NEURAL INFORMATION PROCESSING SYSTEMS

DeBaRA: Denoising-Based 3D Room Arrangement Generation

Léopold Maillard[1,2], Nicolas Sereyjol-Garros, Tom Durand[2], Maks Ovsjanikov[1]

[1]LIX, École Polytechnique, IP Paris [2]Dassault Systèmes

NeurIPS 2024

l'X ÉCOLE POLYTECHNIQUE

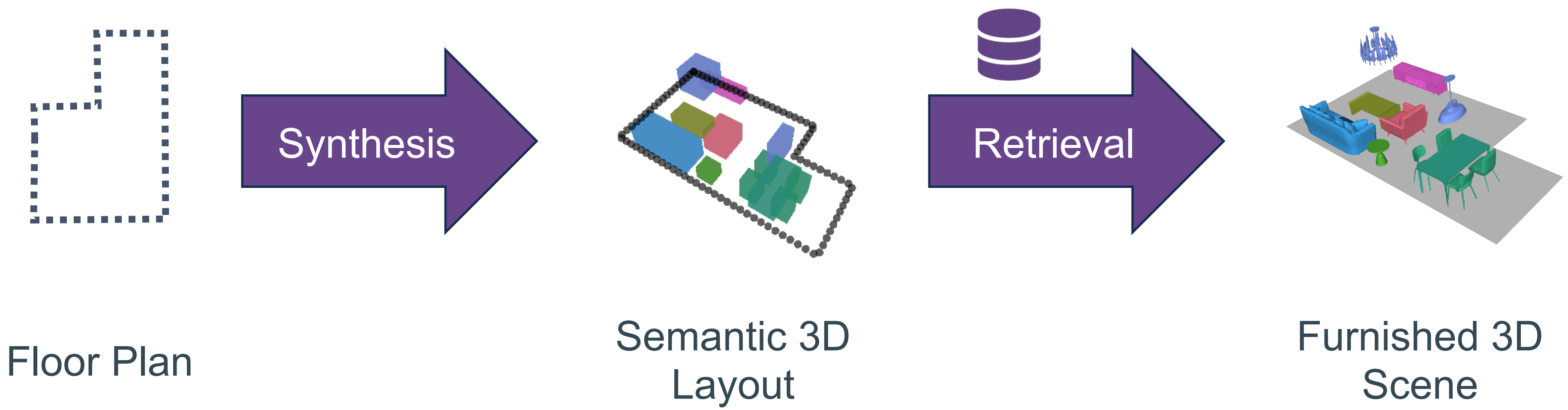
INSTITUT POLYTECHNIQUE DE PARIS

DASSAULT SYSTEMES

Introduction

Task

Controllable **3D Indoor Scene Synthesis** – generating realistic layouts of semantic 3D objects in a bounded environment.



Introduction

Challenges

- Inherent complexity of object **interactions**.
- Requirement to fulfill **spatial**, **ergonomic** and **functional** constraints.
- Limited amount of **available data**.

Background

Existing methods synthesize rooms **autoregressively** [1]

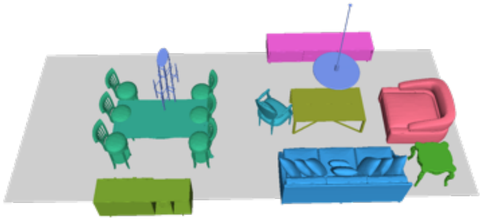
- Which is known to easily fall into local minima

Or by using off-the-shelf **diffusion models** that predicts all the object attributes, both spatial and semantic, within a single framework [2]

- Which lacks of data-efficiency and 3D reasoning considerations.

Typical Failure Cases

Autoregressive [1]



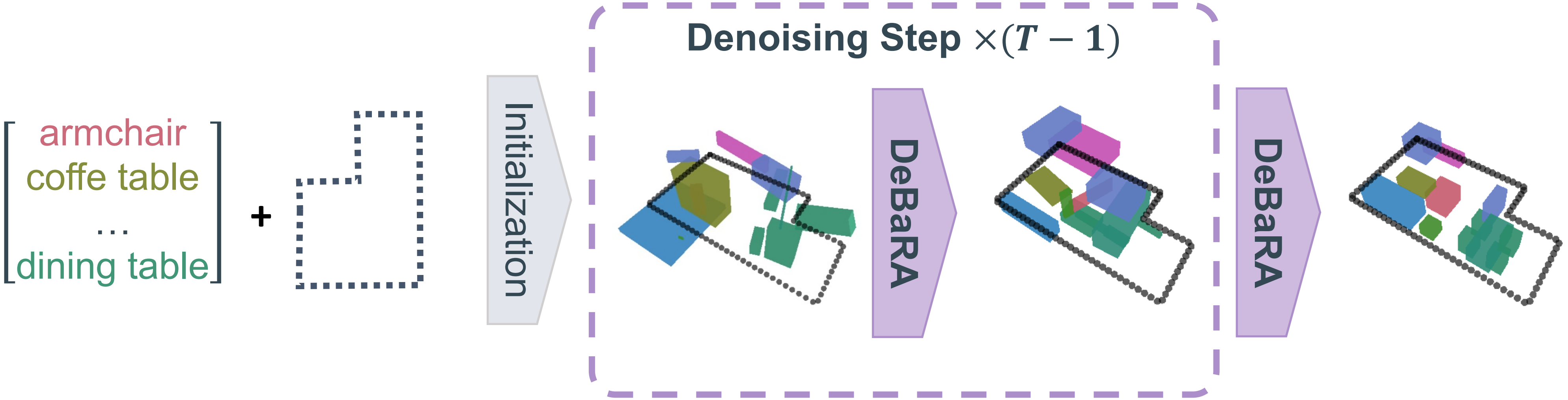
Off-the-shelf Diffusion [2]

[1] Paschalidou et al. *ATISS: Autoregressive Transformers for Indoor Scene Synthesis*, in NeurIPS 2021
[2] Tang et al. *DiffuScene: Denoising Diffusion Models for Generative Indoor Scene Synthesis*, in CVPR 2024

