, KAICHUN, et al. "Haisor: Human-aware indoor scene optimization via deep reinforcement learning". *ACM Transactions on Graphics (TOG)* 43.2 (2024), 1–17 16, 21, 22, 24.

[SYW*25] SUN, KAIFAN, YANG, BINGCHEN, WONKA, PETER, et al. "RelTriple: Learning Plausible Indoor Layouts by Integrating Relationship Triples into the Diffusion Process". *arXiv:2503.20289* (2025) 4, 16, 17, 23, 24.

[SYZ*17] SONG, SHURAN, YU, FISHER, ZENG, ANDY, et al. "Semantic scene completion from a single depth image". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, 1746–1754 13.

[SZZ*24] SUN, QI, ZHOU, HANG, ZHOU, WENGANG, et al. "Forest2Seq: Revitalizing order prior for sequential indoor scene synthesis". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2024, 251–268 15, 23.

[TBSD09] TUTENEL, TIM, BIDARRA, RAFAEL, SMELIK, RUBEN M, and DE KRAKER, KLAAS JAN. "Rule-based layout solving and its application to procedural interior generation". *CASA workshop on 3D advanced media in gaming and simulation*. 2009 12.

[TNM*24] TANG, JIAPENG, NIE, YINYU, MARKHASIN, LEV, et al. "DiffuScene: Denoising diffusion models for generative indoor scene synthesis". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, 20507–20518 4, 6–8, 10, 16, 17, 21, 23.

[TPW*25a] TAM, HOU IN IVAN, PUN, HOU IN DEREK, WANG, AUSTIN T, et al. "SceneEval: Evaluating semantic coherence in text-conditioned 3D indoor scene synthesis". *arXiv:2503.14756* (2025) 7, 21, 23.

[TPW*25b] TAM, HOU IN IVAN, PUN, HOU IN DEREK, WANG, AUSTIN T., et al. "SceneMotifCoder: Example-driven Visual Program Learning for Generating 3D Object Arrangements". *Proceedings of the IEEE Conference on 3D Vision (3DV)*. 2025 5, 17, 18, 22.

[VKE*16] VAN DEN OORD, AARON, KALCHBRENNER, NAL, ESPEHOLT, LASSE, et al. "Conditional image generation with pixelcnn decoders". *Advances in neural information processing systems* 29 (2016) 3.

[VKK16] VAN DEN OORD, AÄRON, KALCHBRENNER, NAL, and KAVUKCUOGLU, KORAY. "Pixel recurrent neural networks". *International Conference on Machine Learning (ICML)*. PMLR. 2016, 1747–1756 3.

[VSP*17] VASWANI, ASHISH, SHAZEER, NOAM, PARMAR, NIKI, et al. "Attention is all you need". *Advances in neural information processing systems* 30 (2017) 3.

[VVN*24] VUONG, AN DINH, VU, MINH NHAT, NGUYEN, TOAN, et al. "Language-driven scene synthesis using multi-conditional diffusion model". *Advances in neural information processing systems* 36 (2024) 22, 23.

[WDP*23] WEI, QIUHONG ANNA, DING, SIJIE, PARK, JEONG JOON, et al. "LEGO-Net: Learning regular rearrangements of objects in rooms". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, 19037–19047 16.

[WIR*25] WU, QIRUI, ILIASH, DENYS, RITCHIE, DANIEL, et al. "Diorama: Unleashing Zero-shot Single-view 3D Scene Modeling". *Proc. of International Conference on Computer Vision (ICCV)*. 2025. DOI: 10.48550/arXiv.2411.19492 9, 19, 20.

[WLD*18] WEISS, TOMER, LITTENEKER, ALAN, DUNCAN, NOAH, et al. "Fast and scalable position-based layout synthesis". *Transactions on Visualization and Computer Graphics (TVCG)* 25.12 (2018), 3231–3243 12, 13, 21.

[WLLS22] WEI, XINYUE, LIU, MINGHUA, LING, ZHAN, and SU, HAO. "Approximate convex decomposition for 3D meshes with collision-aware concavity and tree search". *ACM Transactions on Graphics (TOG)* 41.4 (2022), 1–18 9.

[WLW*19] WANG, KAI, LIN, YU-AN, WEISSMANN, BEN, et al. "PlanIT: Planning and instantiating indoor scenes with relation graph and spatial prior networks". *ACM Transactions on Graphics (TOG)* 38.4 (2019), 1–15 4–6, 15, 16, 21–24.

[WLY*24] WU, ZHENNAN, LI, YANG, YAN, HAN, et al. "BlockFusion: Expandable 3D scene generation using latent tri-plane extrapolation". *ACM Transactions on Graphics (TOG)* 43.4 (2024), 1–17 11.

