arXiv:2303.03565v2 [cs.CV] 2 Jun 2023

CLIP-Layout：基于语义家具嵌入的风格一致的室内场景合成

刘景宇 熊文翰 IanJones 聂逸鑫 Anchit Gupta Barlas O˘guz Meta AI{jingyuliu, xwhan, ijones, ynie, anchit,barlaso}@meta.com

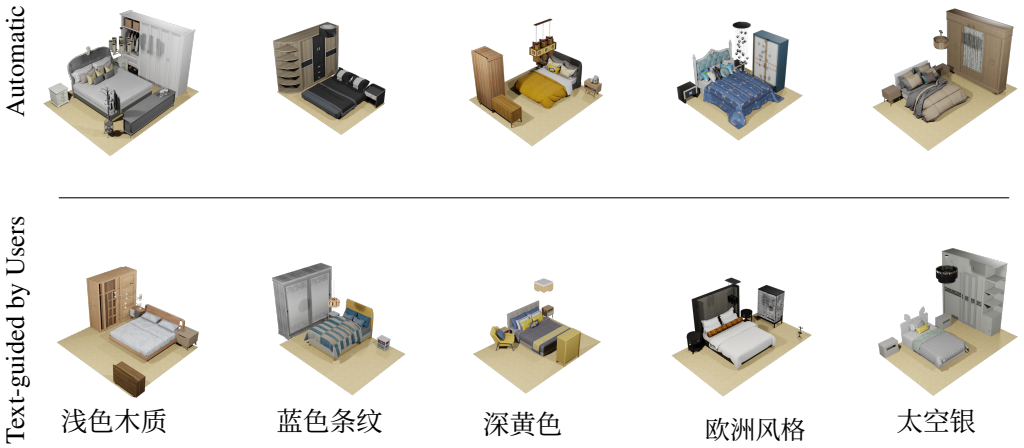


图1：**自动场景合成：**给定平面图和可选的风格描述提示作为条件，我们的模型可以合成兼具风格一致性、多样性和视觉真实性的房间。

摘要

室内场景合成需要在平面图上自动、恰当地挑选并摆放家具，使场景看起来真实且在功能上合理。这类场景可用作沉浸式3D体验中的家居环境，也可用于训练具身智能体。现有方法通过家具的标注类别（如床、椅子或桌子）来生成符合上下文的家具组合。无论是基于启发式规则还是学习得到的方法，这些方法都忽略了物体的实例级视觉属性，因此可能生成视觉上不够协调的场景。

本文介绍了一种自回归场景模型[1]，该模型基于 CLIP 的通用图像嵌入，可输出实例级预测。这使我们能够学习颜色和风格匹配等视觉对应关系，并生成在功能上更合理、在美学上更令人愉悦的场景。在 3D-FRONT数据集上进行评估时，我们的模型在场景合成方面取得了最先进水平，并将自动补全指标提升了 50% 以上。此外，基于嵌入的方法支持零样本文本引导场景合成与编辑，并能轻松泛化到训练期间未见过的家具。

[1]代码将在录用后发布。

预印本。审稿中。

1 引言

自动生成房间、住宅和建筑等室内空间的三维模型具有许多重要应用 [1]。这些应用包括创建虚拟环境，以在元宇宙中带来沉浸式体验；生成虚拟测试平台，用于训练和评估具身智能体；以及辅助现实世界中物理空间的设计。设计良好的室内场景合成模型需要生成满足以下条件的场景：

1. 在物理和功能上合理。家具应摆放得当，例如床应放在地板上，而不是悬浮在空中；冰箱应靠墙放置，而不是摆在房间中央，等等。家具的组合应符合房间类型，例如不应把床放在浴室里，也不应把多盏落地灯并排摆放。2. 视觉协调且美观。人们在布置场景时关注的不只是功能属性，还会考虑两件家具的风格是否匹配、颜色之间的对比是否协调等。模型应能够捕捉这类属性。3. 多样且具有泛化能力。现实世界中的室内空间在类型和复杂程度上差异很大，从典型住宅中的简易卧室到开放式办公空间，不一而足。这类空间中可能出现的独特物体数量几乎无穷无尽。优秀的模型需要能够捕捉这种多样性，并能很好地泛化到训练时未见过的新家具库。

以往的工作要么使用程序化建模技术 [2, 3, 4, 5, 6, 7, 8, 9, 10] 进行合成，要么将该问题视为场景图生成 [11, 12, 13, 14, 15, 16, 17, 18, 19]。这两类方法都依赖人工编码的规则和启发式规则来确保结果合理。尽管这些方法效率很高，但生成场景的多样性最终受限于有限的手工规则集。如果将某些美学约束显式编码，这些方法便能够捕捉它们；不过，这仅在所需属性包含于对象元数据中时才可行，而且效果有限。

近期还有一类工作（我们的方法也属于这一类）提出从数据中学习家具选择与摆放方式 [20, 21, 22, 23]。随着训练数据增多，这种方法具有更强的泛化能力。然而，以往的工作对场景分布的建模仅达到对象类别这一层级，因而难以学习颜色和风格等独特的家具属性。这类模型能够学会将椅子摆在桌子旁边，却无法推荐一把具体的椅子。这与人类设计室内空间的方式形成对比：人们会花费大量精力，从相似家具的不同款式中做出选择，确保风格协调、整体构图赏心悦目。

为了捕捉这种细微的视觉对应关系，我们需要超越简单的类别标签，采用更丰富的物体视觉表征。为此，我们使用物体在多个视角下进行多次渲染后得到的汇聚式 CLIP [24] 图像嵌入。使用在来自互联网的大规模图像数据上进行预训练的图像编码器，使我们能够表示现实世界中物体和风格的全部多样性。我们特别选择 CLIP 模型，是因为它还支持文本引导的编辑功能。我们训练了一个自回归模型 CLIP-Layout，该模型在每一步输出预测的家具嵌入及其摆放位置。在推理过程中，我们利用预测的嵌入检索下一个家具，并根据场景中已有的内容对其进行条件约束，同时还可以接受用户提供的可选风格描述文本引导。

CLIP-Layout 可以从数据中学习生成自然、多样且风格一致的场景（图1）。无需重新训练，它就能轻松泛化到新的家具，并支持文本引导的场景合成应用（见第4.3节）。在实例级别对物体进行建模还赋予了 CLIP-Layout 更强的表示能力，从而实现更准确的场景合成。在 3D-FRONT数据集 [25], 上进行评估时，与此前的最先进方法相比，我们的自动补全精确率和召回率均得到显著提升，相对提升幅度为 22% 至 129%。

