

# METASCENES: Towards Automated Replica Creation for Real-world 3D Scans

Huangyue Yu<sup>1,\*</sup> Baoxiong Jia<sup>1,\*</sup> Yixin Chen<sup>1,\*</sup> Yandan Yang<sup>1,†</sup> Puhao Li<sup>1,3,†</sup> Rongpeng Su<sup>1,4,†</sup>  
 Jiaxin Li<sup>1,2</sup> Qing Li<sup>1</sup> Wei Liang<sup>2</sup> Song-Chun Zhu<sup>1</sup> Tengyu Liu<sup>1</sup> Siyuan Huang<sup>1</sup>

<sup>1</sup>State Key Laboratory of General Artificial Intelligence, BIGAI <sup>2</sup>Beijing Institute of Technology

<sup>3</sup>Tsinghua University <sup>4</sup>University of Science and Technology of China

<https://meta-scenes.github.io/>

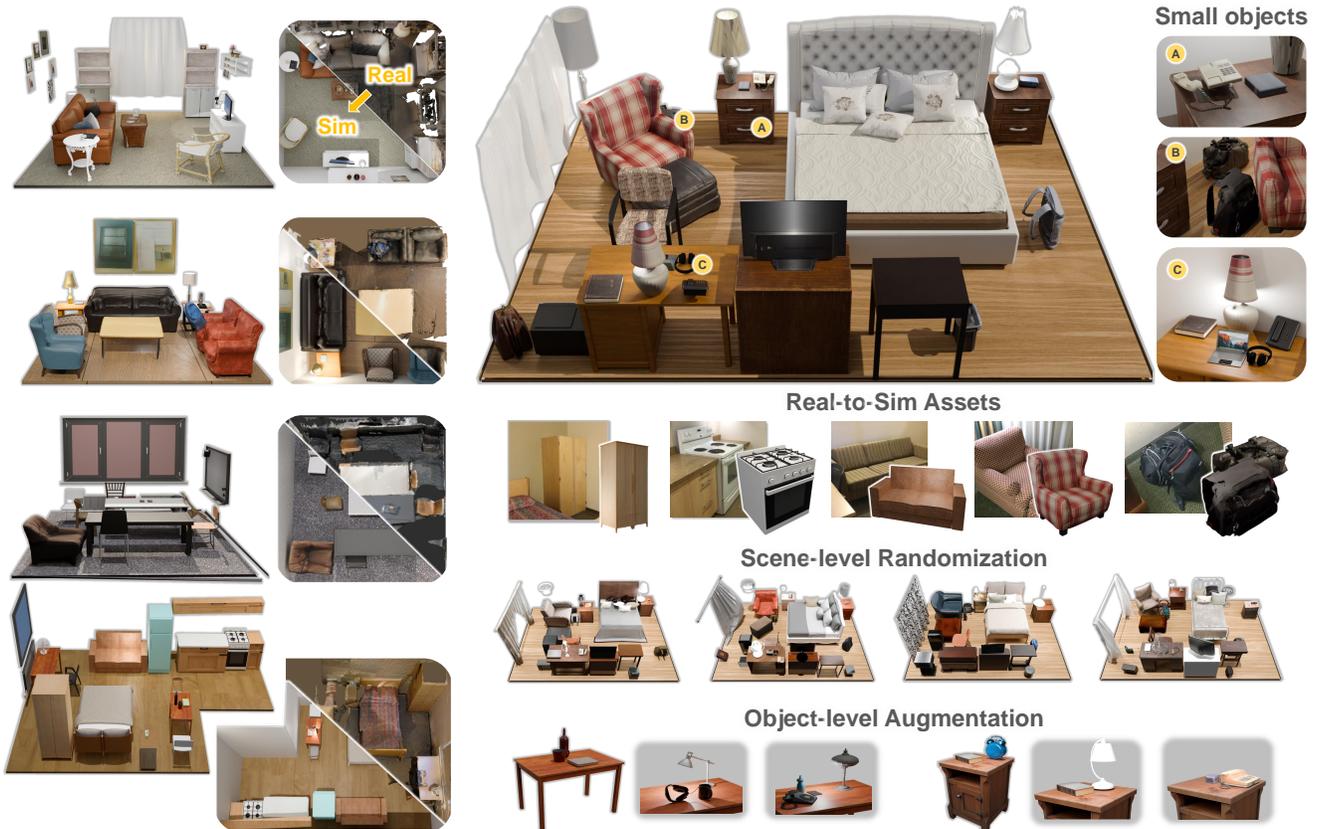


Figure 1. **Overview of METASCENES**, a large-scale simulatable 3D scene dataset constructed by replacing objects in real-world 3D scans with realistic and high-quality object assets retrieved or reconstructed from diverse sources.

## Abstract

*Embodied AI (EAI) research requires high-quality, diverse 3D scenes to effectively support skill acquisition, sim-to-real transfer, and generalization. Achieving these quality standards, however, necessitates the precise replication of real-world object diversity. Existing datasets demon-*

*strate that this process heavily relies on artist-driven designs, which demand substantial human effort and present significant scalability challenges. To scalably produce realistic and interactive 3D scenes, we first present **METASCENES**, a large-scale simulatable 3D scene dataset constructed from real-world scans, which includes 15366 objects spanning 831 fine-grained categories. Then, we introduce **SCAN2SIM**, a robust multi-modal alignment model, which enables the automated, high-quality replacement of assets, thereby eliminating the reliance on artist-driven designs for scaling*

\* indicates equal contribution as first authors.

† indicates equal contribution as secondary authors.

*3D scenes. We further propose two benchmarks to evaluate METASCENES: a detailed scene synthesis task focused on small item layouts for robotic manipulation and a domain transfer task in vision-and-language navigation (VLN) to validate cross-domain transfer. Results confirm METASCENES’s potential to enhance EAI by supporting more generalizable agent learning and sim-to-real applications, introducing new possibilities for EAI research.*

## 1. Introduction

Recent advancements in Embodied AI (EAI) research have been closely tied to the development of high-quality 3D scenes [13, 42, 72], which are essential for enabling agents to learn various skills [18, 25, 29, 79, 87, 91] in simulative environments. As the demand increases for more diverse agent skills, improved skill generalization, and robust sim-to-real (Sim2Real) transfer capabilities, there is a growing need to enhance the scale [13, 19, 102], realism [12, 40, 80], interactability [54, 59, 101], and complexity of 3D scenes to better support a wide range of EAI tasks. However, despite recognizing these crucial features, meeting these quality requirements for 3D scenes largely depends on artist-driven designs, which demand substantial human effort and present significant scalability challenges. This situation underscores a central question in 3D scene research within the context EAI: *How can we scalably produce realistic and interactable 3D scenes that support diverse agent skill learning?*

The major barrier to scaling high-quality artist-designed 3D scenes lies in the diversity of everyday objects and their intricate layout arrangements, particularly small items [43, 96], which are less studied compared to large furniture [20]. Such features are exceptionally difficult to replicate due to the limited availability of diverse object assets and the inherent challenge of learning these complex arrangements with either rule-based [13, 71, 102] or generative models [66, 85, 101], especially given the limited data. As a result, many efforts adopt a real-to-sim pipeline and aim to convert real-world 3D scans [2, 9, 104] that naturally contain such information into virtual replicas by replacing scanned objects with simulatable counterparts (*e.g.*, CAD models) [1, 12, 94]. However, this conversion remains challenging since the limited diversity and quality of available synthetic assets [5] provide no direct equivalent for real-world scanned objects, requiring trade-offs between accuracy in object shape and texture versus attributes like category, location, and orientation. Such “inaccurate” replacements without proper candidate selection rationales recorded provide limited guidance on a general principle for asset replacement in developing automated replica creation pipelines.

Identifying these critical issues in automating the creation of 3D simulatable scene replicas from real-world scans, we propose METASCENES, a large-scale simulatable 3D scene dataset converted from real-world scans. METASCENES

features diverse object types, detailed and realistic layouts (including small items), and visually accurate appearances with physical plausibility ensured. Drawing inspiration from recent advancements in object-level modeling, both from retrieval-based [14, 15, 99] and generative [37, 86, 111] perspectives, we construct a diverse set of potential candidates for each scanned object in the scene, significantly improving the quality, diversity and the degree of variation from the original scanned objects of candidate assets compared to prior works. More importantly, we guide human annotators to rank all potential candidates for each object, providing ground truth for human preference subtle equivalence identified like geometry, texture, or functionality during optimal asset replacement. As demonstrated in our experiments, these annotations not only enable the learning of a powerful multi-modal alignment model, SCAN2SIM, for optimal asset selection, establishing a strong baseline for automated replica creation, but also offer new insights on augmenting these synthetic scenes with object-level randomizations, which renders new potentials for improving the generalizability of agents’ learned skills.

To further explore the potential of METASCENES, we propose two challenging downstream benchmarks to validate the quality of 3D scenes in METASCENES and report key findings within the context of EAI research when equipped with large-scale, realistic simulatable 3D scenes. First, we introduce a novel task, Micro-Scene Synthesis, which extends existing scene-synthesis benchmarks [19] with a special focus on synthesizing small item layouts, crucial for robot manipulation learning [31, 47, 48]. Second, we use domain transfer in vision-language navigation (VLN) [18, 25] as a proxy task to validate the quality of METASCENES scenes by the superior performance of models learned on METASCENES when conducting cross-domain or Sim2Real transfer. We also reveal that navigating to small items is a significant limitation of current VLN models, which could potentially be improved with METASCENES. In summary, our contributions can be summarized as follows:

- We introduce METASCENES, a large-scale simulatable 3D scene dataset constructed by replacing objects in real-world 3D scans with realistic and high-quality object assets from diverse sources to support EAI research.
- With detailed annotations of candidate object selection and transformation during replacement, we enable the learning and evaluation of automated simulatable replica creation pipelines, providing strong baselines as references.
- We meticulously design two challenging tasks, detailed scene synthesis and domain transfer VLN, to validate and leverage the potential of large-scale, realistic simulatable scenes, uncovering new challenges for the field.

