Comput. Aided Geom. Des. 123 (2025) 102489

Contents lists available at ScienceDirect

Computer Aided Geometric Design

journal homepage: www.elsevier.com/locate/cagd

SceneFlow: Synthesizing indoor scenes via geometry-enhanced flow matching

Wenming Wu [a,b], Akang Shen [a], Yanzhe Yin [a], Zixiang Chen [a], Gaofeng Zhang [a], Liping Zheng [a,*]

[a] Hefei University of Technology, Hefei, 230601, Anhui, PR China
[b] Anhui Province Key Laboratory of Industry Safety and Emergency Technology (Hefei University of Technology), Hefei, 230601, Anhui, PR China

ARTICLE INFO

Editor: W. Wang

Keywords:
Indoor scene synthesis
Flow matching
Neural network

ABSTRACT

Recent autoregressive and diffusion models are used to generate indoor scenes. However, they face challenges in maintaining geometric consistency and achieving real-time performance. Autoregressive models struggle with global dependencies, leading to functional inconsistencies, while diffusion-based methods suffer from slow sampling speeds. Motivated by simpler and smoother sampling trajectories, we propose SceneFlow, a novel indoor scene synthesis framework based on Flow Matching, which learns a flow to gradually move objects from an initial chaotic state to a structured layout. To further enhance geometric consistency, we introduce a geometry enhancement strategy comprising a non-overlap constraint to reduce object overlap and a geometry refinement to align objects with room boundaries. We evaluate our approach using the 3D-FRONT dataset. Extensive evaluations demonstrate that our method outperforms state-of-the-art methods in visual quality, geometric accuracy, and generation efficiency. Our method provides a practical solution for applications requiring high-quality indoor scene generation with real-time responsiveness.

1. Introduction

Indoor scene synthesis is to arrange objects within a given interior space to generate 3D scenes that meet functional requirements, geometric constraints, and aesthetic principles. High-quality indoor scene generation holds significant value in video games, virtual reality, and architectural design. Over the years, researchers have explored various data-driven methods Fisher et al. (2012); Kermani et al. (2016); Zhang et al. (2021c,a) to automate indoor scene synthesis. Thanks to the rapid development of deep learning technology, substantial progress has been made. Despite significant breakthroughs, a noticeable *reality gap* still exists between synthesized scenes and the realistic. One of the key challenges is to directly synthesize plausible and realistic indoor scenes without relying on manual or heuristic post-processing. This includes resolving geometric issues such as overlaps, out-of-bounds, and misalignments. Additionally, diffusion-based methods typically exhibit low generation efficiency, particularly as scene complexity and scale increase.

Current generative models for indoor scene synthesis can be broadly categorized into autoregressive and non-autoregressive. Autoregressive models Wang et al. (2021); Paschalidou et al. (2021); Min et al. (2024); Sun et al. (2025) typically decompose

[*] Corresponding author.
E-mail addresses: wwming@hfut.edu.cn (W. Wu), g.zhang@hfut.edu.cn (G. Zhang), zhenglp@hfut.edu.cn (L. Zheng).

https://doi.org/10.1016/j.cagd.2025.102489
Received 23 March 2025; Received in revised form 15 October 2025; Accepted 3 November 2025

Available online 7 November 2025

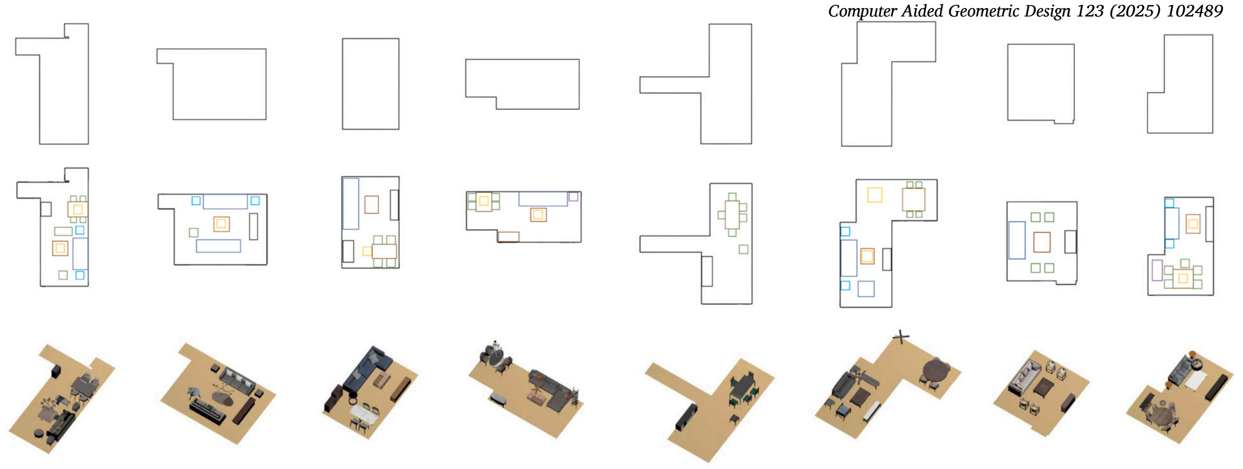


Fig. 1. Indoor scenes generated by SceneFlow. It is a flow-based generative model for indoor scene synthesis that efficiently generates high-quality, geometrically consistent layouts conditioned on an input floorplan, incorporating a geometry enhancement strategy to improve object placement and aesthetic quality. **Top**: input room floorplan; **Middle**: Sketch of the generated scene; **Bottom**: Rendering of the generated scene.

the scene synthesis task into a conditional probability generation process. The model incrementally inserts objects step by step to complete the layout generation. While this iterative process ensures local rationality in object placement, it struggles to model global dependencies, resulting in functional inconsistencies in the layout and difficulties in forming cohesive functional groupings. Furthermore, since the model relies on a locally optimal placement strategy, it is prone to errors such as cross-boundary violations. In contrast, non-autoregressive models Zhang et al. (2020); Gao et al. (2023); Chattopadhyay et al. (2023) use the generative model (e.g., VAE Kingma (2013)) for direct indoor scene synthesis. Recently, diffusion-based generative models Tang et al. (2024a); Hu et al. (2024); Yang et al. (2024) have become the mainstream approach for indoor scene synthesis. However, their inference process requires hundreds to thousands of complicated iterative denoising steps, resulting in long inference times. This limitation poses a challenge for interactive design applications that demand real-time responsiveness. Moreover, the generation path of diffusion models is heavily influenced by noise perturbations. As a result, misalignment, overlapping, and layout inconsistency may occur, violating local geometric constraints and ultimately reducing the functional and aesthetic quality of the generated scenes.

In this paper, we propose a novel indoor scene synthesis framework called SceneFlow based on Flow Matching Lipman et al. (2023), aiming to learn a flow to map samples from an input distribution to the target distribution. However, diffusion models are generally regarded as models that can learn to remove the random noise added to the training data gradually. Their generation process can also be explained by referring to flow-based models, that is, moving the initial distribution to the target distribution. The difference is that diffusion models introduce additional randomness by adding noise during the generation process. This corresponds to a Stochastic Differential Equation (SDE) in mathematics, while flow-based models are defined by a simpler Ordinary Differential Equation (ODE). In comparison to diffusion models, flow-based models directly learn a continuous transformation. This enables SceneFlow to achieve a smooth transformation from the initial state to the final layout. The path of SceneFlow is thereby shorter, smoother, and more efficient. As a result, SceneFlow requires efficient sampling steps and achieves faster inference while maintaining high generation quality. To further improve the geometric and aesthetic quality of generated scenes, we introduce a geometry enhancement module in the SceneFlow framework. This module includes two key components: (1) Non-overlap constraint, aiming to control the “degree of overlap” between objects, thereby improving the functional rationality and aesthetic quality of the generated scene. (2) Geometry refinement, ensuring that the objects are neatly arranged and well-aligned with room boundaries, significantly reducing cases where indoor objects extend beyond the room’s boundary. Technically, the non-overlap constraint is implemented as a differentiable loss term that penalizes the area of intersection between predicted object bounding boxes, encouraging the model to position objects without unreasonable overlaps. The geometry refinement module adopts a Transformer-based correction network, which takes the initially predicted geometric attributes (position, size, orientation) as input and iteratively adjusts these attributes to ensure boundary adherence and spatial alignment.

SceneFlow is a novel flow-based generative model for indoor scene synthesis, capable of generating high-quality indoor scenes conditioned by a floorplan, as shown in Fig. 1. The effectiveness of SceneFlow has been verified through a series of scene synthesis tasks. Compared with state-of-the-art methods, SceneFlow demonstrates superior performance in multiple metrics, including generation quality, sampling time, and geometric consistency. Experimental results show that SceneFlow exhibits significant advantages in visual aesthetics and geometric consistency. Our contributions are as follows:

- We propose a flow-based indoor scene synthesis framework. Unlike traditional autoregressive and diffusion-based generative methods, SceneFlow achieves faster generation and significantly improves the functional rationality and visual aesthetics of the generated scenes.
- We propose a geometry enhancement strategy, which incorporates the non-overlap constraint and the geometry refinement. This strategy ensures the geometric regularity, rationality, and consistency of object placement in indoor scenes.

2. Related work

This paper focuses primarily on deep learning-based methods for indoor scene synthesis. For a comprehensive review of other technologies, the reader is referred to a recent survey Patil et al. (2024).