2 相关工作

2.1 场景合成

室内场景合成的目标是生成同时满足功能和美学标准的家具布局 [1]。早期尝试 [2, 3, 4, 5] 主要依赖手工设计规则和约束

来放置物体。此后，人们探索了程序化建模 [9, 8, 10] 和概率文法 [6, 7] 等多种技术。这些技术能够生成令人印象深刻的样本，但在构图时可能忽视视觉连贯性。 [26, 27]并非在空间中放置明确的物体，而是尝试使用隐式函数表示场景，以对空间中三维点的占用情况和属性进行建模。 [28] 使用文生图模型，并持续对图像进行对齐和融合，将其转化为房间。尽管隐式方法生成的结果很逼真，但对于需要推理单个物体或涉及与场景进行物理交互的下游任务而言，它们可能不太理想。另一类研究 [11, 12, 13, 14, 15, 16, 19, 17, 29] 专注于设计专用架构并使用图来捕捉场景的底层结构。近年来，许多研究开始考虑场景合成中超越预测家具类别和姿态的其他方面。 [30, 31] 旨在纳入经过标注的人体动作， [32] 解决将多个输入房间合并为一个新房间、同时最大化其共同功能空间的问题， [33] 提出了一种在场景数据有限时也能表现良好的模型。受早期数据驱动研究的启发，这些研究使用卷积神经网络和基于Transformer的架构进行自回归生成 [20, 21, 22, 23], , 我们也希望让模型进一步考虑家具的视觉线索，并生成风格一致的场景。与我们的研究方向类似， [18]使用预训练的体积模块，仅对无纹理的三维点云进行编码。但 CLIP-Layout 适用于任何可渲染的三维形状，且只需要家具模型及其变换作为标签。在 [22, 23], 的基础上，我们的模型使用置换不变Transformer [23, 34] 生成一组家具预测结果，其中包含语义嵌入和姿态。

2.2 多模态嵌入

能够以语义和视觉信息编码三维形状，是构建具备美学感知的场景的关键。针对三维形状表示已有诸多研究工作，例如使用不规则潜在网格 [35], 三平面特征 [36], 和神经场 [37]。然而，这些方法更关注拓扑信息而非语义信息，并且需要大量三维或多视图数据进行训练。在二维领域，CLIP [24] 从数亿个图文对中学习文本和图像的联合嵌入空间。CLIP由文本编码器和图像编码器组成，并通过对比损失进行优化，以确保具有相似潜在语义的图像和文本被编码为在嵌入空间中距离更近的特征向量。借助这一大规模预训练的多模态CLIP编码器，我们可以将带纹理形状嵌入这一任务从三维转移至二维，泛化到未见过的家具实例，并利用学习到的各种视觉模式的先验知识来完成文本驱动应用。

3 方法

3.1 问题设定

令 $X = X_1, \ldots\ldots, X_2$ 表示一组场景，其中每个场景 $X_i = (O_i, F_i)$由无序的物体集合 $O_i = \{o_j^i\}_{j=1}^M$及其地面布局 F_i构成。我们的目标是对场景 $p_\theta(X)$的分布进行建模，可表示为:

$$p_\theta(X_i) = p(F_i)p_\theta(O_i|F_i) = p(F_i) \sum_{O \in \pi(O_i)} \prod_{j \in O} p_\theta(o_j^i | o_{<j}^i, F_i) \tag{1}$$

其中， $p(F_i)$是给定的场景平面图先验分布， $p_\theta(o_j^i|o_{<j}^i, F_i)$是在给定平面图以及第 i 个部分场景中的所有物体的条件下，第 j 个物体出现的概率； $\pi(\cdot)$是置换算子。我们的目标是最大化该函数的对数似然。遵循[23], , 我们将求和转换为乘积，得到目标函数:

$$\log p_\theta(X_i) = \log(\prod_{O \in \pi(O_i)} \prod_{j \in O} p_\theta(o_j^i | o_{<j}^i, F_i)) = \sum_{O \in \pi(O_i)} \sum_{j \in O} \log p_\theta(o_j^i | o_{<j}^i, F_i) \tag{2}$$

从求和改为求积在计算上更可行，也可能促使所有排列都具有更高的密度[23]。在我们的实现中，每个家具实例都有一个语义嵌入和一个三维变换，其中包括平移、尺寸以及绕垂直轴旋转的角度。平面图采用二值掩码的形式，用于标示家具摆放的边界。

3.2 架构

我们的模型主要由三个部分组成：1）第 3.2.1 节中用于房间平面图、变换和平移嵌入的编码器；2）第 3.2.2 节中用于场景内所有编码后实例集合的 Transformer解码器；以及 3）第 3.2.3 节中用于预测每个时间戳处下一个实例的解码器。

3.2.1 平面图、变换与嵌入编码器

我们的合成过程以给定的平面图为条件，该平面图在方形网格上编码为二值掩码。随后，将此掩码图像输入预训练ResNet-18特征提取器 [38], ，其最终层用作平面图的特征向量 F。

每个家具实例都有一个七维张量，用于表示其在三维空间中的变换：三维变换、三维尺寸以及沿垂直轴的一维旋转。我们仅对沿垂直轴的旋转进行建模，因为数据集中的所有家具都仅以这种方式旋转，并且在需要时将其推广到三维旋转也很直接。根据训练数据集统计信息，并考虑第3.3节讨论的可能场景增强，我们将所有变换线性归一化到 $[0, 1]$ 中。对于属性的每个维度，我们遵循 [34]中的固定正弦-余弦位置编码方案 $\gamma(\cdot)$。家具实例还具有语义嵌入向量 h （第3.2.4节），我们使用多层感知机将其投影到更小的空间中。编码后的属性被拼接成单一特征向量，并通过线性方式将其投影到512维，即 C_j；该向量表示第j个经过变换的物体实例的特征，随后将被输入Transformer解码器。

3.2.2 Transformer解码器

在 [23], 中，我们使用仅解码器式Transformer [34, 39, 40] ，且不采用任何位置编码，因为我们希望模型忽略给定家具集合的顺序，从而生成置换不变特征。Transformer的输入集合为 $I = \{F\} \cup \{C_j\}_{j<i} \cup q$ ，用于预测第 i个物体实例，其中 $q \in \mathbf{R}^{64}$ 是一个学习查询向量[23]。

3.2.3 物体实例解码器

在每个步骤中，我们首先预测一个停止标记，以决定模型是否应停止添加更多家具。这通过一个简单的多层感知机头实现，其输出为两个logits。随后，将该输出附加到$\hat{}q$ ，并将其输入嵌入解码器，以预测下一个物体实例；嵌入解码器输出一个与原始对象嵌入维度相同的向量。基于此，我们将其投影到数据集中所有可能家具的输出空间中。通过对预测的原始嵌入 h 与所有家具嵌入的点积应用softmax，我们输出该词汇表$\hat{}$上的一个分布。在物体实例检索过程中，我们应用温度缩放和Top-K采样。

一旦确定停止条件和实例嵌入，模型便会以级联方式相应地输出平移、旋转角度和尺寸。我们选择将所有变换属性建模为逻辑斯谛分布混合，如 [41, 23] 所示，因为实验发现，这种方法的表现优于高斯混合等其他设计方案。

3.2.4 对象嵌入

近年来，已有许多研究探讨三维物体的表示学习 [42, 43, 36, 44]。尽管这些方法在各自针对的应用中非常有效，但它们需要大量三维数据进行训练，而我们可能无法获得所需规模的数据。此外，这些表示旨在用于网格表面重建，不一定适合我们的应用。为了捕捉真实室内场景中可能出现的丰富多样的物体和风格，我们提出利用一种现有的预训练图像表示，该表示基于来自网络的大规模图像数据进行训练。具体而言，我们选用多模态 CLIP 模型 [24], ，其额外优势在于提供了对齐的文本嵌入，从而支持文本引导合成和编辑等多模态应用。