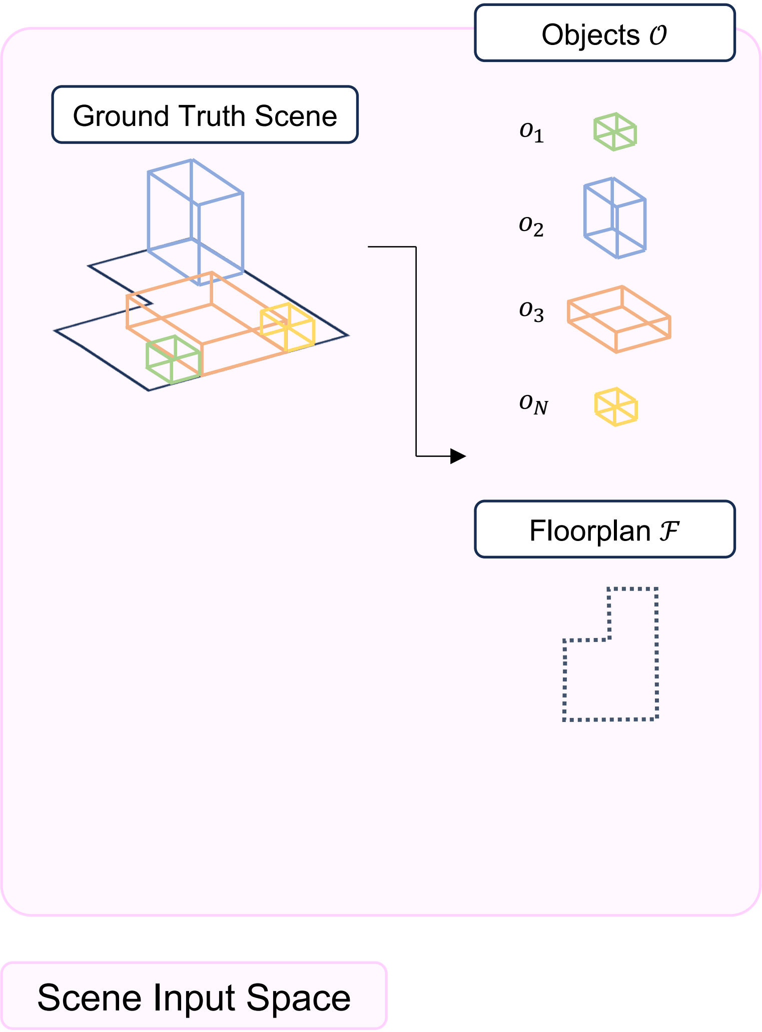
Introduction

Motivation

In contrast, we propose a diffusion-based method that focus solely on accurately establishing the critical **spatial features** (position, rotation and dimension) of objects, represented as **3D bounding boxes**, from a **floor plan** and a **list of categories.**



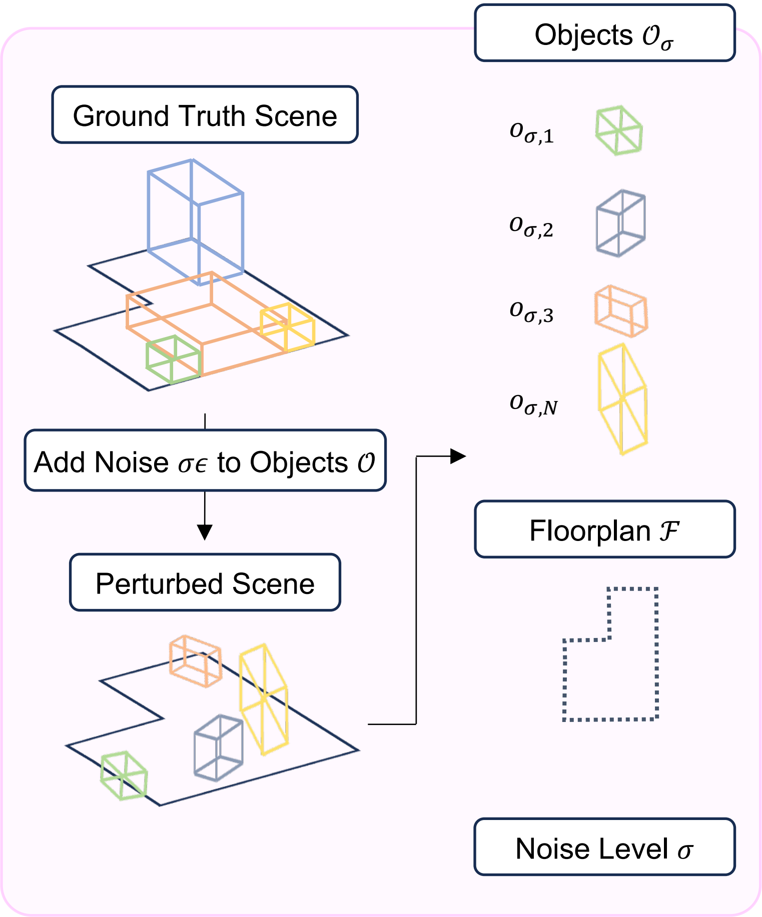
Training Pipeline



3D Scene Representation

A 3D Scene S is defined by a floor plan $\mathcal{F}$ and a set of N objects $\mathcal{O} = \{o_1, \ldots, o_N\}$, each being represented by a category c_i and bounding box **spatial** features $x_i = (p_i, r_i, d_i)$.