2.1. *Autoregressive scene synthesis*

Autoregressive scene synthesis involves sequentially generating each object in the scene, with the generation process modeled as a conditional probability. This approach utilizes previously generated objects to guide the synthesis of subsequent objects. Most existing methods for indoor scene synthesis Ritchie et al. (2019); Wang et al. (2019, 2021); Paschalidou et al. (2021); Zhang et al. (2021b); Leimer et al. (2022); Para et al. (2023); Li et al. (2024); Min et al. (2024); Sun et al. (2025) adopt this strategy. SceneFormer Wang et al. (2021) is a Transformer-based framework for indoor scene synthesis that represents the scene as a sequence of objects. It generates objects autoregressively, predicting each object's properties based on previously generated ones. Similarly, ATISS Paschalidou et al. (2021) also employs an autoregressive Transformer Vaswani et al. (2017) to generate unordered sets of objects in indoor scenes and progressively synthesize the properties of objects. LayoutEnhancer Leimer et al. (2022) integrates expert knowledge into a Transformer-based generative model, synthesizing indoor scenes from imperfect data. In contrast, our method does not require any expert knowledge. Recently, Forest2Seq Sun et al. (2025) uses a tree-to-sequence approach for indoor scene synthesis, organizing indoor objects into hierarchical scene forests, which guides an autoregressive Transformer for order-aware scene generation. Autoregressive indoor scene synthesis often struggles to capture global dependencies. In contrast, our flow-based framework adopts a non-autoregressive approach, enabling more comprehensive and efficient synthesis from the local to the global.

2.2. *Non-autoregressive scene synthesis*

Non-autoregressive scene synthesis generates all objects in the scene simultaneously Zhang et al. (2020); Luo et al. (2020); Purkait et al. (2020); Gao et al. (2023); Chattopadhyay et al. (2023); Wei et al. (2023); Song et al. (2024). Zhang et al. Zhang et al. (2020) uses a feed-forward neural network to generate indoor scenes by mapping latent space inputs to hybrid representations that combine object arrangements and image-based features. Luo et al. Luo et al. (2020) employs a conditional VAE to generate indoor scenes from scene graphs, integrating a differentiable renderer for layout refinement using 2D projections like depth and semantic maps. SceneHGN Gao et al. (2023) uses a hierarchical graph network with a recursive VAE to jointly generate indoor scenes, incorporating room layouts, functional regions, object arrangements, and fine-grained object geometry in a unified framework. Existing non-autoregressive scene synthesis methods face challenges in maintaining local coherence among objects, often resulting in disjointed or unnatural layouts with issues such as misalignment, boundary violations, and object overlap. In contrast, our method achieves superior local consistency by leveraging a flow-based framework, enabling more cohesive and geometrically constrained scene generation.

2.3. *Diffusion-based scene synthesis*

Diffusion models Ho et al. (2020) are a class of deep generative models that synthesize complex data by gradually denoising a random noise input. In recent years, diffusion models have gained prominence in indoor scene synthesis Tang et al. (2024a); Zhai et al. (2024); Hu et al. (2024); Yang et al. (2024); Zhai et al. (2025); Maillard et al. (2024); Meng et al. (2024). DiffuScene Tang et al. (2024a) employs a diffusion model to generate indoor scenes by iteratively refining unordered object attributes, including location, size, orientation, and geometry. MiDiffusion Hu et al. (2024) employs a mixed discrete-continuous diffusion model to synthesize indoor scenes, iteratively denoising object semantics and geometric attributes. PhyScene Yang et al. (2024) utilizes a guided diffusion model to synthesize indoor scenes by iteratively denoising layouts, incorporating physics-based guidance to ensure the generated scenes are physically plausible and interactive. However, these methods suffer from slow inference times due to their complicated denoising process, making them less suitable for real-time applications. Additionally, their denoising process can lead to misalignments, overlaps, and layout inconsistencies. In contrast, our method offers faster, more efficient generation by using a transformation, significantly reducing the sampling time and improving geometric consistency.

2.4. *Flow-based generative model*

Flow-based Kobyzev et al. (2020); Liu et al. (2022); Albergo and Vanden-Eijnden (2023); Lipman et al. (2023); Tong et al. (2023) and diffusion-based models both transform data distributions, but diffusion-based models denoise from noise to data space through multistep generation, while flow-based models employ a more efficient sampling method, making them better suited for real-time generation. The most related work to ours is LayoutFlow Guerreiro et al. (2025), which uses a Flow Matching framework to generate 2D (e.g., document) layouts. While our method shares conceptual similarities with LayoutFlow, our method addresses the significantly more challenging task of 3D indoor scene synthesis, which involves more complex spatial constraints, functional requirements, and aesthetic considerations compared to 2D document layouts. To further strengthen geometric consistency and collision avoidance—challenges less critical in 2D layout tasks—we introduce a geometry enhancement strategy that directly learns geometric relationships from data. As demonstrated by the ablation studies in Section 5.4, this learned geometry enhancement surpasses traditional heuristic approaches, significantly enhancing the realism and functional coherence of generated scenes. Therefore, our contributions extend meaningfully beyond LayoutFlow, integrating advanced geometric reasoning within a sophisticated generative framework.

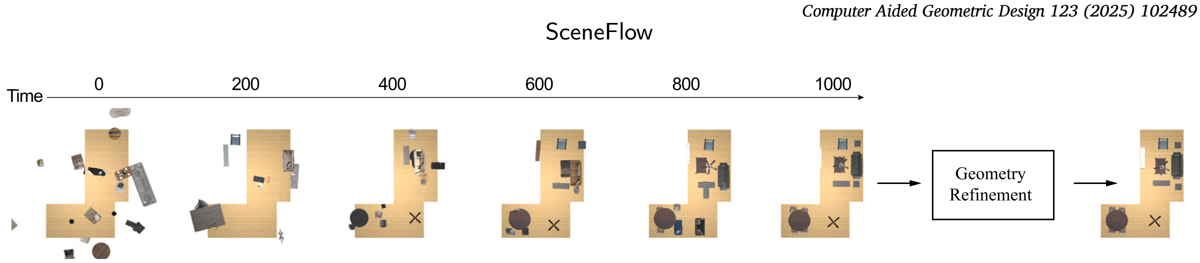


Fig. 2. Overview of SceneFlow for indoor scene synthesis. Starting from an initial chaotic scene at $t = 0$, objects are progressively refined over time through a flow-based generation process, ensuring improved spatial arrangement by reducing overlaps and boundary violations. A Geometry Refinement module is then applied to fine-tune object positions, further enhancing alignment and geometric consistency, resulting in a realistic and well-structured indoor scene.

3. Preliminary

Flow Matching Lipman et al. (2023) achieves the transformation between probability distributions, which is essential for generation tasks. At the core of Flow Matching is the concept of "flow", which defines the mapping between two distributions. Let $p_0(x)$ represent a known source distribution (e.g., Gaussian distribution) and $p_1(x)$ be a more complex target distribution, both d-dimensional. For a data point $x \in \mathbb{R}^d$ from the source distribution. The flow $\phi : [0,1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is defined by the Ordinary Differential Equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\phi_t(x)) = v_t(\phi_t(x)), \quad \phi_0(x) = x_0 \tag{1}$$

Here, v_t is a time-dependent vector field that constructs the flow. The flow $\phi_t(x)$ describes how the initial sample x is transported as time progresses. A simple initial distribution can be transformed into a more complex distribution through the push-forward equation:

$$p_t = \left[\phi_t\right]_* p_0 \tag{2}$$

the push-forward operator $*$ is defined by:

$$\left[\phi_t\right]_* p_0 = p_0(\phi_t^{-1}(x)) \det\left(\frac{\mathrm{d}\phi_t^{-1}}{\mathrm{d}t}(x)\right) \tag{3}$$

If the flow ϕ_t satisfies the Eq. (2), the vector field v_t is said to generate the probability path p_t.

Flow Matching uses neural networks to predict a vector field $\overline{v_t}(x)$, and the loss function $\mathcal{L}_{FM}$ is given by the expectation over the uniformly sampled time t and the data point x along the probability path p_t:

$$\mathcal{L}_{FM} = \mathbb{E}_{t\sim\mathcal{U}(0,1),x\sim p_t(x)} \left\|\overline{v_t}(x) - v_t(x)\right\|^2 \tag{4}$$

However, since p_t and v_t are generally not available, Lipman et al. (2023) introduces the CFM (Conditional Flow Matching). The CFM loss function $\mathcal{L}_{CFM}$ is defined as:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t\sim\mathcal{U}(0,1),z\sim q(z),x\sim p_t(x|z)} \left\|\overline{v_t}(x) - v_t(x|z)\right\|^2 \tag{5}$$

Here, $v_t(x|z)$ is the conditional vector field generating the conditional probability path $p_t(x|z)$. $q(z)$ is the distribution over the latent variable z.

By introducing CFM, the previously inaccessible vector field $v_t(x)$ is replaced by a conditional vector field $v_t(x|z)$. This enables training a neural network using conditional vector fields and their associated probability paths. Optimizing CFM achieves the same effect as optimizing the original Flow Matching, making it feasible to design and optimize a suitable conditional probability path and vector field.

4. Method

4.1. Overview

We propose SceneFlow, a generative model based on Flow Matching, designed to generate 3D indoor scene distributions. Given a floorplan of an empty room (e.g., a living room), our method aims to populate the room with objects while ensuring functional coherence and aesthetic spatial arrangement. The overview of our method is shown in Fig. 2.