为了使用现成的二维图像表示来编码三维物体，我们依赖从多个视点渲染物体，并对这一组渲染图的嵌入进行池化。对于每个物体实例，我们首先对其进行归一化，确保它位于居中的单位立方体内，且朝向和向上方向正确。然后，我们手动将漫反射、环境光和镜面反射

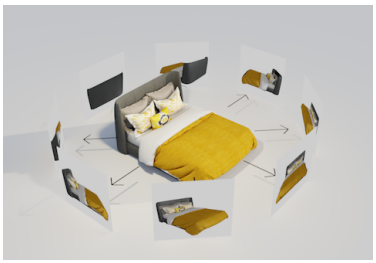


图2：**对象嵌入：** CLIP-Layout通过将从八个标准方向渲染的三维网格图像输入CLIP图像编码器，计算其语义嵌入。

属性设为固定值，以确保一致性。随后，我们从上半球上的八个规范相机方向（方位角）渲染该归一化形状的图像，并固定相机仰角和距离，同时采用相同的照明和着色设置（详见附录B），如图2所示。得到的八张图像 $\{I_i\}_8$ 将输入预训练CLIP图像编码器[24]，该编码器接收一张RGB图像并输出一个$\mathbf{R}^{512}$张量。最终的嵌入是八个独立嵌入的归一化平均值，$h = \frac{1}{8}\sum_{i=1}^{8} normalized(\text{CLIP}(I_i))$。我们发现，这种编码方法在图像检索或基于文本的对象检索中表现良好，可作为风格感知场景生成器的有力基础。

使用预训练的图像编码器，我们能够表示每个家具实例的细致视觉特征，同时不受对象格式的限制。由于我们不再依赖类别标签，因此也降低了对每件家具类别标注的数据要求。

3.3 训练与推理

为了在训练过程中优化公式（2）中的目标，对于数据集中的每个场景，我们首先从场景中选取一个大小为 T 的随机子集，并对其进行随机排列。随后，我们使用各自的编码器对该物体子集和平面图进行编码。在编码特征的条件下，我们的网络首先预测是否添加任何新物体，然后解码其嵌入和变换。对于停止标记，我们采用带标签平滑的标准二元交叉熵损失。对于嵌入，我们首先计算预测嵌入与家具空间中所有实例嵌入的点积，然后使用交叉熵损失。最后，由于变换被建模为逻辑斯蒂混合分布，我们对变换采用解析形式的负对数似然。

为了提高训练数据的质量和多样性，我们尝试了以随机角度旋转场景、沿 x 或 z 轴进行镜像，以及实例变换抖动。我们发现最后一种方法效果较差，因此在实验中采用前两种方法的组合。

推理时，我们向网络提供从给定先验分布中采样得到的平面图，并可像场景补全任务那样额外提供一组初始家具。从候选池中选出新物体后，我们会将相应的真实嵌入反馈给网络作为输入，并与上一步的输出一同用于预测其他属性。最后，我们根据预测的位姿变换家具，并将其添加到场景中。

4 实验

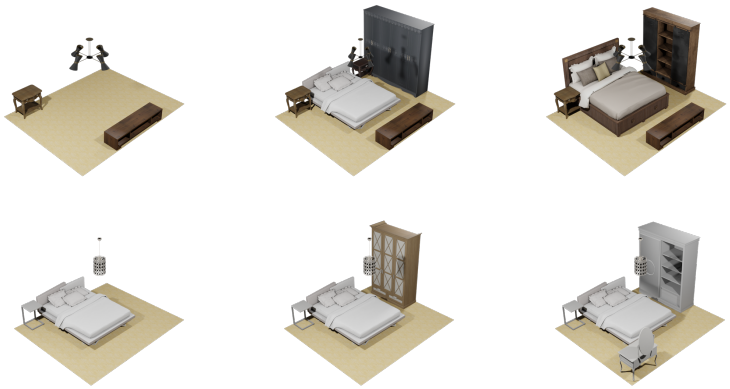
本节中，我们评估 CLIP-Layout 在部分场景补全、无条件生成、文本引导生成和编辑任务上的表现。我们将 ATISS [23] 作为主要基线，并使用其已发布的代码库，在我们采用的数据集版本上重新训练其模型，以进行公平比较。

4.1 数据集

我们在所有实验设置中均使用 3D-FRONT[25] 作为数据集。该数据集由人工设计的室内场景组成，并包含来自互补数据集 3D-FUTURE[45]中每个家具物体的标注。对于每个场景，我们只需要每件家具的变换和原始网格模型的标注。为便于学习和公平比较，我们遵循了 [23] 中采用的相同数据集筛选流程，例如移除尺寸不自然的房间、家具拥挤的房间

表1：筛选后的数据集统计信息

	旧数据集		新数据集		
房间类型	总计	训练集	Test	总计	家具
卧室	5996	3879	162	4041	2398
餐厅	2625	723	177	900	2247
图书馆	622	230	56	286	464
客厅	2962	621	192	813	2380



部分　　ATISS　　Ours

图3：**部分场景补全：**从左到右各列分别为部分场景、ATISS 补全后的场景，以及由我们的模型生成的场景。CLIP-Layout 不仅能够区分颜色信息，还能区分家具材质和形状。

以及形状不合理的房间。不幸的是，经过筛选后，这个数据集的新版本所包含的数据比原始版本更少，而原始版本现已无法获取[2]（数据集统计信息汇总于表 1）。

4.2 基于平面图的场景合成

We 首先评估我们的场景模型根据平面图生成合理且多样的 e
sce场景的能力，其中部分场景中可选择性地包含已有家具。

4.2.1 定量评估

为了定量评估模型的性能，我们旨在比较模型补全部分场景的结果与真实测试集有多接近。对于每个测试场景，我们随机移除 N 个物体，并让模型进行补全。我们重点关注两个指标：

1. 检索指标： 模型通过从资产库中挑选并摆放 M 新物体来补全部分场景。为了衡量模型检索真实物体的能力，我们报告 M 检索物体相对于 N真实家具集合的精确率和召回率。2. 场景分布间距离： 对于三维场景，我们采用了与第 3.2.4 节类似的设置，从场景中渲染出八张图像作为视觉表示。随后，我们使用 FID 和 CLIP-FID 分数 [46] 比较真实测试场景与自动补全场景的图像集合。基于预训练视觉表示的 Inception 距离分数原则上可以衡量场景的整体质量，包括自然摆放和连贯性。我们认为，透视渲染图能更好地呈现场景的视觉细节，而以往工作中使用的俯视图则只能

[2]https://github.com/nv-tlabs/ATISS/issues/18

表2：单步部分场景补全的指标

房间类型	模型	FID(↓)	CLIP-FID(↓)	精确率(↑)	召回率(↑)
卧室	ATISS	8.66	0.33	25.5%	23.0%
	CLIP-Layout	**7.02**	**0.25**	**43.1%**	**36.7%**
餐厅	ATISS	8.87	0.27	17.5%	14.8%
	CLIP-Layout	**6.36**	**0.19**	**32.2%**	**26.9%**
图书馆	ATISS	33.25	1.68	10.9%	9.6%
	CLIP-Layout	**26.23**	**1.31**	**24.8%**	**22.1%**
客厅	ATISS	7.38	0.19	11.2%	9.6%
	CLIP-Layout	**5.01**	**0.13**	**21.3%**	**15.7%**

