

# LiveScene: Language Embedding Interactive Radiance Fields for Physical Scene Rendering and Control

**Delin Qu<sup>1,2\*</sup>**   **Qizhi Chen<sup>2,3\*</sup>**   **Pingrui Zhang<sup>2</sup>**   **Xianqiang Gao<sup>2</sup>**   **Bin Zhao<sup>2,4</sup>**

**Dong Wang<sup>2†</sup>**   **Xuelong Li<sup>2</sup>**

<sup>1</sup>Fudan University      <sup>2</sup>Shanghai AI Laboratory

<sup>3</sup>Southern University of Science and Technology

<sup>4</sup>Northwestern Polytechnical University

## Abstract

This paper aims to advance the progress of physical world interactive scene reconstruction by extending the interactive object reconstruction from single object level to complex scene level. To this end, we first construct one simulated and one real scene-level physical interaction dataset containing 28 scenes with multiple interactive objects per scene. Furthermore, to accurately model the interactive motions of multiple objects in complex scenes, we propose **LiveScene**, the first scene-level language-embedded interactive neural radiance field that efficiently reconstructs and controls multiple interactive objects in complex scenes. LiveScene introduces an efficient factorization that decomposes the interactive scene into multiple local deformable fields to separately reconstruct individual interactive objects, achieving the first accurate and independent control on multiple interactive objects in a complex scene. Moreover, we introduce an interaction-aware language embedding method that generates varying language embeddings to localize individual interactive objects under different interactive states, enabling arbitrary control of interactive objects using natural language. Finally, we evaluate LiveScene on the constructed datasets **OmniSim** and **InterReal** with various simulated and real-world complex scenes. Extensive experiment results demonstrate that the proposed approach achieves SOTA novel view synthesis and language grounding performance, surpassing existing methods by +9.89, +1.30, and +1.99 in PSNR on CoNeRF Synthetic, OmniSim #chanllenging, and InterReal #chanllenging datasets, and +65.12 of mIOU on OmniSim, respectively. Project page: <https://livescenes.github.io>.

## 1 Introduction

Interactive objects are prevalent in our daily lives, and modeling interactable scenes from real physical world plays an essential role in various research fields, including content generation [38, 41, 40, 61, 9], animation [69, 44, 35, 42], virtual reality [59, 13, 26, 66], robotics [1, 20, 4, 21, 46], and world understanding [63, 19, 62, 18, 8]. This paper tackles the challenging and rarely explored task of reconstructing and controlling multiple interactive objects in complex scenes from a single, casually captured monocular video without previous independent modeling of geometry and kinematics.

Prior research on interactable scene modeling, such as CoNeRF [25] and K-Planes [54], typically adopts a joint modeling approach, combining spatial coordinates and all interaction variables as input and representing interactive scene by either implicit MLPs or feature planes. Meanwhile, CoGS [73] learns parameter offsets for different scene parts using multiple independent MLPs after establishing a 3D deformable Gaussian scene. However, these methods primarily focus on capturing interactions for a single object within a clear background, such as a single drawer, toy car, or face [76, 25, 73, 78]. As modeling extends from single objects to multiple objects in complex scenes, as shown in Figure. 1, the

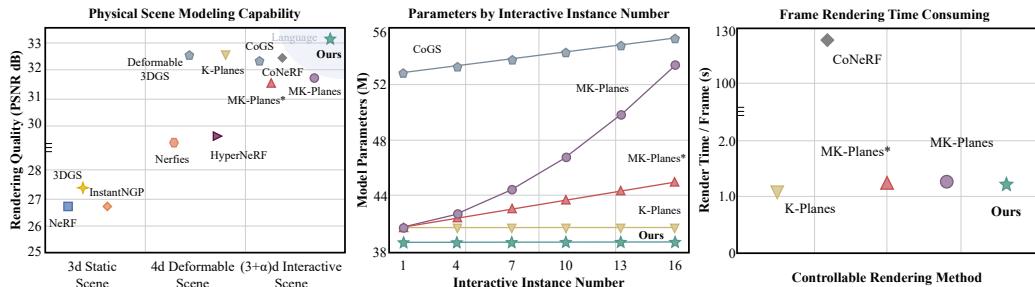


Figure 1: Scene modeling capability, render performance, and efficiency comparisons among various methods. In contrast to static or deformable scenes, LiveScene is capable of modeling  $(3 + \alpha)d$  interactive scenes with language grounding and achieving SOTA rendering quality on OmniSim dataset. Notably, LiveScene exhibits a significant advantage in parameter efficiency, maintaining a constant complexity regardless of interactive instance number and satisfying rendering time at  $1200 \times 1200$  resolution. In contrast, other methods, such as CoGS [73] and MK-Planes [54], exhibit a linear or quadratic increase in parameter quantity.

interaction spaces become increasingly high-dimensional and intricate, complicating these methods for accurate modeling and significantly increasing computational time and memory cost, e.g., 4× A100 GPU for 2 weeks to converge training in CoNeRF [25] and 500M Gaussian storage for a regular indoor living room in CoGS [73]. Moreover, natural language is an intuitive and necessary interface for interacting with 3D scenes, but language embedding of interactive scenes faces an even more daunting challenge: interaction variation inconsistency. For instance, methods like LERF [28], OpenNeRF [77], and GARField [29], which distill CLIP features into static 3D fields, suffer from significant failures when confronted with scene topology structure changes induced by interactions, such as the distinct structures variation of a cabinet before and after opening.

To address these challenges, we propose the first scene-level language embedded interactive scenes reconstruction and control radiance fields LiveScene, which compresses the high-dimensional interaction space into compact 4D feature planes, reducing model parameters while improving optimization effectiveness. Specifically, LiveScene models the multiple object interaction space via a novel high-dimensional factorization by decomposing the interactive scene into multiple local deformable fields, which model individual interactive objects by multi-scale 4D deformable feature planes. Moreover, to achieve independent and accurate control of interactive objects, we propose a multi-scale interaction probability sampling strategy to sample the interaction-relevant 3D points in factorized local deformable fields. As for the nature language interacting interface, we introduce an interaction-aware language embedding method that generates varying language embedding to localize and control interactive objects under arbitrary interaction states, which is the first attempt to control interactive objects using nature language. Finally, we construct the first scene-level physical interaction dataset **OmniSim** and **InterReal**, containing 28 scenes with overall 70 interactive objects for evaluation. Our approach achieves state-of-the-art novel view synthesis quality, outperforming existing best methods by +9.89, +1.30, and +1.99 in PSNR on the CoNeRF Synthetic, OmniSim #chanllenging, and InterReal #chanllenging subsets, respectively. Surpassing LeRF [28], LiveScene significantly improves language embedding grounding accuracy by +65.12 of mIOU on the OmniSim dataset. Notably, our method maintains a lightweight, constant model parameter of 39M, scaling well with increasing scene complexity, as shown in Figure. 1. The contributions can be summarized as follows:

- We introduce **LiveScene**, the first scene-level language embedded interactive radiance field that efficiently reconstructs and controls complex physical scenes, allowing for the control of multiple interactive objects within the neural scene with diverse interaction variations and even facilitating language-based interaction.
- We propose a factorization and sampling technique that decomposes interactive scenes into local deformable fields and samples the interaction-relevant 3D points to control individual interactive objects in a complex scene. Furthermore, we introduce an interaction-relevant language embedding method that generates interaction-relevant varying language embeddings to localize and control interactive objects.
- We construct the first scene-level physical interaction dataset **OmniSim** and **InterReal**, containing **28 subsets and 70 interactive objects** for evaluation. Extensive experiments demonstrate that the proposed approach achieves SOTA performance in various tasks such as novel view synthesis, video frame interpolation, and scene interactive control, showcasing robust interaction capabilities.

## 2 Related Work

**Dynamic Scene Representation.** Extending NeRF [45] to dynamic scene reconstruction has made significant progress. Related methods can generally be categorized into *time-varying methods*, *deformable-canonical methods*, and *hybrid representation methods*. The time-varying methods [10, 11, 37, 49, 51, 64, 75] typically model the radiance field directly over time, but struggle to separate dynamic and static objects. Deformable-canonical methods [14, 36, 50, 71] decouple dynamic deformable field and static canonical space, modeling 4D by warping points with deformable field to query the canonical features. However, these methods face challenges in scene topology changes [54]. Hybrid representation methods, on the other hand, have achieved high-quality reconstruction and fast rendering by utilizing time-space feature planes [56, 54, 5], 4D hash encoding [67], dynamic voxels [68], or triple fields [58]. Recently, several works [43, 72, 70, 32] have introduced 3D gaussians [26] into dynamic scene reconstruction, achieving high-quality real-time rendering speeds. However, these methods are limited to reconstructing dynamic scenes and lack the ability to control and understand interactive scenes.

**3D Vision-language Fields.** Vision-language foundational models [53, 6, 48, 24, 30] with strong generalizability and adaptability inspires numerous language embedded scene representation for 3D scene understanding [3, 80], such as open-vocabulary segmentation [79, 39, 31, 16, 65, 29], 3D visual question answering [17, 8, 23, 22], and 3D language grounding [20, 55, 7, 22]. LeRF [28] is the first to achieve open-vocabulary 3D queries by combining CLIP [53] and DINO [6] with NeRF through feature distillation. Open-NeRF [77] introduces an integrate-and-distill paradigm and leverages hierarchical embeddings to address 2D knowledge-distilling issues from SAM. LEGaussians [57] and LangSplat [52] integrate semantic features into 3D gaussians [26] and achieve precision language query and efficient rendering. However, these methods are limited to static scene understanding and fail to generalize when the interactive scene topology changes.

**Controllable Scene Representation.** Manipulating reconstructed assets or neural fields is of significant importance for avatar and robotic tasks [12, 2, 15, 25, 33]. CoNeRF [25] pioneered this effort by extending HyperNeRF [50] and introduce a fine-grained controllable neural field with 2D attribute mask and value annotations. CoNFies [74] proposes an automatic controllable avatar system and accelerates rendering by distilling. More recently, CoGS [73] leveraged 3D Gaussians [26] to achieve real-time control of dynamic scenes without requiring explicit control signals. However, these methods typically lack natural language interaction capabilities, relying solely on manual control. Furthermore, most works focus on single or few object interactions, disregarding the interaction between different scene parts, limiting their real-world applications.

## 3 Methodology

We aim to establish a representation that models  $\alpha$  interactive articulated objects in a complex scene  $\chi$  from a casually captured monocular video  $\{\mathbf{I}_i\}_{i=1}^T$  via rendering-based self-supervised manner. Control variables  $\kappa = [\kappa_1, \kappa_2, \dots, \kappa_\alpha]$  indicating object motion states, and camera poses  $\{\mathbf{P}\}_{i=1}^T$  of each video frame are given. The overview of LiveScene is shown in Figure. 2. Section. 3.1 introduces the high-dimensional interactive space modeling and challenges. Section. 3.2 presents a multi-scale interactive space factorization and sampling strategy to compress the high-dimensional interactive space into local 4D deformable fields to model complicated interactive motions of individual objects. Section. 3.3 introduces an interaction-aware language embedding method to localize and control interactive objects using natural language.

### 3.1 Interactive Space

Assuming a non-rigidly interactive scene  $\chi$  with  $\alpha$  control variables  $\kappa = [\kappa_1, \kappa_2, \dots, \kappa_\alpha]$  corresponding to  $\alpha$  objects, we delineate its representation by a high-dimensional function:

$$\mathbf{y} = \rho(\mathbf{x}, \kappa, \mathcal{H}; \boldsymbol{\theta}), \quad (1)$$

where  $\rho$  is the model of the representation,  $\mathbf{x} \in \mathbb{R}^3$  are spatial coordinates,  $\kappa \in \mathbb{R}^\alpha$  are control variables,  $\mathcal{H}$  is a set of optional additional parameters (e.g., the view direction), and  $\boldsymbol{\theta}$  stores the scene information. The function outputs scene properties  $\mathbf{y}$  for the given position  $\mathbf{x}$  and  $\kappa$  sampling from a ray  $\mathbf{r}$ , where  $\mathbf{y}$  can be represented with color, occupancy, signed distance, density, and BRDF

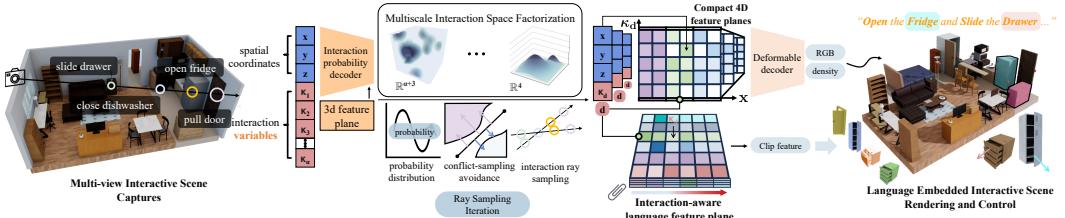


Figure 2: The overview of LiveScene. Given a camera view and control variable  $\kappa$  of one specific interactive object, a series 3D points are sampled in a local deformable field that models the interactive motions of this specific interactive object, and then the interactive object with novel interactive motion state is generated via volume-rendering. Moreover, an interaction-aware language embedding is utilized to localize and control individual interactive objects using natural language.

parameters. In this paper, we focus on color, probability, and language embedding. Distinguishing from 3d static scene or 4d dynamic scene modeling, the sampling point  $p = [x|\kappa] \in \mathbb{R}^{3+\alpha}$  in interactive scene is high-dimensional and variable in topological structure, complicating the scene feature storage  $\theta$  and the optimization of representation model  $\rho$ , leading to significant time-consuming or memory-intensive training in [25, 73].