Scene representation The indoor scene is defined in a world coordinate system, with the origin at the center of the room floor. The ground plane is the XZ plane, and the Y-axis points upward. All object positions and dimensions are normalized to a standardized unit cube. In our experiments, the actual scene space (ranging from 0–3.5 meters for bedrooms and 0–6 meters for living and dining rooms) is linearly mapped to the normalized coordinate range of $[-1, 1]$. Each indoor scene $\mathcal{S}$ is composed of its floorplan $\mathcal{F}$ and a set

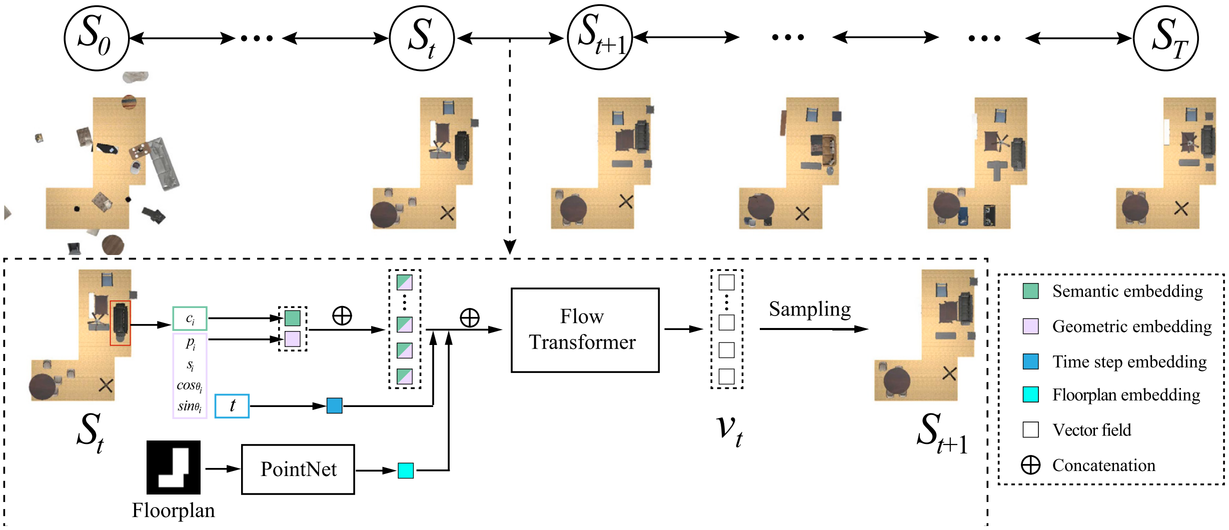
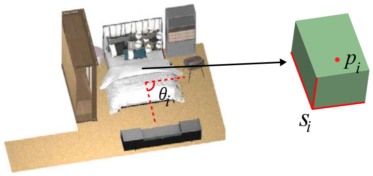


Fig. 3. Our method generates realistic 3D indoor scenes by learning a continuous flow from an initial chaotic state to a structured layout. Starting with an initial scene sampled from a Gaussian distribution, the model constructs a flow path over time, where intermediate samples are obtained by interpolating between the initial scene and the ground truth. A Transformer-based network predicts the conditional vector field at each time step, guiding objects toward their optimal positions. During inference, the scene iteratively evolves along the predicted vector field, ensuring that objects are realistically arranged within spatial boundaries. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

of indoor objects $\mathcal{O} = \{o_1, o_2, ..., o_N \mid \mathcal{F}\}$, N is the maximum number of objects in the scene. We represent the floorplan $\mathcal{F}$ as a grayscale map with dimensions of 256×256, where the pixel values within the interior region of the room are set to 1, while the exterior region is set to 0. Indoor object o_i has the semantic attribute, i.e., object category label c_i and the geometric attributes $g_i = [p_i, s_i, \theta_i]$, where $p_i \in R^3$ represents the central position of the object, $s_i \in R^3$ represents the bounding box size of the object, and θ_i represents the placement angle of the object, as shown in the right inset. Following Tang et al. (2024b), we describe the angle as a two-dimensional vector composed of both the sine and cosine values. Ultimately, each object o_i can be described by concatenating all its attributes together, *i.e.*, $o_i = [c_i, p_i, s_i, cos\theta_i, sin\theta_i] \in R^D$, where D is the dimensions of all attributes. An “empty” label is added to accommodate scenes with fewer than N objects, and the geometric attribute is defined as zero for any “empty” object, ensuring a consistent representation.



Methodology To tackle the challenges proposed in the introduction chapter, our method focuses on two aspects. First, we introduce a geometry-enhanced generation strategy that directly incorporates spatial constraints into the generation process, minimizing the need for post-processing. By designing a model capable of capturing and optimizing geometric relationships, we ensure that the generated indoor scenes naturally avoid common issues such as overlaps and misalignment, thereby improving the usability and realism of the scenes. Second, we employ a flow-based method, leveraging more efficient sampling and optimization techniques to accelerate the generation process. This significantly improves the efficiency compared to traditional diffusion-based methods, especially when handling large-scale and complex scenes. The primary contribution of our work is introducing a flow-based generative framework for efficient and plausible 3D indoor scene synthesis. On the other hand, our key novelty lies in integrating the geometry enhancement strategy within the generative framework. This integration enables SceneFlow to learn smoother and more efficient transformation paths, directly and significantly improving geometric consistency.

4.2. Flow-based scene synthesis

The framework of our flow-based scene synthesis is shown in Fig. 3.

Attribute & floorplan encoding We separately encode the semantic and geometric attributes of the objects within the scene. Both the semantic encoder and the geometric encoder are constructed using Multi-Layer Perceptrons (MLPs) Hornik (1991), which output the semantic and geometric attributes of the objects as 512-dimensional vectors, respectively. These encoders process the scene’s semantic information, such as object types and relations, and geometric features, including spatial configurations and sizes. After encoding, the resulting semantic and geometric attributes are concatenated into a unified attribute embedding. Following LEGO-Net Wei et al. (2023), we utilize PointNet Qi et al. (2017) to encode the input floorplan. Specifically, we uniformly sample 250 points from the floorplan boundary and input these points into PointNet for processing. Each point is characterized by (1) its 2D position (x, z) on the normalized XZ plane and (2) a 2D unit normal vector representing the outward direction. These points form a 250×4 matrix, which is fed into a PointNet encoder to extract a 512-dimensional feature vector. This approach enables PointNet to effectively capture the geometric structure and boundary information of the floorplan, producing a 512-dimensional embedding that comprehensively preserves the geometric and topological properties of the original layout. Compared to image-based feature encoders, PointNet excels in capturing the geometric features of room boundaries, providing more accurate spatial constraints for the subsequent scene generation tasks.

Non-overlap constraint In a well-designed indoor scene, an object should occupy distinct and appropriate spatial regions. To enforce geometric regularity in the generated layouts, we introduce a non-overlap constraint loss function during the training process. This loss function penalizes collisions between different objects, resulting in a more realistic and credible distribution of objects. We consider pairwise overlaps between objects. We denote the geometric constraint loss as $L_{overlap}$. We first compute the intersection and union of the bounding boxes of two indoor objects. Let V_i and V_j represent the volumes of the bounding boxes of the two indoor objects, o_i and o_j, and $V_{overlap(i,j)}$ denote the volume of their overlap. The union volume can then be calculated as:

$$V_{union(i,j)} = V_i + V_j - V_{overlap(i,j)} \tag{6}$$

The 3D overlap loss between the two objects can be expressed as:

$$L_{overlap} = \sum_{(o_i,o_j)\in S} \frac{V_{overlap}(i,j)}{V_{union}(i,j)} \tag{7}$$

This loss function penalizes the model whenever the overlap between objects exceeds an acceptable threshold, thereby encouraging spatially distinct placements of objects in the scene. By incorporating this loss into the training process, we guide the model toward generating more realistic and non-overlapping object configurations. Note that $\mathcal{L}_{overlap}$ is applied to intermediate predicted samples S_t during training, rather than to the final outputs or target samples. The samples S_t are obtained by linear interpolation: $S_t = (1-t)S_0 + tS_T$, where $t \sim \mathcal{U}(0,1)$, S_0 is sampled from a standard normal distribution (noise), and S_T is the ground-truth layout. By computing pairwise bounding box IoU in real time and backpropagating gradients, this loss directly optimizes the predicted object positions and sizes by the generator. Although $\mathcal{L}_{overlap}$ is not used during inference, the geometric constraints it imposes on the intermediate states S_t during training ensure that the final layouts generated exhibit low overlap.

Training We begin by sampling an initial scene $\mathcal{S}_0 \in \mathbb{R}^{N\times D}$ from a prior distribution p_0, such as a Gaussian distribution. This initial sample is typically very chaotic, with objects overlapping and some objects extending beyond the boundaries. In addition, we randomly select a time t, which determines the intermediate sample $\mathcal{S}_t \in \mathbb{R}^{N\times D}$ along the flow trajectory during training. To construct the flow path, we follow a simple linear trajectory during training. The intermediate sample is computed by linearly interpolating between the initial sample $\mathcal{S}_0$ and the ground truth $\mathcal{S}_T \in \mathbb{R}^{N\times D}$. This interpolation defines the flow of the scene from its chaotic initial state to a more structured final state, aligning the generated scene closer to the target distribution. Thus, the construction process ensures smooth transitions from the initial chaotic scene to the desired scene layout, guiding the scene generation process in a controlled manner. Our generation module consists of an 8-layer Transformer. The inputs include the encoded scene, the time step, and the input floorplan, all of which are embedded. We train the generation module to output the predicted conditional vector field v_t. Simply put, the generation module learns a direction for the data sample, enabling the sample to move toward a better prediction during the sampling process. Given the linear interpolation path:

$$S_t = (1-t)S_0 + tS_T \tag{8}$$

The corresponding conditional vector field is defined as:

$$v_t(S_t \mid S_T) = \frac{dS_t}{dt} = S_T - S_0 \tag{9}$$

The resulting CFM loss for this linear trajectory is:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t\sim U(0,1),\, z\sim q(z),\, x\sim p_t(x|z)} \left\| \overline{v}_t(x) - v_t(x|z) \right\|^2 \tag{10}$$

where $\overline{v}_t(x)$ and $v_t(x|z)$ represent the estimated vector fields for the sample and the conditional input, respectively. This formulation explicitly guides the learning process for flow matching along a linear trajectory. We define the loss function as a combination of Eq. (10) and Eq. (7), with α as the weight:

$$L_{total} = L_{CFM} + \alpha L_{overlap} \tag{11}$$

The first part is computed by the Mean Squared Error (MSE) between the predicted vector field and the ground truth vector field. The second part represents the non-overlap constraint loss.