表3：卧室类型部分场景补全的指标

预填充 #	模型	FID(↓)	CLIP-FID(↓)	精确率(↑)	召回率(↑)
1 件家具	ATISS	**16.38**	**0.80**	21.6%	23.0%
	CLIP-Layout	17.83	1.03	**36.5%**	**31.6%**
2 件家具	ATISS	14.87	0.71	26.3%	29.5%
	CLIP-Layout	**14.08**	**0.67**	**40.7%**	**42.1%**
3 件家具	ATISS	13.51	0.64	26.9%	31.2%
	CLIP-Layout	**12.11**	**0.53**	**40.4%**	**38.2%**
4 件家具	ATISS	12.53	0.63	26.7%	31.4%
	CLIP-Layout	**10.80**	**0.46**	**38.4%**	**41.9%**
5 件家具	ATISS	11.84	0.59	26.2%	31.5%
	CLIP-Layout	**9.88**	**0.42**	**39.1%**	**42.5%**

提供有关布局的信息。我们以 512×512 分辨率渲染图像，以增强视觉辨识度。

在表3中，我们总结了“卧室”类型的场景补全结果。我们逐步移除家具，直到场景中剩下1至5个实例。我们分别报告场景中已有物体数量不同时的指标。当这个数量较少时（例如1至2个），任务更接近无条件生成，因为模型在构建完整场景时拥有更大的自由度。随着场景中保留的家具增多（例如4至5个），问题会受到更多约束，因为能够一致地补全场景的方式更少。我们发现，CLIP-Layout在大多数情况下表现良好，精确率平均提升了53%，召回率平均提升了34%。

从FID分数来看，CLIP-Layout在大多数情况下优于基线，而且随着预置家具数量增加，提升幅度也会增大。对于场景严重不完整的情况，尤其是仅有1件家具的设置，我们的方法在FID分数上不如基线。我们推测，模型可能过度校准于精确率而非召回率，因此在这些情况下容易过早停止。我们将在第5节进一步讨论局限性和潜在解决方案。

所有房间类型的完整指标见附录表4。表2汇总了所有房间类型的结果，其中我们仅要求模型在不完整的场景中补全一件家具。所有定量结果均为使用固定随机种子进行五次运行的平均值。与基线相比，CLIP-Layout 在所有房间类型和各项指标上均有显著优势。尽管其他房间类型的数据集规模较小，但由于提升幅度较大，这些结果仍具有显著意义。

4.2.2 定性结果

在本节中，我们提供了一些视觉示例，以展示我们的模型生成的场景优于基线。图4对比了两个模型从平面图进行（无条件）场景合成的结果，图3则展示了部分场景补全（

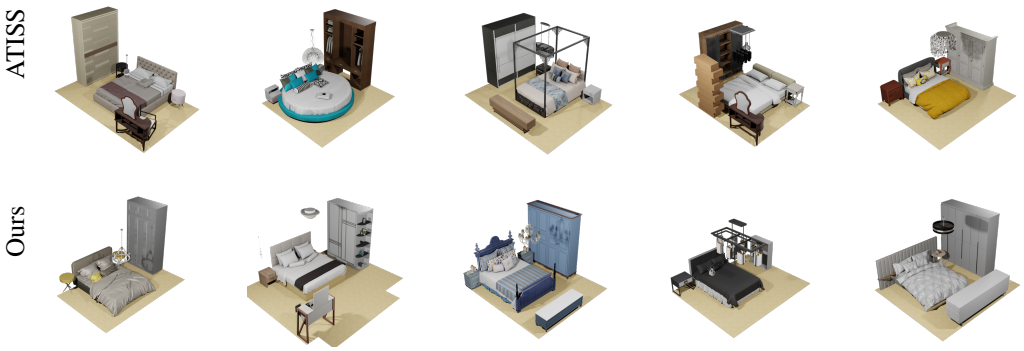


图4：**从房间掩码进行场景合成**：由平面图生成的场景视觉示例。顶排为 ATISS 生成的样本，底排为 CLIP-Layout() 生成的样本。

更多示例以及与真实值的对比见附录图9、10和11）。请注意，在这两种情况下，尽管基线模型能够生成看似合理的房间布局配置，却无法捕捉数据集中的风格信息。CLIP-Layout能够生成遵循特定模式（如风格和颜色）的场景，因此样例在艺术表现上更加逼真，设计也更加出色，例如家具风格统一为现代风格，或采用某种主导色（更多CLIP-Layout示例见图1）。

4.3 文本引导应用

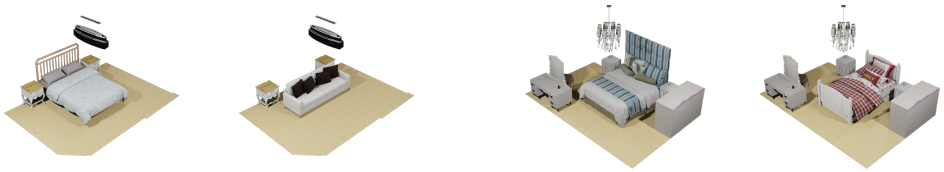
得益于家具语义 CLIP 嵌入，CLIP-Layout 能够自然地执行许多文本条件任务，例如场景合成和文本引导家具替换，并取得不错的零样本性能。

4.3.1 文本引导场景合成

我们首先展示模型能够接收用户指定的风格提示，并据此生成场景。具体而言，我们将文本提示嵌入为编辑向量 t，再将其缩放至与当前预测嵌入 $\hat{e}$ 的范数相同的长度，随后通过权重为 w 的线性插值将二者合并，最后用于物体检索。在我们的试验中，我们发现权重取 0.2 到 0.5 效果较好。为避免干扰模型学到的风格一致性，我们在每个时间戳对权重因子应用指数衰减。这种方法有几个优点：1）模型能够利用学到的风格一致性先验，在后续阶段减少对提示的依赖；2）当目标家具与数据分布相距过远时，模型不会受到编辑向量太大影响，仍可从可用选项中选择最佳项。图1通过示例提示定性展示了生成场景的多样性和忠实度（更多示例见附录图8）。CLIP-Layout 既能适应“木质”或“条纹”等简单的材质或纹理线索，也能响应“欧洲风格”等更抽象的设计提示。据我们所知，这是首次成功展示在零样本设置下具备实例级操控能力的文本引导场景合成。

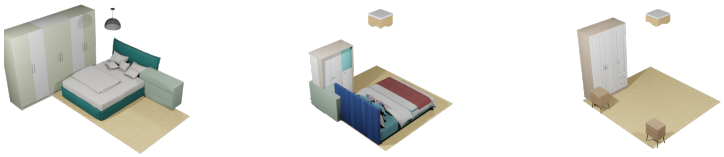
4.3.2 文本引导的家具替换

除了合成整个场景外，CLIP-Layout 还能够通过文本修改特定实例。为此，我们使用文本描述检索目标物体，并通过同一个 CLIP 嵌入器对文本描述进行嵌入。如图5所示，我们展示了若干示例，表明模型能够可靠地检索相关家具实例，并据此进行摆放。请注意，CLIP-Layout 会根据实例级信息调整新物体的摆放位置，并可在需要时重新调整，而不是盲目地在原位置替换之前的物体（例如，将床换成休闲沙发时）。更多示例见附录图7。