[WMV*25] WEI, YAO, MIN, MARTIN RENQIANG, VOSSELMAN, GEORGE, et al. "Planner3D: LLM-enhanced graph prior meets 3D indoor scene explicit regularization". *IEEE transactions on pattern analysis and machine intelligence* (2025) 17, 23.

[WQL*24] WANG, YIAN, QIU, XIAOWEN, LIU, JIAGENG, et al. "Architect: Generating vivid and interactive 3D scenes with hierarchical 2D inpainting". *Advances in Neural Information Processing Systems* 37 (2024), 67575–67603 5, 7, 8, 19–23.

[WSCR18] WANG, KAI, SAVVA, MANOLIS, CHANG, ANGEL X, and RITCHIE, DANIEL. "Deep convolutional priors for indoor scene synthesis". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 1–14 5, 6, 9, 15, 16, 21, 22, 24.

[WXC*24] WANG, YUFEI, XIAN, ZHOU, CHEN, FENG, et al. "RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation". *International Conference on Machine Learning (ICML)*. 2024, 51936–51983 9, 17, 18, 21, 23.

[WXC*25] WEN, BEICHEN, XIE, HAOZHE, CHEN, ZHAOXI, et al. "3D Scene Generation: A Survey". *arXiv:2505.05474* (2025) 2.

[WYN21] WANG, XINPENG, YESHWANTH, CHANDAN, and NIESSNER, MATTHIAS. "SceneFormer: Indoor scene generation with transformers". *Proc. of International Conference on 3D Vision (3DV)*. IEEE. 2021, 106–115 4, 10, 15, 16, 21, 22, 24.

[WZC*24] WANG, CAN, ZHONG, HONGLIANG, CHAI, MENGLEI, et al. "Chat2Layout: Interactive 3D Furniture Layout with a Multimodal LLM". *arXiv:2407.21333* (2024) 9, 17, 18, 21–24.

[WZJ*23] WANG, WEIQI, ZHAO, ZIHANG, JIAO, ZIYUAN, et al. "Rearrange indoor scenes for human-robot co-activity". *International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, 11943–11949 12, 13.

[XCF*13] XU, KUN, CHEN, KANG, FU, HONGBO, et al. "Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–15 14, 15, 22.

[XCHL24] XIE, HAOZHE, CHEN, ZHAOXI, HONG, FANGZHOU, and LIU, ZIWEI. "CityDreamer: Compositional generative model of unbounded 3D cities". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, 9666–9675 11.

[XCHL25] XIE, HAOZHE, CHEN, ZHAOXI, HONG, FANGZHOU, and LIU, ZIWEI. "Generative Gaussian splatting for unbounded 3D city generation". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 6111–6120 11.

[XHH*23] XU, RUI, HUI, LE, HAN, YUEHUI, et al. "Scene graph masked variational autoencoders for 3D scene generation". *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, 5725–5733 21, 23, 24.

[XLW*23] XU, JIAZHENG, LIU, XIAO, WU, YUCHEN, et al. "ImageReward: Learning and evaluating human preferences for text-to-image generation". *Advances in neural information processing systems* 36 (2023), 15903–15935 21, 24.

[XSF02] XU, KEN, STEWART, JAMES, and FIUME, EUGENE. "Constraint-based automatic placement for scene composition". *Graphics Interface*. Vol. 2. Citeseer. 2002, 25–34 9, 12, 13.

[XZL*24] XIA, XIAO, ZHANG, DAN, LIAO, ZIBO, et al. "SceneGenAgent: Precise industrial scene generation with coding agent". *arXiv preprint arXiv:2410.21909* (2024) 17, 22.

[YCC*24] YAO, ZHIHAN, CHEN, YUHANG, CUI, JIAHAO, et al. "Conditional room layout generation based on graph neural networks". *Computers & Graphics* 122 (2024), 103971 5, 21.

[YDH*24] YU, HONG-XING, DUAN, HAOYI, HUR, JUNHWA, et al. "WonderJourney: Going from anywhere to everywhere". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 6658–6667 7, 11.

[YDH*25] YU, HONG-XING, DUAN, HAOYI, HERRMANN, CHARLES, et al. "Wonderworld: Interactive 3D scene generation from a single image". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 5916–5926 7, 11.

[YGZT21] YANG, MING-JIA, GUO, YU-XIAO, ZHOU, BIN, and TONG, XIN. "Indoor scene generation from a collection of semantic-segmented depth images". *Proc. of International Conference on Computer Vision (ICCV)*. 2021, 15203–15212 6, 15, 16, 21, 23, 24.