Testing After training, given a scene $\mathcal{S}_t$, the time step t, and the floorplan $\mathcal{F}$, our model can predict the vector field v_t. In simple terms, this vector field represents the direction pointing to the position of the next time step. During the inference, we sample an initial scene $\mathcal{S}_0$ from a Gaussian distribution and then generate new scenes by solving the Ordinary Differential Equation in Eq. (1) regarding the flow. In practice, for a fixed number of sampling time steps T, the initial scene $\mathcal{S}_0$ iteratively moves along the direction predicted by our model. The generation process can be described as follows:

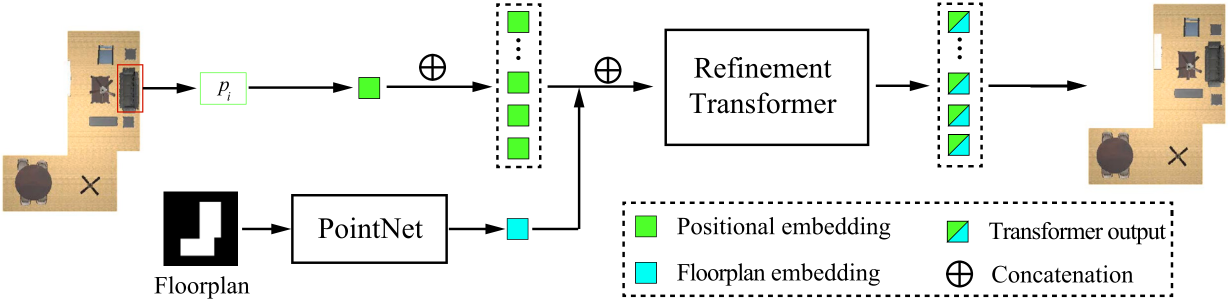


Fig. 4. The geometry refinement model fine-tunes object positions and size to resolve overlaps and boundary violations in generated scenes. Using Gaussian noise to simulate training data, a dedicated Refinement Transformer learns to adjust positions and size by minimizing the deviation from ground truth. Once trained, it refines intermediate results, ensuring objects are realistically placed, overlap-free, and aligned within room boundaries for visually and functionally coherent layouts.

$$\mathcal{S}_{\frac{t+1}{T}} = \mathcal{S}_{\frac{t}{T}} + \frac{1}{T} v_t \tag{12}$$

where $t \in [0, T-1]$, indicating the number of iterations the initial scene $\mathcal{S}_0$ has undergone. During inference, PointNet effectively captures the boundary characteristics of the room layout, while the Transformer-based generation module models global layout dependencies through multi-head attention mechanisms. As a result, SceneFlow consistently generates physically plausible layouts that minimize object overlaps and out-of-boundary violations.

4.3. Geometry refinement

To obtain scenes that closely adhere to the geometric guidelines of real-world interiors, additional geometry enhancement strategies are required. We have proposed the non-overlap constraint, which plays a crucial role in the flow-based scene generation, as mentioned above. While the non-overlap constraint helps reduce the overlap between indoor objects, additional refinements are necessary to achieve more precise and aesthetically pleasing scenes. To this end, after the flow-based generation, we consider performing further geometry refinement on the generated scenes, focusing on adjusting the geometric attributes of the layout. This aims to correct any remaining geometric imperfections, such as objects being positioned outside the input floorplan or unreasonable sizes, and bring the final results closer to the realistic indoor arrangements. This geometry refinement strategy is illustrated in Fig. 4. Specifically, a well-trained fine-tuning model is used to post-process the flow-based generation to further optimize the indoor scene.

Training data construction To effectively simulate the intermediate results produced by the flow-based generative model and facilitate training of the refinement model, we generate noisy scenes by introducing standard Gaussian noise to the ground-truth positions and sizes of indoor objects. The angle attributes are not perturbed because object orientations in these intermediate results are already sufficiently accurate. These noisy scenes serve as our training data. Specifically, given the ground truth object geometric attributes g_{gt}, the noisy data g_{nosiy} is computed as:

$$g_{noisy} = g_{gt} + \lambda N(0, \sigma^2) \tag{13}$$

where λ is a scaling coefficient that controls the noise level, and $N(0, \sigma^2)$ represents standard Gaussian noise with zero mean and variance σ^2. This process introduces controlled perturbations to the ground truth geometric attributes, simulating intermediate results with noise. The noisy data g_{noisy} serves as the training dataset for the Refinement Transformer, enabling it to learn how to refine object geometric attributes and correct geometric inconsistencies effectively.

Transformer-based refinement The generation process involves encoding the geometric attributes of the intermediate scene using a geometric encoder, inputting the encoded result into a Transformer, called the Refinement Transformer, to generate new geometric attributes, and finally decoding the new geometric attributes through a geometric decoder to obtain the final geometric attributes of objects in the scene. The training objective is to minimize the difference between the Transformer prediction and the ground truth.

The training loss for the Refinement Transformer is defined as the mean squared error (MSE) between the predicted and ground truth geometric attributes. For each object o_i in scene $\mathcal{S}$, let $g_i = [p_i, s_i]$ represent the ground truth 3D position p_i and size s_i, and $\overline{g}_i$ denote the corresponding prediction from the refinement module. The loss is formulated as:

$$L_{ref} = \sum_{o_i \in \mathcal{S}} \|\overline{g}_i - g_i\|^2 = \sum_{o_i \in \mathcal{S}} \left\|\overline{p}_i - p_i\right\|^2 + \left\|\overline{s}_i - s_i\right\|^2 \tag{14}$$

Once trained, the Refinement Transformer can be used to refine the intermediate results of our layout generation process. These intermediate results, which may contain suboptimal or noisy geometric attributes, are fed into the Refinement Transformer. The network then outputs adjusted geometric attributes that are more realistic, resulting in a more organized and visually appealing scene.

The geometry refinement, along with its specialized transformer module for geometric adjustment, plays a vital role in improving the quality of generated indoor furniture layouts. By simulating intermediate results with Gaussian noise and training the Refinement

Transformer on this data, we achieve a powerful and effective method for refining object geometric attributes. This approach ensures that the final layouts are not only geometrically accurate but also visually appealing and functional.

5. Experiment and evaluation

5.1. Implementation

We have implemented SceneFlow in PyTorch and used the Adam optimizer (Kingma and Ba, 2015) for training. The model is trained separately on different room types (bedrooms, living rooms, and dining rooms), and all models are trained on an NVIDIA GeForce RTX 3090 with a batch size of 128. The learning rate is initialized to 2e-4 and decays during training: on the bedroom dataset, the learning rate decays every 10,000 epochs by a factor of 0.5, with a total training duration of 50,000 epochs; for the living room and dining room datasets, the decay occurs every 15,000 epochs over a total of 100,000 epochs.

We evaluate our method using the 3D-FRONT Fu et al. (2021a) dataset, which is a large-scale synthetic indoor scene dataset consisting of 6,813 houses and 14,629 rooms, each furnished with high-quality 3D models sourced from the 3D-FUTURE Fu et al. (2021b) dataset. Following ATISS Paschalidou et al. (2021), we train and evaluate our method on three types of room layouts: 4,041 bedrooms, 813 living rooms, and 900 dining rooms. For each room type, 80% of the rooms are used for training and the remaining for testing. The corresponding 3D object is retrieved from the 3D-FUTURE dataset Fu et al. (2021b) by selecting the object within the predicted semantic category whose size most closely matches the predicted dimensions.

5.2. Qualitative evaluation

We compare our method with three state-of-the-art methods:

- **ATISS** Paschalidou et al. (2021): An autoregressive model that uses the Transformer to generate objects sequentially in the scene.
- **DiffuScene** Tang et al. (2024b): A diffusion-based model that follows the DDPM (Denoising Diffusion Probabilistic Model) Ho et al. (2020) framework.
- **MiDiffusion** Hu et al. (2024): A discrete-continuous diffusion-based model that combines the DDPM and D3PM (Discrete Denoising Diffusion Probabilistic Model) Austin et al. (2021).
- **PhyScene** Yang et al. (2024): A framework for synthesizing physically interactive 3D scenes in embodied AI, ensuring realistic object interactions and physical plausibility for agent training and evaluation.

We conduct a qualitative evaluation of the scenes synthesized by different methods. To ensure fairness, we use multiple room boundaries for testing. Given the same room boundaries as inputs, we compare the generated scenes from different methods, as shown in Fig. 5. From the results, in the scenes generated by ATISS (the first column in the figure), the overlapping of furniture is quite severe. For example, in the first room, the cabinet and dining table overlap, and in the fourth room, the single-seat sofa and multi-seat sofa also overlap significantly. Additionally, some furniture pieces exceed the room boundaries, such as the cabinet in the first room. The scenes in the second column are generated by DiffuScene. While the overlapping of furniture improves compared to ATISS, the issue of objects exceeding the boundaries remains quite prominent. For instance, in the first room, both the large and small cabinets, and in the second room, the sofa and cabinet extend beyond the boundaries. The scenes in the third column are generated by MiDiffusion Hu et al. (2024). Compared to DiffuScene, the issue of objects exceeding the boundaries is somewhat mitigated, but there are still cases where furniture exceeds the boundaries, such as the chairs in the first room and the chandelier in the third room. Moreover, the overlapping of objects in these scenes is even more pronounced than in those generated by DiffuScene. The scenes in the fourth column are generated by PhyScene, where the overlapping is less frequent. However, there are still instances of objects exceeding the boundaries, such as the TV stand in the first room and the cabinet in the fourth room. From another perspective, there is a certain connection between the phenomenon of objects exceeding the boundaries and the overlapping of objects. When objects exceed the boundaries, their distribution tends to be more scattered, thus reducing the probability of overlapping among them. In contrast, our method achieves a good balance between controlling the extent to which objects exceed the boundaries and controlling the overlapping of objects, resulting in more realistic and reasonable generated scenes.

5.3. Quantitative evaluation

Distribution evaluation We follow previous works Paschalidou et al. (2021); Tang et al. (2024b); Yang et al. (2024) to evaluate the plausibility and diversity of the synthesized indoor scenes. We utilize the models from 3D-FUTURE to render each scene into a top-down floorplan image of size 256×256. A series of evaluation metrics is employed to comprehensively assess the quality of the generated scenes:

- FID (Fréchet Inception Distance): This metric measures the fidelity of indoor scenes by calculating the distribution distance between the generated and real scene images in the feature space. The lower the FID, the closer the distribution of the generated scenes is to that of the real scenes, indicating better performance of the method.
- KID (Kernel Inception Distance): Similar to FID, KID measures the distribution difference between generated and real images. The key difference is that KID calculates the distance using kernel methods, whereas FID uses the Fréchet distance.