## 2. Related Work

**3D Indoor Scene Datasets** The development of 3D scene datasets has been central to computer vision research due to its crucial role in understanding and interacting with the real physical world. Early datasets leveraged RGB-D cameras [4, 9, 27] to build large collections of scanned indoor scenes, enabling tasks in 3D semantic and geometrical reasoning [16, 33, 38, 78, 88, 89]. However, the quality limitations of these capture devices and the static nature of the scenes limit their utility for EAI applications. To address limitations, recent efforts have focused on creating higher-quality 3D indoor scenes, either by directly designing them in simulative environments [22, 40, 44, 68] or by using high-resolution capture devices during scanning [2, 80, 104] and providing extra annotations for object geometry and dynamics [59]. These datasets have significantly advanced EAI research, particularly in embodied reasoning [11, 57, 79], navigation [25, 34, 35, 82], and manipulation [22, 28, 40]. Nonetheless, such high-quality scene curation remains labor-intensive, prompting efforts to generate realistic 3D scenes via rule-based or generative models [13, 66, 71, 85, 101, 102]. Despite their scalability, these synthetic scenes present a significant Sim2Real gap [40] due to limited diversity and realism. As scaling becomes increasingly important in both 3D scene-centric [32, 90] and EAI research [13, 28, 64], a scalable approach to constructing realistic, simulatable, and diverse 3D scenes is urgently needed.

**3D Asset Modeling** Recent years have witnessed significant progress in the development of 3D asset modeling [14, 46, 53, 67, 75, 83, 84, 111]. The curation of large-scale object CAD asset libraries, such as Objaverse [14] and Objaverse-XL [15] effectively addresses the diversity and quality limitations present in earlier datasets like ABO [8] and ShapeNet [5], thus paving the way for new research directions in 3D asset generation including text-to-shape [37] and image-to-shape generation [26, 46, 53, 97, 113]. Among the two directions, image-to-shape generation has received considerably more attention given the fast development of 2D diffusion models [24, 55, 56, 75] and multi-view object representations like NeRF [60, 62] and Gaussian Splatting [39]. These methods leverage the power of pre-trained 2D diffusion models to generate multi-view images of an object which could be used for learning multi-view representations [21, 26, 46, 53, 97] or use them as guidance functions for directly learning 3D multi-view representations [67, 83]. However, adopting such methods for 3D scene reconstruction remains a challenging task due to the complexity of modeling individual objects, especially in the presence of severe occlusions. This challenge has led to the development of various models aimed at reconstructing 3D scenes from scene images [7, 52, 63, 110]. Despite the improving

mesh reconstruction quality, these methods often produce physically implausible mesh predictions for object instances. A recent approach, PhyRecon [61], addresses this issue by introducing physical loss functions in simulators for reconstruction supervision. Nevertheless, the reconstructed scenes still lack essential information such as object texture and accurate geometry, which limits the applicability of these methods in scaling 3D scenes for EAI tasks.

**Real-to-Sim 3D Scene Creation** Creating realistic and diverse simulatable 3D scenes from real-world data is a long-standing task. Prior work [50, 81] addresses scene understanding by annotating images with 3D models using keypoint correspondences, while others [6, 30, 63] use single RGB images to jointly optimize the size, location, orientation and appearance for 3D objects in the scene. Despite aiming for holistic scene understanding, these methods lack the robustness and generalizability to produce image-aligned 3D objects necessary for EAI research, which demands realistic 3D objects in diverse environments. To tackle the challenges of object modeling in 3D scenes, several large-scale datasets [20, 40, 58, 94] are proposed with dense annotations of matched 3D assets. However, they face challenges with limited asset variety, *e.g.*, Scan2CAD [1] that converts ScanNet [9] into 3D CAD models in ShapeNet [5], and struggle with scalability due to the substantial manual work required for adjusting, selecting, or even designing 3D assets [40], especially articulated ones [87]. These challenges highlight the need for automated scene-creation pipelines, while existing methods, such as ACDC [10] that uses foundation models for object matching, struggle in more complex, realistic scenarios and rely heavily on existing asset datasets. We argue the key to solving this challenge is to alleviate the dependence on existing assets in a scalable way, where we propose an automatic pipeline that replaces objects in real-world scans with assets from object-level reconstruction or retrieval.

## 3. METASCENES

In this section, we detail the construction of the **METASCENES** dataset, covering data collection, annotation, and post-optimization, and present an overview of our collection pipeline in Fig. 2. We also outline our design for SCAN2SIM, a powerful baseline pipeline for automated replica creation, leveraging ground-truth annotations available in **METASCENES**.

### 3.1. Data Acquisition

In **METASCENES**, we aim to automatically convert real-world 3D scans into replicas in simulative environments by reconstructing the layout of scenes as well as replacing scanned objects with simulatable 3D assets. Specifically, we choose the ScanNet [9] dataset as the major data source for

Table 1. **Comparison with 3D scene datasets.** We provide a comprehensive comparison between **METASCENES** and existing datasets, noting that “Recon.” indicates whether the dataset utilizes reconstructed 3D assets.

Dataset	Scene			Object				Asset Candidates	Physical Optimization
	Source	#Rooms	Real	CAD Source	#Cat	Recon.	#Objects		
Scan2CAD [1]	ScanNet [9]	706	✓	ShapeNet [5]	35	✗	14225	✗	✗
OpenRooms [49]	ScanNet [9]	706	✓	ShapeNet [5]	44	✗	16014	✗	✗
R3DS [94]	Matterport3D [4]	370	✓	ShapeNet [5], Wayfair [76]	110	✗	19050	✗	✗
CAD-Estate [58]	YouTube	19512	✓	ShapeNet [5]	49	✗	100882	✗	✗
RoboTHOR [12]	Artist design	89	✗	IKEA	44	✗	731	✗	✗
BVS [22]	BEHAVIOR-1K [45]	1000	✗	BEHAVIOR-1K [45]	1937	✗	6685	✗	✓
ReplicaCAD [82]	Replica [80]	90	✓	Artist design	39	✗	2293	✗	✓
HSSD-200 [40]	Floorplanner	211	✗	Floorplanner	466	✗	18656	✗	✗
3D-FRONT [19]	Artist design	18968	✗	3D-FUTURE [19]	49	✗	13151	✗	✗
<b>METASCENES</b>	ScanNet [9]	706	✓	Objaverse [14]	831	✓	15366	✓	✓

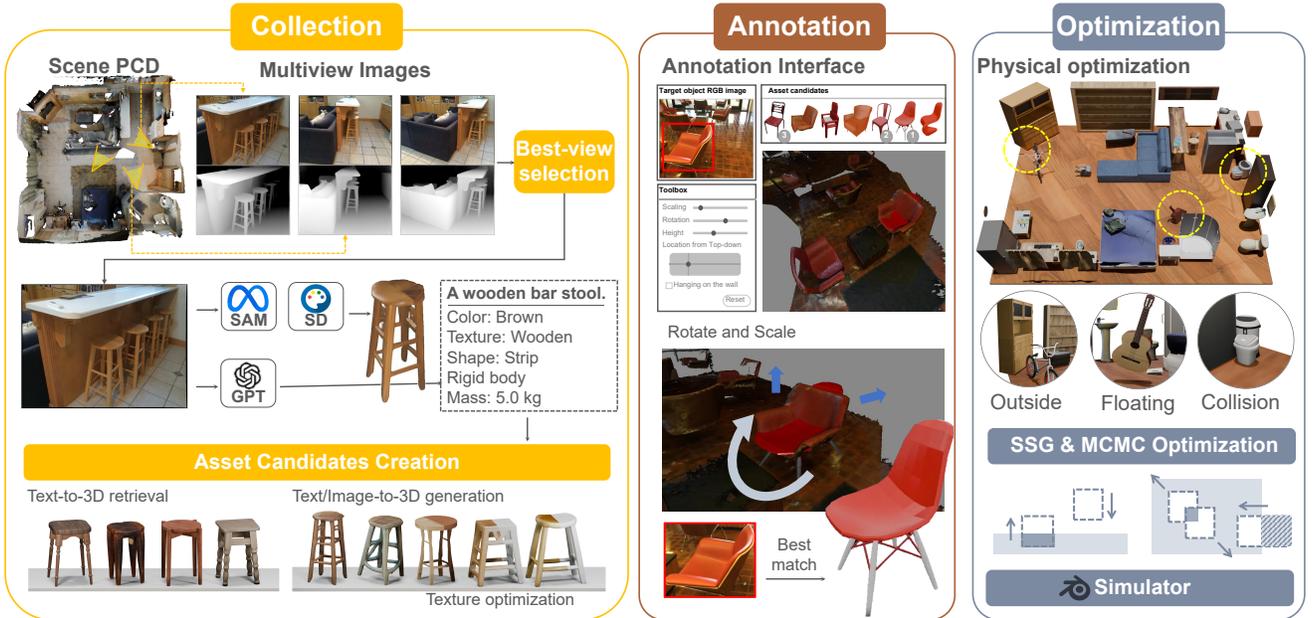


Figure 2. **The construction of METASCENES.** **METASCENES** is composed of three sequential steps: (i) *Collection*, where we gather diverse 3D asset candidates for each real-world object in the scan; (ii) *Annotation*, where annotators rank and select the best-matching 3D asset for each object based on visual similarity and geometric fit; and (iii) *Optimization*, where selected assets undergo post-processing and global optimization to ensure full interactivity and physical plausibility in simulation environments.

real-world scans and construct the **METASCENES** dataset with the following main steps:

**Room Layout Estimation** To obtain simulatable replicas, we first reconstruct the floor plan of each real-world scene using the 3D scene point clouds. Specifically, we employ two types of methods: (i) an end-to-end method following [106], which uses a pre-trained layout transformer to predict the floor plan, walls, and ceilings from the 3D point cloud; and (ii) a heuristic-based method, which uses the maximum area covering all object contours as the room’s floor plan. During post-optimization, the second method serves as a backup solution in case of incomplete room point clouds or inaccurate predictions from the first method.

**Object Asset Curation** For each scanned object in the scene, we aim to find diverse and high-quality simulatable

3D assets that can serve candidates for replacements, closely matching the original objects. To achieve this, we use the capability of vision-language foundation models [41, 103] to generate rich multi-modal descriptions for each scanned object. First, we leverage 3D point clouds and depth maps to select the 2D view with the clearest visibility and minimal occlusion for each object. Then, we use SAM [41] to generate 2D masks of the objects, feeding these masked images into GPT-4V [103] to produce detailed captions describing object texture, color, physical properties, and more. With this descriptive information, we apply recent advancements in object-level modeling to gather asset candidates through three main types of methods: (i) *Text-to-3D generation* methods where we use the detailed text prompts of the object to generate object meshes via models like Shape-E [37]; (ii) *image-to-3D generation* methods where we use the 2D ob-

ject image as the input condition to generate object meshes using methods like TripoSR [86], InstantMesh [95], and Michelangelo [113]; and (iii) *text-to-3D retrieval* methods where we retrieve object assets from online large-scale data sources like Objaverse with methods like Uni3D [114] and ULIP [98]. To further refine the quality and realism of generated meshes, we apply texture optimization methods, such as Paint3D [107], to enhance the color fidelity and surface texture of generated meshes. We provide more details for data collection and a full list of methods used for asset curation in the *supplementary*.