[YHT*23] YI, HONGWEI, HUANG, CHUN-HAO P, TRIPATHI, SHASHANK, et al. "MIME: Human-aware 3D scene generation". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, 12965–12976 4, 10, 16, 21, 23.

[YJZH24] YANG, YANDAN, JIA, BAOXIONG, ZHI, PEIYUAN, and HUANG, SIYUAN. "PhyScene: Physically interactable 3D scene synthesis for embodied ai". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, 16262–16272 4, 7, 8, 16, 17, 21, 23, 24.

[YLD*25] YANG, YIXUAN, LUO, ZHEN, DING, TONGSHENG, et al. "LLM-driven Indoor Scene Layout Generation via Scaled Human-aligned Data Synthesis and Multi-Stage Preference Optimization". *arXiv preprint arXiv:2506.07570* (2025) 18.

[YLH*23] YAN, KAI, LUAN, FUJUN, HAŠAN, MILOŠ, et al. "PSDR-room: Single photo to scene using differentiable rendering". *SIGGRAPH Asia Conference Papers*. 2023, 1–11 20.

[YLW*24] YAN, HAN, LI, YANG, WU, ZHENNAN, et al. "Frankenstein: Generating semantic-compositional 3D scenes in one tri-plane". *SIGGRAPH Asia 2024 Conference Papers*. 2024, 1–11 24.

[YLZ*24] YANG, YIXUAN, LU, JUNRU, ZHAO, ZIXIANG, et al. "LLplace: The 3D Indoor Scene Layout Generation and Editing via Large Language Model". *arXiv:2406.03866* (2024) 17, 18, 22.

[YRY*23] YENAMANDRA, SRIRAM, RAMACHANDRAN, ARUN, YADAV, KARMESH, et al. "HomeRobot: Open Vocabulary Mobile Manipulation". *arXiv preprint arXiv:2306.11565* (2023) 9.

[YSW*24] YANG, YUE, SUN, FAN-YUN, WEIHS, LUCA, et al. "Holodeck: Language guided generation of 3D embodied AI environments". *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, 16227–16237 4, 6–10, 17, 18, 21, 23.

[YWL*22] YE, SIFAN, WANG, YIXING, LI, JIAMAN, et al. "Scene synthesis from human motion". *SIGGRAPH Asia 2022 Conference Papers*. 2022, 1–9 4, 21, 23.

[YYT*11] YU, LAP FAI, YEUNG, SAI KIT, TANG, CHI KEUNG, et al. "Make it Home: Automatic Optimization of Furniture Arrangement". *ACM SIGGRAPH Asia Conference Proceedings* 30.4 (2011) 8, 14, 15, 22.

[YYT15] YU, LAP-FAI, YEUNG, SAI-KIT, and TERZOPOULOS, DEMETRI. "The Clutterpalette: An interactive tool for detailing indoor scenes". *Transactions on Visualization and Computer Graphics (TVCG)* 22.2 (2015), 1138–1148 14, 15.

[YYW*12] YEH, YI-TING, YANG, LINGFENG, WATSON, MATTHEW, et al. "Synthesizing open worlds with constraints using locally annealed reversible jump mcmc". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–11 8, 12, 13.

[YZLP24] YE, ZHAODA, ZHENG, XINHAN, LIU, YANG, and PENG, YUXIN. "RelScene: A Benchmark and baseline for Spatial Relations in text-driven 3D Scene Generation". *Proceedings of the 32nd ACM International Conference on Multimedia*. 2024, 10563–10571 23.

[YZY*25] YAO, KAIXIN, ZHANG, LONGWEN, YAN, XINHAO, et al. "CAST: Component-aligned 3D scene reconstruction from an RGB image". *ACM Transactions on Graphics (TOG)* 44.4 (2025), 1–19 19, 20.

[ZCC*21] ZHANG, CHENG, CUI, ZHAOPENG, CHEN, CAI, et al. "DeepPanoContext: Panoramic 3D scene understanding with holistic scene context graph and relation-based optimization". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 12632–12641 4, 20.

[ZCZ*21] ZHANG, CHENG, CUI, ZHAOPENG, ZHANG, YINDA, et al. "Holistic 3D scene understanding from a single image with implicit representation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 8833–8842 20.

[ZHG*16] ZHAO, XI, HU, RUIZHEN, GUERRERO, PAUL, et al. "Relationship templates for creating scene variations". *ACM Transactions on Graphics (TOG)* 35.6 (2016), 1–13 13, 22.

[ZHL*19] ZHANG, SUIYUN, HAN, ZHIZHONG, LAI, YU-KUN, et al. "Active arrangement of small objects in 3D indoor scenes". *IEEE transactions on visualization and computer graphics* 27.4 (2019), 2250–2264 14, 21, 24.