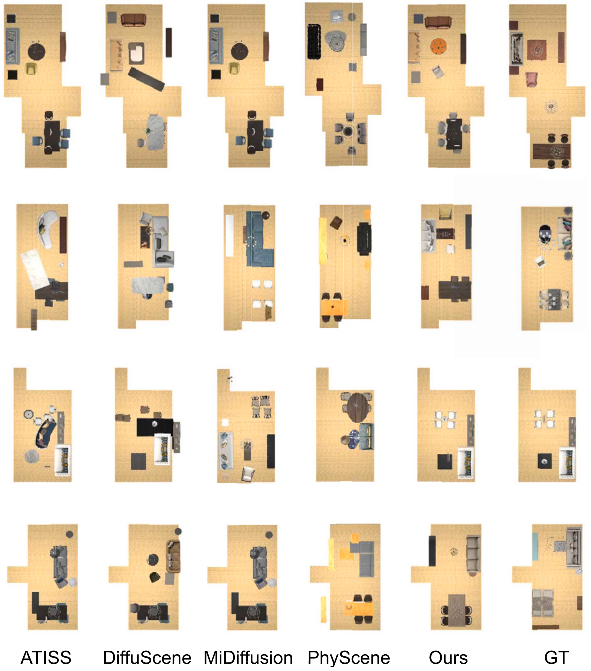


Fig. 5. Qualitative evaluation on the living room generation. We conduct the qualitative evaluation of our method and the baseline methods. From left to right, we show the indoor scenes of ATISS, DiffuScene, MiDiffusion, PhyScene, our method, and the ground truth.

Table 1
Quantitative evaluation on three kinds of room scene generation. We conduct a distribution evaluation of ATISS, DiffuScene, MiDiffusion, and our method using four metrics: FID, KID, CA, and KL. The best results are shown in bold.

Method	Living room				Dining room				Bedroom			
	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)
ATISS	54.34	0.0035	54.24%	0.01861	59.73	0.00316	54.16%	0.02652	85.93	0.00273	58.49%	0.02637
DiffuScene	52.75	0.0032	54.18%	0.01732	57.16	0.00285	55.21%	0.01865	82.68	0.00264	56.79%	0.01683
MiDifusion	50.58	0.0031	53.54%	**0.01693**	53.29	0.00275	52.27%	**0.01702**	81.43	0.00285	54.54%	**0.00884**
PhyScene	50.46	0.0027	54.06%	0.01813	55.28	0.00281	54.16%	0.01950	81.25	0.00254	55.23%	0.01967
SceneFlow (Ours)	**49.94**	**0.0021**	**52.11%**	0.02107	**52.43**	**0.00204**	**52.24%**	0.02442	**80.62**	**0.00246**	**53.81%**	0.02107

- CA (Scene Classification Accuracy): We report the CA over 10 runs of the random sampling of predicted layouts for both training and testing. The closer the CA is to 50%, the better the generated scene.
- KL (Kullback-Leibler Divergence of Category): We use KL to evaluate the difference between the object label distribution of the generated scenes and the real scenes. A smaller KL value indicates a label distribution that is closer to that of the real scenes.

For all similarity metrics above, we use the Inception-v3 network Szegedy et al. (2015), pre-trained on ImageNet, as the feature extractor and classifier. (1) FID/KID feature extraction: The features used to compute FID and KID are taken from the pre-logits layer of the Inception-v3 network (i.e., the layer immediately preceding the final classification layer). (2) CA/KL logit extraction: The logits used for computing CA and KL are taken from the final output layer of the Inception-v3 network. CA is calculated based on the predicted scene class probabilities derived from these logits.

The results are presented in Table 1. ATISS struggles with local decision-making when handling complex scenes, making it difficult to optimize the overall scene layout globally. As a result, ATISS achieves the highest FID and KID values among all compared methods. DiffuScene, on the other hand, directly models the distribution of the entire scene rather than generating objects one by one. This global modeling capability allows diffusion models to better manage object relationships within complex scenes, leading to lower FID and KID values compared to ATISS. PhyScene, which employs a guided diffusion model, further improves performance, achieving lower FID, KID, and CA values compared to both ATISS and DiffuScene. In the comparison of synthesized layout images, our method demonstrates improvements over baseline methods across the FID, KID, and CA metrics for all three room types, indicating that our approach generates more realistic and orderly results. However, regarding the KL metric, our method does not exhibit a significant improvement over other methods. This may be due to the randomness in sampling, as all methods achieve relatively low KL values.

Geometry evaluation We further evaluate the geometry of the generated indoor scenes to compare the reasonableness of object placement across different methods:

Table 2
Quantitative evaluation on three kinds of room scene generation. We conduct a geometry evaluation of ATISS, DiffuScene, MiDiffusion, PhyScene, and our method using three metrics: OOB, OLN, and OBJ. The best results are shown in bold. The ground truth is also shown.

Method	Living room			Dining room			Bedroom		
	OOB (↓)	OLN (↓)	OBJ ($\to GT$)	OOB (↓)	OLN (↓)	OBJ ($\to GT$)	OOB (↓)	OLN (↓)	OBJ ($\to GT$)
ATISS	10.59	5.56	12.03	9.96	5.85	**11.23**	5.95	1.62	4.92
DiffuScene	12.48	**4.53**	11.95	15.21	**4.37**	10.98	6.46	1.57	5.34
MiDiffusion	9.25	5.18	12.22	6.11	4.93	10.91	4.72	1.53	**5.18**
PhyScene	9.64	5.64	12.10	7.06	5.34	10.86	4.58	1.66	5.31
SceneFlow (Ours)	**8.52**	5.43	11.93	**5.49**	5.21	10.63	**4.15**	**1.44**	4.75
Ground Truth	1.51	3.74	11.67	0.73	4.37	11.12	3.37	1.45	5.22

Table 3
Efficiency evaluation on three kinds of room scene generation. We compare the generation time (ms) of ATISS, DiffuScene, MiDiffusion, PhyScene, and our method. The best results are shown in bold.

Method	Living room	Dining room	Bedroom
ATISS	351	343	187
DiffuScene	322	327	293
MiDiffusion	196	195	145
PhyScene	338	340	303
SceneFlow (Flow matching)	26	20	18
SceneFlow (Geometry enhancement)	5	5	4
SceneFlow (Full method)	**31**	**25**	**22**

- Percentage of Out-of-Bounds Objects (OOB): This represents the proportion of objects that exceed the boundaries of the scene. The lower the OOB (%), the fewer objects that exceed the boundaries in the generated scene.
- Overlapping Objects Number (OLN): We generate 1,000 scenes for each specified room type and then calculate the average number of overlapping objects in each indoor scene. The lower the OLN, the fewer overlaps there are between objects.
- Number of Generated Objects (OBJ): This metric calculates the number of objects in each room and compares it with the ground truth. The closer OBJ is to the ground truth, the more the number of objects generated aligns with the real scene.

We present the evaluation results in Table 2. Firstly, for the two types of rooms, namely the living room and the dining room, our method performs the best in terms of OOB. That is, in the scene generation tasks for these two types of rooms, the phenomenon of objects exceeding the boundaries is relatively less. Compared to ATISS and PhyScene, our method has a lower OLN, indicating that our method is more effective in reducing object overlap in the scenes. DiffuScene has the lowest OLN, but its OOB is the highest. The reason is that the objects in scenes generated by DiffuScene are relatively scattered, and a significant number of objects exceed the boundaries. Therefore, the overlapping phenomenon among objects is relatively less. As for OBJ, there is no significant difference among the methods. Then, for the bedroom, our method also performs relatively well. However, the differences among all methods are not substantial. The reason is that compared to the other two room types, the layout of the bedroom is relatively simple, and all methods can more easily learn the characteristics of the scene. Overall, our method shows significant improvement in object placement compared to other methods, with a greater emphasis on boundary constraints and object compatibility. More importantly, we can generate a scene in a very short time, which is crucial for tasks that require real-time or near-real-time synthesis. Note that these metrics mainly capture global spatial validity but do not explicitly decompose errors into finer components, such as precise orientation accuracy or translational deviations. In future work, we plan to extend our evaluation framework to incorporate additional fine-grained geometric metrics—such as orientation consistency (e.g., mean angular deviation from ground truth) and position error (e.g., average Euclidean distance between predicted and true centroids)—to more comprehensively quantify the geometric precision of generated scenes.

Efficiency evaluation Finally, we compare the time required to generate scenes using different methods to test the generation efficiency of each approach. Table 3 presents the generation time required by different methods to produce the same scene. Due to the varying complexity of different room types, the time required to generate scenes of each type varies, even for the same method. ATISS, which places objects iteratively, takes the longest time to generate living rooms and dining rooms. Bedrooms, which contain fewer objects, require slightly less time. DiffuScene reduces the generation time compared to ATISS for living rooms and dining rooms. However, it takes longer for bedrooms as the number of sampling steps remains constant across all scene types. MiDiffusion achieves shorter generation times for all room types compared to ATISS and DiffuScene. PhyScene has longer generation times for all room types compared to DiffuScene, primarily due to its more complex denoising process. We have further evaluated the generation efficiency of our method by separately reporting the computation times for both the flow matching and geometry refinement stages. Specifically, the average generation time for flow matching is only 26 ms for living rooms, 20 ms for dining rooms, and 18 ms

Table 4
Ablation study on three kinds of room scene generation. We conduct a distribution evaluation of G.R. Only, SceneFlow w/o G.E., SceneFlow w/o G.R., SceneFlow w/o N.C., Simulated Annealing, and our method using four metrics: FID, KID, CA, and KL. The best results are shown in bold.