### 3.2. Data Annotation and Processing

**Data Annotation** With 3D asset candidates generated, we guide human annotators to rank these candidates based on their suitability as replacements for the original scanned objects. Ranking criteria focus on geometric similarity and visual appearance (*e.g.*, material and texture), with annotators referencing point clouds and multi-view images of the scanned objects. Leveraging the ranking information, we perform scene- and object-level augmentation by replacing each highest-ranked candidate with one of the top five alternatives, as shown in Fig. 1. Additionally, we instruct annotators to place the best replacement asset into the 3D scene, adjusting orientation and scale as needed for the optimal fit. A visualization of our annotation pipeline is shown in Fig. 2, with further details on the annotation process provided in the *supplementary*.

**Physics-based Optimization** To further ensure the physical plausibility of object placements, we perform a physics-based optimization by first constructing a 3D hierarchical scene-graph from the scene point clouds following [32]. These scene-graphs encode spatial relations (*e.g.*, support, embedding, containment) as constraints. To assess the quality of the scene-graphs, we manually verified spatial relations in 10 randomly sampled scenes and observed 96.3% accuracy. Given the complexity of optimizing layouts with these constraints using gradient-based methods, we employ Markov-Chain Monte-Carlo (MCMC) sampling guided by both the scene-graph and also the physical violations like collisions to adjust object positions. Finally, we import the optimized scenes into Blender, where we add physical properties like material types and masses for each object prompted from foundation models, to enhance the physical realism of the reconstructed scene. Pseudo code for the MCMC process and additional details are provided in the *supplementary*.

### 3.3. Dataset Statistics and Quality analysis

We provide a detailed comparison between METASCENES and existing datasets in Tab. 1. METASCENES includes 15366 object instances derived from 7328 unique 3D assets.

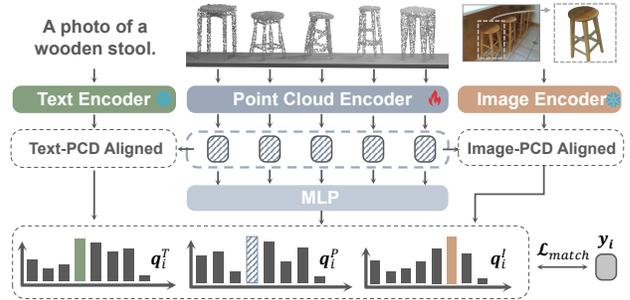


Figure 3. **Overview of our optimal asset retrieval model.** We provide a multi-modal alignment model to retrieve the best asset from candidates.

For each object, we provide a minimum of six asset candidates, resulting in a total of 98423 unique 3D assets in the dataset. These objects covering 831 fine-grained object categories in 706 replicated scenes spanning various room types. It also includes rich semantic information for each object, entailing their physical properties such as mass, material, and bounciness, along with 21 types of spatial relationships and detailed textual descriptions. We believe these comprehensive annotations can significantly enhance the value of METASCENES for EAI tasks.

We further verify the quality of the replicated scenes with quantitative analysis based on Chamfer Distance (CD) metrics, we can show we significantly outperforms previous methods like Scan2Cad in not only diversity but also accuracy. Specifically, the replicated objects in our scenes more closely match the originals, with an average similarity score of 0.25 in METASCENES compared to 0.35 in Scan2CAD.

### 3.4. The SCAN2SIM Pipeline

In this section, we detail the proposed SCAN2SIM pipeline for automated simulatable replica creation for real-world 3D scans. As described in Sec. 1, the major challenges of designing such a pipeline lie in: (i) the selection of the optimal asset for replacing the target scanned object, and (ii) aligning the location, size, and orientation of the selected asset to the scanned object. We describe our solution to these challenges as follows:

**Optimal Asset Retrieval** Based on the ground truth optimal asset selection annotation in METASCENES, we learn a multi-modal alignment model to retrieve the best asset candidate from a set of candidate assets. For each object,  $i$  in the scene, we construct quadruples  $\langle I_i, T_i, \mathbb{P}_i, \mathbf{y}_i \rangle$ , where  $I_i$  is the object image,  $T_i$  is the text description,  $\mathbb{P}_i = \{P_i^1, \dots, P_i^L\}$  is the set of  $L$  potential candidate point clouds, and  $\mathbf{y}_i$  is a one-hot vector indicating the best match. We then design a multi-modal contrastive model to learn optimal asset retrieval. First, we extract image and text features,  $\mathbf{h}_i^I$  and  $\mathbf{h}_i^T$ , with frozen image and text encoders from [99]. Next, we adopt a learnable 3D encoder  $\mathcal{E}_P$  to extract point cloud feature  $\mathbf{h}_{i,k}^P = \mathcal{E}_P(P_i^k)$  for each candidate

$P_i^k \in \mathbb{P}_i$ . We compute the matching score between each candidate and the corresponding image or text with:

$$\mathbf{q}_i^r = [\langle \mathbf{h}_{i,1}^P, \mathbf{h}_i^r \rangle, \dots, \langle \mathbf{h}_{i,L}^P, \mathbf{h}_i^r \rangle], \quad r \in \{I, T\}. \quad (1)$$

Additionally, we compute a matching score  $\mathbf{q}_i^P$  directly from the point cloud by passing  $\{\mathbf{h}_{i,l}^P\}_{l=1}^L$  through a learnable MLP, to prevent the case where no image or text is available. We supervise model learning with the following loss and provide an illustrative visualization of our model in Fig. 3:

$$\mathcal{L}_{\text{match}} = - \sum_i \mathbf{y}_i \cdot \log \sigma(\mathbf{q}_i^I + \mathbf{q}_i^T + \mathbf{q}_i^P). \quad (2)$$

To better align point cloud features with image or text features across different scenes and object instances, we add an additional supervisory signal by creating a new set of candidates  $\mathbb{P}'_i$  consisting of the original best candidate and candidates randomly sampled from different scenes. We follow Eq. (1) to calculate a similar matching score  $\mathbf{q}_i^{I'}$  and  $\mathbf{q}_i^{T'}$  for the auxiliary loss:

$$\mathcal{L}_{\text{aux}} = - \sum_i \mathbf{y}'_i \log \sigma(\mathbf{q}_i^{I'} + \mathbf{q}_i^{T'}). \quad (3)$$

The final learning objective is  $\mathcal{L} = \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{aux}}$ .

**Object Pose Alignment** We adopt a heuristic-based asset placement pipeline for aligning the best-retrieved asset into the scene. First, we translate the center of the best retrieved asset  $\mathbf{c}_{\text{asset}}$  to the center of the real-world scanned object  $\mathbf{c}_{\text{real}}$ . Next, we scale the asset so the longest side of the asset bounding box  $\mathbf{x}_{\text{asset}}$  matches that of the scanned object  $\mathbf{x}_{\text{real}}$ . Finally, we rotate the asset around the up-axis in 30-degree intervals, finding the minimal rotation angle that best aligns  $\mathbf{x}_{\text{asset}}$  and  $\mathbf{x}_{\text{real}}$ .

## 4. Experiments

### 4.1. Automated Replica Creation

**Settings** We first evaluate the automated creation of replicas from real-world 3D scans in the following two settings:

(i) *Optimal Asset Selection*, where the target is to select the best asset from a candidate pool given the target image, text description and scanned point cloud. We compare SCAN2SIM against state-of-the-art multimodal alignment methods, which match the modality from the input to the modality from the candidates. For example, **I+T↔I** indicates matching with the **Image** and **Text** of the input with the candidate assets using rendered **Images**. We report the Top-1 and Top-5 accuracy, along with similarity metrics, *i.e.*, Chamfer Distance (CD), Enhanced Chamfer Distance (ECD), Intersection over Union (IoU) of 3D bounding box and Color Histograms. Evaluation is conducted on the **METASCENES** test set, covering 2497 objects where each one contains 10 asset candidates to choose from.

(ii) *Object Pose Alignment*, where we evaluate the performance of our model SCAN2SIM and ACDC [10] in recovering the correct scale and rotation of the asset given the original image and scan. ACDC uses Dino-V2 [65] to select the best-matched orientation and then apply a render-and-compare method to determine the asset’s scale. For evaluation, we report the pose alignment difference measured in CD, IoU, Size Error ( $m^3$ ), and Scale Error ( $m$ ). We evaluate on 30 scenes from **METASCENES** and 10 scenes in ScanNet++[104]. The ground truth for ScanNet++ scenes is annotated following the same procedure in Sec. 3.2.

For more experiment details, refer to *supplementary*.

**Results & analyses.** We present the quantitative results of *asset selection* in Tab. 2 and *pose alignment* in Tab. 3, with the following key observations:

- The results in Tab. 2 indicate that our SCAN2SIM pipeline, which aligns the text and image inputs with candidate 3D point clouds (I+T↔P), achieves the highest performance across all metrics. This indicates that training with the ranking annotations of our dataset significantly improves the performance of optimal asset selection, as compared with ULIP2, which is trained on large-scale Objaverse [14] with the same modality alignment, fails to fulfill this task whereas our model achieves a Top-1 accuracy of 28.4%.
- The large-scale models, *e.g.*, CLIP and GPT-4V, realize the second-best performance, indicating their strong generalizability on the text and image alignment. In contrast, methods relying on single-modality alignment underperform in both accuracy and similarity. For example, I↔I methods struggle due to the challenges of capturing detailed 3D geometric structures with a single 2D image, while P↔P methods with powerful encoders PointBert and PointNet++, are limited by discrepancies in distribution between real-scanned point clouds and the 3D asset sampling, leading to suboptimal results.
- Tab. 3 reveals that accurately estimating the transformation of assets using 2D images alone is challenging, as real-world objects are often occluded. These occlusions can lead to incorrect orientation estimations from render-and-compare in ACDC. SCAN2SIM mitigates this issue by optimizing poses based on the scanned object point clouds, providing more stable and robust 3D spatial information for object geometry and orientation. Fig. 4 shows that our model offers more reliable asset selection among baselines, enabling automatic digital replica creation in ScanNet++.

### 4.2. Micro-Scene Synthesis

**Overview** Current research [51, 66, 85, 101, 108] in indoor scene synthesis primarily focuses on generating layouts for large furniture, such as table, wardrobe, and sofa. However, due to the lack of training data, none of them talks about the arrangement of smaller objects, which we believe is essential for enhancing the realism of the scene and its practical

Table 2. **Quantitative evaluation on optimal asset selection.** We used different colors to highlight the top three methods for each metric.