### 3.2 Multi-scale Interaction Space Factorization

For a  $(3 + \alpha)$ -dimensional interactive space  $\chi = \mathbb{R}^{(3+\alpha)}$  containing  $\alpha$  control variables, we aim to explicitly represent the high-dimensional space in a concise and compact storage, thereby reduce memory usage and improve optimization. As illustrated in Figure. 2, interactive objects exhibit mutual independence, and interaction features  $\mathcal{F}_\alpha \in \theta$  unevenly distribute in the  $(3 + \alpha)$ -dimensional interactive space  $\chi$  and aggregate into cluster centers  $c_i, i \leq \alpha$ . Thus, there exists a set of hyperplanes that partition the space into disjoint regions  $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_\alpha\}$ , with each subregion  $\mathcal{R}_i$  containing a local 4D deformable field  $\theta_d$ , as shown in Figure. 3. Hence, the interaction features  $\theta(p)$  at sampling point  $p \in \mathbb{R}^{3+\alpha}$  can be projected into a compact 4-dimensional space  $\mathbb{R}^4$  in Figure. 3 by a transformation  $T$ :

$$\Theta(\theta(p), T) \rightarrow \theta_d(p_d), \quad p \in \mathbb{R}^{(3+\alpha)}, \quad p_d \in \mathbb{R}^4, \quad (2)$$

where  $\Theta$  represents the operation of projecting high-dimensional features into a 4-dimensional space, and  $T$  denotes the corresponding projection matrix.

**Multi-scale Interactive Ray Sampling.** Solving  $\Theta; T$  is complex and unwise, we consider using ray sampling to perform the projection. As shown in Figure. 2, assuming a ray  $r(t) = [\mathbf{o}_x|\mathbf{o}_\kappa] + t[\mathbf{d}_x|\mathbf{d}_\kappa]$  with origin  $\mathbf{o}_x \in \mathbb{R}^3$ ,  $\mathbf{o}_\kappa \in \mathbb{R}^\alpha$ , and direction  $\mathbf{d}_x \in \mathbb{R}^3$ ,  $\mathbf{d}_\kappa \in \mathbb{R}^\alpha$ , the ray intersects with the interaction instances  $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_\alpha\}$  at a point  $p = [x|\kappa] \in \mathbb{R}^{3+\alpha}$ , where  $x \in \mathbb{R}^3$  is the 3D position and  $\kappa \in \mathbb{R}^\alpha$  is the interaction variables. For a given intersection point  $p$ , the deformable features  $\mathcal{F}_d$  can be retrieved from the corresponding local 4D deformable field  $\theta_d$  by:

$$\mathcal{F}_d = \theta_d(p_d), \quad p_d = [x|\kappa_d], \quad d = \arg \max_i \Theta(\kappa, \theta_s(x))_i, \quad (3)$$

where  $\theta_s(x)$  is the probability features at position  $x$  from 3d feature planes, and  $\arg \max_i \Theta(\kappa, \theta_s(x))_i$  maps the interaction variables  $\kappa$  to the most probable cluster region  $\mathcal{R}_d$  in the 4D space, and can be optimized by minimizing the focal loss of mask  $\tilde{y}_m$  across all the training camera views and images:  $\arg \min_{\Theta, \theta_s} \mathcal{L}_{\text{focal}}(\tilde{y}_m, \Theta(\kappa, \theta_s(x)))$  in supplemental appendix D. Next, the

deformable features  $\mathcal{F}_d$  are used to compute the local deformable scene properties  $y$  at the sampling point  $p$ . In this way, the high-dimensional interaction features  $\mathcal{F}_\alpha$  are factorized into a 4D space  $\mathcal{F}_d$  by a lightweight transformation  $\Theta$  modeling with interaction probability decoder and 3d feature planes in Figure. 2, supervised by deformable masks  $\tilde{y}_m$ . Moreover, leveraging K-Planes [54], the multiple 4d local deformable space can be further compressed in only  $C_4^2 = 6$  feature planes. We iteratively sample from coarse to fine within the multi-scale feature plane, retrieving the maximum probability interaction variables  $\kappa_d$  and indices  $d$  at each scale.

**Feature Repulsion and Probability Rejection.** A latent issue is that optimizing the interaction probability decoder with varying masks can lead to blurred boundaries in the local deformable field, further causing ray sampling and feature storage conflicts. As illustrated in Figure. 4(a), consider

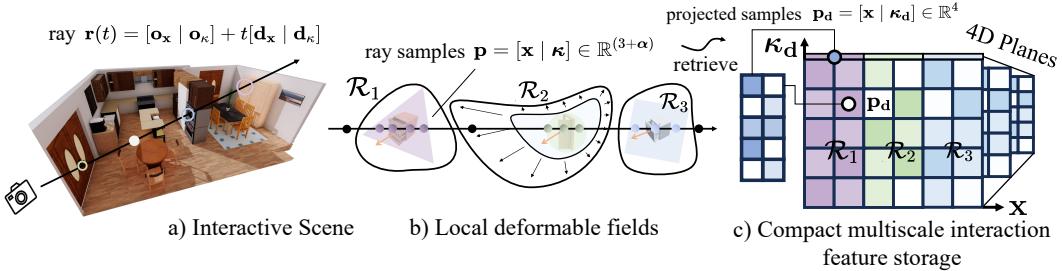


Figure 3: Illustration of the proposed hyperplanar factorization for compact storage and efficient rendering. We maintain multiple local deformable fields  $\mathcal{F}_d$  for each interactive object  $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_\alpha\}$  in the 4D space, and project the high-dimensional interaction features  $\mathcal{F}_\alpha$  into a compact 4D space  $\mathcal{F}_d$ , which can be further compressed into multiscale feature planes.

two adjacent local deformable regions  $\mathcal{R}_i$  and  $\mathcal{R}_j$ , and a point  $\mathbf{p}$  in high-dimensional space, suppose  $\mathbf{p}$  moves from the cluster center  $\mathbf{c}_i$  of  $\mathcal{R}_i$  towards the cluster center  $\mathbf{c}_j$  of  $\mathcal{R}_j$ , then the probability of  $\mathbf{p}$  belonging to  $\mathcal{R}_i$  gradually decreases, while the probability of  $\mathbf{p}$  belonging to  $\mathcal{R}_j$  increases. To avoid sampling conflicts and feature oscillations at the boundaries, we introduce a repulsion loss to amplify the feature differences between distinct deformable scenes, thereby promoting the separation of deformable field:

$$\mathcal{L}_{\text{repuls}} = \frac{1}{\|\mathbf{c}_i - \mathbf{c}_j\| + \delta} \text{ELU}(s - \|(\tilde{\mathbf{y}}_m^i \odot \tilde{\mathbf{y}}_m^j)(\mathcal{F}_s^i - \mathcal{F}_s^j)\|), \quad (4)$$

where  $\tilde{\mathbf{y}}_m^i$  and  $\tilde{\mathbf{y}}_m^j$  are the groundtruth mask of rays, and  $\mathcal{F}_s^i$  and  $\mathcal{F}_s^j$  are the last-layer features of interaction probability decoder in Figure. 2. In training iteration, we randomly select ray pairs  $(\mathbf{r}_i, \mathbf{r}_j)$  and apply  $\mathcal{L}_{\text{repuls}}$  to enforce the separation of interactive probability features across local deformable spaces. Meanwhile, we incorporate spatial information by weighting  $\mathcal{L}_{\text{repuls}}$  with  $\frac{1}{\|\mathbf{c}_i - \mathbf{c}_j\| + \delta}$ , thereby encouraging the model to focus on local sampling conflicts. The centroid of ray pairs  $\mathbf{c}_i$  and  $\mathbf{c}_j$  can be obtained by  $\sum_{i=1}^{i=M} w_i \mathbf{x}_i$ , where  $w_i$  and  $\mathbf{x}_i$  are the weight and 3D position of each sample, and  $M$  is the sample number. Additionally, a probability rejection operation is proposed to truncate the low-probability samples if the deformable probability at  $\mathbf{p}$  is smaller than threshold  $s$ :  $d = \arg \max_i \Theta(\boldsymbol{\kappa}, \theta_s(\mathbf{x}))_i$  if  $\Theta(\boldsymbol{\kappa}, \theta_s(\mathbf{x}))_i \geq s$  else  $-1$ . The rejected sample is considered to be lying beyond the hyperplane and does not belong to the deformable field. As shown in Figure. 4(b), the proposed operations help the model achieve higher rendering quality, demonstrating their effectiveness in alleviating boundary sampling conflicts.

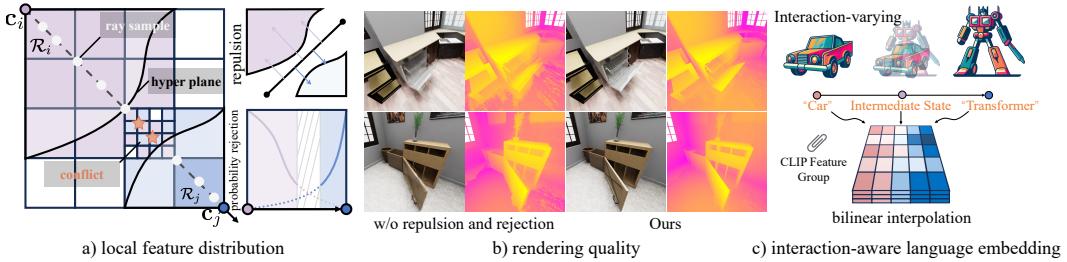


Figure 4: Illustration of a) boundary sampling conflicts, b) rendering quality comparison, and c) interaction-aware language embedding.

### 3.3 Interaction-Aware Language Embedding

Language embedding in interactive scenes is complex and storage-intensive, as 3D distillation faces the dual challenges of high-dimensional optimization and interaction scene variation inconsistency, such as the distinct topological structures of a transformer before and after morph in Figure. 4(c), leading to the failure of SAM [30] segmentation or LERF [28] grounding. Leveraging the proposed multi-scale interaction space factorization in 3.2, we efficiently store language features in lightweight planes by indexing them according to maximum probability sampling instead of 3D fields in LERF. For any sampling point  $\mathbf{p}$ , we project it onto  $\mathbf{p}_d = [\mathbf{x} \mid \boldsymbol{\kappa}_d]$ , retrieving a local language feature group by  $d$ , and perform bilinear interpolation using  $\boldsymbol{\kappa}_d$  to obtain a language embedding that adapts to interactive variable changes from surrounding clip features. Our method has a storage complexity of  $\mathcal{O}(C \times \alpha \times \text{dim})$ , much smaller than language distillation methods like LERF that operate in 3D scenes, where  $\text{dim}$  is the dimension of CLIP feature.

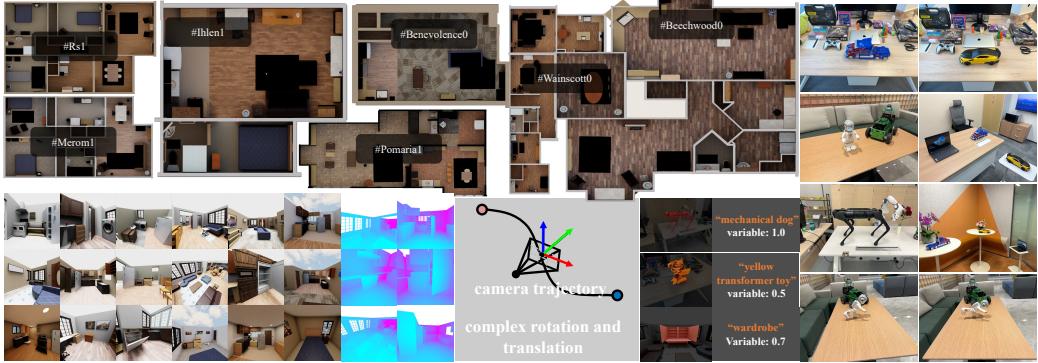


Figure 5: Overview of the OmniSim and InterReal datasets.