Training Pipeline



Scene Input Space

3D Scene Representation

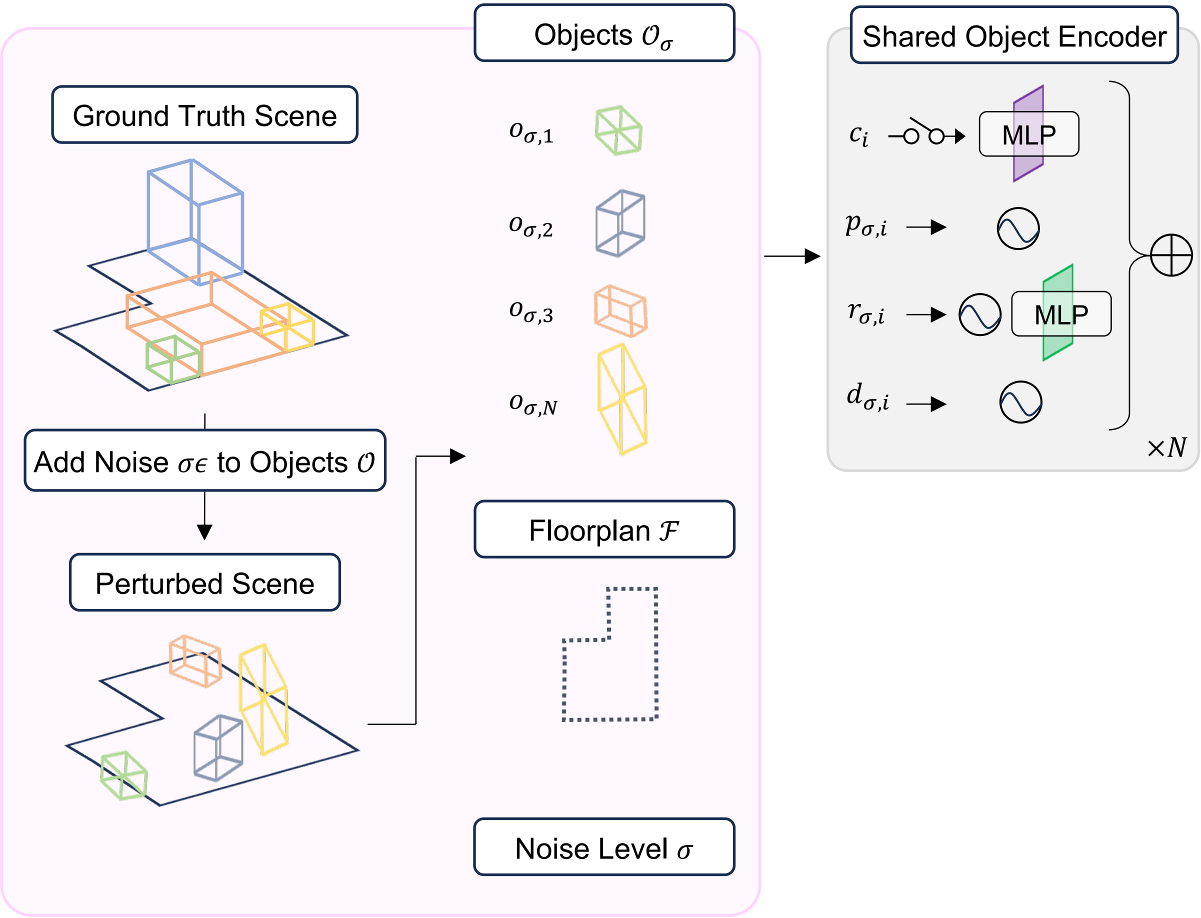
A 3D Scene S is defined by a floor plan $\mathcal{F}$ and a set of N objects $\mathcal{O} = \{o_1, \ldots, o_N\}$, each being represented by a category c_i and bounding box **spatial** features $x_i = (p_i, r_i, d_i)$.

Learning 3D Layouts from Room Bounds

We use a score-based approach to yield a **conditional generative model** that outputs 3D object bounding boxes from their semantic categories and input floor plan.

A noise-conditioned denoiser $D_\theta(x_\sigma; \mathcal{F}, c, \sigma)$ maps **noisy spatial features** $x_\sigma = x + \sigma\epsilon$ to their *clean* counterparts x.

Training Pipeline



Modeling the Unconditional Density

During training, input object categories are randomly **dropped** to model both the **class-conditional** and **unconditional** 3D layout distributions.