Method	Modality		Accuracy		Similarity			
	Input	Cand.	Top-1(%) $\uparrow$	Top-5(%) $\uparrow$	CD $\downarrow$	ECD $\downarrow$	IoU $\uparrow$	Color Hist. $\downarrow$
SSIM [92]	I	I	6.3	44.4	0.24	0.31	0.40	48.10
LPIPS [112]			5.9	45.5	0.24	0.30	0.40	48.01
Uni3D [114]	I	P	11.1	51.8	0.23	0.29	0.45	39.22
ULIP-2 [99]			12.0	59.8	0.22	0.28	0.44	42.36
ICP [3]	P	P	9.2	52.5	0.24	0.30	0.40	41.34
Point-BERT [105]			9.5	51.6	0.22	0.28	0.47	43.48
PointNet++ [69]			11.8	52.5	0.22	0.28	0.49	37.50
Uni3D [114]	T	P	10.2	51.9	0.26	0.32	0.43	37.14
ULIP-2 [99]			14.3	60.3	0.19	0.25	0.52	32.34
CLIP [70]	T	I	14.9	66.6	0.21	0.27	0.51	28.02
GPT-4V [103]			16.5	59.9	0.19	0.26	0.52	32.66
ACDC [10]	I+T	I	12.3	36.6	0.21	0.27	0.47	37.92
ULIP-2 [99]	I+T	P	13.1	57.7	0.20	0.26	0.49	37.49
SCAN2SIM			28.4	76.0	0.17	0.23	0.60	24.65

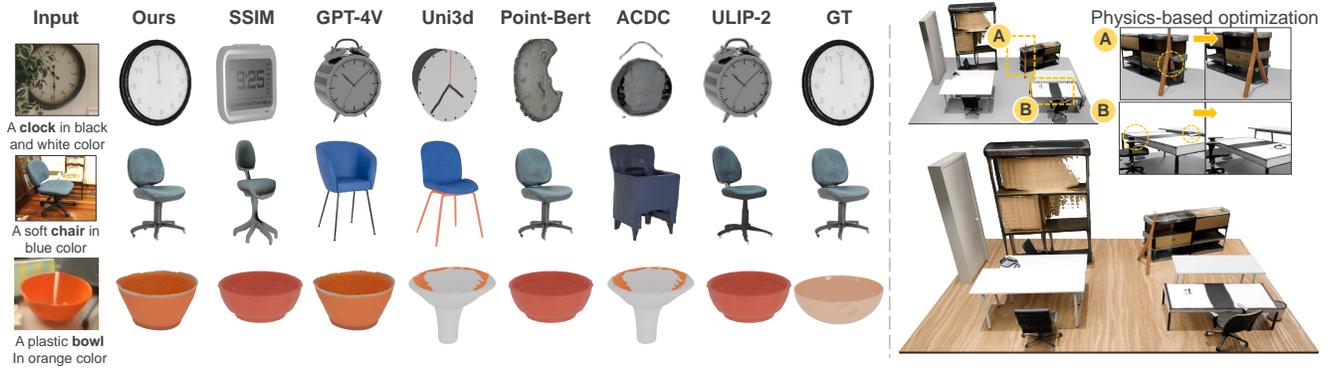


Figure 4. **Automated replica creation.** We visualize the optimal asset selection results in **METASCENES** (left), and a digital replica automatically created via **SCAN2SIM** on **ScanNet++**, before (top) and after physics-based optimization (bottom).

Table 3. **Quantitative evaluation on object pose alignment.** Note that "Size Err." represents the size discrepancy between each aligned object and its real-world counterpart, while "Scale Err." refers to the scene-level size discrepancy.

Dataset	Method	Size Err. $\downarrow$	IoU $\uparrow$	CD $\downarrow$	Scale Err. $\downarrow$
<b>METASCENES</b>	ACDC [10]	0.34	0.29	0.21	0.17
	<b>SCAN2SIM</b>	<b>0.26</b>	<b>0.35</b>	<b>0.20</b>	<b>0.17</b>
ScanNet++ [104]	ACDC [10]	0.55	0.24	0.26	0.13
	<b>SCAN2SIM</b>	<b>0.36</b>	<b>0.40</b>	<b>0.21</b>	<b>0.13</b>

applicability. Leveraging the abundant small objects and realistic arrangements in **METASCENES**, we propose a new task, **Micro-Scene Synthesis**: generating plausible layouts of small objects atop a given piece of large furniture.

**Settings** We follow the setting of scene synthesis and benchmark this new task by adopting three popular methods: **ATISS** [66], **DiffuScene** [85], and **PhyScene** [101]. For metrics, we follow the previous works and report Fréchet In-

ception Distance [23] (FID), Scene Classification Accuracy (SCA), and Category KL divergence (CKL). We also adopt the collision rate of both objects  $Col_{obj}$  and scenes  $Col_{scene}$ , and use  $R_{out}$  to evaluate the rate of small objects outside the plane of large furniture [101].

**Results** From Tab. 4, **ATISS** has the best SCA and  $R_{out}$  score, which means the generated layouts are more accurate and similar to the dataset. On the contrary, **DiffuScene** and **PhyScene** show greater diversity, with better scores on CKL. Meanwhile, **PhyScene** shows effectiveness in reducing object collision by introducing additional physics guidance, producing lower  $Col_{obj}$  and  $Col_{scene}$ . We visualize the generated examples from **PhyScene** in Fig. 5(a), which shows realistic and diverse object-level generation with the given large furniture. Finally, we combine large-object scene synthesis with **Micro-Scene Synthesis** to achieve room-level genera-



Figure 5. **Micro-Scene Synthesis results.** We visualize the generated results in a) **Object-Level** with the generated small objects given the large furniture. b) **Room-Level** by first generating the room layout, and then generating small objects atop the large objects.

Table 4. **Benchmark results on Micro-Scene Synthesis.** These three methods show different advantages on different metrics.

Method	FID↓	SCA↓	CKL↓	Col <sub>obj</sub> ↓	Col <sub>scene</sub> ↓	Rout↓
ATISS [66]	33.25	<b>0.631</b>	0.121	0.645	0.68	<b>0.015</b>
DiffuScene [85]	<b>30.63</b>	0.772	<b>0.037</b>	0.657	0.68	0.078
PhyScene [101]	<b>30.63</b>	0.767	0.039	<b>0.395</b>	<b>0.45</b>	0.074

tion. Fig. 5(b) shows the synthesized whole room from PhyScene by first generating the large-object layout with training on 3D-Front [19] and generating the small-object layout for each large object with training on METASCENES.

### 4.3. Embodied Navigation in 3D scenes

**Overview** Previous work [18, 73, 74, 93, 100] shows that imitating shortest path trajectories in simulation enables embodied agents to develop effective navigation skills. However, current datasets [13] are often procedurally generated scenes rather than real-world environments, limiting their applicability for real-world settings. In contrast, our dataset, METASCENES, offers more realistic environments that better capture the complexities of real-world layouts and object variations, and can be seamlessly incorporated into simulation platforms. To demonstrate the validity of our dataset, we train agents using different data sources and evaluate their generalizability within the AI Habitat [77] environment.

**Settings** We have three settings for imitation navigation training: 1) ProcTHOR [13], a procedurally generated scene dataset 2) METASCENES, and 3) a combination of both. For evaluation, we split METASCENES into *In-domain Scenes*, which is used during training, and *Heldout Scenes*, which remain unseen. We further test on 10 scenes from ScanNet++ as a completely *Held-out Domain*. We choose the state-of-the-art navigation model SPOC [18] as the shared agent baseline. We report Success Rate (SR), Episode Length (EL), Curvature, Success Weighted by Episode Length (SEL), and Success Weighted by Path Length (SPL) to evaluate the agent’s capabilities on exploration and planning efficiency.

**Results** Tab. 5 shows that the model trained solely on METASCENES performs better in the *Heldout Scenes* while

Table 5. **Cross-domain embodied navigation. METASCENES improves generalization in unseen real scenes.**

Benchmark	Data Source	SR(%)↑	EL↓	Curvature↓	SEL↑	SPL↑
In-domain Scenes	ProcTHOR [13]	52.43	25.34	0.38	50.00	43.81
	METASCENES	58.00	23.40	0.17	55.00	51.39
	Both	<b>59.07</b>	<b>22.78</b>	<b>0.21</b>	<b>55.94</b>	<b>52.28</b>
Heldout Scenes	ProcTHOR [13]	51.21	25.73	0.33	48.43	43.82
	METASCENES	<b>52.64</b>	<b>25.57</b>	<b>0.14</b>	<b>49.62</b>	<b>45.55</b>
	Both	51.36	25.58	0.22	48.33	44.78
Heldout Domains	ProcTHOR [13]	45.33	28.56	0.38	42.90	37.58
	METASCENES	<b>50.67</b>	<b>26.56</b>	<b>0.25</b>	<b>47.78</b>	<b>44.33</b>
	Both	46.67	26.95	0.27	43.43	41.51

the model trained on both datasets demonstrates the highest SR in *In-domain Scenes*. This indicates that ProcPHOR is more likely to cause overfitting while METASCENES allows for improved generalization to unseen real scenes. This is further validated by the *Heldout Domains* experiments, where training on METASCENES results in a 5.34% SR increase over the ProcTHOR. The EL, SPL, and SEL further show that our dataset leads to paths more closely aligned with the ideal shortest trajectory, indicating more efficient navigation with superior smoothness from the curvature metric. We further evaluate the sim2real capability of our agents in real-world environments, with more qualitative results in *supplementary*.

## 5. Conclusion

In this work, we presented METASCENES, a large-scale simulatable 3D scene dataset that advances EAI by providing high-quality, interactable, and realistic 3D scenes. Using detailed annotations, we developed SCAN2SIM, a multi-modal alignment model that supports the creation and evaluation of automated real-to-sim replication pipelines. Additionally, we introduced two benchmarks: Micro-Scene Synthesis and cross-domain VLN, which validate METASCENES’s effectiveness and value in addressing key challenges within EAI. METASCENES represents a step forward in scalable and realistic scene generation, laying the groundwork for robust scene understanding and more generalized agent skills.