Method	Living room				Dining room				Bedroom			
	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)
G.R. Only	62.79	0.0159	57.20%	0.0379	59.70	0.0150	57.84%	0.0391	85.65	0.0056	58.26%	0.0340
SceneFlow w/o G.E.	50.58	0.0033	52.73%	**0.0206**	52.97	0.00216	53.26%	0.02383	81.49	0.00267	54.93%	0.0246
SceneFlow w/o G.R.	50.26	0.0026	53.16%	0.02216	52.80	0.00209	53.15%	0.02431	80.92	0.00259	54.27%	0.0221
SceneFlow w/o N.C.	50.37	0.0023	52.64%	0.02647	52.48	0.00213	52.86%	**0.02254**	80.84	0.00294	53.92%	0.0276
Simulated Annealing	50.73	0.0034	54.30%	0.02155	52.93	0.00227	53.51%	0.02548	81.23	0.00265	**53.66%**	0.0258
SceneFlow (Full method)	**49.94**	**0.0021**	**52.11%**	0.02107	**52.43**	**0.00204**	**52.24%**	0.02442	**80.62**	**0.00246**	53.81%	**0.02107**

for bedrooms. The geometry refinement stage requires an additional 5 ms, 5 ms, and 4 ms, respectively. The total time for the full SceneFlow method is thus 31 ms, 25 ms, and 22 ms for these three room types. In comparison, all baseline methods (such as ATISS, DiffuScene, MiDiffusion, and PhyScene) require substantially more time, often by an order of magnitude. Removing the geometry refinement module further reduces the overall scene generation time, enabling faster synthesis while maintaining competitive performance. These results demonstrate the high efficiency of our method, especially in comparison to existing state-of-the-art methods. In contrast to diffusion-based methods that require hundreds to thousands of stochastic iterative denoising steps, our method leverages Flow Matching, which employs a deterministic ordinary differential equation (ODE) formulation. This deterministic framework enables SceneFlow to directly learn a smoother and more concise transformation path from the initial chaotic scene distribution to the structured target layout, significantly reducing both the required number of sampling iterations and computational overhead. Consequently, SceneFlow achieves substantial reductions in generation time across all room types, outperforming diffusion-based methods by several orders of magnitude.

5.4. Ablation study

Regarding the geometry enhancement strategy, we have conducted ablation experiments under the following conditions: (1) without the geometry enhancement strategy (SceneFlow w/o G.E.); (2) without the non-overlap constraint (SceneFlow w/o N.C.); (3) without the geometry refinement (SceneFlow w/o G.R.). In addition, to demonstrate that our geometry refinement outperforms traditional heuristic methods, we include a comparison with the Simulated Annealing algorithm. We compare against Simulated Annealing as a representative non-gradient method for constrained layout optimization. It is easy to implement and requires no gradients, making it a practical baseline for layout optimization. It has been widely validated as an effective, robust, and model-agnostic method Yu et al. (2011); Merrell et al. (2011); Wu et al. (2018). Thus, Simulated Annealing is a widely accepted neutral baseline, ensuring unbiased ablation comparisons. Specifically, the Simulated Annealing algorithm is initialized with temperature $T = 100$, using SceneFlow without geometry refinement (SceneFlow w/o G.R.) as the initial solution state S, and performs 50 iterations per temperature level by generating new candidate solutions through perturbations of geometric attributes. During testing, no ground truth (GT) should be available to guide the optimization. Therefore, the objective function is redefined as a non-overlap loss based solely on the current scene, computed as the sum of intersection areas among object bounding boxes. This formulation encourages the generation of non-overlapping and spatially plausible layouts. Candidate solutions are probabilistically accepted based on the magnitude of improvement or degradation of the objective function. The temperature is gradually reduced according to a predefined cooling schedule until it falls below 0.1, at which point the algorithm terminates. To ensure fairness, both our geometry refinement and the Simulated Annealing algorithm start from the same initial solution derived from SceneFlow w/o G.R. On the other hand, we have conducted an additional experiment called G.R. Only, where the geometry refinement module is directly applied to the initial random samples $\mathcal{S}_0$ (sampled from a Gaussian distribution) without passing them through the flow model. This experiment aims to test whether the geometry refinement module alone could transform a disordered initialization into a structured layout.

The results of these ablation experiments are shown in Table 4 and Table 5. The results in Table 4 indicate that each component of the geometry enhancement strategy impacts the scenes. The G.R. Only variant performs the worst, confirming that the refinement module alone cannot produce coherent structures without the flow model's structural guidance. When SceneFlow is detached from the non-overlap constraint or geometry refinement, the FID, KID, and CA values increase. When both the non-overlap constraint and geometry refinement are used together, the lowest FID, KID, and CA values are achieved. Moreover, our method achieves superior results compared to the Simulated Annealing algorithm, highlighting the effectiveness of our geometry refinement. This demonstrates that the geometry enhancement significantly contributes to generating scenes that are more realistic and functionally coherent. From the results in Table 5, it can be observed that when using SceneFlow without the geometry enhancement strategy (SceneFlow w/o G.E.), the evaluation metrics OOB and OLN significantly increase, indicating an increase in out-of-bounds or overlap objects within the scenes. When using SceneFlow without the non-overlap constraint (SceneFlow w/o N.C.), the evaluation metrics OLN increase, demonstrating a significant effect in reducing the object overlap, effectively decreasing the phenomenon of overlapping between objects. When using SceneFlow without the geometry refinement (SceneFlow w/o G.R.), the evaluation metrics OOB increase, indicating an increase in out-of-bounds objects, confirming that the geometry refinement plays an important role in adjusting the geometric attributes of objects. Also, our geometry refinement clearly outperforms the Simulated Annealing algorithm. The G.R. Only variant exhibits the highest geometric errors, demonstrating that the refinement module alone cannot recover realistic layouts without flow-based structural priors. Our method, which uses both the non-overlap constraint and geometry refinement, achieves

Table 5
Ablation study on three kinds of room scene generation. We conduct a geometry evaluation of G.E. Only, SceneFlow w/o G.E., SceneFlow w/o G.R., SceneFlow w/o N.C., Simulated Annealing, and our method using three metrics: OOB, OLN, and OBJ. The best results are shown in bold. The ground truth is also shown.

Method	Living room			Dining room			Bedroom		
	OOB (↓)	OLN (↓)	OBJ ($\rightarrow GT$)	OOB (↓)	OLN (↓)	OBJ ($\rightarrow GT$)	OOB (↓)	OLN (↓)	OBJ ($\rightarrow GT$)
G.R. Only	9.27	5.94	11.29	6.60	5.33	10.62	5.41	1.53	5.72
SceneFlow w/o G.E.	8.89	6.10	**11.59**	5.76	5.48	**10.93**	4.45	1.58	5.66
SceneFlow w/o G.R.	8.92	5.66	11.96	5.70	5.33	10.91	4.52	1.47	5.54
SceneFlow w/o N.C.	8.73	6.13	12.03	5.53	5.45	10.70	4.22	1.55	**4.97**
Simulated Annealing	8.92	6.71	11.88	5.70	5.49	10.91	4.54	1.72	4.88
SceneFlow (Full method)	**8.52**	**5.43**	11.93	**5.49**	**5.21**	10.63	**4.15**	**1.44**	4.75
GT	1.51	3.74	11.67	0.73	4.37	11.12	3.37	1.45	5.22

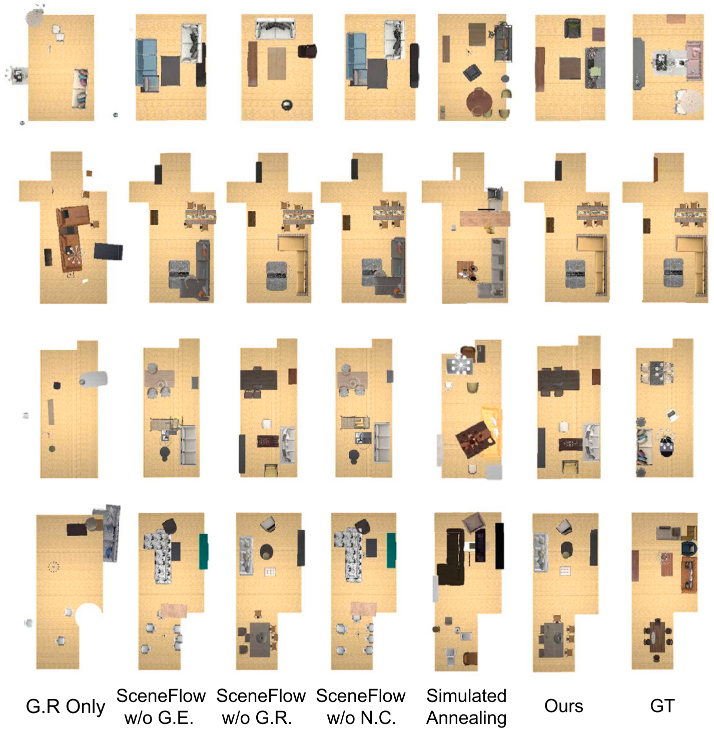


Fig. 6. Ablation study on the living room generation. We conduct the ablation study of our method and the baseline methods. From left to right, we show the indoor scenes of G.R., SceneFlow w/o G.E., SceneFlow w/o G.R., SceneFlow w/o N.C., Simulated Annealing, our method, and the ground truth.

the lowest OLN and OOB values. In other words, SceneFlow demonstrates superior performance in reducing the object overlap and out-of-bounds.

Fig. 6 presents some results from the ablation experiments. The results show that (G.R. Only), when applied independently, the geometry refinement module performs poorly—producing incoherent layouts due to the lack of structural context. This confirms that the flow model is essential for providing a reasonable initial structure that enables effective refinement. The results in the second column demonstrate that using the non-overlap constraint reduces the overlap in the scenes. The results in the third column show that the geometry refinement can correct improperly placed objects, adjusting those that originally exceeded the boundaries. The fourth column shows results obtained with the Simulated Annealing algorithm, where noticeable object overlap can be observed. The last column displays the effects of using both the non-overlap constraint and geometry refinement simultaneously, with significant improvements in both controlling the overlap and preventing objects from out-of-bounds.

We have conducted an ablation study between PointNet and ResNet (ResNet-18) for the task of 2D floorplan encoding, including a direct comparison between SceneFlow w/ ResNet and SceneFlow w/ PointNet, as shown in Table 6 and Table 7. The results clearly demonstrate that SceneFlow w/ PointNet outperforms SceneFlow w/ ResNet across most evaluation metrics and room types. Specifically, in the living room and dining room datasets, which are characterized by limited training data and larger room sizes, the PointNet-based model achieves significant performance gains. For example, SceneFlow w/ PointNet achieves lower FID (49.94 vs. 52.15 in living room; 52.43 vs. 53.67 in dining room) and KID scores (0.0021 vs. 0.0043 in living room; 0.00204 vs. 0.0029 in dining room), and higher CA (52.11% vs. 53.64% in living room; 52.24% vs. 53.12% in dining room) compared to SceneFlow w/

Table 6
Quantitative evaluation on different floorplan encoding methods: SceneFlow w/ PointNet and SceneFlow w/ ResNet using four metrics: FID, KID, CA, and KL. The best results are shown in bold.