Conditioning Dropout

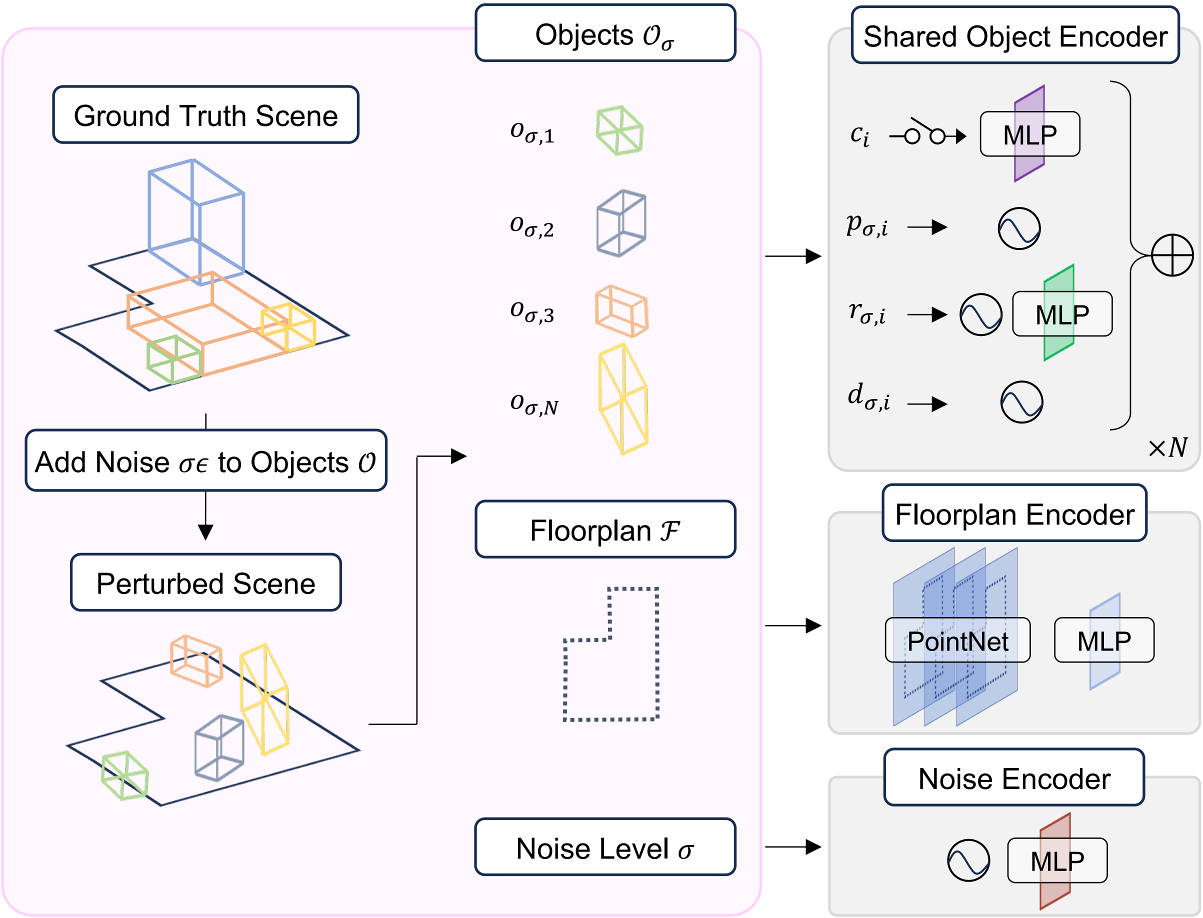
Scene Input Space

Trainable Module

Concatenation

Positional Encoding

Training Pipeline



Scene Input Space
Trainable Module
Concatenation
Positional Encoding

Modeling the Unconditional Density

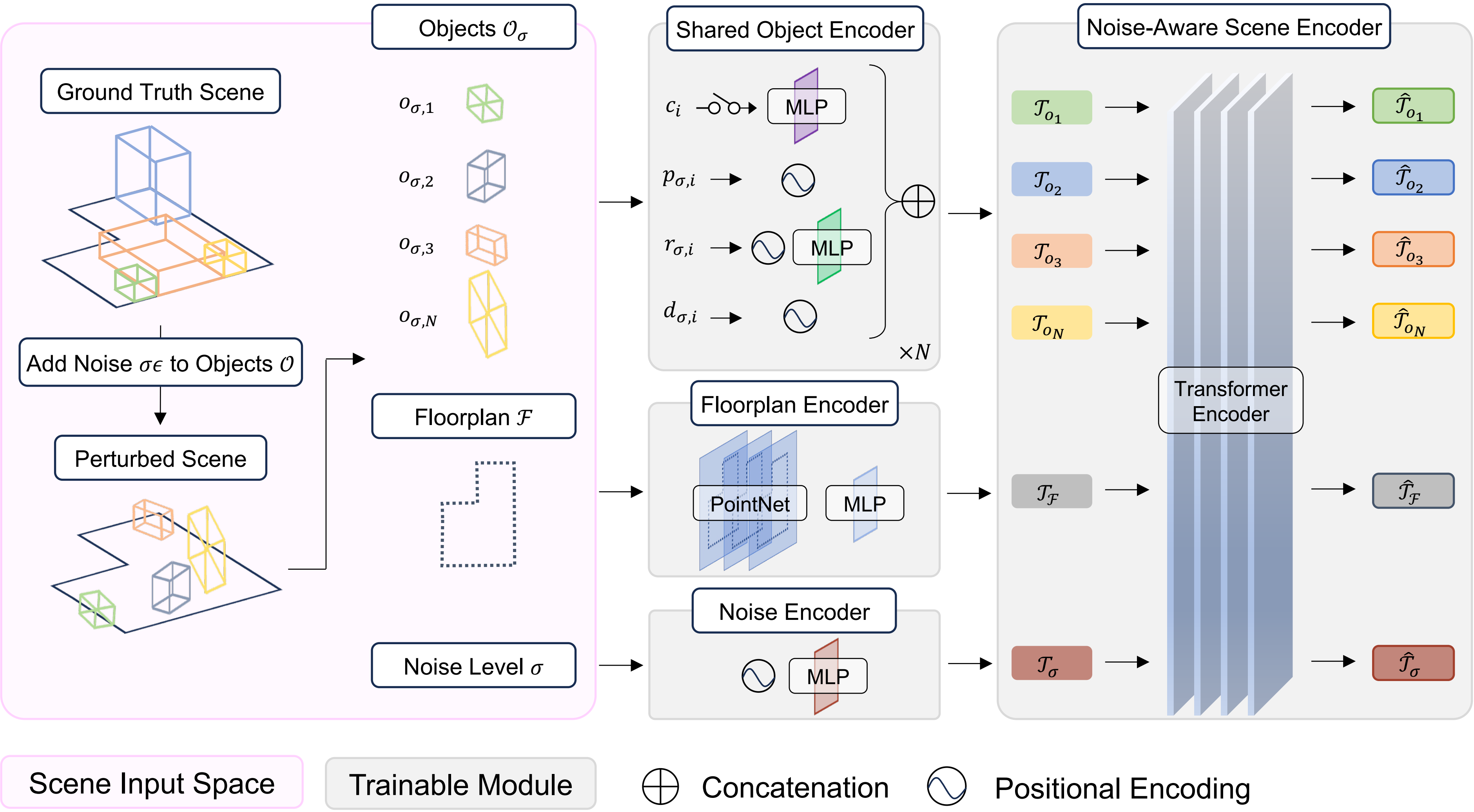
During training, input object categories are randomly **dropped** to model both the **class-conditional** and **unconditional** 3D layout distributions.

Conditioning Dropout

Denoising Network Architecture

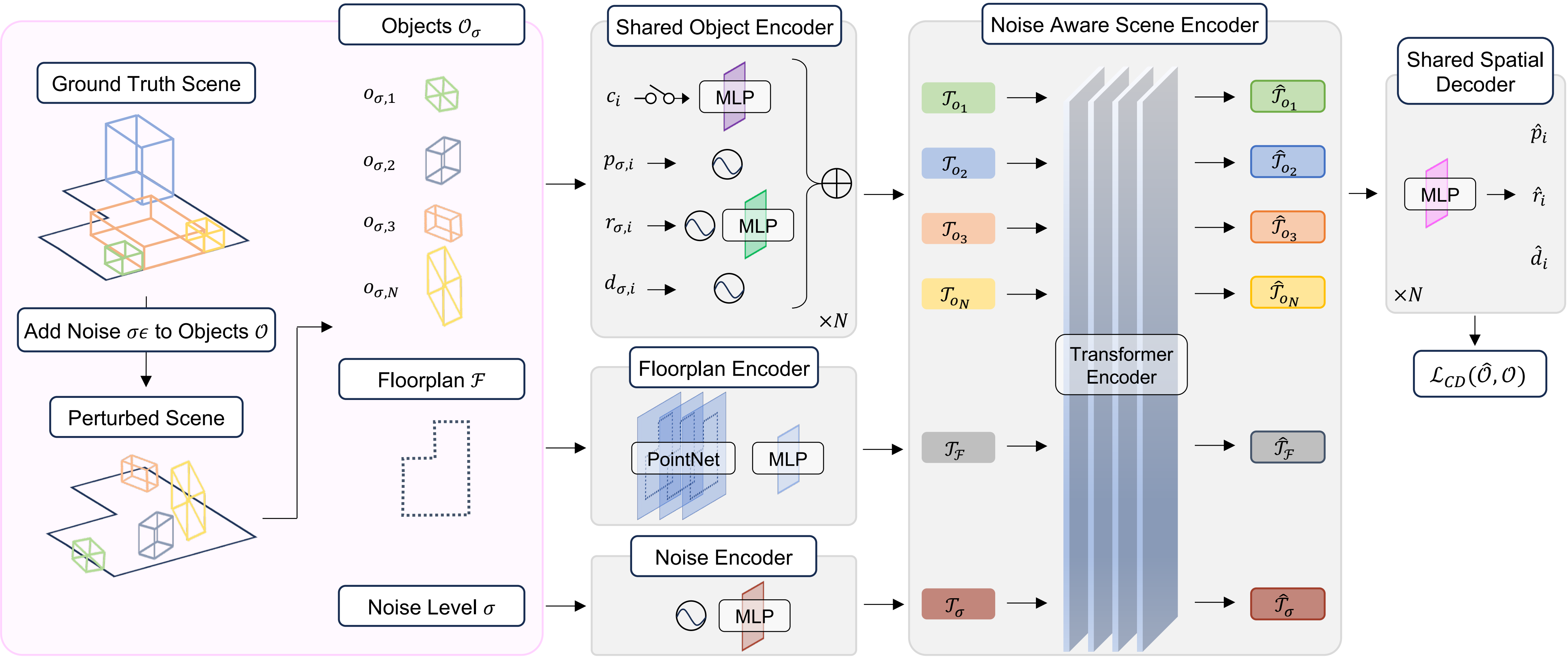
The floor plan $\mathcal{F}$, noise level σ and corresponding perturbed objects $\mathcal{O}_\sigma$ are processed by respective **encoders** to form an **unordered set** of embeddings…