## References

- [1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#), [3](#), [4](#), [A1](#)
- [2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. [2](#), [3](#)
- [3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. [7](#)
- [4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. [3](#), [4](#)
- [5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [2](#), [3](#), [4](#)
- [6] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *International Conference on Computer Vision (ICCV)*, 2019. [3](#)
- [7] Yixin Chen, Junfeng Ni, Nan Jiang, Yaowei Zhang, Yixin Zhu, and Siyuan Huang. Single-view 3d scene reconstruction with high-fidelity shape and texture. In *International Conference on 3D Vision (3DV)*, pages 1456–1467. IEEE, 2024. [3](#)
- [8] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21126–21136, 2022. [3](#)
- [9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. [2](#), [3](#), [4](#)
- [10] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Acdc: Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024. [3](#), [6](#), [7](#), [A6](#)
- [11] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [3](#)
- [12] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#), [4](#)
- [13] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Proctor: Large-scale embodied ai using procedural generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#), [3](#), [8](#)
- [14] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2023. [2](#), [3](#), [4](#), [6](#)
- [15] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. [2](#), [3](#)
- [16] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet: A deep learning label fusion method for multi-atlas segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2019. [3](#)
- [17] Ainaz Eftekhari, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Ani Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. *arXiv preprint arXiv:2311.04193*, 2023. [A7](#)
- [18] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [8](#), [A7](#)
- [19] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *International Conference on Computer Vision (ICCV)*, pages 10933–10942, 2021. [2](#), [4](#), [8](#)
- [20] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision (IJCV)*, 129:3313–3337, 2021. [2](#), [3](#)
- [21] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. [3](#)
- [22] Yunhao Ge, Yihe Tang, Jiashu Xu, Cem Gokmen, Chengshu Li, Wensi Ai, Benjamin Jose Martinez, Arman Aydin, Mona Anvari, Ayush K Chakravarthy, et al. Behavior vision suite: Customizable dataset generation via simulation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [3](#), [4](#)
- [23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two

- time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 7
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3
- [25] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3
- [26] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 3
- [27] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*, pages 92–101. Ieee, 2016. 3
- [28] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 3
- [29] Jiangyong Huang, Baoxiong Jia, Yan Wang, Ziyu Zhu, Xiongkun Linghu, Qing Li, Song-Chun Zhu, and Siyuan Huang. Unveiling the mist over 3d vision-language understanding: Object-centric evaluation with chain-of-analysis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [30] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2018. 3
- [31] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023. 2
- [32] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision (ECCV)*, 2024. 3, 5, A6
- [33] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3
- [34] Nan Jiang, Zimo He, Zi Wang, Hongjie Li, Yixin Chen, Siyuan Huang, and Yixin Zhu. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. 3
- [35] Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [36] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Yolo by ultralytics. <https://github.com/ultralytics/ultralytics>, 2023. A1
- [37] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 3, 4
- [38] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *International Conference on Computer Vision (ICCV)*, 2017. 3
- [39] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3
- [40] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4, A9
- [41] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 4, A1
- [42] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2
- [43] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *International Conference on Robotics and Automation (ICRA)*, 2011. 2
- [44] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021. 3
- [45] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabriel Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 4
- [46] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 3
- [47] Puhao Li, Tengyu Liu, Yuyang Li, Muzhi Han, Haoran Geng, Shu Wang, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Ag2manip: Learning novel manipulation skills with agent-agnostic visual and action representations. *arXiv preprint arXiv:2404.17521*, 2024. 2
- [48] Puhao Li, Yingying Wu, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Song-Chun Zhu, Tengyu Liu, and

- Siyuan Huang. Controlmanip: Few-shot manipulation fine-tuning via object-centric conditional control. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025. 2
- [49] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, et al. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *arXiv preprint arXiv:2007.12868*, 2020. 4
- [50] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 2992–2999, 2013. 3
- [51] Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations (ICLR)*, 2024. 6
- [52] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision (ECCV)*, 2022. 3
- [53] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *International Conference on Computer Vision (ICCV)*, 2023. 3
- [54] Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Building interactable replicas of complex articulated objects via gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2025. 2
- [55] Ruijie Lu, Yixin Chen, Yu Liu, Jiayang Tang, Junfeng Ni, Diwen Wan, Gang Zeng, and Siyuan Huang. Taco: Taming diffusion for in-the-wild video amodal completion. *arXiv preprint arXiv:2503.12049*, 2025. 3
- [56] Ruijie Lu, Yixin Chen, Junfeng Ni, Baoxiong Jia, Yu Liu, Diwen Wan, Gang Zeng, and Siyuan Huang. Movis: Enhancing multi-object novel view synthesis for indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3
- [57] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [58] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Cad-estate: Large-scale cad model annotation in rgb videos. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 4
- [59] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgb-d scanning for 3d environments with articulated objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3
- [60] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3
- [61] Junfeng Ni, Yixin Chen, Bohan Jing, Nan Jiang, Bin Wang, Bo Dai, Puhao Li, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Phyrecon: Physically plausible neural scene reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3
- [62] Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3
- [63] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3
- [64] Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3
- [65] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6, A6
- [66] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3, 6, 7, 8, A6
- [67] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3
- [68] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dal-laire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. 3
- [69] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 7, A6
- [70] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 7, A6
- [71] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3
- [72] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Un-

- dersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 2
- [73] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8
- [74] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8
- [75] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [76] Shrenik Sadalgi. Wayfair’s 3d model api. <https://www.aboutwayfair.com/tech-innovation/wayfairs-3d-model-api>, 2016. [Online; accessed 15-Nov-2023]. 4
- [77] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *International Conference on Computer Vision (ICCV)*, 2019. 8
- [78] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 3
- [79] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3
- [80] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 3, 4
- [81] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2974–2983, 2018. 3
- [82] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 4, A9
- [83] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3
- [84] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 3
- [85] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6, 7, 8, A6
- [86] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2, 5
- [87] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024. 2, 3
- [88] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [89] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [90] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [91] Yan Wang, Baoxiong Jia, Ziyu Zhu, and Siyuan Huang. Masked point-entity contrast for open-vocabulary 3d scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [92] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 7
- [93] Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alex Schwing. Bridging the imitation gap by adaptive insubordination. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 8
- [94] Qirui Wu, Sonia Raychaudhuri, Daniel Ritchie, Manolis Savva, and Angel X Chang. R3ds: Reality-linked 3d scenes for panoramic scene understanding. *arXiv preprint arXiv:2403.12301*, 2024. 2, 3, 4, A1
- [95] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d

- mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. [5](#)
- [96] Mutian Xu, Pei Chen, Haolin Liu, and Xiaoguang Han. To-scene: A large-scale dataset for understanding 3d tabletop scenes. In *European Conference on Computer Vision (ECCV)*, 2022. [2](#)
- [97] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. [3](#)
- [98] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189, 2023. [5](#)
- [99] Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27091–27101, 2024. [2](#), [5](#), [7](#), [A4](#)
- [100] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *International Conference on Learning Representations (ICLR)*, 2023. [8](#)
- [101] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#), [6](#), [7](#), [8](#), [A6](#)
- [102] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#)
- [103] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of Imms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023. [4](#), [7](#), [A1](#)
- [104] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *International Conference on Computer Vision (ICCV)*, pages 12–22, 2023. [2](#), [3](#), [6](#), [7](#)
- [105] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [7](#), [A6](#)
- [106] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the dots: Floorplan reconstruction using two-level queries. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 845–854, 2023. [4](#)
- [107] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4252–4262, 2024. [5](#), [A1](#)
- [108] Guangyao Zhai, Evin Pinar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. Echoscene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision (ECCV)*, 2025. [6](#)
- [109] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. [A7](#)
- [110] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [3](#)
- [111] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 2024. [2](#), [3](#)
- [112] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [7](#)
- [113] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [3](#), [5](#)
- [114] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024. [5](#), [7](#), [A4](#)

# METASCENES: Towards Automated Replica Creation for Real-world 3D Scans

## Supplementary Material

### A. The METASCENES Dataset

#### A.1. Data Acquisition details

**Small objects capturing** METASCENES includes numerous small objects, a category that existing datasets [1, 94] often fail to capture effectively. We follow a structured approach to identify and capture small objects that may be difficult to locate within a scene. First, we manually curate a list of support objects—such as tables and shelves—that are likely to either support or contain small objects. Next, we utilize SAM [41] to generate 2D masks for these support objects. These masked images are then input into GPT-4V [103] to prompt potential small objects that may be positioned on or within these support objects. Finally, we employ YOLO-v8 [36] to detect and segment these small objects within the scene. The prompt used to guide GPT-4V in capturing small objects is presented in Tab. A1.

**Object captions generation** To generate detailed object captions that describe object attributes, we employ GPT-4V [103] for description prompting. The object captions are categorized into two types: *Object appearance*, which detail visual characteristics such as color, shape, and texture. *Physical attribute*, which cover attributes like physics properties, mass, friction and bounciness. These two types of captions comprehensive coverage of object features, enabling a nuanced understanding of each object’s role within the scene. We show some examples in Tab. A2. The prompt used to guide GPT-4V in generating *physical attribute* captions is presented in Tab. A1.

**Asset candidates curation** To replace each object with simulatable 3D assets, our goal is to identify diverse, high-quality candidates that closely resemble the original objects. For each scanned object, we generate 10 asset candidates using a combination of methods: text-to-3D generation, image-to-3D generation, and text-to-3D retrieval. The models for generating these 10 candidates are detailed in Fig. A1. These candidates ensure a balance of variety and fidelity, offering multiple options for replacement that enhance realism and physical plausibility. We show additional qualitative examples of asset candidates in our METASCENES dataset in Fig. A5.

For texture optimization, we refine the UV unwrapping process to improve the handling of complex object shapes. Instead of using the open-source UV-Atlas tool, as adopted in Paint3D [107]. We employ Blender’s Smart UV unwrapping to preprocess images. This approach generates a UV map with fewer fragments and greater stability, facilitating smoother and more effective texture optimization. This re-

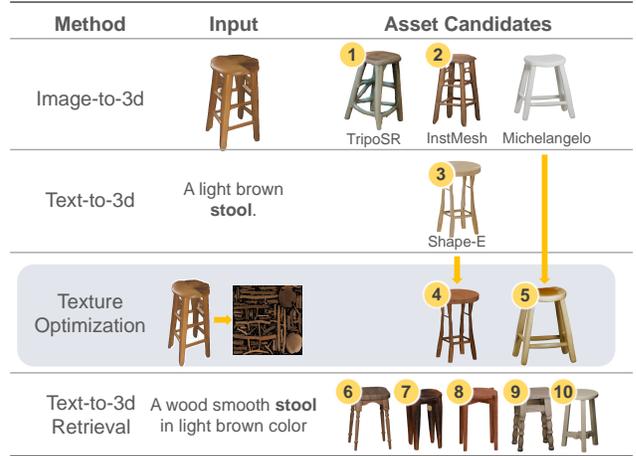


Figure A1. **Models for generate asset candidates.** For each object, we generate 10 asset candidates (labeled as 1–10 in the figure) by leveraging a combination of approaches: text-to-3D generation, image-to-3D generation, and text-to-3D retrieval.

finement is particularly beneficial for assets with intricate geometries, ensuring more consistent and visually appealing texture mapping.