(a) 从床变为“黑白休闲沙发” (b) 从蓝色床变为“红条纹床”

图5：**使用文本提示进行家具替换：** 每对图像左侧为原始场景，右侧为替换家具后的场景，图像下方为文本提示。



(a) 越界 (b) 双重放置 (c) 提前停止

图6： **常见的失败示例**。

5 局限性

由于训练数据集（表1）相对较小，我们的模型有时无法生成合理的结果。图6列出了三种常见的失败模式：图6a展示了家具被放置在平面图之外的情况。尽管这种场景在视觉上并不真实，我们发现训练数据中也存在类似的不合理场景。由于我们的模型旨在完全由数据驱动，如果有需要，我们将把在数据集中注入人类设计先验的工作留待未来，并采用学习方法 [47] 或手动约束。有时，模型会像图6b那样放置重复的家具。我们推测这类似于文本生成模型中不必要的重复短语。最后一种常见问题是图6c所示的提前停止，这是在标签分布严重不平衡的情况下，为停止标记单独训练一个预测头所产生的副作用。我们认为，通过改进架构、增加训练数据，或对合成长度进行显式校准，或许可以解决这些不足。遗憾的是，构建具有密集标注的大规模、高质量室内场景数据集既昂贵又耗时。这或许能成为投入数据收集工作的良好动因，或者促使人们探索无监督和半监督方法。

6 结论

本文提出了 CLIP-Layout，这是一种具有语义家具嵌入的风格一致室内场景模型，能够合成和自动补全场景，使其符合数据集的风格与美学，同时保留家具配置。得益于多模态特性和语义家具嵌入，CLIP-Layout 能够完成许多由用户控制的任务，例如文本引导的家具替换，以及具有良好零样本性能的场景合成。我们通过定性和定量评估表明，在不同设置下，我们的模型均优于基线。据我们所知，我们的模型首次将家具实例的视觉特征纳入考量。我们希望 CLIP-Layout 能够扩展到规模更大、更多样化的数据集，并应用于包括完整建筑室内空间在内的新任务，从而支持许多需要自动场景合成的下游任务，例如导航智能体训练、虚拟世界构建、空间计算平台和建筑设计。

参考文献

[1] Song Hai Zhang、Shao Kui Zhang、Yuan Liang 和 Peter Hall。3D室内场景合成综述。《计算机科学与技术学报》，34(3):594–608，2019年5月。[2] Richard W. Bukowski 和 Carlo H. Séquin。对象关联：一种简单实用的虚拟3D操作方法。载于*ACM交互式3D*图形与游戏研讨会，1995年。[3] Ken Xu、James Stewart 和 Eugene Fiume。用于场景构成的基于约束的自动放置。第25–34页，2002年5月。[4] Tobias Germer 和 Martin Schwarz。用于实时漫游的家具程序化布局。《计算机图形论坛》，28，2009年。[5] Paul Merrell、Eric Schkufza、Zeyang Li、Maneesh Agrawala 和 Vladlen Koltun。运用室内设计准则进行交互式家具布局。载于*ACM SIGGRAPH 2011*论文集，SIGGRAPH '11，美国纽约州纽约市，2011年。计算机协会。[6] Amlan Kar、Aayush Prakash、Ming-Yu Liu、Eric Cameracci、Justin Yuan、Matt Rusiniak、David Acuna、Antonio Torralba 和 Sanja Fidler。Meta-sim：学习生成合成数据集，2019年。[7] Jeevan Devaranjan、Amlan Kar 和 Sanja Fidler。Meta-sim2：通过无监督学习进行场景结构学习，以生成合成数据，2020年。[8] Aayush Prakash、Shaad Boochoon、Mark Brophy、David Acuna、Eric Cameracci、Gavriel State、Omer Shapira 和 Stan Birchfield。结构化域随机化：通过上下文感知合成数据弥合现实差距，2018年。[9] Josh Tobin、Rachel Fong、Alex Ray、J onas Schneider、Wojciech Zaremba 和 Pieter Abbeel。通过域随机化将深度神经网络从仿真迁移到现实世界，2017年。[10] Matt Deitke、Eli VanderBilt、Alvaro Herrasti、Luca Weihs、Jordi Salvador、Kiana Ehsani、Winson Han、Eric Kolve、Ali Farhadi、Aniruddha Kembhavi 和 Roozbeh Mottaghi。ProcThor：利用程序化生成实现大规模具身人工智能，2022年。[11] Pascal Müller、Peter Wonka、Simon Haegler、Andreas Ulmer 和 Luc Van Gool。建筑物的程序化建模。《*ACM*图形学汇刊》，25(3):614–623，2006年。[12] Jerry O. Talton、Yu Lou、Steve Lesser、Jared Duke、Radomír M˘ech 和 Vladlen Koltun。Metropolis程序化建模。《*ACM*图形学汇刊》，30(2)，2011年。[13] Manyi Li、Akshay Gadi Patil、Kai Xu、Siddhartha Chaudhuri、Owais Khan、Ariel Shamir、Changhe Tu、Baoquan Chen、Daniel Cohen-Or 和 Hao Zhang。Grains：用于室内场景的生成式递归自编码器，2018年。[14] Kai Wang、Yu-An Lin、Ben Weissmann、Manolis Savva、Angel X. Chang 和 Daniel Ritchie。Planit：利用关系图和空间先验网络规划并实例化室内场景。《*ACM*图形学汇刊》，38(4)，2019年。[15] Yang Zhou、Zachary While 和 Evangelos Kalogerakis。Scenegraphnet：用于3D室内场景增强的神经消息传递，2019年。[16] Andrew Luo、Zhoutong Zhang、Jiajun Wu 和 Joshua B. Tenenbaum。场景布局的端到端优化，2020年。[17] Pulak Purkait、Christopher Zach 和 Ian Reid。Sg-vae：用于生成新室内场景的场景文法变分自编码器，2019年。[18] Zaiwei Zhang、Zhenpei Yang、Chongyang Ma、Linjie Luo、Alexander Huth、Etienne Vouga 和 Qixing Huang。通过混合表示进行场景合成的深度生成建模，2018年。[19] Song-Hai Zhang、Shao-Kui Zhang、Wei-Yu Xie、Cheng-Yang Luo 和 Hong-Bo Fu。通过离散且精确的布局模式提取实现快速 3D 室内场景合成，2020 年。[20] Daniel Ritchie、Kai Wang 和 Yu-an Lin。通过深度卷积生成模型实现快速灵活的室内场景合成，2018年。[21] Kai Wang、Manolis Savva、Angel X. Chang 和 Daniel Ritchie。用于室内场景合成的深度卷积先验分布。《*ACM*图形学汇刊》，37(4)，2018年。[22] Xinpeng Wang、Chandan Yeshwanth 和 Matthias Nießner。Sceneformer：使用Transformer生成室内场景，2020年。[23] Despoina Paschalidou、Amlan Kar、Maria Shugrina、Karsten Kreis、Andreas Geiger 和 Sanja Fidler。ATISS：用于室内场景合成的自回归Transformer，2021年。[24] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger 和 Ilya Sutskever。通过自然语言监督学习可迁移视觉模型，2021年。