Method	Living room				Dining room				Bedroom			
	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)	FID (↓)	KID (↓)	CA (→ 50%)	KL (↓)
SceneFlo w/ ResNet	52.15	0.0043	53.64%	**0.02097**	53.67	0.00290	53.12%	**0.02351**	81.29	0.00302	54.35%	0.0288
SceneFlow w/ PointNet	**49.94**	**0.0021**	**52.11%**	0.02107	**52.43**	**0.00204**	**52.24%**	0.02442	**80.62**	**0.00246**	**53.81%**	**0.02107**

Table 7
Geometry evaluation on different floorplan encoding methods: SceneFlow w/ PointNet, SceneFlow w/ ResNet, and the ground truth (GT) using three metrics: OOB, OLN, and OBJ. The best results are shown in bold.

Method	Living room			Dining room			Bedroom		
	OOB (↓)	OLN (↓)	OBJ (→ GT)	OOB (↓)	OLN (↓)	OBJ (→ GT)	OOB (↓)	OLN (↓)	OBJ (→ GT)
SceneFlow w/ ResNet	9.34	5.64	12.06	5.90	5.42	**10.77**	4.46	**1.42**	4.71
SceneFlow w/ PointNet	**8.52**	**5.43**	**11.93**	**5.49**	**5.21**	10.63	**4.15**	1.44	**4.75**
GT	1.51	3.74	11.67	0.73	4.37	11.12	3.37	1.45	5.22

Table 8
Ablation study on different timesteps using the computation time (TIME) and three metrics: FID, OOB, and OLN. The best results are shown in bold.

T	Living room				Dining room				Bedroom			
	TIME	FID (↓)	OOB (↓)	OLN (↓)	TIME	FID (↓)	OOB (↓)	OLN (↓)	TIME	FID (↓)	OOB (↓)	OLN (↓)
20	6	58.78	12.59	7.56	5	60.08	11.96	5.78	4	85.45	6.44	1.88
40	13	54.23	11.24	6.23	10	58.42	10.40	5.66	9	84.16	5.25	1.72
60	18	50.80	9.25	5.88	15	54.20	8.18	5.38	13	82.80	5.01	1.58
80	25	50.24	8.64	5.70	20	53.18	7.35	5.34	17	81.45	4.58	1.50
100	31	**49.94**	8.52	**5.43**	25	52.43	**5.49**	5.21	22	80.62	**4.15**	**1.44**
120	37	49.97	**8.51**	5.52	30	**52.41**	5.52	5.19	28	80.60	4.14	1.50

ResNet. Geometry evaluation in Table 7 further confirms these advantages: the PointNet-based model achieves lower OOB (8.52 vs. 9.34 in living room; 5.49 vs. 5.90 in dining room) and OLN values, indicating fewer out-of-bounds and overlapping objects. On the bedroom dataset, where the training data is more abundant and room sizes are smaller, PointNet's performance is slightly inferior to ResNet-18 in some metrics, but the difference is marginal (e.g., FID 80.62 vs. 81.29; OOB 4.15 vs. 4.46). Overall, SceneFlow with PointNet achieves the best or highly competitive results across all room types and most metrics. These results support our design choice of using PointNet as the floorplan encoder for enhanced geometric feature extraction and model generalizability.

To investigate the effect of different timesteps, we present an ablation study across different timesteps $T = \{20, 40, 60, 80, 100, 120\}$ (Table 8). Increasing T leads to steady improvements in FID, OOB, and OLN, but these gains quickly saturate. For instance, FID improves from 49.94 at T = 100 to only 49.97 at T = 120 in the living room, and the OOB drops slightly from 8.52% to 8.51%. OLN is also minimized at T = 100. However, further increasing T results in a linear rise in inference time (from 31 ms at T = 100 to 37 ms at T = 120), representing an 18% slowdown with negligible quality improvement. Thus, T = 100 offers an optimal trade-off between efficiency and generation quality.

5.5. Diverse generation

Our method can generate multiple plausible scenes based on the same room floorplan, and all these scenes are reasonable. We present some of the results in Fig. 7. For different types of rooms, our method can generate various reasonable scenes. For example, in the second row of the figure, which represents a living room, the generated scenes include different types of sofas and dining tables, with some positioned closer to the upper side and others closer to the lower side. Although the types and positions of the objects vary, all scenes are neatly organized and reasonable. Another example is the bedroom in the third row, where the generated scenes feature diverse types of beds, and while the first three columns include wardrobes, the fourth column does not, catering to different needs. Overall, the scenes generated by our method not only exhibit aesthetically pleasing object arrangements but also demonstrate strong diversity.

5.6. Scene completion

Our method is also capable of generating complete scenes from partial scenes. In the scene completion task, the user provides a partial scene containing M known (placed) objects. Internally, the model still utilizes N object slots: M slots are filled with the attributes (category, position, size, orientation) of the known objects, and the remaining slots are initialized as "empty" objects. The model's objective is to transform these empty slots into plausible non-empty objects (i.e., to generate the missing furniture) or to keep them empty if no further objects are needed. Fig. 8 shows a comparison of the completion effects of different methods. The first column of the figure presents the partial scenes we provided, with different rooms containing varying types and quantities of

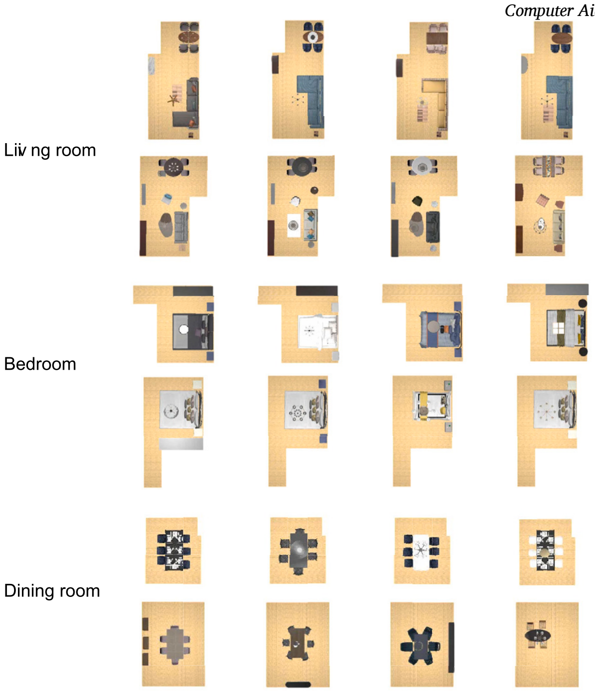


Fig. 7. Diversity generation on three kinds of room scene generation. Our method can generate multiple plausible scenes based on the same room floorplan.

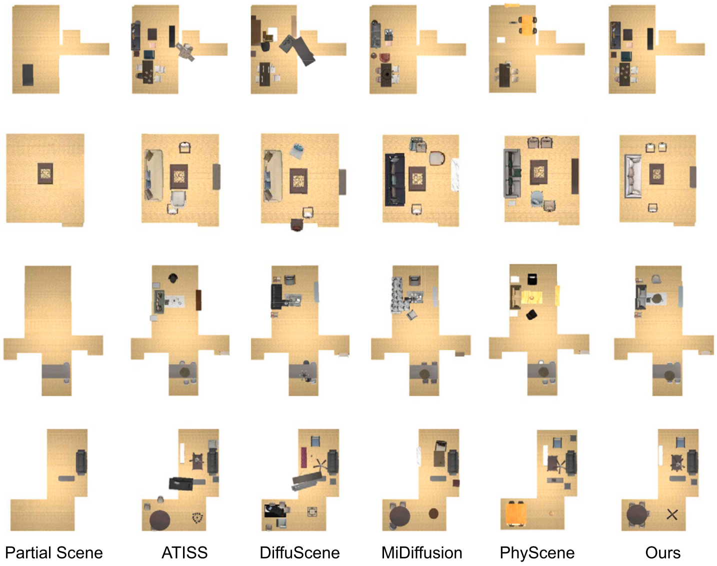


Fig. 8. Scene completion on the living room generation. We conduct the evaluation of our method and the baseline methods. From left to right, we show the partial scene, the generated scenes of ATISS, DiffuScene, PhyScene, and our method.

furniture, used to verify the performance of each method under different conditions. For ATISS, this is a relatively simple task. As ATISS is an autoregressive model, the existing objects allow the model to skip the first few iterations, reducing the likelihood of errors, and thus performing better than before. The issues with DiffuScene persist; even with existing objects, the newly generated objects still exhibit a significant problem of going out of bounds. PhyScene incorporates guidance during the denoising process and performs better in this task, but there are still a few instances of improperly placed objects. Our method, even in the context of partial scenes, demonstrates the best performance, with fewer instances of object overlap and out-of-bounds objects compared to other methods.

6. Conclusion

We propose SceneFlow, a flow-based generative framework for indoor scene synthesis, complemented by a geometric enhancement strategy to improve generation quality. Extensive experiments demonstrate that SceneFlow outperforms state-of-the-art methods in quality and efficiency. The geometric enhancement strategy is effective across different methods, improving spatial consistency in generated scenes.