Training Pipeline



… fed as input to a noise-conditioned **transformer** encoder

Training Pipeline



Scene Input Space
Trainable Module
$\oplus$ Concatenation
Positional Encoding

Training Pipeline

3D Spatial Objective

We propose a novel Chamfer formulation that does not penalize **permutation** of 3D object bounding boxes sharing **the same semantic category**.

$$\mathcal{L}_{CD}(\hat{\mathcal{O}}, \mathcal{O}) = \frac{1}{2N}\left(\sum_{\hat{o}\in\hat{\mathcal{O}}} \min_{o\in\mathcal{O}} l(\hat{o}, o) + \sum_{o\in\mathcal{O}} \min_{\hat{o}\in\hat{\mathcal{O}}} l(\hat{o}, o)\right)$$

where $l(\hat{o}, o) = \|\hat{x} - x\|_2^2 + \underbrace{\boldsymbol{\kappa}\big(\mathbf{1} - \boldsymbol{\delta_c}(\hat{\boldsymbol{o}}, \boldsymbol{o})\big)}_{\text{Semantic Penalty}}$ and $\kappa \gg 1$

Training Pipeline

Ablations

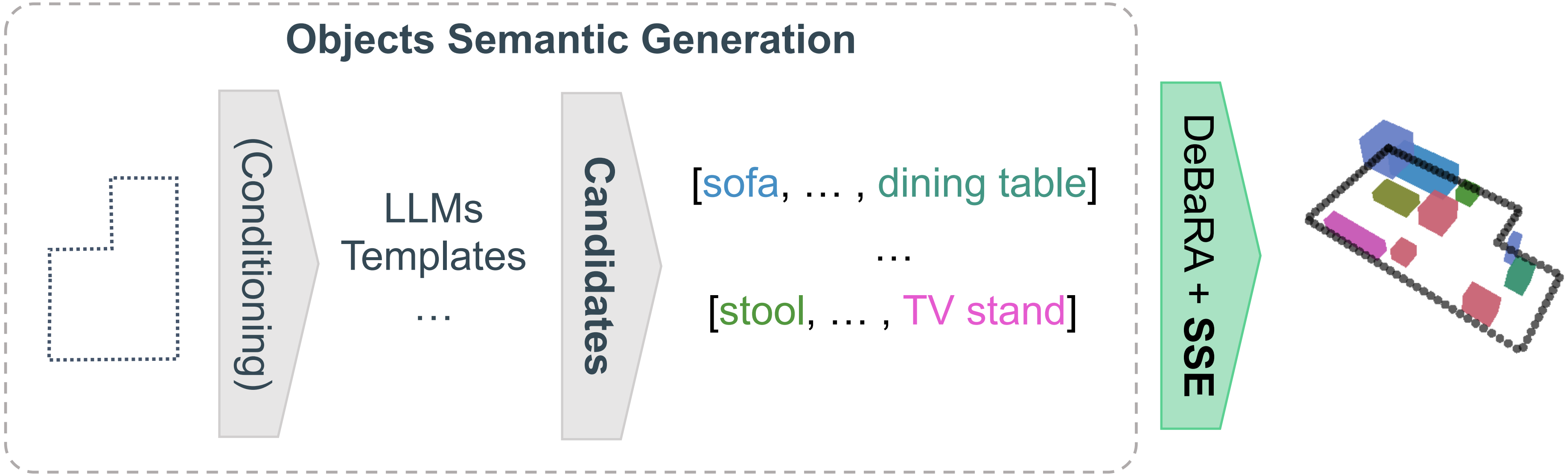
Ablation Setting		*Living Rooms*				*Dining Rooms*			
$\mathcal{L}(\hat{\mathcal{O}}, \mathcal{O})$	$p_{dropout}$	FID ↓	KID ↓	SCA %	OBA ↓	FID ↓	KID ↓	SCA %	OBA ↓
MSE	0.0	21.66	6.55	70.9	237.0	23.89	5.51	56.9	136.5
CD	0.0	21.76	7.05	71.7	225.1	25.21	6.75	59.4	294.7
ours	0.0	19.89	4.82	**63.5**	220.0	22.60	4.87	53.4	159.4
ours	**0.2**	**18.89**	**3.57**	68.3	**167.8**	**22.04**	**4.41**	**52.4**	**132.8**

*For SCA, values closer to 50% are better.
**OBA is the cumulated out-of-bounds objects area computed across the test subset, in m^2.

Self Score Evaluation (SSE)

Motivation

Input set of object categories can be *provided* by external sources such as a LLM [3]. **SSE** is a method to select the sets that lead to the most realistic scenes.



[3] Feng et al. *LayoutGPT: Compositional Visual Planning and Generation with Large Language Models*, in NeurIPS 2023

Self Score Evaluation (SSE)

Method

Input set of object categories can be *provided* by external sources such as a LLM [3]. **SSE** is a method to select the sets that lead to the most realistic scenes.

It consists in evaluating C conditioning categories *candidates*, where each candidate is associated to a 3D layout sampled from the **class-conditional** density:

$$\text{candidates} = \left\{\left(c_j, x_j \sim p_\theta\left(x|\mathcal{F}, c_j\right)\right)\right\}_{j=1}^{C}$$

The optimal conditioning candidate c^* is derived from a density estimate of its corresponding 3D spatial layout x^* provided by the **unconditional** model:

$$x^* = \arg\min_{x_i} \mathbb{E}_{\epsilon,\sigma}[\mathcal{L}_{CD}\{D_\theta(x_i + \sigma\epsilon; \mathcal{F}, \emptyset, \sigma), x_i\}]$$

[3] Feng et al. *LayoutGPT: Compositional Visual Planning and Generation with Large Language Models*, in NeurIPS 2023

Self Score Evaluation (SSE)

Algorithm

Similar to Diffusion Classifiers [4], we compute a **Monte Carlo estimate** of each *candidate* expectation using T_{SSE} fixed (σ, ϵ) pairs.