#### A.2. Data Annotation and processing details

**Human annotation** We outline a typical annotation workflow that begins with a real-world scene represented as a point cloud. Annotators freely pan the camera to explore the entire scene, with an overlaid interface that remains synchronized with their view. The annotation process involves the following three sequential steps:

- (i) **Selection:** Annotators select an object from the list of unannotated objects. Once an object is selected, a panel displays a list of candidate 3D assets corresponding to the object. Annotators are instructed to evaluate and identify the best-matching 3D asset based on visual and geometric similarity.
- (ii) **Transformation:** The selected 3D asset is automatically integrated into the scene with a preprocessed scale and orientation. Annotators can then refine the placement by adjusting the asset’s position, height, scale, and rotation to ensure accurate alignment with the point cloud and image.
- (iii) **Ranking:** Annotators rank the remaining 9 candidate assets, identifying the top 2–5 objects that also closely match the real-world object. As shown in Fig. A2.

We recruited annotators to ensure the quality and accuracy

Table A1. Prompts used in METASCENES.

Purpose	Prompt
Small object capturing	<p>You will be provided with an <b>image</b> containing a <b>label</b>. Your task is to carefully analyze the image and list the items present on the surface of the <b>label</b>.</p> <p>Please ensure that you only include items that are on its surface and not those nearby. If you think there is nothing on this <b>label</b>, please return an empty list.</p> <p>Each item should be described in a concise and accurate manner and returned in JSON format.</p> <p>Each item’s JSON object should include the following fields:</p> <ul style="list-style-type: none"> <li>- item: The name of the object</li> <li>- color: The color of the object</li> </ul> <p>Example Output:</p> <p>If there is a black mouse pad and a red cup on the table, your output should be:</p> <pre>[{ 'item': 'mouse pad', 'color': 'black' }, { 'item': 'cup', 'color': 'red' }]</pre> <p><b>Image:</b> A real-world image containing a table.</p> <p><b>Label:</b> Table</p>
Physical attribute	<p>Given the following object <b>label</b> and its <b>size</b>, please output the <b>physics attributes</b> of the object in strict JSON format, including:</p> <p>Physics Properties: Classify the object into one of the following categories:</p> <p>Rigid Body (e.g., Table, Chair, Book, Ball, Cup, Box, Door)</p> <p>Cloth (e.g., T-shirt, Curtain, Tablecloth, Flag, Bed sheet, Towel, Pants)</p> <p>Soft Body (e.g., Jelly, Soft toy, Rubber ball, Cushion, Slime, Foam, Balloon)</p> <p>Mass: Estimate the mass of the object based on its label and bbox size. The mass value should be a float number.</p> <ul style="list-style-type: none"> <li>- For small objects (e.g., ball, book), the mass should be between 0.1 to 5.0.</li> <li>- For medium objects (e.g., table, chair), the mass should be between 5.0 to 50.0.</li> <li>- For large objects (e.g., building, vehicle), the mass should be above 50.0, depending on the object’s real properties.</li> </ul> <p>Friction: Assign a friction value between 0 and 1 based on the object type. The friction value should be a float number:</p> <ul style="list-style-type: none"> <li>- 0.0: No friction (completely smooth, slides freely).</li> <li>- 0.1 - 0.3: Low friction (slight resistance, still easy to slide).</li> <li>- 0.4 - 0.6: Medium friction (noticeable resistance, sliding becomes difficult).</li> <li>- 0.7 - 1.0: High friction (almost no sliding, quickly stops after collision).</li> <li>- &gt; 1.0: Super high friction (very high resistance, may "stick" together, preventing sliding).</li> </ul> <p>Bounciness: Assign an integer value of 0 or 1 to indicate whether the object bounces or not:</p> <ul style="list-style-type: none"> <li>- 0: Does not bounce.</li> <li>- 1: Bounces.</li> </ul> <p>Output Format: Please format your output strictly as JSON, ensuring that mass and friction are float values, and bounciness is an integer:</p> <pre>{ 'physics_attributes': 'category':{Rigid Body   Cloth   Soft Body}, 'mass': [float], 'friction': [float], 'bounciness':[int]}</pre> <p><b>Object Label:</b> Chair</p> <p><b>Object Size:</b> [1.2, 1.0, 0.6]</p>

of the 3D scene reconstruction process. Annotators were instructed to follow these detailed guidelines: (i) *Object Matching*. Annotators were required to select 3D assets that closely align with the observed categories, shapes, and sizes of the objects in the scene. Accurate matching between the original objects and their replica creations is critical for maintaining realism. (ii) *Object Consistency*. For objects with uniform appearance across the scene, the same 3D asset must be consistently selected for replacement. (iii) *Spatial*

*Accuracy*. Each object must be placed and oriented to match its position in the 3D point cloud and accompanying image as closely as possible. Annotators were instructed to avoid misplacements, such as collisions between objects or floating artifacts, to the greatest extent feasible.

To ensure the accuracy and reliability of the annotation results, we implemented a quality control process as follows: For each batch of annotated data, 10% of the samples are randomly selected for accuracy verification. If more than

Table A2. **Examples of object captions in METASCENES.** Note that ‘Friction’ assign a friction value between 0 and 1 based on the object type and ‘Bounciness’ assign an integer value of 1 or 0 to indicate whether the object bounces or not.

Image	Object Appearance	Physical Attributes
	A fabric and plastic soft office chair in red color.	<ul style="list-style-type: none"> <li>• Rigid body</li> <li>• Mass: 20 kg</li> <li>• Friction: 0.5</li> <li>• Bounciness: 0</li> </ul>
	A fabric soft blanket in white color.	<ul style="list-style-type: none"> <li>• Cloth</li> <li>• Mass: 10 kg</li> <li>• Friction: 0.3</li> <li>• Bounciness: 0</li> </ul>
	A fabric smooth pillow in multi-colored.	<ul style="list-style-type: none"> <li>• Soft Body</li> <li>• Mass: 1 kg</li> <li>• Friction: 0.3</li> <li>• Bounciness: 0</li> </ul>
	A fabric soft stuffed animal in brown color.	<ul style="list-style-type: none"> <li>• Soft Body</li> <li>• Mass: 0.5 kg</li> <li>• Friction: 0.3</li> <li>• Bounciness: 1</li> </ul>

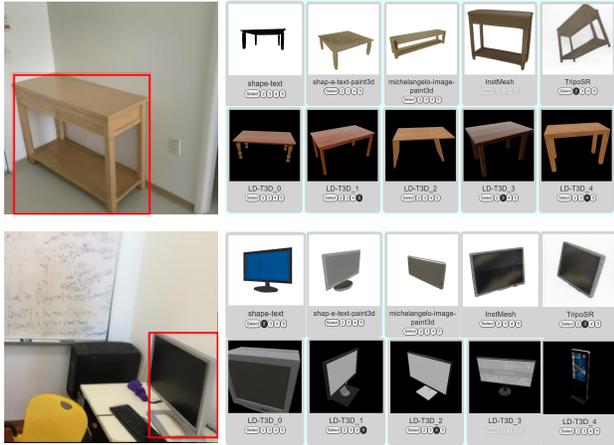


Figure A2. **Annotation interface of object ranking.** Once the best-match asset is selected, annotators are asked to rank the remaining 9 candidate assets.

98% of the inspected samples pass the reviewer’s validation, the batch is deemed acceptable. Otherwise, the annotators are required to re-label the entire batch to address potential errors and meet the quality standards.

**Physics-based Optimization** We use Markov Chain Monte Carlo (MCMC) to traverse the non-differentiable solution space, optimizing the horizontal and vertical placement of objects to prevent issues like collisions or floating objects. See Algorithm 1 for the pseudo code. To quantify collisions for  $m$  objects in scene  $\mathbb{S}$ , we compute the collision loss as follows:

$$L = \sum_{i=1}^m \sum_{j=i+1}^m \text{IoU}(\text{BBox}(o_i), \text{BBox}(o_j)), \quad (\text{A1})$$

where  $\text{BBox}(\cdot)$  represents the 3D bounding box of object, and  $\text{IoU}$  denotes the *Intersection over Union* metric. The loss  $L$  aggregates the pairwise  $\text{IoU}$  values for all unique object pairs. This formulation allows the optimization process to iteratively minimize  $L$ , effectively reducing collisions and ensuring proper spatial arrangements in the scene.

#### Algorithm 1: MCMC Optimization Algorithm

---

**Input** : Scene  $\mathbb{S}$  with  $m$  objects at their initial positions, where  $\mathbb{S} = \{o_1, o_2, \dots, o_m\}$

**Output** : Scene  $\mathbb{S}$  with  $m$  objects at their optimized positions.

- 1:  $n \leftarrow 0$  {Initialize MCMC step counter}
- 2:  $T \leftarrow \{t_1, t_2, t_3, t_4\}$  {Set of possible movements along parameter axes}
- 3:  $L_0 \leftarrow \text{CalculateCollisionLoss}(\mathbb{S})$  {Initial collision loss}
- 4:  $L_{\min} \leftarrow L_0$  {Track the minimum collision loss}
- 5: **while**  $L_n > 0$  **and**  $n < \text{MaxStep}$  **do**
- 6:   **for**  $i = 1$  **to**  $m$  **do**
- 7:     Randomly select a movement  $t \in T$  and apply it to object  $o_i$
- 8:     **if**  $o_i$  remains within scene boundaries **then**
- 9:       Compute the new position for  $o_i$
- 10:        $L_n^i \leftarrow \text{CalculateCollisionLoss}(\mathbb{S})$  {Collision loss after moving  $o_i$ }
- 11:       **if**  $L_n^i < L_{\min}$  **then**
- 12:          Update the position of  $o_i$
- 13:           $L_{\min} \leftarrow L_n^i$  {Update the minimum loss}
- 14:       **else**
- 15:          Revert the position of  $o_i$
- 16:       **end if**
- 17:     **end if**
- 18:   **end for**
- 19:    $n \leftarrow n + 1$  {Increment the MCMC step counter}
- 20: **end while**

---

### A.3. METASCENES statistics

We present histograms showing the distribution of object counts and object categories per scene in Fig. A6a