[25] Huan Fu、Bowen Cai、Lin Gao、Lingxiao Zhang、Jiaming Wang Cao Li、Zengqi Xun、Chengyue Sun、Rongfei Jia、Binqiang Zhao 和 Hao Zhang。《3D-FRONT：具有布局和语义的3D家具房间》，2020年。[26] Zhaoxi Chen、Guangcong Wang 和 Ziwei Liu。《SceneDreamer：从二维图像集合生成无界3D场景》，2023年。[27] Miguel Angel Bautista、Pengsheng Guo、Samira Abnar、Walter Talbott、Alexander Toshev、Zhuoyuan Chen、Laurent Dinh、Shuangfei Zhai、Hanlin Goh、Daniel Ulbricht、Afshin Dehghan 和 Josh Susskind。《Gaudi：用于沉浸式3D场景生成的神经架构师》，2022年。[28] Lukas Höllein、Ang Cao、Andrew Owens、Justin Johnson 和 Matthias Nießner。《Text2Room：从二维文生图模型中提取带纹理的3D网格》，2023年。[29] Mohammad Keshavarzi、Aakash Parikh、Xiyu Zhai、Melody Mao、Luisa Caldas 和 Allen Y. Yang。《SceneGen：利用场景图先验进行生成式上下文场景增强》，2020年。[30] Siyuan Qi、Yixin Zhu、Siyuan Huang、Chenfanfu Jiang 和 Song-Chun Zhu。《使用随机文法进行以人为中心的室内场景合成》，2018年。[31] Hongwei Yi、Chun-Hao P. Huang、Shashank Tripathi、Lea Hering、Justus Thies 和 Michael J. Black。《Mime：具备人体感知的3D场景生成》，2022年。[32] Mohammad Keshavarzi、Michael Zollhoefer、Allen Y. Yang、Patrick Peluse 和 Luisa Caldas。《用于混合现实远程临场的互场景合成》，2022年。[33] Mohammad Keshavarzi、Flaviano Christian Reyes、Ritika Shrivastava、Oladapo Afolabi、Luisa Caldas 和 Allen Y. Yang。《通过GSACNet进行上下文场景增强与合成》，2021年。[34] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。《注意力就是一切》，2017年。[35] Biao Zhang、Matthias Nießner 和 Peter Wonka。《3DILG：用于3D生成建模的不规则潜在网格》，2022年。[36] Jun Gao、Tianchang Shen、Zian Wang、Wenzheng Chen、Kangxue Yin、Daiqing Li、Or Litany、Zan Gojcic 和 Sanja Fidler。《GET3D：一种从图像中学习的高质量带纹理3D形状生成模型》，2022年。[37] Yiheng Xie、Towaki Takikawa、Shunsuke Saito、Or Litany、Shiqin Yan、Numair Khan、Federico Tombari、James Tompkin、Vincent Sitzmann 和 Srinath Sridhar。《视觉计算及其他领域中的神经场》，2021年。[38] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。《用于图像识别的深度残差学习》，2015年。[39] A. Katharopoulos、A. Vyas、N. Pappas 和 F. Fleuret。《Transformer是RNN：具有线性注意力的快速自回归Transformer》。载于国际机器学习大会（*ICML*）论文集，2020年。[40] A. Vyas、A. Katharopoulos 和 F. Fleuret。《采用聚类注意力的快速Transformer》。*arXiv*预印本 *arXiv:2007.04825*，2020年。[41] Tim Salimans、Andrej Karpathy、Xi Chen 和 Diederik P. Kingma。《PixelCNN++：通过离散逻辑斯谛混合似然及其他改进提升PixelCNN》，2017年。[42] Aditya Sanghi、Hang Chu、Joseph G. Lambourne、Ye Wang、Chin-Yi Cheng、Marco Fumero 和 Ka- mal Rahimi Malekshan。《CLIP-Forge：迈向零样本文本到形状生成》，2021年。[43] Ben Poole、Ajay Jain、Jonathan T. Barron 和 Ben Mildenhall。《DreamFusion：使用二维扩散进行文本到3D》。*arXiv*，2022年。[44] Xiaohui Zeng、Arash Vahdat、Francis Williams、Zan Gojcic、Or Litany、Sanja Fidler 和 Karsten Kreis。《LION：用于3D形状生成的潜在点扩散模型》，2022年。[45] Huan Fu、Rongfei Jia、Lin Gao、Mingming Gong、Binqiang Zhao、Steve Maybank 和 Dacheng Tao。《3D-FUTURE：带纹理的3D家具形状》，2020年。[46] Gaurav Parmar、Richard Zhang 和 Jun-Yan Zhu。《关于混叠缩放及GAN评估中的意外细节》。载于计算机视觉与模式识别会议，2022年。[47] Kurt Leimer、Paul Guerrero、Tomer Weiss 和 Przemyslaw Musialski。《LayoutEnhancer：从不完美数据生成优质室内布局》。载于亚洲计算机图形学会议*2022*论文集，SA '22，纽约，美国，2022年。计算机协会。[48] Katharopoulos Angelos 和 Despoina Paschalidou。simple-3dviz。https://simple-3dviz.com，2020年。[49] Blender在线社区。*Blender*——一款*3D*建模与渲染软件包。Blender基金会，Blender基金会（荷兰），阿姆斯特丹，2018年。

7 CLIP-Layout 附录：基于语义家具嵌入的风格一致室内场景合成

A 完整场景补全指标

在第4.2.1节中，我们列出了卧室类型的指标，因为卧室场景拥有最多具有正确布局和家具数量的数据。这里我们在表2中提供所有房间类型的完整评估指标表。总体而言，关于卧室场景得出的结论同样适用于其他房间类型。

B 渲染设置

为便于可复现性，本节详细说明本实验特有的渲染设置。我们首先讨论网格物体材质标准化过程，然后介绍用于物体和场景八视角渲染的相机设置。

由于模型数据集 [45] 中有许多实例具有异常的材质属性，例如漫反射为 0 或镜面反射为 1，这会使渲染图像看起来过暗或过亮。为了更一致地渲染图像，并使其能够泛化至所有网格，我们在从数据集加载网格时手动调整材质属性。在实验中，我们发现将环境光、漫反射和镜面反射设置为 (0.4, 0.4, 0.1) 的比例，能够获得最一致的照明效果。
我们规定了从八个标准方向渲染物体或场景图像以创建对象嵌入并评估模型时所使用的设置。我们使用 [48] 作为渲染软件包，因其高效且简洁，并使用 EGL 作为后端。我们在表5中汇总了相机设置，未指定的所有参数均采用默认值。

请注意，评估时我们使用 [48] 作为渲染软件包，以提高效率；但论文中展示的所有图均经过后期处理，使用 Blender[49] 的 CYCLE 引擎在三点布光设置下进行光线追踪渲染。此外，我们固定了平面图的纹理，以确保评估期间的选择不会影响任何指标。

C 更多视觉示例

本节中，我们提供了更多由模型在不同设置下生成的场景视觉样例，以补充前文的定量与定性结果及分析：

1. 文本引导家具替换：图7展示了通过描述新家具的文本提示，将现有场景中的某件家具替换为新家具的示例。2. 文本引导的合成：图8列出了更多示例，展示模型如何生成符合文本提示中视觉线索的场景。3. 根据平面布局进行场景合成：图9展示了模型根据测试集平面图生成的更多场景示例，此外，图中还给出了一些 ATISS 的样例作为对比。4. 部分场景补全：图10将我们的模型与基线在部分场景补全任务中进行了比较（与3相同），定性地展示了相较于 ATISS 的美学优势。图11展示了我们的模型与真实场景的对比效果。请注意，我们的模型不仅能够忠实地重建真实场景，还能泛化到其他家具。