Algorithm 1 Self Score Evaluation

Require: a diffusion prior D_θ trained with conditioning dropout and by optimizing $\mathcal{L}_{CD}$
Input: conditioning candidates $\{\boldsymbol{c}_j\}_{j=1}^{C}$, number of score evaluation trials T_{sse}

```
sample x_j ~ p_θ(x|F, c_j) for each candidate c_j using iterative sampling
initialize scores[c_j] = list() for each c_j
for trial t = 1, ..., T_sse do
    sample σ ~ N(0, σ_s); ε ~ N(0, I)
    for candidate c_k, sample x_k do
        scores[c_k].append(L_CD[D_θ(x_k + σε, ; F, ∅, σ), x_k])
    end for
end for
return arg min_{c_j} mean(scores[c_j])
```

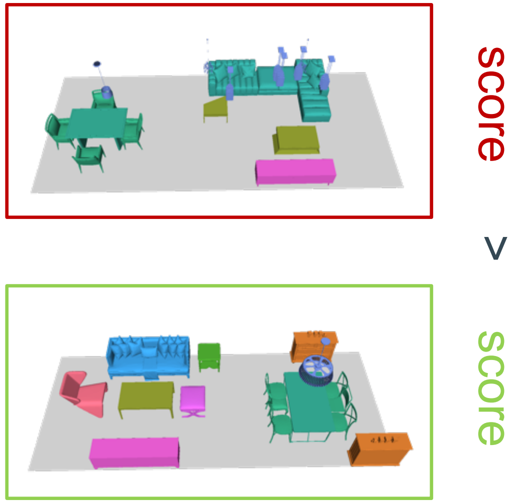
$$\textbf{sample } \boldsymbol{x}_j \sim p_\theta(\boldsymbol{x}|\mathcal{F}, \boldsymbol{c}_j) \text{ for each candidate } \boldsymbol{c}_j \text{ using iterative sampling}$$
$$\textbf{initialize } \texttt{scores}[\boldsymbol{c}_j] = \texttt{list}() \text{ for each } \boldsymbol{c}_j$$
$$\textbf{for } \text{trial } t = 1, \dots, T_{\text{sse}} \textbf{ do}$$
$$\quad \textbf{sample } \sigma \sim \mathcal{N}(0, \sigma_s); \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\quad \textbf{for } \text{candidate } \boldsymbol{c}_k, \text{ sample } \boldsymbol{x}_k \textbf{ do}$$
$$\qquad \texttt{scores}[\boldsymbol{c}_k].\text{append}(\mathcal{L}_{CD}[D_\theta(\boldsymbol{x}_k + \sigma\boldsymbol{\epsilon}, ; \mathcal{F}, \emptyset, \sigma), \boldsymbol{x}_k])$$
$$\quad \textbf{end for}$$
$$\textbf{end for}$$
$$\textbf{return } \arg\min_{\boldsymbol{c}_j} \texttt{mean}(\texttt{scores}[\boldsymbol{c}_j])$$

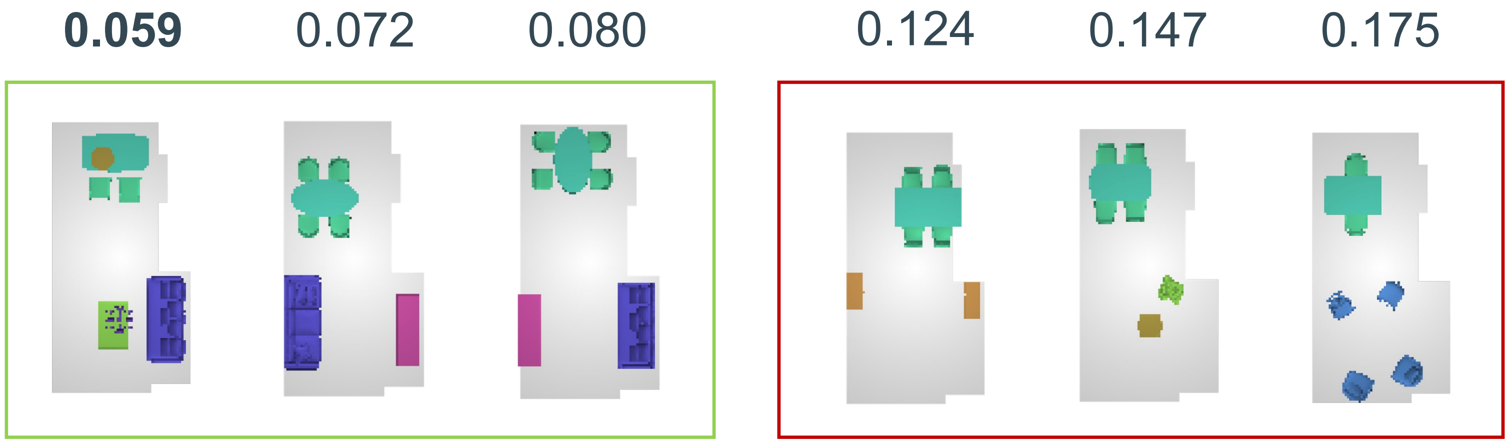
[4] Li et al. *Your Diffusion Model is Secretly a Zero-Shot Classifier*, in ICCV 2023

Self Score Evaluation (SSE)

Application Results

Candidate sets of object categories can be automatically generated by a LLM, and using SSE, further **selected** to generate a plausible 3D layout, or automatically **discarded**.