## 4 Dataset

To our knowledge, existing view synthetic datasets for interactive scene rendering are primarily limited to a few interactive objects [76, 25, 73, 78] due to necessitating a substantial amount of manual annotation of object masks and states, making it impractical to scale up to real scenarios involving multi-object interactions. To bridge this gap, we construct two scene-level, high-quality annotated datasets to advance research progress in reconstructing and understanding interactive scenes: **OmniSim** and **InterReal**, as shown in Figure. 5. Besides, we use the CoNeRF Synthetic and Controllable [25] dataset for evaluation as well. **1) OmniSim Dataset** is rendered through OmniGibson [34] simulator, leveraging 7 indoor scene models: #rs, #ihlen, #beechwod, #merom, #pomaria, #wainscott and #benevolence. By varying the rotation vectors of the articulated objects’ joints and the camera’s trajectory within the scene, we generated 20 high-definition subsets, each consisting of RGBD images, camera trajectory, interactive object masks, and corresponding object state quantities at each time step. **2) InterReal Dataset** is captured from 8 real Interactable scenes and finely annotated with interaction variables and masks, camera poses encompassing multiple objects, and articulated motion variables. More details can be found in the supplementary.

## 5 Experiment

In this section, we evaluate **LiveScene** with existing SOTA methods on CoNeRF Synthetic and Controllable [25] datasets, and the proposed OmniSim and InterReal datasets, demonstrating our superiority in scene-level interactive scenarios reconstruction and language grounding. Detailed implementation can be found in the supplemental materials.

We compare LiveScene with the existing 3D static rendering methods [45, 47, 27, 26], 4D deformable methods [54, 50, 49], and controllable scene reconstruction methods [25, 73]. Note that we reimplemented CoGS [73] based on Deformable Gaussian [72] since the official code is unavailable. Additionally, we extended K-Planes [54] from  $C_4^2$  planes to  $C_{3+\alpha}^2$  planes, denoted as MK-Planes, where  $\alpha$  represents the number of interactable objects in dynamic scenes. By leveraging the fact that each instance occupies a distinct region, we further compressed the model, denoted as MK-Planes\*, requiring only  $3 + 3\alpha$  planes. Detailed implementations can be found in the supplemental materials.

### 5.1 View Synthesis Quality Comparison

**Evaluation on CoNeRF Synthetic and Controllable Datasets.** We report the quantitative results on CoNeRF Synthetic and Controllable scenes in Table. 1. LiveScene outperforms all the existing methods in PSNR, SSIM, and LPIPS metrics on CoNeRF Synthetic scenes with a large margin. In particular, LiveScene achieves 43.349, 0.986, and 0.011 in PSNR, SSIM, and LPIPS, respectively, outperforming the second-best method by 9.894, 0.009, and 0.053. On CoNeRF Controllable, LiveScene achieves the best PSNR of 32.782 and comparable SSIM and LPIPS to the SOTA methods. The results demonstrate the effectiveness of LiveScene in modeling object-level interactive scenarios.

**Evaluation on OmniSim and InterReal Datasets.** OmniSim dataset is categorized into 3 interaction level subsets: #easy, #medium, and #challenging, based on the number of interactive objects in each scene. As shown in Table. 2, LiveScene achieves the best PSNR, SSIM, and LPIPS on all interaction level subsets of OmniSim, with average PSNR, SSIM, and LPIPS of 33.158, 0.962, and 0.074, respectively. Notably, substantial performance degradation is observed across all methods as the quantity and complexity of interactive objects increase, e.g., CoGS [73] experiences a 3.641 dB PSNR

Table 1: **Quantitative results on CoNeRF synthetic and controllable datasets.** LiveScene achieves the best results in all metrics on synthetic scenes and the best PSNR on the controllable datasets.

Method	CoNeRF Synthetic			CoNeRF Controllable		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [45]	25.299	0.843	0.197	28.795	0.951	0.210
InstantNGP [47]	27.057	0.903	0.230	26.391	0.884	0.278
3DGS [26]	32.576	0.977	0.077	25.945	0.834	0.414
NeRF + Latent [45]	28.447	0.939	0.115	32.653	0.981	0.182
Nerfies[49]				32.274	0.981	0.180
HyperNeRF[50]	25.963	0.854	0.158	32.520	0.981	0.169
K-Planes [54]	33.301	0.933	0.150	31.811	0.912	0.262
CoNeRF-M[25]	27.868	0.898	0.155	32.061	0.979	0.167
CoNeRF[25]	32.394	0.972	0.139	32.342	0.981	0.168
CoGS [73]	33.459	0.960	0.064	32.601	0.983	0.164
LiveScene (Ours)	<b>43.349</b>	<b>0.986</b>	<b>0.011</b>	<b>32.782</b>	0.932	0.186

Table 2: **Quantitative results on OmniSim Dataset.** LiveScene outperforms prior works on most metrics and achieves the best PSNR on the #challenging subset with a significant margin.

Method	#Easy Sets			#Medium Sets			#Challenging Sets			#Avg (all 20 Sets)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [45]	25.817	0.906	0.167	25.645	0.928	0.138	26.364	0.927	0.128	25.776	0.916	0.153
InstantNGP [47]	25.704	0.902	0.183	25.627	0.930	0.140	26.367	0.920	0.143	25.706	0.914	0.164
HyperNeRF [50]	30.708	0.908	0.316	31.621	0.936	0.265	27.533	0.897	0.318	30.748	0.917	0.299
K-Planes [54]	32.841	0.952	0.093	32.548	0.954	0.100	29.833	0.937	0.118	32.573	0.952	0.097
CoNeRF [25]	32.104	0.932	0.254	33.256	0.951	0.207	30.349	0.923	0.238	32.477	0.939	0.234
MK-Planes*	31.630	0.948	0.098	31.880	0.951	0.104	26.565	0.887	0.218	31.477	0.946	0.106
MK-Planes	31.677	0.948	0.098	32.165	0.952	0.099	29.254	0.933	0.119	31.751	0.949	0.099
CoGS [73]	32.315	0.961	0.108	32.447	<b>0.965</b>	0.086	28.701	<b>0.970</b>	<b>0.073</b>	32.187	<b>0.963</b>	0.097
LiveScene (Ours)	<b>33.221</b>	<b>0.962</b>	<b>0.072</b>	<b>33.262</b>	<b>0.965</b>	<b>0.072</b>	<b>31.645</b>	0.948	0.093	<b>33.158</b>	0.962	<b>0.074</b>

drop from #easy to #challenging. While LiveScene maintains a relatively stable high performance across all subsets, demonstrating its robustness in modeling complex interactive scenarios.

We also divided InterReal dataset into 2 subsets: #medium and #challenging. In Table. 3, CoGS [73] underperforms compared to LiveScene on the #medium subset and fails to converge when faced with long camera trajectories and a large number of interactive objects in the scene (#challenging), highlighting the limitation of existing methods in modeling real-world interactive scenarios. In contrast, LiveScene achieves the highest PSNR of 28.436 and the lowest LPIPS of 0.185 on the #challenging subset, indicating its superiority in modeling real-world large-scale interactive scenarios.

Figure. 6 presents the novel view synthesis results of LiveScene and the SOTA methods on OmniSim dataset. The results reveal that LiveScene generates more detailed results than SOTA methods, particularly in complex interactive scenarios. For instance, on the #pomaria scene featuring an openable dishwasher, CoNeRF [25] fails to capture details, while CoGS [73] and MK-Planes\* exhibit residual artifacts. In contrast, our method accurately reconstructs the internal details. Another challenge arises in the #rs scene, where other methods struggle to reconstruct distant and static objects. In comparison, our method not only overcomes the challenging problem of dramatic topology changes in interactive scenes but also maintains the ability of high-quality static scene reconstruction.

Table 3: **Quantitative results on InterReal Dataset.** Our method outperforms others in most settings, with a significant advantage of PSNR, SSIM, and LPIPS on the #challenging subset.

Method	#Medium Sets			#Challenging Sets			#Avg (all 8 Sets)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [45]	20.816	0.682	0.190	21.169	0.728	0.337	20.905	0.694	0.227
InstantNGP [47]	21.700	0.776	0.215	21.643	0.745	0.338	21.686	0.769	0.245
HyperNeRF [50]	25.283	0.671	0.467	25.261	0.713	0.517	25.277	0.682	0.480
K-Planes [54]	27.999	0.813	0.177	26.427	<b>0.756</b>	<b>0.331</b>	27.606	0.799	0.215
CoNeRF [25]	27.501	0.745	0.367	26.447	0.734	0.472	27.237	0.742	0.393
CoGS [73]	30.774	<b>0.913</b>	0.100	X	X	X	<b>30.774</b>	<b>0.913</b>	0.100
LiveScene (Ours)	<b>30.815</b>	0.911	<b>0.066</b>	<b>28.436</b>	<b>0.846</b>	<b>0.185</b>	30.220	0.895	<b>0.096</b>

## 5.2 Interactive Scenes Geometry Comparison

To evaluate the completeness of the topological structure of interactive objects, we employ the depth L1 error metric. As shown in Figure. 7, our method outperforms SOTA methods on scenes from OmniSim. While existing methods excel in RGB image rendering, they struggle with depth structure representation. Specifically, CoNeRF [25] performs relatively well in #seq08 and #seq15 but fails in large-scale scenes (#seq17 and #seq19). CoGS [73] exhibit notable artifacts in the depth map. Moreover, MK-Planes\* also fails to recover depth around interactive objects. In contrast, our method achieves the lowest depth error and renders satisfying depth maps, demonstrating its accurate interactive scene modeling capabilities.

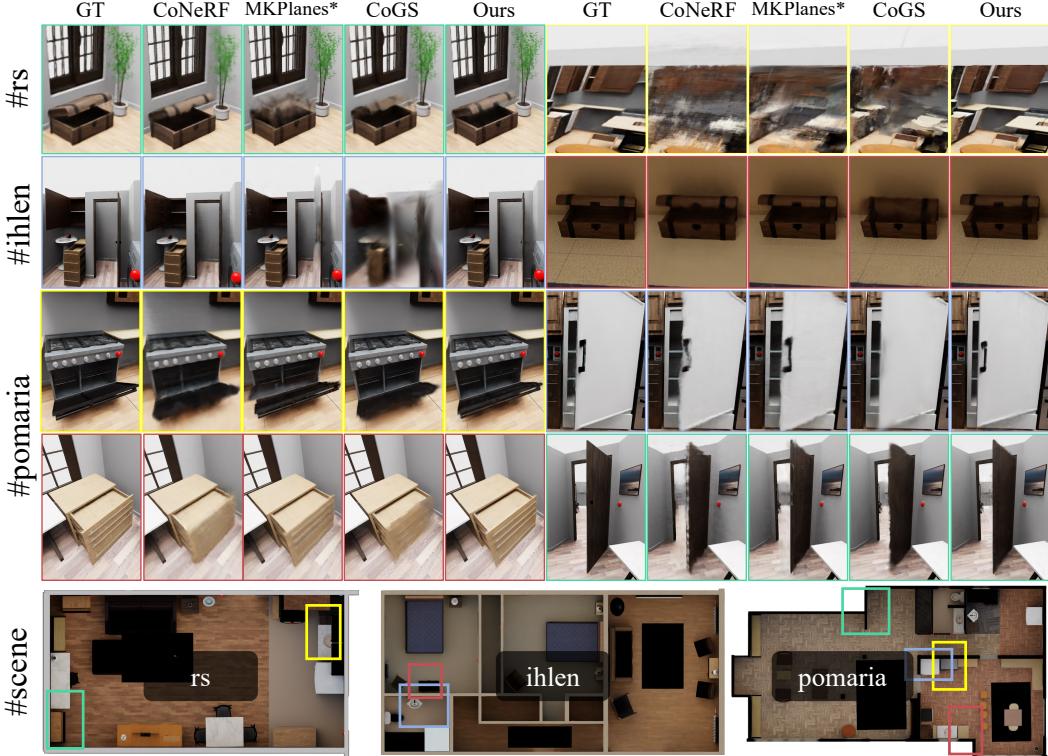


Figure 6: **View Synthesis Visualization on OmniSim Dataset.** We compare our method with SOTA methods on RGB rendering across three scenes: #rs, #ihlen, and #pomaria. Boxes of different colors represent distinct interactive objects within the scene.