Despite its strengths, SceneFlow has limitations. It relies on pre-defined object representations, which may struggle with unconventional object configurations, and the geometric enhancement strategy focuses primarily on spatial regularity, lacking consideration for higher-level semantic relationships. The generalization to highly complex or irregular spaces needs further validation. Although SceneFlow effectively generates realistic scenes, its generation process lacks the capability to adaptively control and adjust the types and quantities of furniture based on user preferences. This limitation restricts SceneFlow's flexibility in handling various dynamic scenarios. Additionally, our current method does not explicitly consider the impacts of doors, windows, and internal circulation paths on furniture arrangements, limiting its ability to fully capture realistic constraints and interactions within indoor spaces. Addressing these factors remains a valuable direction for future work. In SceneFlow, the non-overlap constraint serves as a fundamental geometric regularizer to eliminate large-scale implausible collisions, such as furniture interpenetration. Currently, functional overlaps are not explicitly considered in our experiments. Nevertheless, our framework can be readily extended to handle functional overlaps by introducing a hierarchical generation framework. For instance of a trash bin under a table, we first generate the table (and other relevant objects), and then generate the trash bin conditioned on the presence of the table, thus supporting functional overlap in a natural and structured way. Minor orientation errors of furniture may occur, common in recent scene synthesis methods. Object orientation prediction remains challenging, as it is sensitive to stochastic noise and incomplete spatial reasoning during generation, and is further exacerbated by dataset quality and quantity.

Future work will address these challenges by incorporating more flexible object representations, semantic reasoning, and support for interactive scene editing and multi-room layouts. Expanding the dataset diversity and integrating real-time user feedback will further enhance SceneFlow's adaptability and applicability to complex, dynamic scenes. On the other hand, future work could explicitly integrate collision constraints between furniture into the generation process, allowing these intermediate scenes to serve as meaningful guidance for practical tasks like motion planning Xiong et al. (2020) or room layout transformations, enhancing the versatility and applicability of SceneFlow. In future work, we plan to introduce a dedicated post-processing module to refine object orientations via functional and contextual cues (e.g., room layout and inter-object relationships) for more consistent, physically meaningful alignments.

CRediT authorship contribution statement

Wenming Wu: Writing – review & editing, Writing – original draft, Project administration, Methodology, Funding acquisition, Conceptualization. **Akang Shen:** Writing – original draft, Validation, Methodology, Data curation. **Yanzhe Yin:** Writing – review & editing, Investigation. **Zixiang Chen:** Writing – review & editing, Investigation. **Gaofeng Zhang:** Writing – review & editing. **Liping Zheng:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive suggestions and comments. This work was supported by the National Natural Science Foundation of China (Grant No. 62372152), the Fundamental Research Funds for the Central Universities of China (Grant No. PA2025GDSK0035), and the Open Project Program of the State Key Laboratory of CAD&CG, Zhejiang University (Grant No. A2412).

Data availability

Data will be made available on request.

References

Albergo, M.S., Vanden-Eijnden, E., 2023. Building normalizing flows with stochastic interpolants. In: The Eleventh International Conference on Learning Representations (ICLR).

Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R., 2021. Structured denoising diffusion models in discrete state-spaces. Adv. Neural Inf. Process. Syst. 34, 17981–17993.

Chattopadhyay, A., Zhang, X., Wipf, D.P., Arora, H., Vidal, R., 2023. Learning graph variational autoencoders with constraints and structured priors for conditional indoor 3d scene generation. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 785–794.

Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., Hanrahan, P., 2012. Example-based synthesis of 3d object arrangements. ACM Trans. Graph. 31, 1–11.

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

Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D., 2021b. 3d-future: 3d furniture shape with texture. Int. J. Comput. Vis. 129, 3313–3337.

Gao, L., Sun, J.M., Mo, K., Lai, Y.K., Guibas, L.J., Yang, J., 2023. Scenehgn: hierarchical graph networks for 3d indoor scene generation with fine-grained geometry. IEEE Trans. Pattern Anal. Mach. Intell. 45, 8902–8919.

Guerreiro, J.J.A., Inoue, N., Masui, K., Otani, M., Nakayama, H., 2025. Layoutflow: flow matching for layout generation. In: European Conference on Computer Vision. Springer, pp. 56–72.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4, 251–257.

Hu, S., Arroyo, D.M., Debats, S., Manhardt, F., Carlone, L., Tombari, F., 2024. Mixed diffusion for 3d indoor scene synthesis. arXiv preprint arXiv:2405.21066.

Kermani, Z.S., Liao, Z., Tan, P., Zhang, H., 2016. Learning 3d scene synthesis from annotated rgb-d images. Comput. Graph. Forum 35, 197–206.

Kingma, D.P., 2013. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.

Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1–13.

Kobyzev, I., Prince, S.J., Brubaker, M.A., 2020. Normalizing flows: an introduction and review of current methods. IEEE Trans. Pattern Anal. Mach. Intell. 43, 3964–3979.

Leimer, K., Guerrero, P., Weiss, T., Musialski, P., 2022. Layoutenhancer: generating good indoor layouts from imperfect data. In: SIGGRAPH Asia 2022 Conference Papers, pp. 1–8.

Li, Y., Xu, P., Ren, J., Shao, Z., Huang, H., 2024. Gltscene: global-to-local transformers for indoor scene synthesis with general room boundaries. In: Computer Graphics Forum. Wiley Online Library, p. e15236.

Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M., 2023. Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (ICLR).

Liu, X., Gong, C., Liu, Q., 2022. Flow straight and fast: learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (ICLR).

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

Maillard, L., Sereyjol-Garros, N., Durand, T., Ovsjanikov, M., 2024. Debara: denoising-based 3d room arrangement generation. arXiv preprint arXiv:2409.18336.

Meng, Q., Li, L., Nießner, M., Dai, A., 2024. Lt3sd: latent trees for 3d scene diffusion. arXiv preprint arXiv:2409.08215.

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., Koltun, V., 2011. Interactive furniture layout using interior design guidelines. ACM Trans. Graph. 30, 1–10.

Min, W., Wu, W., Zhang, G., Zheng, L., 2024. Funcscene: function-centric indoor scene synthesis via a variational autoencoder framework. Comput. Aided Geom. Des. 111, 102319.

Para, W.R., Guerrero, P., Mitra, N., Wonka, P., 2023. Cofs: controllable furniture layout synthesis. In: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–11.

Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., Fidler, S., 2021. Atiss: autoregressive transformers for indoor scene synthesis. Adv. Neural Inf. Process. Syst. 34, 12013–12026.

Patil, A.G., Patil, S.G., Li, M., Fisher, M., Savva, M., Zhang, H., 2024. Advances in data-driven analysis and synthesis of 3d indoor scenes. In: Computer Graphics Forum. Wiley Online Library, p. e14927.

Purkait, P., Zach, C., Reid, I., 2020. Sg-vae: scene grammar variational autoencoder to generate new indoor scenes. In: Computer Vision – ECCV 2020. Springer, pp. 155–171.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660.

Ritchie, D., Wang, K., Lin, Y.a., 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6182–6190.

Song, C., Wang, J., Chen, S., Li, H., Jiang, Z., Yang, B., 2024. Non-autoregressive transformer with fine-grained optimization for user-specified indoor layout. Eng. Appl. Artif. Intell. 133, 108024.

Sun, Q., Zhou, H., Zhou, W., Li, L., Li, H., 2025. Forest2seq: revitalizing order prior for sequential indoor scene synthesis. In: European Conference on Computer Vision. Springer, pp. 251–268.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567.

Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M., 2024a. Diffuscene: denoising diffusion models for generative indoor scene synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20507–20518.

Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M., 2024b. Diffuscene: denoising diffusion models for gerative indoor scene synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1–21.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., Bengio, Y., 2023. Improving and generalizing flow-based generative models with minibatch optimal transport. In: ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Curran Associates Inc., pp. 6000–6010.

Wang, K., Lin, Y.A., Weissmann, B., Savva, M., Chang, A.X., Ritchie, D., 2019. Planit: planning and instantiating indoor scenes with relation graph and spatial prior networks. ACM Trans. Graph. 38, 1–15.

Wang, X., Yeshwanth, C., Nießner, M., 2021. Sceneformer: indoor scene generation with transformers. In: 2021 International Conference on 3D Vision (3DV). IEEE, pp. 106–115.

Wei, Q.A., Ding, S., Park, J.J., Sajnani, R., Poulenard, A., Sridhar, S., Guibas, L., 2023. Lego-net: learning regular rearrangements of objects in rooms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19037–19047.

Wu, W., Fan, L., Liu, L., Wonka, P., 2018. Miqp-based layout design for building interiors. In: Computer Graphics Forum. Wiley Online Library, pp. 511–521.

Xiong, G., Fu, Q., Fu, H., Zhou, B., Luo, G., Deng, Z., 2020. Motion planning for convertible indoor scene layout design. IEEE Trans. Vis. Comput. Graph. 27, 4413–4424.

Yang, Y., Jia, B., Zhi, P., Huang, S., 2024. Physcene: physically interactable 3d scene synthesis for embodied ai. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16262–16272.

Yu, L.F., Yeung, S.K., Tang, C.K., Terzopoulos, D., Chan, T.F., Osher, S.J., 2011. Make it home: automatic optimization of furniture arrangement. ACM Trans. Graph. 30, 86.

Zhai, G., Örnek, E.P., Chen, D.Z., Liao, R., Di, Y., Navab, N., Tombari, F., Busam, B., 2025. Echoscene: indoor scene generation via information echo over scene graph diffusion. In: European Conference on Computer Vision. Springer, pp. 167–184.

Zhai, G., Örnek, E.P., Wu, S.C., Di, Y., Tombari, F., Navab, N., Busam, B., 2024. Commonscenes: generating commonsense 3d indoor scenes with scene graphs. Adv. Neural Inf. Process. Syst. 36.

Zhang, S.H., Zhang, S.K., Xie, W.Y., Luo, C.Y., Yang, Y.L., Fu, H., 2021a. Fast 3d indoor scene synthesis by learning spatial relation priors of objects. IEEE Trans. Vis. Comput. Graph. 28, 3082–3092.

Zhang, S.K., Li, Y.X., He, Y., Yang, Y.L., Zhang, S.H., 2021b. Mageadd: real-time interaction simulation for scene synthesis. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 965–973.

Zhang, S.K., Xie, W.Y., Zhang, S.H., 2021c. Geometry-based layout generation with hyper-relations among objects. Graph. Models 116, 101–104.

Zhang, Z., Yang, Z., Ma, C., Luo, L., Huth, A., Vouga, E., Huang, Q., 2020. Deep generative modeling for scene synthesis via hybrid representations. ACM Trans. Graph. 39, 1–21.