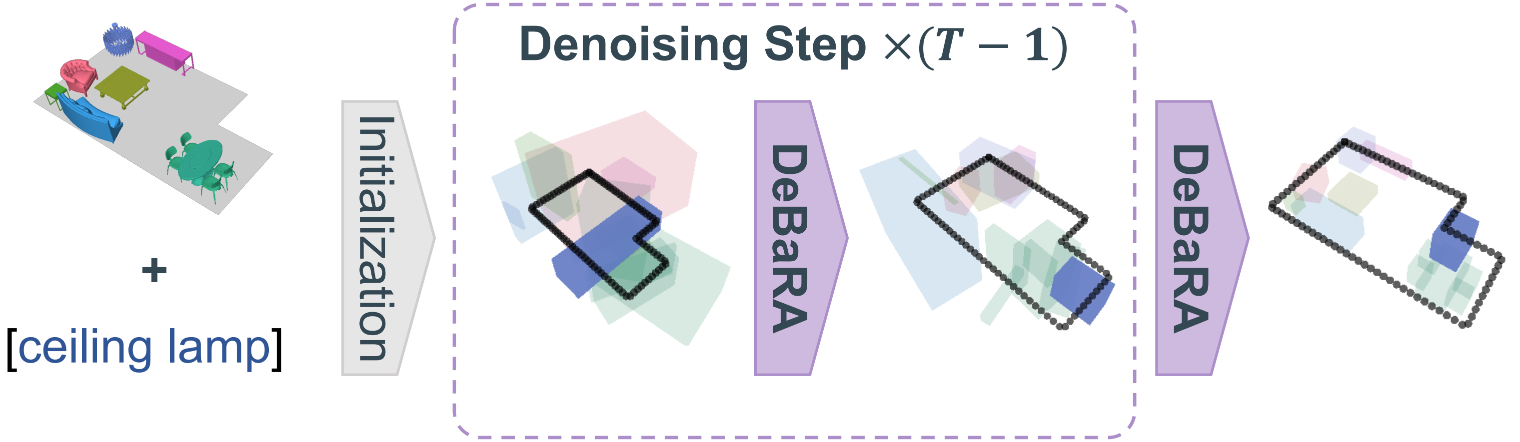


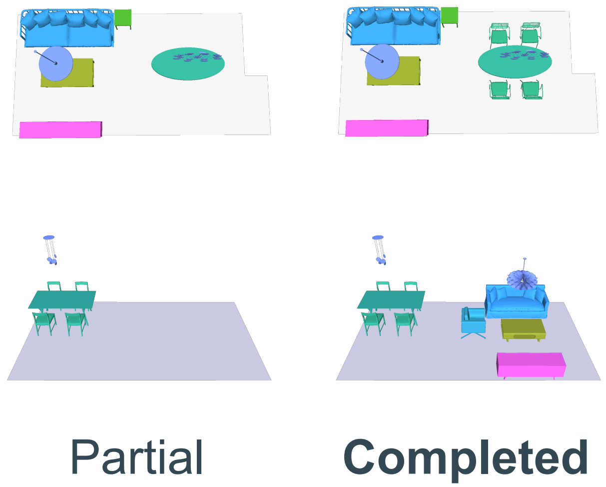
Top-down view of scenes generated by DeBaRA from LLM-generated candidates and their associated SSE scores.

Other Application Scenarios

A trained DeBaRA model can be leveraged to perform several downstream applications, by tweaking the initial sampling noise level $\sigma_{\max}$ and / or performing object or attribute-level **layout inpainting**.

Scene Completion

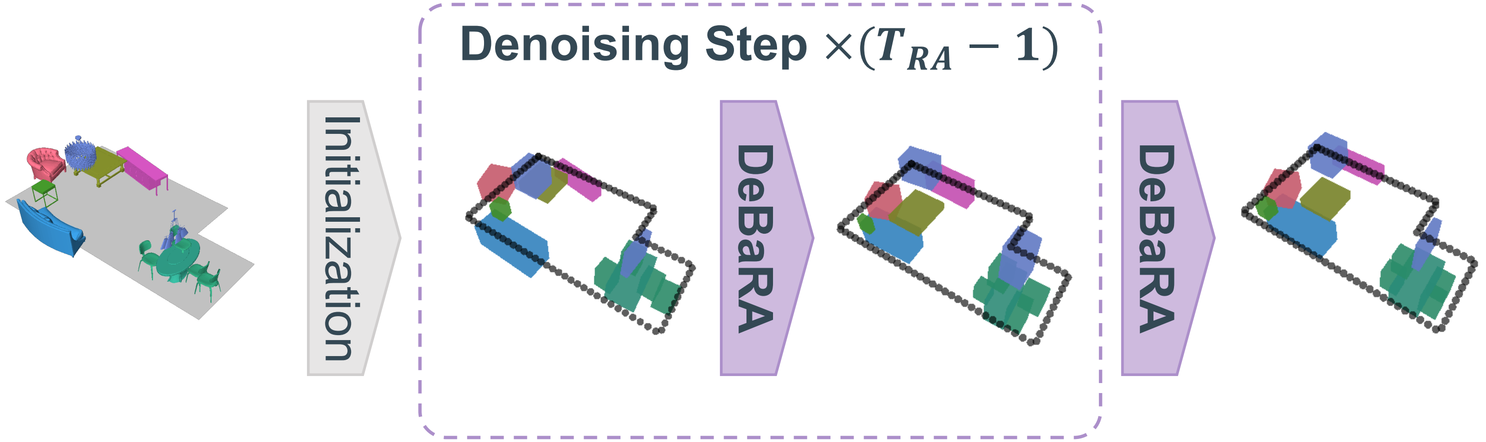


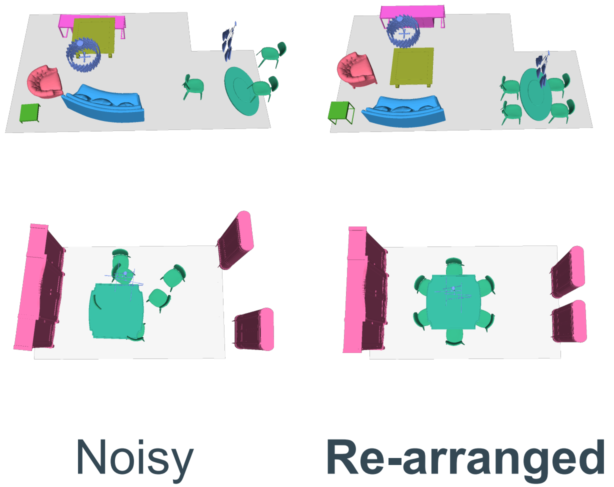


Other Application Scenarios

A trained DeBaRA model can be leveraged to perform several downstream applications, by tweaking the initial sampling noise level $\sigma_{\max}$ and / or performing object or attribute-level layout inpainting.