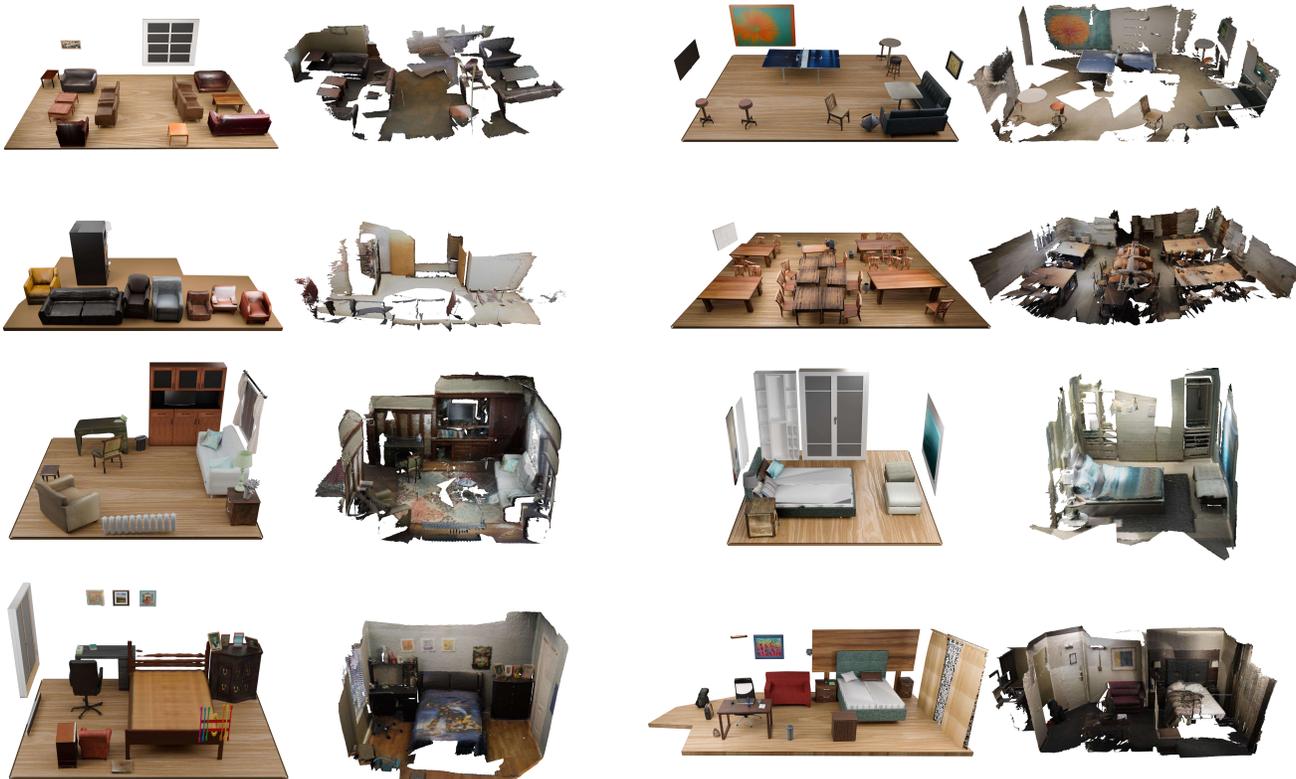


Figure A3. **Scene examples.** We compare the scenes in METASCENES (left) with its original 3D point cloud (right). Note that layouts are set to be invisible.

and Fig. A6b. Additionally, we include a box plot illustrating the distribution of physical sizes (measured in volume,  $m^3$ ) for the top 50 most frequent object categories in Fig. A7. Fig. A4 shows a word cloud visualization of categories in METASCENES, with the text font size representing the total count of unique object instances in each category. We see that our dataset contains a diverse set of object categories. Qualitative examples of scenes from our METASCENES dataset can be found in Fig. A3. For the efficiency of dataset creation, the end-to-end preprocessing of a scene with 39 preprocessed object candidates takes approximately 12 minutes. The time for object candidate creation depends on the reconstruction model used. Each annotator takes about 2 minutes to annotate one object.

## B. Experiment Details

### B.1. Automated Replica Creation

**Model Training** We train our optimal asset retrieval model using a training set of 600 scenes, which includes a total of 13125 objects. For point cloud encoding, we finetune the PointBERT pretrained on [99], and for image and text encoding, we utilized OpenCLIP. During training, we applied standard data augmentation techniques to the 3D point



Figure A4. **Word cloud of object categories in METASCENES.** Font sizes indicate unique instance count per category.

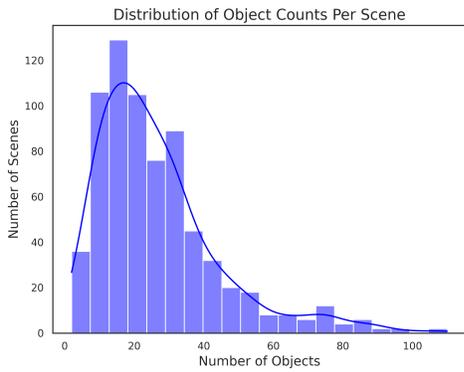
clouds, such as random point dropping, scaling, shifting, and rotational perturbations, to enhance model robustness.

**Baselines** We detail the setup for the comparative models, through two key components: *Optimal Asset Selection* and *Object Pose Alignment*.

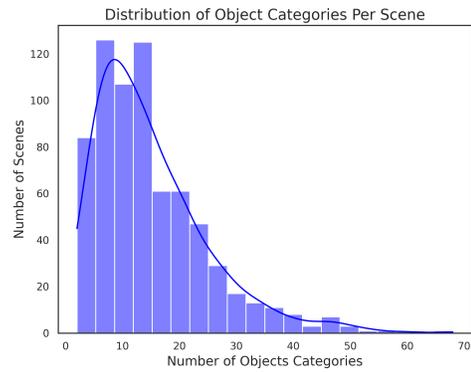
(i) *Optimal Asset Selection.* We evaluate METASCENES against state-of-the-art multimodal alignment methods, as summarized in Tab. 2 in the main paper. For the Uni3D [114] baseline, we use OpenCLIP with the model configuration



Figure A5. Overview of our asset candidates. Note that “\*” indicates texture optimization.



(a) Distribution of Object Counts Per Scene.



(b) Distribution of Object Categories Per Scene.

Figure A6. Object statistics in METASCENES.

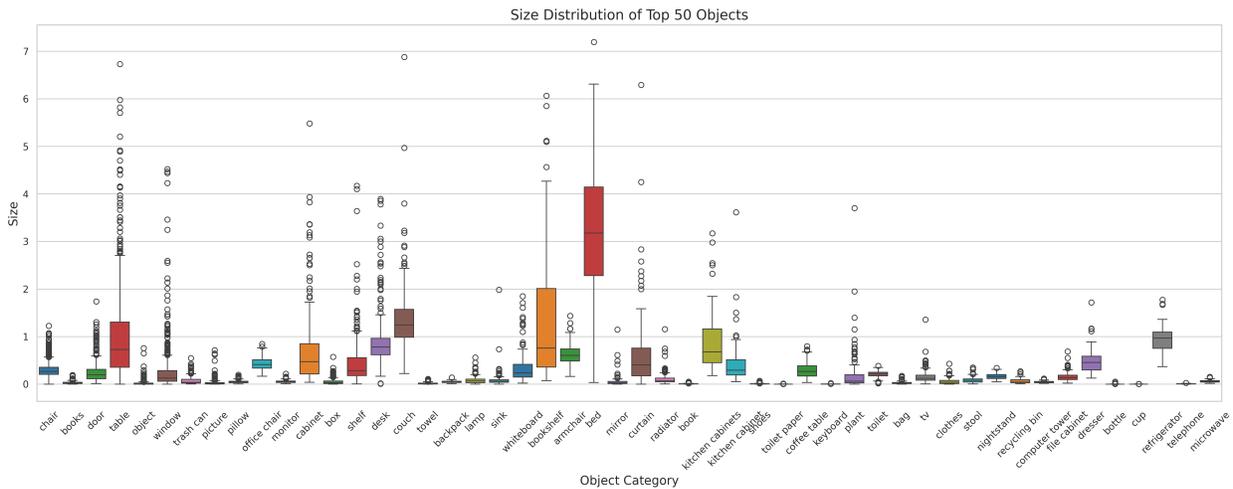


Figure A7. Box plot of the physical size distribution. This shows a wide range of object sizes, with the size distribution clearly highlighting a significant contrast between larger and smaller objects.



Figure A8. **Diverse results of the micro-scene synthesis.** The model is capable of generating varied layouts for the same large furniture.

“EVA02-E-14-plus” as the image and text encoder. This advanced Transformer-based model is pre-trained to reconstruct robust language-aligned visual features through masked image modeling, enabling strong cross-modal alignment capabilities. The Point-BERT [105] is pre-trained on the ModelNet40 dataset, while PointNet++ [69] is pre-trained on the SceneVerse [32] dataset. For the ACDC [10] framework, we employ CLIP and DINO-v2 [65] to identify the best-matching assets.

(ii) *Object Pose Alignment.* In the ACDC framework, we first utilize DINO-v2 to determine the optimal orientation of the asset. Once the best orientation is selected, we apply a render-and-compare method to adjust the asset’s scale. Specifically, after identifying the optimal orientation, we scale the asset across a range of factors from 0.5 to 1.5 and render both the asset and the corresponding real-world object into the 2D image. The asset’s scale is then determined by comparing the 2D bounding box sizes of the rendered asset and the real-world object in the 2D image, with the best-matching scale corresponding to the minimal discrepancy between the two boxes.

**Metrics** We detail the metrics used in our experiment as follows: Chamfer Distance (CD) measures the average distance between point clouds. Enhanced Chamfer Distance (ECD) extends CD by incorporating curvature and geometric features to better capture fine details. Bounding Box Intersection over Union (Bbox IoU) calculates the intersection over union for the 3D bounding boxes of the assets. Color Histograms (Color Hist) compute the Kullback-Leibler divergence between the color distributions of the selected and ground truth assets.

## B.2. Micro-Scene Synthesis

**Data Processing** We preprocess METASCENES by dividing the rooms into micro-scenes. Each micro-scene contains

one large object and several corresponding smaller objects placed on it. We retain the large object categories similar to those in 3D-FRONT, such as “sofa,” “cabinet,” and “table”. For a small portion of objects with unknown categories, we classify them as “object”. Additionally, we merge over 400 open-vocabulary object names into 60 categories: 25 for large objects and 43 for small objects, with 8 categories shared between them, as shown in Tab. A3. After processing, the micro-scene dataset consists of 1,012 micro-scenes and 773 object assets. The quantity distribution of each category in the preprocessed micro-scene dataset is illustrated in Fig. A10.