	Depth L1 Error ↓				
scene	CoNeRF	MK-Planes*	CoGS	LiveScene	Ours
#seq04	0.029	0.037	0.381	<b>0.018</b>	
#seq08	<b>0.018</b>	0.301	0.655	0.025	
#seq14	0.042	0.103	0.810	<b>0.039</b>	
#seq15	<b>0.019</b>	0.568	0.690	0.021	
#seq17	X	0.282	0.706	<b>0.019</b>	
#seq19	X	0.136	0.689	<b>0.034</b>	
avg	0.027	0.238	0.655	<b>0.026</b>	

Figure 7: **Structure Reconstruction Performance on OmniSim Dataset.** left): Our method surpasses most previous works on chosen subsets. right): LiveScene excels in both reconstructing intricate details of interactive objects and preserving the geometric integrity of static background.

### 5.3 Language Grounding Comparison

We assess the language grounding performance on OmniSim dataset using mIOU metric. Figure. 8 suggests that our method obtains the highest mIOU score, with an average of 86.86. In contrast, traditional methods like LERF [28] encounter difficulties in locating objects precisely, with an average mIOU of 21.74. Meanwhile, 2D methods like SAM [30] fail to accurately segment the whole target under specific viewing angles, as objects appear discontinuous in the image. Conversely, our method perceives the completeness of the object and has clear knowledge of its boundaries, demonstrating its advantage in language grounding tasks.

### 5.4 Ablation Study

In this section, we present ablative studies to investigate the effectiveness of each component in LiveScene. We selected 1 scene from the #medium subset and 2 scenes from the #easy subset for OmniSim dataset, and 1 scene each from the #medium and #challenging settings for InterReal. Notably, ground-truth state quantities are only available in OmniSim, but not in InterReal. Therefore, we use GT quantities on OmniSim and introduce a learnable variable on InterReal to infer state changes. Figure. 9 reports the rendering quality and grounding performance for #1 and #6.

As illustrated in Table. 4, the multi-scale factorization significantly improves the rendering performance on both datasets, with PSNR on OmniSim increasing from 31.74 to 35.094, shown in #1.

		mIOU ↑		
OmniSim	setting	SAM [30]	LERF [28]	Ours
	#easy	61.58	23.60	<b>86.94</b>
	#medium	55.13	19.40	<b>86.32</b>
	#challenging	63.86	19.87	<b>90.41</b>
	#avg	59.11	21.74	<b>86.86</b>

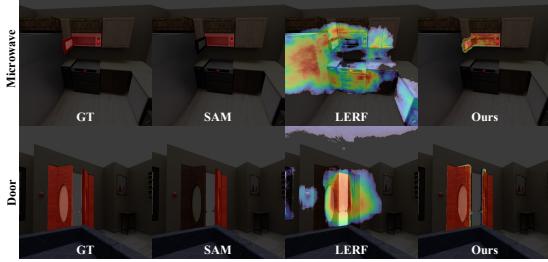


Figure 8: **Language Grounding Performance on OmniSim Dataset** left): Our method gains the highest mIOU score. right): LiveScene’s grounding exhibits clearer boundaries than other methods.

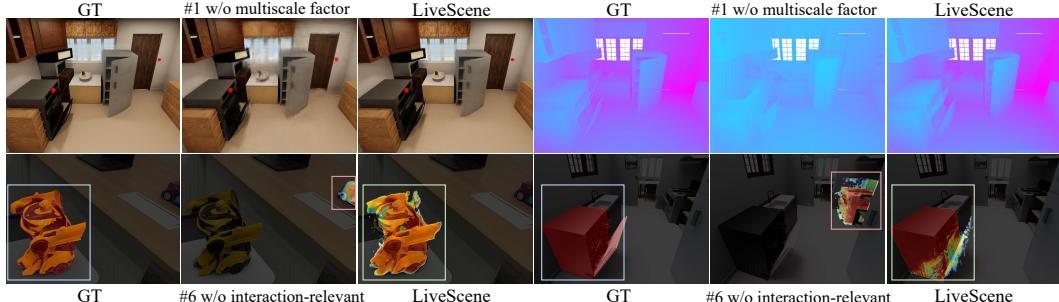


Figure 9: **Rendering and Grounding Performance for #1 and #6.** above): Multi-scale factorization greatly boosts the performance of RGB rendering and geometry reconstruction. below): Without view consistency, the model struggles when objects have similar appearances.

Introducing learnable variables for each frame (#2) yields corresponding improvements on InterReal dataset since this latent code can perceive the change in object states. The feature repulsion loss and probability rejection (#3 and #4) together make rendering quality better in InterReal as well as in OmniSim dataset. As for grounding, #5 shows that rendering embeddings along a ray [31, 79] struggles to locate objects precisely. Ensuring view consistency further boosts grounding performance on OmniSim, as demonstrated in #6.

Table 4: **Ablation Study** on the subset of InterReal and OmniSim Datasets.

#exp	#settings (rendering)						InterReal			OmniSim			
	I	II	III	IV	V	VI	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Depth L1↓
#0							25.329	0.731	0.329	31.74	0.938	0.118	0.238
#1	✓						28.289	0.819	0.226	35.094	0.969	0.059	0.086
#2	✓	✓					29.577	0.865	0.162	—	—	—	—
#3	✓	✓*					29.959	0.883	0.131	34.989	0.969	0.059	0.085
#4	✓	✓*	✓	✓			30.123	0.884	0.132	34.977	0.967	0.061	0.086
LiveScene	✓	✓*	✓	✓	✓		30.591	0.896	0.115	35.254	0.971	0.057	0.042
#settings (grounding)							mIOU ↑			mIOU ↑			
#5	✓	✓*	✓	✓	✓	✓	30.40	—	—	32.87	—	—	—
#6	✓	✓*	✓	✓	✓	✓	93.10	—	—	71.64	—	—	—
LiveScene	✓	✓*	✓	✓	✓	✓	93.02	—	—	78.52	—	—	—

I: multi-scale factorization, II: learnable variable, III: feature repulsion  $\mathcal{L}_{\text{repuls}}$ , IV: probability rejection, V: maximum probability embeds retrieval, VI: interaction-aware language embedding. ✓\* denotes enable II for InterReal but disable for OmniSim.

## 6 Conclusion and Limitation

We present LiveScene, the first language-embedded interactive neural radiance field for complex scenes with multiple interactive objects. A parameter-efficient factorization technique is proposed to decompose interactive spaces into local deformable fields to model individual interactive objects. Moreover, we introduce a novel interaction-aware language embedding mechanism that effectively localizes and controls interactive objects using natural language. Finally, We construct two challenging datasets that contain multiple interactive objects in complex scenes and evaluate the effectiveness and robustness of LiveScene.

**Limitations:** The control ability of LiveScene is limited by label density. Additionally, our natural language control approach is currently restricted to closed vocabulary, and it is inherently tied to the capabilities of the underlying foundation model, such as OpenCLIP. In future work, we plan to extend our method to enable open-vocabulary grounding and control, increasing the model’s flexibility and range of applications.

## A Appendix / supplemental material

### Abstract

This supplementary material accompanies the main paper by providing more details for reproducibility as well as additional evaluations and qualitative results to verify the effectiveness and robustness of LiveScene:

- ▷ **appendix B:** Configurations of OmniSim and InterReal dataset, including scene assets, interaction variables generation, mask and prompts annotation, and dataset visualization.
- ▷ **appendix C:** Video demonstration and anonymous link: <https://livescenes.github.io>.
- ▷ **appendix D:** Additional implementation details.
- ▷ **appendix E:** Additional experimental results, including more ablation studies, detailed view synthesis quality comparison, interactive scenes geometry comparison and language grounding comparison.

## B Configurations of OmniSim and InterReal datasets

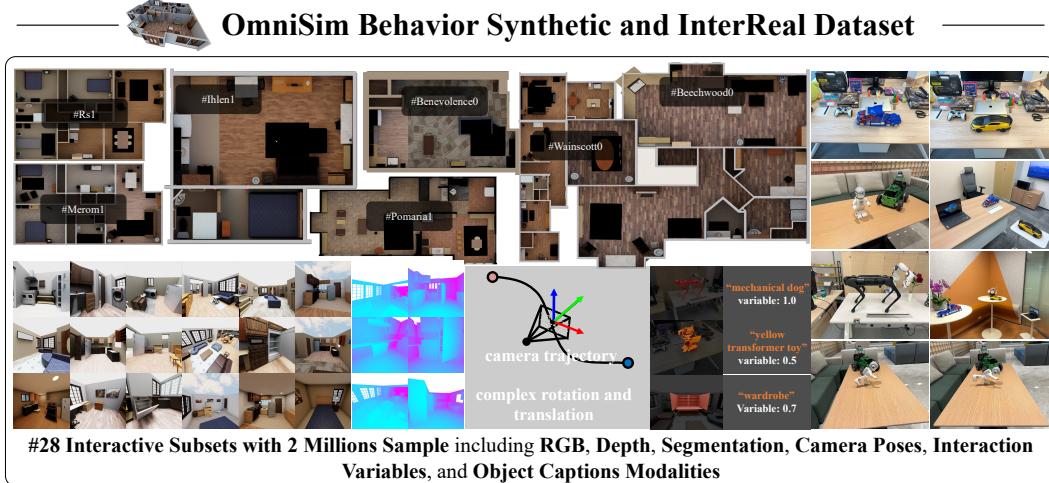


Figure 10: Illustration of the proposed Omniverse behavior synthetic (OminiSim) and Real captured interactive (InterReal) dataset. These datasets are captured in an OmniGibson simulator or real scene and carefully annotated, providing #28 interactive subsets with 2 Million samples, including RGB, depth, segmentation, camera trajectory, interaction variables, and object captions modalities.

**Scene Assets and Generation Pipeline for OmniSim.** We generate the synthetic dataset using the OmniGibson simulator. The dataset consists of 20 interactive scenes from 7 scene models: #rs, #ihlen, #beechwood, #merom, #pomaria, #wainscott, and #benevolence. The scenes feature various interactive objects, including cabinets, refrigerators, doors, drawers, and more, each with different hinge joints.

We configure the simulator camera with an intrinsic parameter set of focal length 8, aperture 20, and a resolution of  $1024 \times 1024$ . By varying the rotation vectors for each joint of the articulated objects, we can observe different motion states of various objects. We generated 20 high-definition subsets, each consisting of RGB images, depth, camera trajectory, interactive object masks, and corresponding object state quantities relative to their "closed" state at each time step, from multiple camera trajectories and viewpoints.

The data is obtained through the following steps: 1) The scene model is loaded, and the respective objects are selected, with motion trajectories set for each joint. 2) Keyframes are set for camera movement in the scene, and smooth trajectories are obtained through interpolation. 3) The simulator is then initiated, and the information captured by the camera at each moment is recorded.

**Scene Assets and Generation Pipeline for InterReal.** InterReal is primarily captured using the Polycam app on an Apple iPhone 15 Pro. We selected 8 everyday scenes and placed various interactive objects within each scene, including transformers, laptops, microwaves, and more. We recorded 8 videos, each at a frame rate of 5FPS, capturing 700 to 1000 frames per video.

The dataset was processed via the following steps: 1) manual object movement and keyframe capture, 2) OBJ file export and pose optimization using Polycam, 3) conversion to a dataset containing RGB images and transformation matrices using Nerfstudio [60], and 4) mask generation for each object in each scene using SAM [30] and corresponding prompts and state quantity labeling for certain keyframes.

**Statistic of OmniSim and InterReal Datasets.** The detailed statistics of the OmniSim and InterReal datasets are shown in Table. 5. The OmniSim dataset consists of 20 interactive scenes, each with 2 to 6 objects, and the InterReal dataset contains 8 real-world scenes, each with 1 to 3 objects. The datasets include RGB, depth, pose, mask, and text prompts modalities, providing a total of 2 million samples for training and evaluation. The objects in the datasets include cabinets, refrigerators, doors, drawers, transformers, laptops, microwaves, and more, with various interactive states and captions.