Scene Re-Arrangement [5]



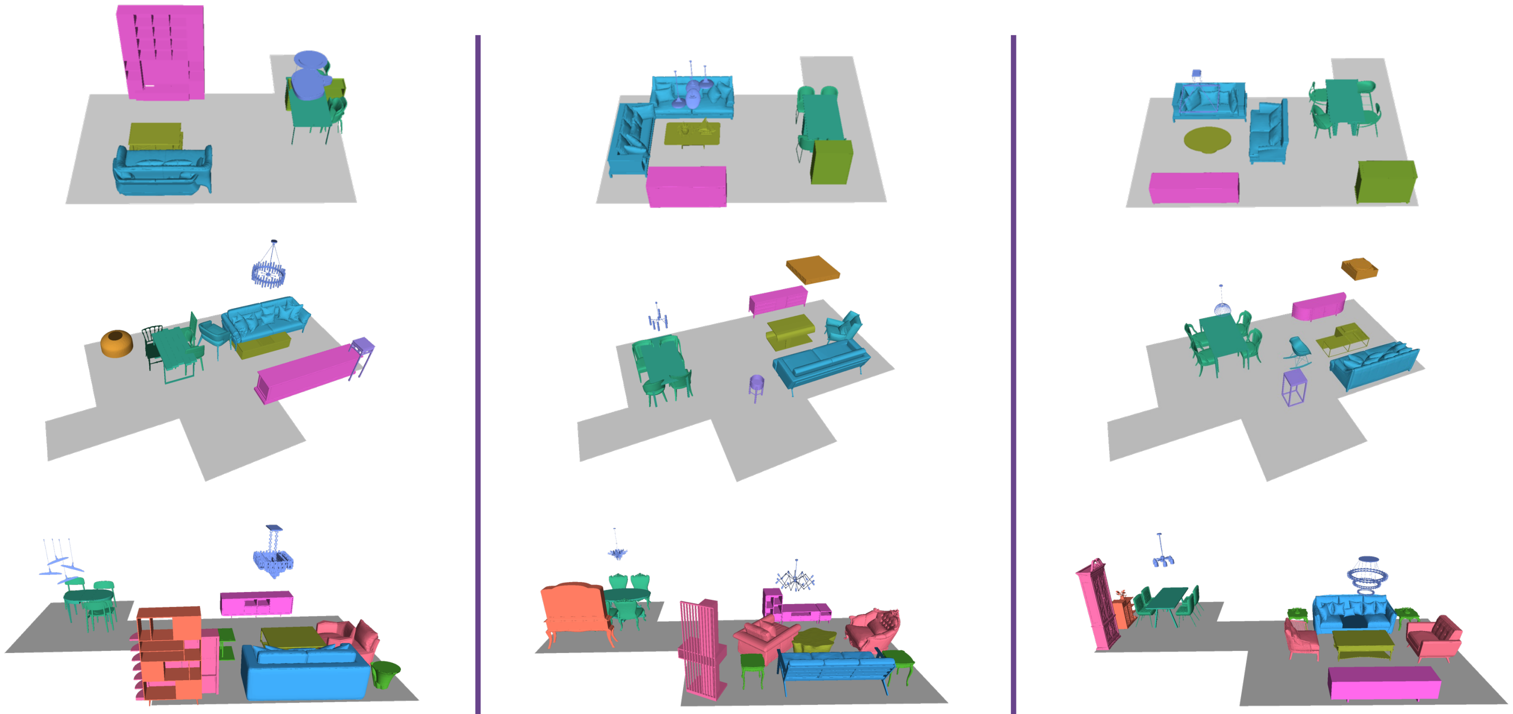


[5] Wei et al. *LEGO-Net: Learning Regular Rearrangements of Objects in Rooms*, in CVPR 2023

Experimental Evaluations

Our quantitative experimental evaluations shows that DeBaRA achieves **state-of-the-art** performance in a range of scenarios including 3D Layout Generation, Scene Synthesis, and Re-arrangement.

3D Layout Generation



ATISS | DiffuScene | **DeBaRA**

Experimental Evaluations

Our quantitative experimental evaluations shows that DeBaRA achieves **state-of-the-art** performance in a range of scenarios including 3D Layout Generation, Scene Synthesis, and Re-arrangement.

3D Layout Generation

Method	*Living Rooms*				*Dining Rooms*			
	FID ↓	KID ↓	SCA %	OBA ↓	FID ↓	KID ↓	SCA %	OBA ↓
ATISS	25.67	8.91	71.8	857.3	28.05	9.26	63.2	702.4
DiffuScene	21.54	6.40	69.7	341.1	23.06	5.35	57.7	266.4
DeBaRA (ours)	**18.89**	**3.57**	**68.3**	**167.8**	**22.04**	**4.41**	**52.4**	**132.8**

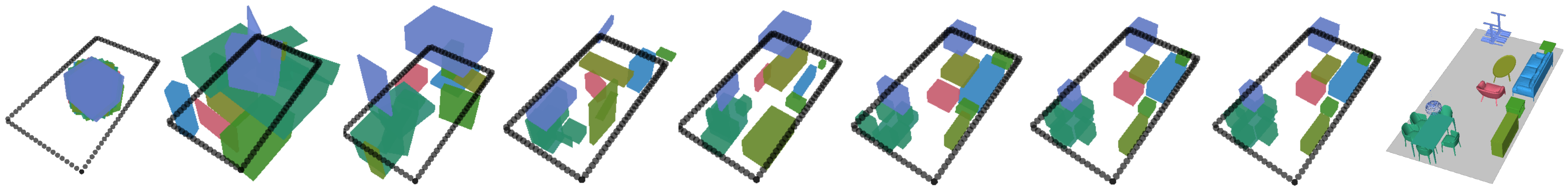
Quantitative evaluation results on the 3D-FRONT [6] dataset.

[6] Fu et al. *3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics*, in ICCV 2021

Sampling

Timestep t	0 (init)	5	10	15	20	30	40	50	**Result**

Layout $\tilde{x}_{\sigma_t}$



Method	Parameters (10^6)	Sampling Time (s)
ATISS	36.1	0.160
DiffuScene*	89.7	32.796
DeBaRA†	12.2	0.488
DeBaRA + SSE‡	12.2	0.894

*__DiffuScene__ uses **DDPM** [7] sampling with 1000 steps.

†**DeBaRA** uses 2nd order **EDM** [8] sampling with 50 steps.

‡**SSE** is implemented with 100 denoising trials.

[7] Ho et al. *Denoising Diffusion Probabilistic Models*, in NeurIPS 2020
[8] Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*, in NeurIPS 2022

Contributions Summary

DeBaRA

A **lightweight score-based model** trained to learn the class-conditional and unconditional densities of 3D layouts in bounded indoor scenes, using a **novel 3D spatial Chamfer objective**.

Self Score Evaluation (SSE)

A **procedure to select the best conditioning inputs** provided by external sources, such as LLMs, using **density estimates** provided by the pretrained generative model.

Controllable Sampling Method

A single model trained following our method can perform **multiple downstream tasks** such as scene completion or re-arrangement, in **real-time** (<1s).

Thank You!

DeBaRA: Denoising-Based 3D Room Arrangement Generation

Léopold Maillard[1,2], Nicolas Sereyjol-Garros, Tom Durand[2], Maks Ovsjanikov[1]

[1]LIX, École Polytechnique, IP Paris [2]Dassault Systèmes