**Model Setting** In our setup, micro-scenes do not require the shape of the floor plan. Therefore, for all three models, *i.e.*, ATISS [66], DiffuScene [85], and PhyScene [101], we exclude the floor plan input and layout encoder. For DiffuScene and PhyScene, we set the maximum number of objects to 24, with the layout of the large furniture provided as the first object vector. The models generate the remaining 23 vectors, including the empty vectors. For ATISS, the model uses the layout of the large furniture as the first object and then sequentially predicts the layouts of the smaller objects. From the 1,012 processed scenes, we randomly select 803 for training and reserve the remaining 208 for testing.

**Diverse Generation Results** We present results with various large furniture pieces in Fig. 5. In addition, we show diverse results for the same large furniture, specifically selecting a table. As shown in Fig. A8, the model is capable of generating varied layouts for the same large furniture.

**Room Type - Object Category Relationship** We train DiffuScene with a text embedding module, where the prompt includes both the large object’s category and the room type. For example: “A **counter** in the **kitchen**”. The text encoder from CLIP [70] is used to embed the prompt. During infer-

Table A3. **List of 60 categories in micro-scene synthesis.** The category for large furniture is marked in green and the category for small object is marked with underline. There are 8 categories shared between both groups.

<u>alarm_clock</u>	<u>bag</u>	<u>basket</u>	<u>bathub</u>	<u>bed</u>	<u>bin</u>
<u>book</u>	<u>bottle</u>	<u>box</u>	<u>bucket</u>	<u>cabinet</u>	<u>can</u>
<u>chair</u>	<u>clothing</u>	<u>coffee_table</u>	<u>computer</u>	<u>cooking_machine</u>	<u>counter</u>
<u>decoration</u>	<u>desk</u>	<u>dining_table</u>	<u>earphone</u>	<u>electronic_devices</u>	<u>end_table</u>
<u>food</u>	<u>instrument</u>	<u>kettle</u>	<u>keyboard</u>	<u>kitchenware</u>	<u>lamp</u>
<u>ledge</u>	<u>monitor</u>	<u>mouse</u>	<u>mouse_pad</u>	<u>mug</u>	<u>nightstand</u>
<u>object</u>	<u>organizer</u>	<u>phone</u>	<u>picture</u>	<u>pillow</u>	<u>plant</u>
<u>refrigerator</u>	<u>remote_control</u>	<u>round_table</u>	<u>shelf</u>	<u>sink</u>	<u>sofa</u>
<u>stool</u>	<u>table</u>	<u>tissue_paper</u>	<u>toilet</u>	<u>tool</u>	<u>towel</u>
<u>toy</u>	<u>tv</u>	<u>tv_stand</u>	<u>wardrobe</u>	<u>washing_machine</u>	<u>washing_stuff</u>

ence, we generate layouts with a fixed large object, specifically a table, while varying the room type, such as “A table in the office”. We calculate the related small object’s category distribution for each room type. The results in Fig. A11 demonstrate that the model has learned distinct category distributions for different room types. For example, “monitor” has the highest probability of appearing in “office”, “cooking\_machine” is most likely in “kitchen”, and “bag” is most often found in “Bookstore/Library”. These findings also validate the effectiveness of our METASCENES.

### B.3. Embodied Navigation in 3D scenes

**Data and Simulation Setup** We use the Habitat simulator for our data generation and simulation. For data generation, we convert all glb format files into the desired format in Habitat. To generate trajectories for training, we randomly sample a start position for the agent and a navigable target object except for walls. For each trajectory, we sample the ground-truth shortest path using PathFinder within the Habitat simulator. Therefore, each trajectory consists of the agent’s start position and end position, the ground-truth shortest path, and the semantics of the target object. Then these trajectories will be used for training the navigation model. In the Habitat simulator, the agent’s action space contains `move forward (0.25m)`, `turn left (30 degrees)`, and `turn right (30 degrees)`.

**Model and Training Details** We use SPOC [18] as our shared model architecture, with SigLIP [109] image and text encoders. We use a 3-layer transformer encoder and decoder and a context window of 10. We evaluate the object navigation task for the SPOC model trained on the ProcTHOR, METASCENES, and Both, within the AI Habitat environment. The dataset consists of 706 scenes which are randomly split into train/test on a 4:1 ratio. We randomly collect 100 trajectories from each training scene and 50 trajectories from each testing scene for train/test data. We train or fine-tune the model on our METASCENES navigation data with a batch size of 256, a learning rate of 0.0001, and 70k training

steps.

**Quantitative Metrics** Following Eftekhari *et al.* [17], we use quantitative metrics containing SR (Success Rate), EL (Episode Length), SEL (Success weighted by Episode Length), SPL (Success weighted by Path Length), and curvature. SR represents the proportion of correctly navigated trajectories with respect to all trajectories. EL indicates how many actions on average are needed to successfully navigate to the target object. SEL and SPL indicate the difference between the ground-truth path and the predicted path by the agent. A larger SEL or SPL value indicates a closer alignment between the ground truth path and the actual path. Curvature measures the smoothness of a trajectory, with larger curvature values indicating a less smooth path. Some qualitative examples of navigation are shown in Fig. A12. Regardless of whether the target object is seen at the beginning, the agent can navigate to the destination correctly.



Figure A9. **The configuration of UP AGV and its environment.** This includes the real-world scene, the scanned scene, and the digital replica.

Table A4. Comparison on VLN experiments with HSSD

Benchmark	Data Source	SR(%) $\uparrow$	EL $\downarrow$	Curvature $\downarrow$	SEL $\uparrow$	SPL $\uparrow$
10 scenes from	HSSD	27.00	33.77	0.39	26.77	23.32
Replica CAD	METASCENES	32.00	33.71	0.46	31.56	26.91

**Real-world Deployment** We deploy the policy trained on METASCENES to a real-world Automated Guided Vehicle (AGV), called UP. For odometry estimation, the vehicle

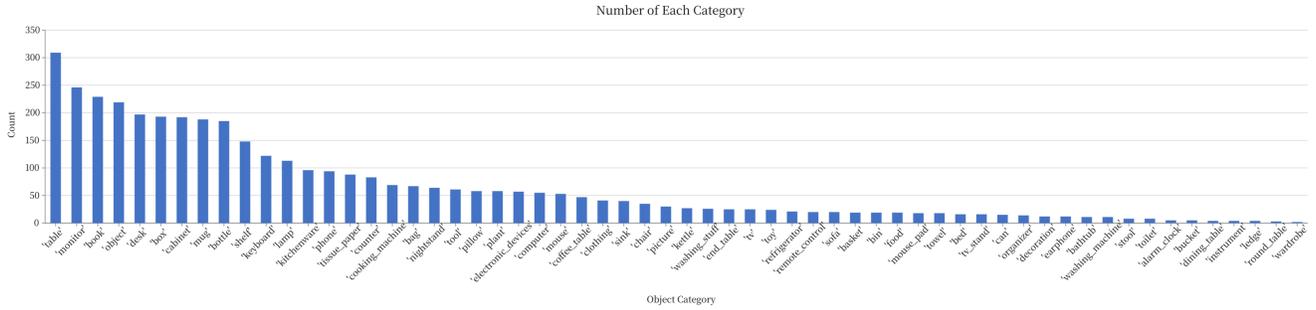


Figure A10. Number of each category in preprocessed micro-scene dataset.

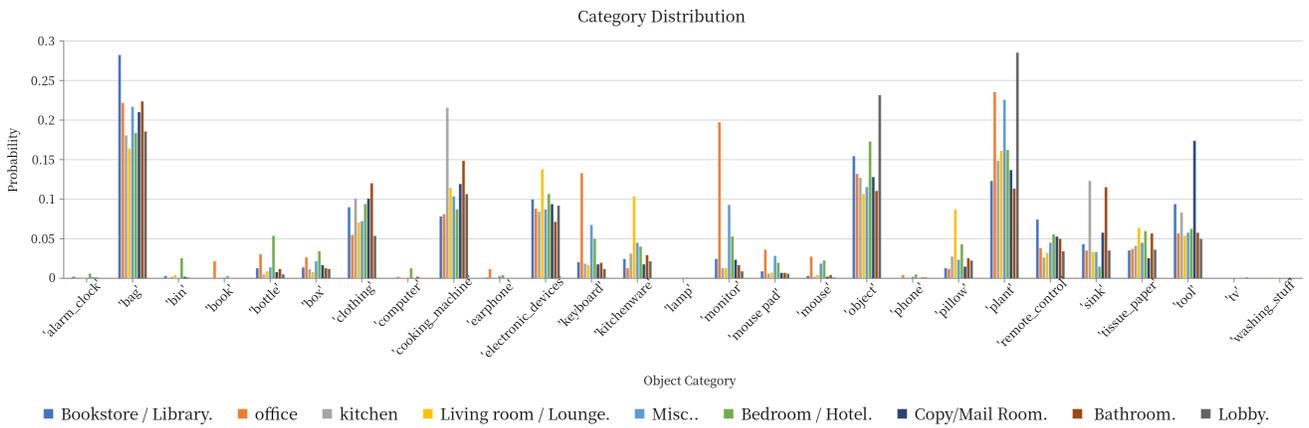


Figure A11. Generated class distribution of different room types. We generate the layout with the same large furniture using the prompt with different room types. Results show the model has learned different class distribution of different room types.

Navigate to the coffee maker



Navigate to the stove



Figure A12. Embodied Navigation. Demonstration of the embodied agent performing goal-directed navigation in Habitat.

Navigate to the coffee maker



Figure A13. Real-world transfer. Demonstration of the embodied agent performing goal-directed navigation in the real world.

combines data from a 2D Lidar, IMU, and wheel speedometer. After receiving the predicted actions from the navigation policy based on the digital replica of the scene, we down-sample these actions at approximately 0.5-meter intervals to create a sequence of local goals. UP plans a trajectory for each local goal and computes the corresponding linear and angular velocities using Dynamic Window Approach (DWA) algorithm, ensuring collision-free execution. The AGV configuration, the real-world scan, and its digital replica are shown in Fig. A9. We present navigation scenarios in Fig. A13, demonstrating that UP successfully reaches the target by transferring the policy in simulation to the real world.

**Comparisons with Other Datasets** We evaluate navigation models pre-trained on METASCENES and HSSD [40] using the replica-CAD [82] dataset in Tab. A4. We randomly selected 10 scenes in the replica-CAD dataset, and randomly sampled the starting point and target object in each scene, collecting 10 trajectories for testing. Finally, the two models are tested on these 100 trajectories and the metrics are calculated. The results confirm that pre-training with our dataset consistently yields superior performance, further verifying our scene quality claim.