Table 5: Statistic of OmniSim and InterReal Datasets.

	datasets	#objects	#frame	#key frame value	rgb	depth	pose	mask	text prompts
OmniSim	#seq001_Rs_int	4	770	770	✓	✓	✓	✓	fridge, microwave, oven, top cabinet
	#seq002_Rs_int	4	2190	2190	✓	✓	✓	✓	fridge, microwave, oven, top cabinet
	#seq003_Ihlen_1_int	4	1610	1610	✓	✓	✓	✓	bottom cabinet, dishwasher, oven, top cabinet
	#seq004_Ihlen_1_int	2	1630	1630	✓	✓	✓	✓	bottom cabinet, cedar chest
	#seq005_Beechwood_0_int	2	1370	1370	✓	✓	✓	✓	bottom cabinet, door
	#seq006_Beechwood_0_int	2	1610	1610	✓	✓	✓	✓	dishwasher, microwave
	#seq007_Beechwood_0_int	3	1450	1450	✓	✓	✓	✓	bottom cabinet, door, top cabinet
	#seq008_Benevolence_1_int	4	1830	1830	✓	✓	✓	✓	door, fridge, microwave, top cabinet
	#seq009_Benevolence_1_int	2	1690	1690	✓	✓	✓	✓	cedar chest, door
	#seq010_Merom_1_int	3	1930	1930	✓	✓	✓	✓	dishwasher, fridge, microwave, top cabinet
	#seq011_Merom_1_int	3	1690	1690	✓	✓	✓	✓	bottom cabinet, top cabinet, door
	#seq012_Pomaria_1_int	2	970	970	✓	✓	✓	✓	bottom cabinet, fridge
	#seq013_Pomaria_1_int	3	770	770	✓	✓	✓	✓	bottom cabinet, fridge
	#seq014_Wainscott_0_int	2	1850	1850	✓	✓	✓	✓	bottom cabinet, cedar chest
	#seq015_Wainscott_0_int	2	1350	1350	✓	✓	✓	✓	bottom cabinet, door
	#seq016_Wainscott_0_int	2	1170	1170	✓	✓	✓	✓	fridge, stove
	#seq017_Benevolence_1_int	6	4590	4590	✓	✓	✓	✓	cedar chest, door, door, fridge, microwave, top cabinet
	#seq018_Benevolence_1_int	2	2050	2050	✓	✓	✓	✓	door, top cabinet
	#seq019_Rs_int	2	1130	1130	✓	✓	✓	✓	fridge, top cabinet
	#seq020_Merom_1_int	2	1990	1990	✓	✓	✓	✓	bottom cabinet, door
InterReal	#demo	5	6267	6267	✓	✓	✓	✓	cedar chest, door, fridge, oven, top cabinet
	#demo001	4	2040	2040	✓	✓	✓	✓	fridge, microwave, oven, top cabinet
	#demo002	3	2395	2395	✓	✓	✓	✓	dishwasher, microwave, top cabinet
	#demo003	3	2480	2480	✓	✓	✓	✓	dishwasher, oven, top cabinet
	#demo004	3	2280	2280	✓	✓	✓	✓	dishwasher, fridge, stove
	#demo005	3	1670	1670	✓	✓	✓	✓	bottom cabinet, cedar chest, door
	#seq001_transformer	1	329	38	✓	✗	✓	✓	yellow toy car
	#seq002_transformer	1	329	43	✓	✗	✓	✓	blue toy car
	#seq003_door	1	355	31	✓	✗	✓	✓	door
	#seq004_dog	1	213	41	✓	✗	✓	✓	black mechanical dog
	#seq005_sit	1	913	25	✓	✗	✓	✓	small white humanoid
	#seq006_stand	1	899	33	✓	✗	✓	✓	small white humanoid
	#seq007_flower	3	620	153	✓	✗	✓	✓	blue toy car, yellow toy car, black laptop
	#seq008_office	4	1087	658	✓	✗	✓	✓	blue toy car, yellow toy car, black laptop, microwave

## C Videos Demonstration and Anonymous Link

We provide a video of our proposed method LiveScene along with this document to demonstrate the interactive scene reconstruction and multimodal control capabilities. Please refer to the anonymous link: <https://livescenes.github.io> for more information.

## D Additional implementation details

**Loss Functions.** In this section, we provide detailed descriptions of the loss functions used in LiveScene:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda_1 \mathcal{L}_{focus} + \lambda_2 \mathcal{L}_{repuls} + \lambda_3 \mathcal{L}_{var} + \lambda_4 \mathcal{L}_{lang} + \lambda_5 \mathcal{L}_{smooth}, \quad (5)$$

**Rendering Loss.** We use the standard NeRF rendering loss, which is the sum of the mean squared error (MSE) between the rendered color and the ground truth color, and the MSE between the rendered depth and the ground truth depth. The loss is computed for each pixel in the image and

averaged over the entire image:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{M} \sum_{i=1}^M \left\| \mathbf{C}_i - \tilde{\mathbf{C}}_i \right\|^2, \quad (6)$$

where  $\mathbf{C}_i$  and  $\tilde{\mathbf{C}}_i$  are the rendered and ground truth RGB values, respectively, and  $M$  is the number of pixels in the image.

**Focal Loss.** Due to the predominance of background regions in the images, we employ focal loss to enhance the model's focus on the relatively smaller interactive mask regions:

$$\mathcal{L}_{\text{focus}} = \beta \cdot \left( 1 - e^{\sum_{i=1}^{\alpha} \tilde{\mathbf{y}}_m^i \log(\mathbf{q}_i)} \right)^{\gamma} \cdot \left( - \sum_{i=1}^{\alpha} \tilde{\mathbf{y}}_m^i \log(\mathbf{q}_i) \right), \quad (7)$$

where  $\tilde{\mathbf{y}}_m^i$  is the ground truth label,  $\mathbf{q}$  is the predicted probability  $\Theta(\kappa, \theta_s(\mathbf{x}))$  rendering from the interactive probability field,  $\beta$  is the balancing factor, and  $\gamma$  is the focusing parameter. In our experiments, we set  $\alpha = 0.5$  and  $\gamma = 1.5$ .

**Repulsion Loss.** To avoid sampling conflicts and feature oscillations at the boundaries, we introduce a repulsion loss to amplify the feature differences between distinct deformable scenes, thereby promoting the separation of deformable field:

$$\mathcal{L}_{\text{repuls}} = \frac{1}{\|\mathbf{c}_i - \mathbf{c}_j\| + \delta} \text{ELU}(s - \|(\tilde{\mathbf{y}}_m^i \odot \tilde{\mathbf{y}}_m^j)(\mathcal{F}_s^i - \mathcal{F}_s^j)\|), \quad (8)$$

where  $\tilde{\mathbf{y}}_m^i$  and  $\tilde{\mathbf{y}}_m^j$  are the groundtruth mask of rays, and  $\mathcal{F}_s^i$  and  $\mathcal{F}_s^j$  are the last-layer features of probability decoder in Figure. 2. In training iteration, we randomly select ray pairs  $(\mathbf{r}_i, \mathbf{r}_j)$  and apply  $\mathcal{L}_{\text{repuls}}$  to enforce the separation of interactive probability features across local deformable spaces. Meanwhile, we incorporate spatial information by weighting  $\mathcal{L}_{\text{repuls}}$  with  $\frac{1}{\|\mathbf{c}_i - \mathbf{c}_j\| + \delta}$ , thereby encouraging the model to focus on local sampling conflicts. The centroid of ray pairs  $\mathbf{c}_i$  and  $\mathbf{c}_j$  can be obtained by  $\sum_{i=1}^{M_w} w_i \mathbf{x}_i$ , where  $M_w$ ,  $w_i$ , and  $\mathbf{x}_i$  are the sample number, weight, and 3D position of each sample. In our experiments, we set  $\delta = 0.5$  and  $s = 1e2$ .

**Interaction Variable MSE.** We follow the value MSE in [25] and use the standard MSE loss to supervise the interaction values training:

$$\mathcal{L}_{\text{var}} = \frac{1}{N} \sum_{i=1}^N \left\| \kappa_i - \tilde{\kappa}_i \right\|^2, \quad (9)$$

where  $\kappa_i$  and  $\tilde{\kappa}_i$  are the predicted and ground truth interaction variables, respectively, and  $N$  is the number of ray samples of a batch. Note that we only apply  $\mathcal{L}_{\text{var}}$  to the InterReal dataset and use learnable variables as inputs to the model due to the lack of dense ground truth interaction variables. In OmniSim, we directly use the ground truth interaction variables as inputs provided by the simulator to achieve precise control.

**Language Embedding L2 Loss.** In LiveScene implementation, we use the huber loss to supervise the language embedding training. But we do not distill the language embedding in the 3D language field but we store the language embedding in the proposed interaction-aware language feature plane. The loss is defined as:

$$\mathcal{L}_{\text{lang}}(\phi(\mathbf{p}), \tilde{\phi}(\mathbf{p})) = \begin{cases} \frac{1}{2}(\phi(\mathbf{p}) - \tilde{\phi}(\mathbf{p}))^2 & \text{if } |\phi(\mathbf{p}) - \tilde{\phi}(\mathbf{p})| \leq \delta \\ \delta \left( |\phi(\mathbf{p}) - \tilde{\phi}(\mathbf{p})| - \frac{1}{2}\delta \right) & \text{otherwise} \end{cases}, \quad (10)$$

where  $\phi(\mathbf{p})$  and  $\tilde{\phi}(\mathbf{p})$  are the predicted and ground truth language embeddings, respectively, and  $\delta$  is the threshold. In our experiments, we set  $\delta = 1.0$ .

**Smoothness Loss.** Inspired by K-Planes [54], we use 1D Laplacian (second derivative) filter to smooth the local deformable field feature plane, which helps to reduce the noise in the deformable field and alleviate feature oscillations and sampling conflicts at the sampling boundary:

$$\mathcal{L}_{\text{smooth}}(\mathbf{p}) = \frac{1}{|L|n^2} \sum_{l,i,k} \left\| \mathbf{p}_l^{i,k-1} - 2\mathbf{p}_l^{i,k} + \mathbf{p}_l^{i,k+1} \right\|_2^2, \quad (11)$$

where  $i$  and  $k$  are indices on the plane resolution  $n$ , and  $l$  is the feature planes index.

**Probability Rejection Operation.** The probability rejection operation is proposed to truncate the low-probability samples if the maximum deformable probability  $\mathbf{q}$  at  $\mathbf{p}$  is smaller than threshold  $s$  and select the background feature directly (-1). The operation is defined as:

$$d = \begin{cases} \arg \max_i \Theta(\boldsymbol{\kappa}, \boldsymbol{\theta}_s(\mathbf{x}))_i & \text{if } \geq s \\ -1 & \text{otherwise} \end{cases}, \quad (12)$$

**Implementation Details.** LiveScene is implemented in Nerfstudio [60] from scratch. We represent the field as a multi-scale feature plane with resolutions of  $512 \times 256 \times 128$ , and feature dimension of 32. The proposal network adopts a coarse-to-fine sampling process, where each sampling step concatenates the position feature and the state quantity as the query for the 4D deformation mask field, which is a 1-layer MLP with 64 neurons and ReLU activation. For InterReal, we introduce additional learnable variables bound to each frame to capture changes in object states within the scene. These variables are represented by a plane with a resolution typically half the frame number, with a feature dimension of 4 for most scenes. For all experiments, we use the Adam optimizer with initial learning rates of 0.01 and a cosine decay scheduler with 512 warmup steps for all networks. We set loss weights as follows:  $\lambda_1 = 1e-3$ ,  $\lambda_2 = 1e-2$ ,  $\lambda_3 = 1e-3$ ,  $\lambda_4 = 1.0$ ,  $\lambda_5 = 1e-3$ . The model is trained for 80k steps on the OmniSim dataset and 100k steps on the InterReal dataset, using a batch size of 4096 rays with 64 samples each. We run the model on an NVIDIA A100 GPU, requiring approximately 4 hours and 40GB of memory.

## E Additional Experimental Results

**Model Parameter Efficiency Comparison.** We compare the number of parameters of LiveScene with other methods in Table. 6, varying the number of interactive objects in the scene. The results show that LiveScene has a constant number of parameters, regardless of the number of interactive objects, making it more efficient than other methods. In contrast, CoGS [73] has a higher base number of parameters and a linear increase with the number of interactive objects. MK-Planes [54] exhibits a quadratic increase in parameters with the number of interactive objects. Although NeRF[45] and InstantNGP [47] have a low number of parameters, they are limited to 3D static scene reconstruction.

Table 6: Model Parameters vs Object Quantity.