[ZHX*24] ZHANG, YUNFAN, HUANG, HONG, XIONG, ZHIWEI, et al. "Style-Consistent 3D Indoor Scene Synthesis with Decoupled Objects". *arXiv preprint arXiv:2401.13203* (2024) 4, 21, 23, 24.

[ZLH*21] ZHANG, SHAO-KUI, LI, YI-XIAO, HE, YU, et al. "MageAdd: Real-time interaction simulation for scene synthesis". *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, 965–973 14.

[ZLH24] ZHOU, JUNSHENG, LIU, YU-SHEN, and HAN, ZHIZHONG. "Zero-shot scene reconstruction from single images with deep prior assembly". *Advances in Neural Information Processing Systems* 37 (2024), 39104–39127 19, 20.

[ZLJ*21] ZHAO, YIZHOU, LIN, KAIXIANG, JIA, ZHIWEI, et al. "Luminous: Indoor scene generation for embodied AI challenges". *arXiv:2111.05527* (2021) 7, 8, 10, 12, 21, 24.

[ZLL*23] ZHANG, SHAO-KUI, LIU, JIA-HONG, LI, YIKE, et al. "Automatic generation of commercial scenes". *Proceedings of the 31st ACM international conference on multimedia*. 2023, 1137–1147 12.

[ZÖC*24] ZHAI, GUANGYAO, ÖRNEK, EVIN PINAR, CHEN, DAVE ZHENYU, et al. "Echoscene: Indoor scene generation via information echo over scene graph diffusion". *Proc. of European Conference on Computer Vision (ECCV)*. Springer. 2024, 167–184 21, 23.

[ZÖW*23] ZHAI, GUANGYAO, ÖRNEK, EVIN PINAR, WU, SHUN-CHENG, et al. "CommonScenes: Generating commonsense 3D indoor scenes with scene graphs". *Advances in neural information processing systems* 36 (2023), 30026–30038 7, 8, 10, 15, 16, 21–23.

[ZSS*25] ZOOK, ALEX, SUN, FAN-YUN, SPJUT, JOSEF, et al. "GRS: Generating robotic simulation tasks from real-world images". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 594–603 20.

[ZTL*23] ZHANG, SHAO-KUI, TAM, HOU, LI, YIKE, et al. "SceneDirector: Interactive scene synthesis by simultaneously editing multiple objects in real-time". *Transactions on Visualization and Computer Graphics (TVCG)* 30.8 (2023), 4558–4569 14.

[ZWK19] ZHOU, YANG, WHILE, ZACHARY, and KALOGERAKIS, EVANGELOS. "Scenegraphnet: Neural message passing for 3D indoor scene augmentation". *Proc. of International Conference on Computer Vision (ICCV)*. 2019, 7384–7392 21, 22.

[ZWWZ25] ZHOU, MENGQI, WANG, XIPENG, WANG, YUXI, and ZHANG, ZHAOXIANG. "RoomCraft: Controllable and Complete 3D Indoor Scene Generation". *arXiv preprint arXiv:2506.22291* (2025) 4, 17, 18, 21, 23, 24.

[ZYM*20] ZHANG, ZAIWEI, YANG, ZHENPEI, MA, CHONGYANG, et al. "Deep generative modeling for scene synthesis via hybrid representations". *ACM Transactions on Graphics (TOG)* 39.2 (2020), 1–21 4–6, 15, 16, 24.

[ZZG*25] ZHENG, KAIZHI, ZHANG, RUIJIAN, GU, JING, et al. "Constructing a 3D Town from a Single Image". *arXiv preprint arXiv:2505.15765* (2025) 11.

[ZZL*24] ZHAO, YIQUN, ZHAO, ZIBO, LI, JING, et al. "RoomDesigner: Encoding anchor-latents for style-consistent and shape-compatible indoor scene generation". *Proc. of International Conference on 3D Vision (3DV)*. IEEE. 2024, 1413–1423 21, 23, 24.

[ZZX*20] ZHANG, SONG-HAI, ZHANG, SHAO-KUI, XIE, WEI-YU, et al. "Fast 3D indoor scene synthesis with discrete and exact layout pattern extraction". *arXiv preprint arXiv:2002.00328* (2020) 14.

[ZZX*21] ZHANG, SONG-HAI, ZHANG, SHAO-KUI, XIE, WEI-YU, et al. "Fast 3D indoor scene synthesis by learning spatial relation priors of objects". *Transactions on Visualization and Computer Graphics (TVCG)* 28.9 (2021), 3082–3092 14.