表 4：所有房间类型的部分场景补全指标

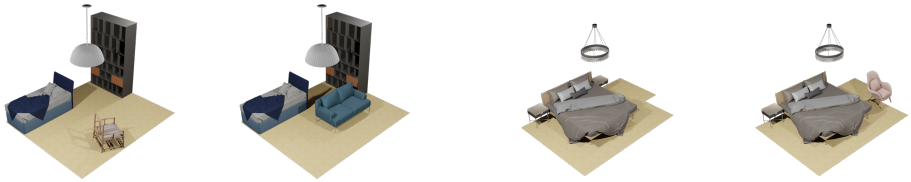
房间类型	预填充数量	模型	FID(↓)	CLIP-FID(↓)	精确率(↑)	召回率(↑)
卧室	1 件家具	ATISS	**16.38**	**0.80**	21.6%	23.0%
		CLIP-Layout	17.83	1.03	**36.5%**	**31.6%**
	2 件家具	ATISS	14.87	0.71	26.3%	29.5%
		CLIP-Layout	**14.08**	**0.67**	**40.7%**	**42.1%**
	3 件家具	ATISS	13.51	0.64	26.9%	31.2%
		CLIP-Layout	**12.11**	**0.53**	**40.4%**	**38.2%**
	4 件家具	ATISS	12.53	0.63	26.7%	31.4%
		CLIP-Layout	**10.80**	**0.46**	**38.4%**	**41.9%**
	5 件家具	ATISS	11.84	0.59	26.2%	31.5%
		CLIP-Layout	**9.88**	**0.42**	**39.1%**	**42.5%**
餐厅	1 件家具	ATISS	21.42	1.21	12.5%	12.9%
		CLIP-Layout	21.66	1.18	**21.8%**	**25.1%**
	2 件家具	ATISS	20.60	1.13	18.7%	19.5%
		CLIP-Layout	20.24	**1.01**	**27.9%**	**31.6%**
	3 件家具	ATISS	19.51	1.02	20.6%	21.7%
		CLIP-Layout	19.19	**0.95**	**29.5%**	**33.5%**
	4 件家具	ATISS	18.61	0.95	21.5%	23.0%
		CLIP-Layout	18.13	**0.90**	**30.1%**	**34.2%**
	5 件家具	ATISS	17.16	0.80	21.2%	23.1%
		CLIP-Layout	17.10	0.78	**29.4%**	**34.5%**
图书馆	1 件家具	ATISS	49.84	3.19	10.7%	11.1%
		CLIP-Layout	**47.56**	**3.08**	**16.7%**	**21.3%**
	2 件家具	ATISS	44.91	2.73	11.2%	11.7%
		CLIP-Layout	**41.60**	**2.60**	**20.4%**	**26.9%**
	3 件家具	ATISS	40.74	2.37	13.9%	15.0%
		CLIP-Layout	**37.78**	**2.27**	**22.6%**	**27.5%**
	4 件家具	ATISS	38.84	2.16	12.7%	14.0%
		CLIP-Layout	**34.93**	**2.03**	**20.7%**	**26.0%**
	5 件家具	ATISS	37.32	2.03	14.4%	14.8%
		CLIP-Layout	**33.18**	**1.93**	**19.6%**	**24.9%**
客厅	1 件家具	ATISS	**17.74**	**0.98**	6.4%	6.9%
		CLIP-Layout	19.98	1.10	**12.3%**	**13.2%**
	2 件家具	ATISS	**17.16**	0.96	8.9%	9.6%
		CLIP-Layout	18.34	**0.91**	**16.2%**	**17.5%**
	3 件家具	ATISS	**16.69**	0.90	10.0%	10.8%
		CLIP-Layout	17.48	**0.85**	**19.7%**	**21.4%**
	4 件家具	ATISS	16.16	0.86	10.6%	12.2%
		CLIP-Layout	16.54	**0.76**	**20.7%**	**22.7%**
	5 件家具	ATISS	15.56	0.83	11.6%	13.1%
		CLIP-Layout	15.92	**0.75**	**21.6%**	**23.9%**

表5：simple-3dviz 的相机设置。

Task	距离	高度	查看	方位角
对象嵌入	3	1	(0,0,0)	八个视图
场景可视化	8	2.5	(0,1,0)	八个视图



(a) 从青色物体变为“实木电视柜”　　(b) 从衣架变为“白色床边脚凳”



(c) 从椅子变为“蓝色现代沙发”　　(d) 从床头柜变为“小型粉色椅子”

图7：**使用文本提示进行家具替换：** CLIP-Layout 可以根据文本提示描述的新实例，将场景中的家具替换为该实例，并进行实例依赖变换。每组图中，左侧为原始场景，右侧为替换家具后的场景，下方为文本提示