Method	# interactive objects						Trend
	1	2	3	4	5	6	
NeRF [45]	13.23	13.23	13.23	13.23	13.23	13.23	Constant
InstantNGP [47]	11.68	11.68	11.68	11.68	11.68	11.68	Constant
K-Planes [54]	35.66	35.66	35.66	35.66	35.66	35.66	Constant
MK-Planes	35.66	35.96	36.32	36.76	37.27	37.86	Quadratic
MK-Planes*	35.66	35.88	36.10	36.32	36.54	36.76	Linear
CoGS [73]	42.24	43.23	44.23	45.23	46.23	47.22	Linear
LiveScene (Ours)	34.52	34.52	34.52	34.52	34.52	34.52	Constant

**View Synthesis Quality Comparison on OmniSim and InterReal dataset** We provide detailed quantitative results on the OmniSim and InterReal datasets in Table. 7 and Table. 8 provide detailed quantitative results on the OmniSim and InterReal datasets, respectively. LiveScene outperforms prior works on most metrics and achieves the best PSNR on the #challenging and #office subsets with a significant margin. Note that the #challenging and #office subsets contain scenes with multiple interactive objects and large deformable fields, which are challenging for existing methods. We report the score as NaN if the model fails to converge or is out of memory during training multiple times.

**More Detailed Rendering Comparison** We provide more detailed visual comparisons, including RGB, depth, and language grounding on the OmniSim and InterReal datasets in Figure. 11, Figure. 12, Figure. 13, and Figure. 14, respectively. Our method surpasses existing approaches by reconstructing more detailed and accurate representations of the objects. In both datasets, LiveScene can generate more accurate and detailed object shapes and textures, especially for scenes with multiple interactive objects and large deformable fields. Compared with LERF [28], our method can generate more accurate language grounding results, which is crucial for interactive object manipulation tasks, demonstrated in Figure. 14.

**Table 7: Detailed Quantitative results on OmniSim Dataset.** LiveScene outperforms prior works on most metrics and achieves the best PSNR on the #challenging subset with a significant margin.

Dataset	Metric	NeRF [45]	Instant-NGP [47]	HyperNeRF [50]	K-Planes [54]	CoNeRF [25]	MK-Planes [54]	MK-Planes* [54]	CoGS [73]	LiveScene
#seq001_Rs	PSNR	25.941	25.768	NaN	33.136	34.035	32.169	32.092	32.211	34.784
#seq001_Rs	SSIM	0.931	0.933	NaN	0.953	0.957	0.946	0.946	0.968	0.974
#seq001_Rs	LPIPS	0.118	0.113	NaN	0.093	0.135	0.110	0.110	0.068	0.048
#seq002_Rs	SSIM	0.917	0.925	0.939	0.960	0.957	0.960	0.959	0.974	0.966
#seq002_Rs	PSNR	22.927	22.765	28.985	30.120	31.092	29.461	29.396	30.254	30.029
#seq002_Rs	LPIPS	0.173	0.158	0.275	0.093	0.233	0.087	0.088	0.065	0.074
#seq003_Ihlen	LPIPS	0.425	0.424	0.270	0.106	0.225	0.087	0.089	0.327	0.068
#seq003_Ihlen	SSIM	0.794	0.808	0.950	0.956	0.963	0.963	0.962	0.879	0.969
#seq003_Ihlen	PSNR	20.059	20.854	33.119	34.462	34.598	33.679	33.653	25.947	35.223
#seq004_Ihlen	LPIPS	0.113	0.123	0.229	0.107	0.185	0.118	0.116	0.072	0.067
#seq004_Ihlen	SSIM	0.943	0.940	0.945	0.948	0.960	0.941	0.941	0.980	0.970
#seq004_Ihlen	PSNR	25.750	25.574	32.691	31.914	34.319	30.926	30.916	33.795	33.393
#seq005_Beechwood	PSNR	26.364	26.367	27.533	29.833	30.349	29.254	26.565	28.701	31.645
#seq005_Beechwood	SSIM	0.927	0.920	0.897	0.937	0.923	0.933	0.887	0.970	0.948
#seq005_Beechwood	LPIPS	0.128	0.143	0.318	0.118	0.238	0.119	0.218	0.073	0.093
#seq006_Beechwood	LPIPS	0.103	0.119	0.314	0.097	0.283	0.095	0.096	0.059	0.077
#seq006_Beechwood	PSNR	27.988	28.150	29.513	31.861	30.058	31.541	31.521	31.272	32.206
#seq006_Beechwood	SSIM	0.938	0.938	0.907	0.951	0.917	0.951	0.951	0.974	0.959
#seq007_Beechwood	SSIM	0.953	0.951	NaN	0.955	0.962	0.942	0.942	0.975	0.976
#seq007_Beechwood	LPIPS	0.080	0.092	NaN	0.083	0.202	0.110	0.108	0.078	0.050
#seq007_Beechwood	PSNR	27.257	27.191	NaN	30.721	32.307	28.288	28.134	32.949	34.266
#seq008_Benevolence	LPIPS	0.104	0.102	0.389	0.085	0.366	0.145	0.135	0.209	0.072
#seq008_Benevolence	SSIM	0.927	0.942	0.878	0.955	0.888	0.919	0.926	0.915	0.962
#seq008_Benevolence	PSNR	30.847	31.800	31.115	36.157	32.684	34.863	35.000	31.055	36.712
#seq009_Benevolence	LPIPS	0.154	0.161	NaN	0.123	0.166	0.162	0.164	0.045	0.056
#seq009_Benevolence	SSIM	0.925	0.930	NaN	0.943	0.964	0.931	0.930	0.981	0.970
#seq009_Benevolence	PSNR	25.277	24.018	NaN	30.860	33.682	30.390	30.291	32.868	33.592
#seq010_Merom	SSIM	0.950	0.946	NaN	0.967	0.951	0.976	0.968	0.979	0.969
#seq010_Merom	PSNR	28.616	28.660	NaN	34.765	34.286	36.532	34.580	34.497	35.190
#seq010_Merom	LPIPS	0.096	0.112	NaN	0.074	0.217	0.036	0.074	0.051	0.070
#seq011_Merom	PSNR	26.732	27.077	NaN	33.394	30.483	32.951	32.910	31.767	33.426
#seq011_Merom	SSIM	0.932	0.933	NaN	0.959	0.932	0.959	0.959	0.968	0.960
#seq011_Merom	LPIPS	0.112	0.117	NaN	0.074	0.246	0.073	0.072	0.091	0.068
#seq012_Pomaria	PSNR	23.273	24.074	31.280	30.462	32.580	30.655	30.626	31.280	32.869
#seq012_Pomaria	LPIPS	0.306	0.259	0.086	0.140	0.276	0.139	0.142	0.086	0.078
#seq012_Pomaria	SSIM	0.823	0.852	0.970	0.929	0.914	0.919	0.918	0.970	0.954
#seq015_Pomaria	LPIPS	0.124	0.128	0.327	0.090	0.248	0.088	0.090	0.074	0.071
#seq015_Pomaria	SSIM	0.921	0.922	0.897	0.956	0.932	0.954	0.953	0.975	0.962
#seq012_Pomaria	PSNR	24.326	24.386	29.596	32.836	31.225	31.500	31.471	33.205	32.030
#seq014_Wainscott	LPIPS	0.145	0.342	0.293	0.093	0.248	0.090	0.091	0.114	0.081
#seq014_Wainscott	SSIM	0.918	0.909	0.911	0.951	0.936	0.953	0.952	0.976	0.958
#seq014_Wainscott	PSNR	28.236	24.296	32.551	34.690	34.297	33.049	33.002	34.963	34.187
#seq015_Wainscott	PSNR	23.201	22.902	31.259	30.979	33.451	30.136	30.089	27.367	30.360
#seq015_Wainscott	LPIPS	0.220	0.219	0.289	0.140	0.229	0.120	0.121	0.219	0.107
#seq015_Wainscott	SSIM	0.885	0.886	0.913	0.938	0.935	0.942	0.942	0.893	0.946
#seq016_Wainscott	LPIPS	0.175	0.194	0.327	0.089	0.260	0.086	0.087	0.066	0.083
#seq016_Wainscott	PSNR	21.953	21.660	28.364	30.414	30.205	28.915	28.710	31.965	29.746
#seq016_Wainscott	SSIM	0.897	0.895	0.909	0.951	0.935	0.952	0.951	0.976	0.955
#seq017_Benevolence	SSIM	0.930	0.937	0.906	0.944	0.927	0.961	0.959	0.978	0.962
#seq017_Benevolence	LPIPS	0.127	0.112	0.291	0.105	0.245	0.076	0.080	0.058	0.072
#seq017_Benevolence	PSNR	27.183	27.295	30.699	31.944	32.549	33.195	33.098	33.664	33.623
#seq018_Benevolence	LPIPS	0.105	0.116	NaN	0.077	0.244	0.082	0.083	0.067	0.083
#seq018_Benevolence	SSIM	0.927	0.924	NaN	0.955	0.925	0.951	0.951	0.969	0.952
#seq018_Benevolence	PSNR	26.011	25.966	NaN	32.517	29.580	30.511	30.504	31.885	31.197
#seq019_Rs	LPIPS	0.120	0.121	0.268	0.097	0.244	0.087	0.090	0.077	0.094
#seq019_Rs	SSIM	0.940	0.944	0.946	0.964	0.953	0.966	0.966	0.980	0.966
#seq019_Rs	PSNR	26.720	28.255	33.551	35.217	34.700	34.758	34.753	36.816	35.323
#seq020_Merom	PSNR	26.856	27.074	NaN	35.185	33.065	32.248	32.209	37.284	33.367
#seq020_Merom	SSIM	0.936	0.943	NaN	0.972	0.954	0.966	0.966	0.985	0.969
#seq020_Merom	LPIPS	0.138	0.126	NaN	0.059	0.199	0.075	0.075	0.047	0.061

## E.1 More Ablation Studies

**Ablation Study on Multi-scale Factorization.** We conduct more ablation studies on the OmniSim dataset to evaluate the effectiveness of the multi-scale factorization. The results show that the multi-scale factorization can improve the model’s performance by capturing the object’s detailed structure and texture. However, the model without multi-scale factorization performs poorly in depth rendering, illustrating the improvements of multi-scale factorization in scene geometric modeling. The results are shown in Figure. 15.

**Ablation Study on Interaction-aware Language Embedding.** We conduct more ablation studies on the OmniSim dataset to evaluate the effectiveness of the interaction-aware language embedding. The results show that the interaction-aware language embedding can effectively improve the model’s performance when encouraging significant scene topological changes. While the model without interaction-aware language embedding fails to ground the correct object because of the lack of interaction-aware information. The results are shown in Figure. 16.

**Maximum Probability Embeds Retrieval.** We conduct more ablation studies on the OmniSim dataset to evaluate the effectiveness of the maximum probability embedding retrieval. The results show that the maximum probability embedding retrieval can improve the model’s performance with higher storage efficiency and training speed, and the grounding results will also be more concentrated in the object region. The fundamental reason is that this method decouples language from the 3D scene to the object level, rather than the entire 3D space. The results are shown in Figure. 17.

Table 8: **Detailed Quantitative results on InterReal Dataset.** Our method outperforms others in most settings, with a significant advantage of PSNR, SSIM, and LPIPS on the #challenging subset.

dataset	Metric	NeR [45]	Instant-NGP [47]	HyperNeR [50]F	K-Planes [54]	CoNeRF [25]	CoGS [25]	LiveScene
#seq01_transformer	PSNR	20.094	20.619	24.651	26.881	27.260	31.067	30.396
#seq01_transformer	SSIM	0.725	0.805	0.638	0.791	0.739	0.943	0.912
#seq01_transformer	LPIPS	0.182	0.167	0.495	0.185	0.355	0.060	0.060
#seq02_transformer	PSNR	21.187	21.474	25.188	26.309	26.040	27.510	28.663
#seq02_transformer	SSIM	0.735	0.743	0.714	0.754	0.720	0.897	0.848
#seq02_transformer	LPIPS	0.371	0.358	0.545	0.341	0.520	0.138	0.181
#seq03_door	LPIPS	0.302	0.319	0.489	0.321	0.425	0.165	0.188
#seq03_door	SSIM	0.721	0.747	0.712	0.759	0.748	0.893	0.844
#seq03_door	PSNR	21.150	21.813	25.334	26.545	26.854	28.435	28.208
#seq04_dog	LPIPS	0.123	0.260	0.538	0.237	0.470	0.104	0.089
#seq04_dog	SSIM	0.643	0.699	0.574	0.736	0.627	0.919	0.868
#seq04_dog	PSNR	23.109	24.483	24.833	27.285	26.159	31.442	29.400
#seq05_sit	LPIPS	0.178	0.236	0.543	0.207	0.463	0.232	0.084
#seq05_sit	PSNR	21.558	24.211	24.944	27.970	26.252	27.169	30.161
#seq05_sit	SSIM	0.480	0.727	0.573	0.773	0.633	0.767	0.886
#seq06_stand	LPIPS	0.196	0.178	0.435	0.107	0.324	0.074	0.049
#seq06_stand	SSIM	0.723	0.819	0.730	0.894	0.815	0.950	0.943
#seq06_stand	PSNR	20.044	20.206	25.691	30.350	28.567	32.455	32.519
#seq07_flower	LPIPS	0.250	0.250	0.316	0.101	0.231	0.071	0.044
#seq07_flower	SSIM	0.785	0.831	0.878	0.920	0.922	0.962	0.960
#seq07_flower	PSNR	20.001	20.652	27.144	29.278	29.850	31.998	32.709
#seq08_office	SSIM	0.736	0.778	0.635	0.763	0.732	0.938	0.899
#seq08_office	PSNR	20.093	20.028	24.433	26.232	26.917	30.513	29.706
#seq08_office	LPIPS	0.210	0.196	0.477	0.223	0.357	0.062	0.069



Figure 11: **View Synthesis Visualization on InterReal Dataset.** We compare our method with SOTA methods on RGB rendering across real scenes. LiveScene obtained more detailed and accurate representations of the objects. While other methods fail to capture the object's shape and cause significant artifacts.

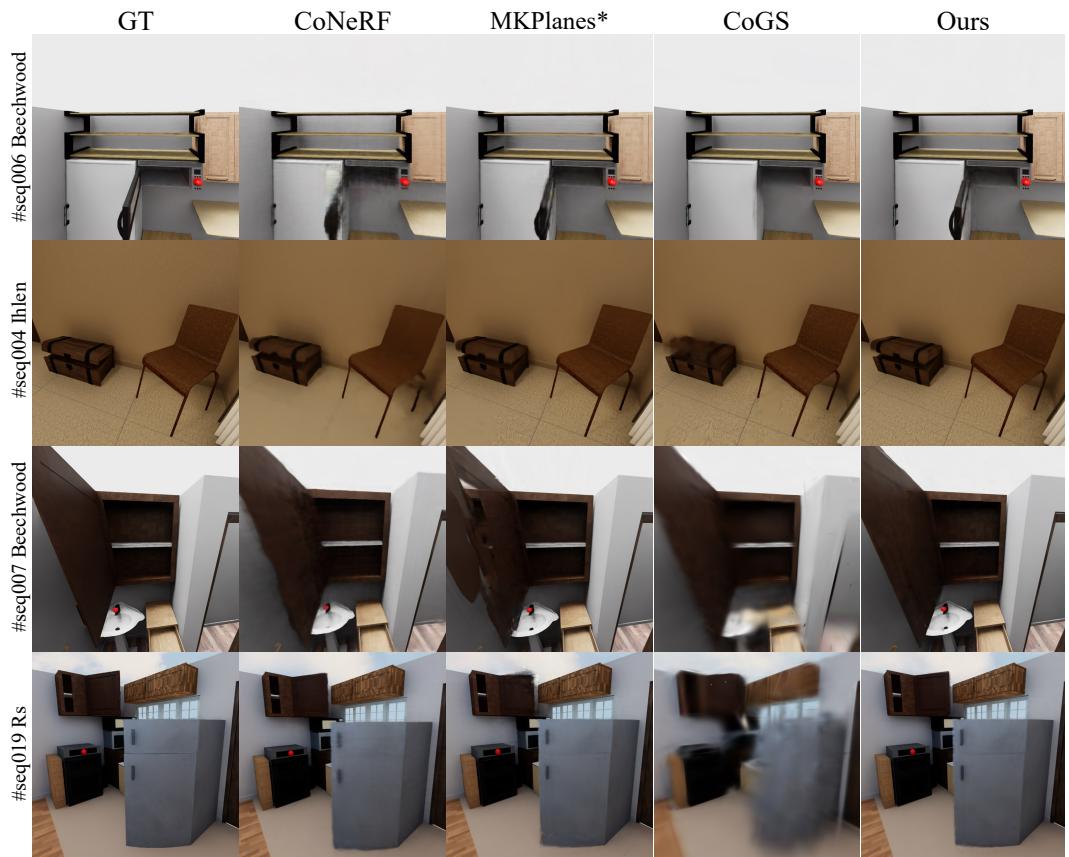


Figure 12: **View Synthesis Visualization on InterReal Dataset.** compared with the other methods, LiveScene reconstructs clear and accurate object shapes and textures.

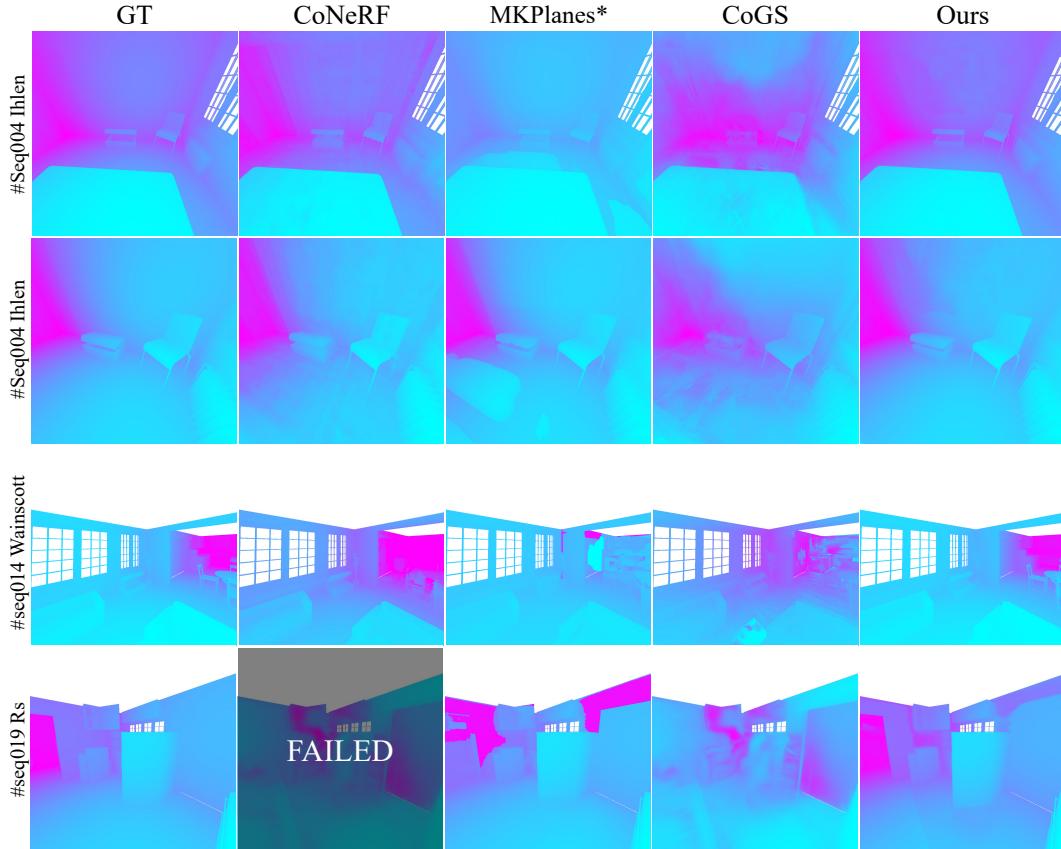


Figure 13: Illustration of the depth map comparison on the OmniSim datasets. Our method can generate more accurate depth maps than other methods, demonstrating the effectiveness of interactive scene reconstruction. In contrast, other methods either fail to capture the object’s shape or cause significant artifacts.

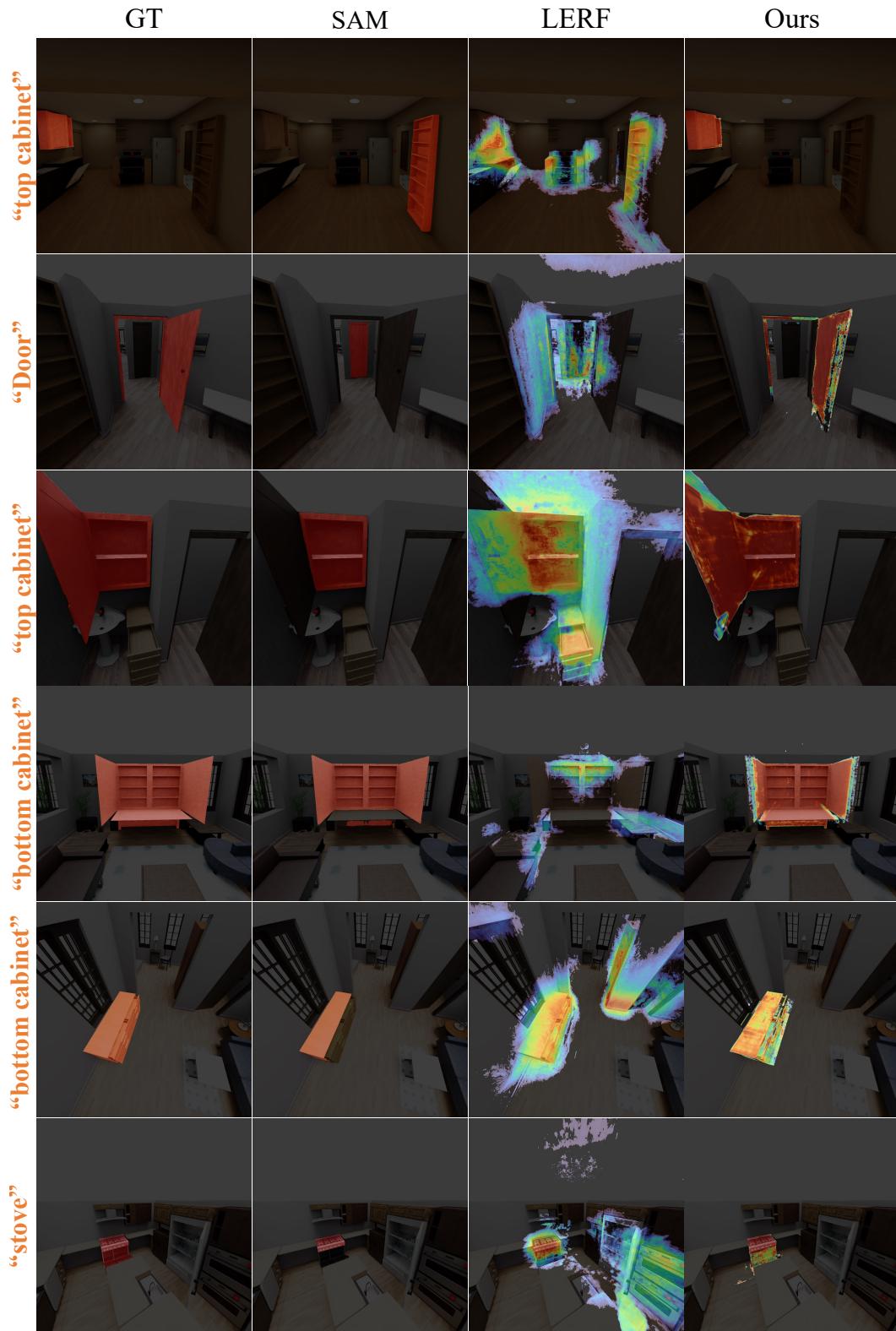


Figure 14: Illustration of the language grounding comparison on the OmniSim datasets. Compared to LeRF, our method can locate more accurate interactive objects, overcoming the obvious inconsistency problem in interactions, while maintaining accurate boundaries. In contrast, LeRF suffers from a diffusion phenomenon in object localization due to changes in object topology structure.