(a) 深色木材　(b) 可爱白色　(c) 彩色　(d) 红色条纹　(e) 三角形



(f) 宝蓝色　(g) 工业风　(h) 童趣粉色　(i) 设计师　(j) 凯蒂猫

图8：**房间掩码场景合成：** 平面图生成场景的视觉示例，其中第一行展示 ATISS 生成的样例，下面两行展示 CLIP-Layout 生成的样例。

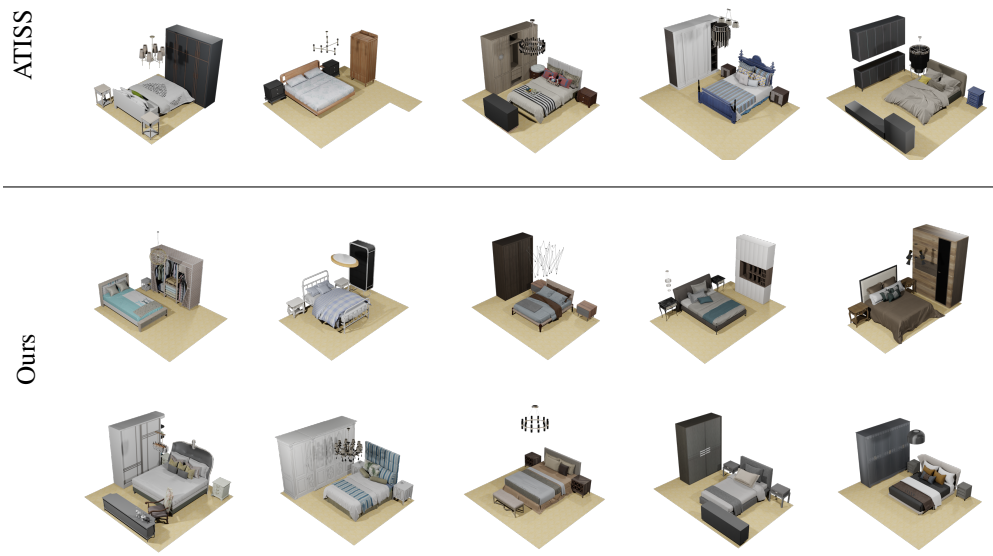


图 9：**基于平面图的场景合成的视觉对比：**顶行展示基线模型生成的样本，分隔线下方展示我们的模型生成的样本。

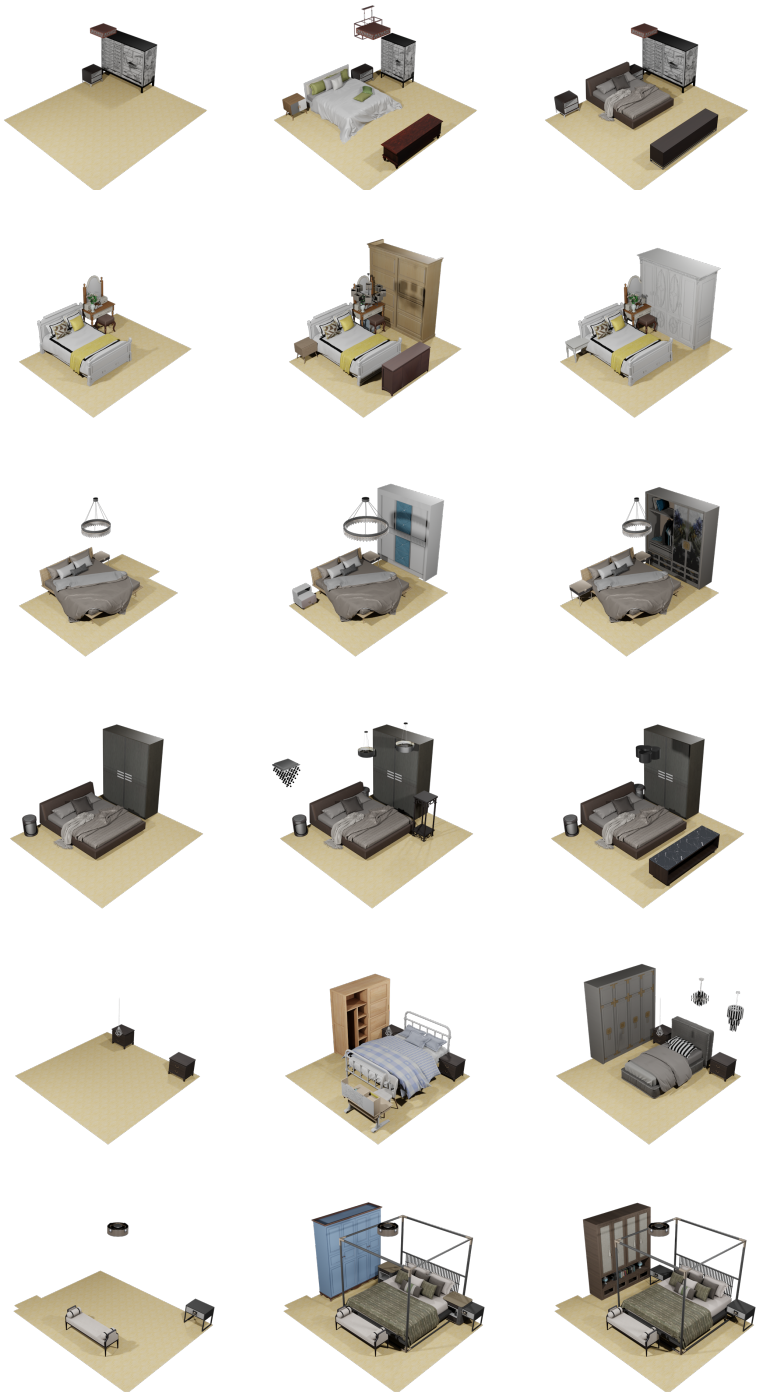


图10：**我们模型与基线的部分场景补全：**从最左列到最右列依次为部分场景、基线补全后的场景以及我们模型补全后的场景。

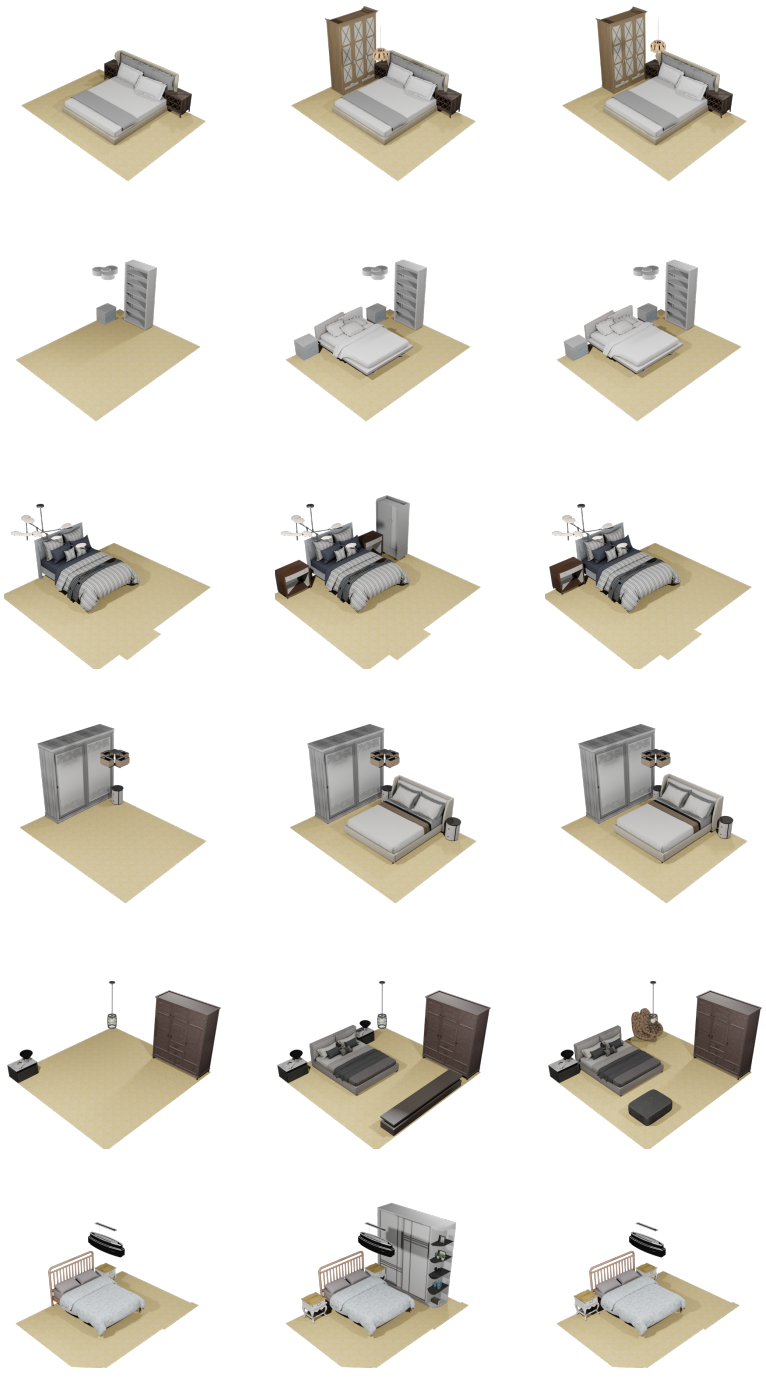


图11：**我们的模型与真实场景的部分场景补全对比：**从最左列到最右列依次为部分场景、由我们的模型补全后的场景和真实场景。