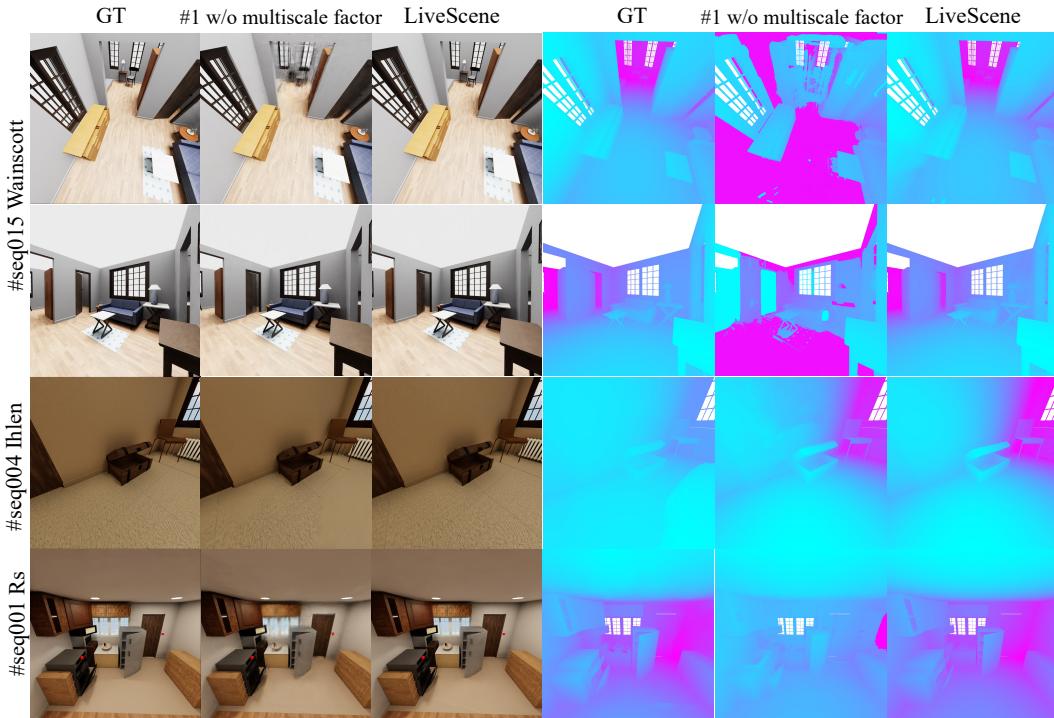


Figure 15: More ablation of multi-scale factorization on the OmniSim dataset. We compare the performance of LiveScene with w/o multiscale factor. The results show that the multi-scale factorization can improve the model’s performance by capturing the object’s detailed structure and texture. However, the model without multi-scale factorization performs poorly in depth rendering, illustrating the improvements of multi-scale factorization in scene geometric modeling.

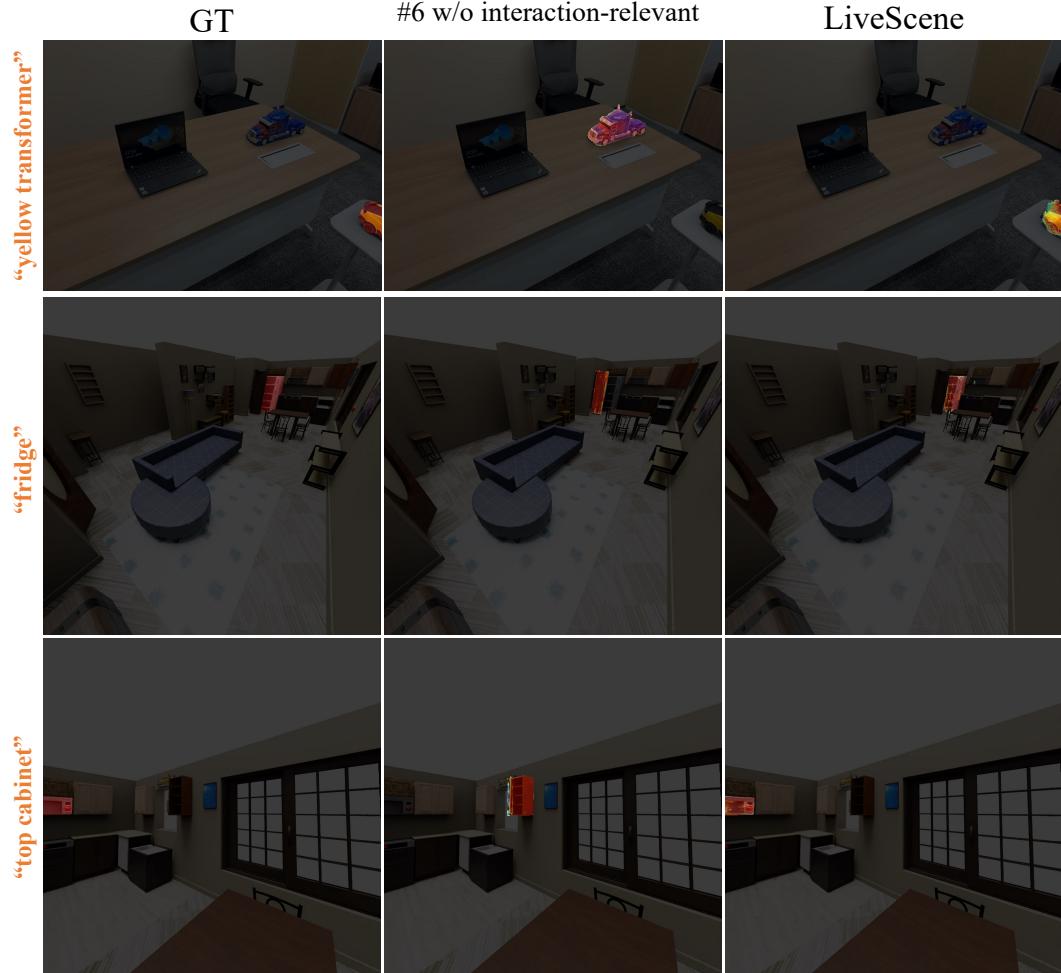


Figure 16: More ablation of interactive object modeling on the OmniSim dataset. We compare the performance of LiveScene with w/o interaction-aware language embedding. The results show that the interaction-aware language embedding can effectively improve the model’s performance when encouraging significant scene topological changes. While the model without interaction-aware language embedding fails to ground the correct object because of the lack of interaction-aware information.



Figure 17: By applying the proposed multiscale factor and maximum probability embedding retrieval, the model achieves better performance with higher storage efficiency and training speed, and the grounding results will also be more concentrated in the object region. The fundamental reason is that this method decouples language from the 3D scene to the object level, rather than the entire 3D space.

## References

- [1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.
- [2] ShahRukh Athar. Rignerf: Fully controllable neural 3d portraits. *ArXiv*, 2022.
- [3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspia Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.
- [5] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. *CVPR*, 2023.
- [6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv: Computer Vision and Pattern Recognition*, 2021.
- [7] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. 2023.
- [8] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning, 2023.
- [9] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023.
- [10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021.
- [11] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.
- [12] Roya Firooz, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *arXiv preprint arXiv:2312.07843*, 2023.
- [13] Carlos Flavián, Sergio Ibáñez-Sánchez, and Carlos Orús. The impact of virtual, augmented and mixed reality technologies on the customer experience. *Journal of Business Research*, 2019.
- [14] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.
- [15] Stephan J. Garbin, Marek Kowalski, V. Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, JingJing Shen, Matthew W. Johnson, and Julien P. C. Valentin. Voltmorph: Realtime, controllable and generalisable animation of volumetric representations. *arXiv.org*, 2022.
- [16] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan. Interactive segmentation of radiance fields. *ArXiv*, 2022.
- [17] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- [18] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv*, 2023.
- [19] Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world, 2024.

- [20] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. 2023.
- [21] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.
- [22] Krishna Murthy Jatavallabhula, Ali Kuwajerwala, Qiao Gu, Mohd. Osama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Varma Keetha, A. Tewari, J. Tenenbaum, Celso M. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba. Conceptfusion: Open-set multimodal 3d mapping. *arXiv.org*, 2023.
- [23] 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. *arXiv preprint arXiv:2401.09340*, 2024.
- [24] Chao Jia, Yafei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [25] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, Andrea Tagliasacchi, Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. 2022.
- [26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *arXiv (Cornell University)*, 2023.
- [28] J. Kerr, C. Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *ArXiv*, 2023.
- [29] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. *arXiv (Cornell University)*, 2024.
- [30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [31] Sosuke Kobayashi, Eiichi Matsumoto, and V. Sitzmann. Decomposing nerf for editing via feature field distillation. *ArXiv*, 2022.
- [32] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. *arXiV*, 2023.
- [33] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, Gerard Pons-Moll, Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. 2023.
- [34] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.
- [35] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision*, pages 419–436. Springer, 2022.
- [36] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.
- [37] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.
- [38] Jingbo Zhang<sup>3</sup> Zhihao Liang<sup>4</sup> Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024.
- [39] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, A. E. Saddik, C. Theobalt, Eric P. Xing, and Shijian Lu. 3d open-vocabulary segmentation with foundation models. *arXiv.org*, 2023.
- [40] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023.
- [41] 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, 2023.

- [42] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.
- [43] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024.
- [44] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *arXiv preprint arXiv:2402.10483*, 2024.
- [45] 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. In *ECCV*, 2020.
- [46] Richard M. Murray, Shankar Sastry, and Zexiang Li. A mathematical introduction to robotic manipulation. 1994.
- [47] T. Müller, Alex Evans, Christoph Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022.
- [48] M. Oquab, Timothée Darzet, T. Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, H. Jégou, J. Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv.org*, 2023.
- [49] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.
- [50] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.
- [51] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.
- [52] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023.
- [53] Alec Radford, JongWook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Askell Amanda, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Cornell University - arXiv; Cornell University - arXiv*, Feb 2021.
- [54] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023.
- [55] Nur Muhammad (Mahi) Shafiuallah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur D. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *ArXiv*, 2022.
- [56] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.
- [57] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. *arXiv preprint arXiv:2311.18482*, 2023.
- [58] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023.
- [59] Jonathan Steuer. Defining virtual reality: dimensions determining telepresence. 1992.
- [60] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, J. Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *ArXiv*, 2023.
- [61] 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.
- [62] Yiwen Tang, Ray Zhang, Zoey Guo, Xianzheng Ma, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5171–5179, 2024.

- [63] Sima Team, Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune, Adrian Collister, Vikki Copeman, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, Yani Donchev, Emma Dunleavy, Martin Engelcke, Ryan Faulkner, Frankie Garcia, Charles Takashi Toyin Gbadamosi, Zhitao Gong, Lucy Gonzales, Karol Gregor, Arne Olav Hallingstad, Tim Harley, Sam Haves, Felix Hill, Ed Hirst, Drew A. Hudson, Steph Hughes-Fitt, Danilo J. Rezende, Mimi Jasarevic, Laura Kampis, Rosemary Ke, Thomas Keck, Junkyung Kim, Oscar Knagg, Kavya Kopparapu, Andrew Kyle Lampinen, Shane Legg, Alexander Lerchner, Marjorie Limont, Yulan Liu, Maria Loks-Thompson, Joseph Marino, Kathryn Martin Cussons, Loïc Matthey, Siobhan Mcoughlin, Piermaria Mendolicchio, Hamza Merzic, Anna Mitenkova, Alexandre Moufarek, and Valeria Oliveira. Scaling instructable agents across many simulated worlds. 2024.
- [64] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.
- [65] Vadim Tschernezki, Iro Laina, Diane Larlus, and A. Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *ArXiv*, 2022.
- [66] Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, Nasif Zaman, Prithul Sarker, Andrew G. Lee, and A. Tavakkoli. The future of ophthalmology and vision science with the apple vision pro. *Eye*, 38:242–243, 2023.
- [67] Feng Wang, Zilong Chen, Guokang Wang, Yafei Song, and Huaping Liu. Masked space-time hash encoding for efficient dynamic scene reconstruction, 2023.
- [68] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023.
- [69] Zian Wang, Tianshang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Muller, and Zan Gojcic. Adaptive shells for efficient neural radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1–15, 2023.
- [70] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023.
- [71] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021.
- [72] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.
- [73] Heng Yu, Joel Julin, Zoltán Á Milacska, Koichiro Niinuma, and László A Jeni. Cogs: Controllable gaussian splatting. *arXiv preprint arXiv:2312.05664*, 2023.
- [74] Heng Yu, Koichiro Niinuma, Laszlo A. Jeni, Heng Yu, Koichiro Niinuma, and Laszlo A. Jeni. Confies: Controllable neural face avatars. 2023.
- [75] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13144–13152, 2021.
- [76] Raza Yunus, Jan Eric Lenssen, Michael Niemeyer, Yiyi Liao, Christian Rupprecht, Christian Theobalt, Gerard Pons-Moll, Jia-Bin Huang, Vladislav Golyanik, and Eddy Ilg. Recent trends in 3d reconstruction of general non-rigid scenes. *arXiv preprint arXiv:2403.15064*, 2024.
- [77] Hao Zhang, Fang Li, and Narendra Ahuja. Open-nerf: Towards open vocabulary nerf decomposition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3456–3465, 2024.
- [78] Chengwei Zheng, Wenbin Lin, and Feng Xu. Editablenerf: Editing topologically varying neural radiance fields by key points, 2023.
- [79] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. 2021.
- [80] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guan Wang, Kaichao Zhang, Cheng Ji, Qi Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, P. Xie, Caiming Xiong, Jian Pei, Philip S. Yu, Lichao Sun Michigan State University, B. University, Lehigh University, M. University, Nanyang Technological University, University of California at San Diego, D. University, U. Chicago, and Salesforce AI Research. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *ArXiv*, 2023.