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

Supplementary Material

Abstract

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

▷ **Section. 7**: Configurations of OmniSim and InterReal dataset, including scene assets, interaction variables generation, mask and prompts annotation, and dataset visualization.

▷ **Section. 8**: Video demonstration and anonymous link: https://livescenes. github.io.

▷ **Section. 9**: Additional implementation details.

▷ **Section. 10**: Additional experimental results, including more ablation studies, detailed view synthesis quality comparison, interactive scenes geometry comparison and language grounding comparison.

7 Configurations of OmniSim and InterReal datasets
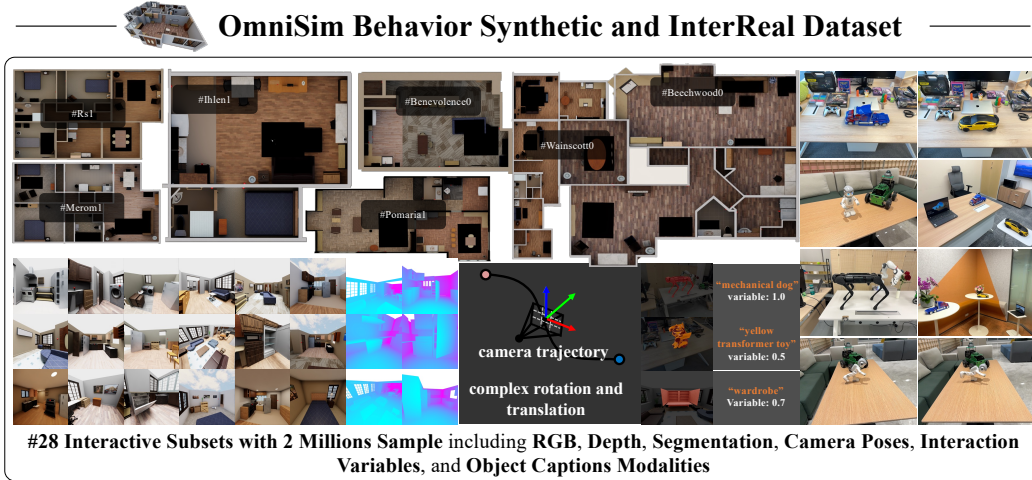
Figure 11: Illustration of the proposed Omniverse behavior synthetic (OminiSim) and Real captured interactive (InterReal) dataset. These datasets are captured in an OmniGibson simulator or real scene and carefully annotated, providing #28 interactive subsets with 2 Million samples, including RGB, depth, 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 × 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.

11

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 [51], and 4) mask generation for each object in each scene using SAM [25] 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	3	1610	1610	✓	✓	✓	✓	bottom cabinet, dishwasher, 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
	#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
InterReal	#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

8 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.

9 Additional implementation details

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

$$\mathcal{L}_{total} = \mathcal{L}_{\text{MSE}} + \lambda_1 \mathcal{L}_{\text{focus}} + \lambda_2 \mathcal{L}_{\text{repuls}} + \lambda_3 \mathcal{L}_{\text{var}} + \lambda_4 \mathcal{L}_{\text{lang}} + \lambda_5 \mathcal{L}_{\text{smooth}}, \qquad (6)$$

12

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:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{C}_i - \tilde{\mathbf{C}}_i \right\|^2, \tag{7}$$

where $\mathbf{C}_i$ and $\tilde{\mathbf{C}}_i$ are the rendered and ground truth RGB values, respectively, and N 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:

$$\mathcal{L}_{\text{focus}} = \beta \cdot \left(1 - e^{\sum_{i=1}^{\alpha} \mathbf{M}_i \log(\hat{\mathbf{P}}_i)} \right)^{\gamma} \cdot \left(-\sum_{i=1}^{\alpha} \mathbf{M}_i \log(\hat{\mathbf{P}}_i) \right), \tag{8}$$

where $\mathbf{M}$ is the ground truth mask label, $\hat{\mathbf{P}}$ is the probability map rendering from the interactive probability field, β is the balancing factor, and γ 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:

$$\mathcal{L}_{\text{repuls}} = \mathbf{ELU}(K - \|(\mathbf{M}_i \odot \mathbf{M}_j)(\mathcal{F}_i - \mathcal{F}_j)\|), \tag{9}$$

where $\mathbf{M}_i$ and $\mathbf{M}_j$ are the ground truth mask of rays, and $\mathcal{F}_i$ and $\mathcal{F}_j$ are the last-layer features of interaction probability decoder in Figure. 2. K is the constant hyperparameters. In training iteration, we randomly select ray pairs and apply $\mathcal{L}_{\text{repuls}}$ to enforce the separation of interactive probability features across local deformable spaces. Our approach draws inspiration from [24], which has shown the effectiveness of repulsive forces in resolving ambiguities in 3D segmentation.

Interaction Variable MSE. We follow the value MSE in [20] and use the standard MSE loss to supervise the interaction values training:

$$\mathcal{L}_{\text{var}} = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{\kappa}_i - \tilde{\boldsymbol{\kappa}}_i\|^2, \tag{10}$$

where $\boldsymbol{\kappa}_i$ and $\tilde{\boldsymbol{\kappa}}$ 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:

$$\mathcal{L}_{\text{lang}}(\phi(\mathbf{p}), \tilde{\phi}(\mathbf{p})) = \begin{cases} \frac{1}{2}(\phi(\mathbf{p}) - \phi(\mathbf{p}))^2 & \text{if } |\phi(\mathbf{p}) - \tilde{\phi}(\mathbf{p})| \leq \delta \\ \delta \left(|\phi(\mathbf{p}) - \tilde{\phi}(\mathbf{p})| - \frac{1}{2}\delta \right) & \text{otherwise} \end{cases}, \tag{11}$$

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

Smoothness Loss. Inspired by K-Planes [10], 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:

$$\mathcal{L}_{\text{smooth}}(\mathbf{p}) = \frac{1}{|L|n^2} \sum_{l,i,k} \left\| \mathbf{p}_l^{i,k-1} - 2\mathbf{p}_l^{i,k} + \mathbf{p}_l^{i,k+1} \right\|_2^2, \tag{12}$$

13

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

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

$$u = \begin{cases} \arg\max_i\{\mathbf{P}_i\}, & \text{if} \quad \mathbf{P}_i \geq s \\ -1 & \text{otherwise} \end{cases} . \tag{13}$$

Implementation Details. LiveScene is implemented in Nerfstudio [51] 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.

10 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 [63] has a higher base number of parameters and a linear increase with the number of interactive objects. MK-Planes [10] exhibits a quadratic increase in parameters with the number of interactive objects. Although NeRF[38] and InstantNGP [39] 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 [38]	13.23	13.23	13.23	13.23	13.23	13.23	Constant
InstantNGP [39]	11.68	11.68	11.68	11.68	11.68	11.68	Constant
K-Planes [10]	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 [63]	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.

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. 12, 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 [20] performs relatively well in #seq08 and #seq15 but fails in large-scale scenes (#seq17 and #seq19). CoGS [63] exhibit notable artifacts in the depth map. Moreover, MK-Planes⋆ also fails to recover

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 [38]	Instant-NGP [39]	HyperNeRF [42]	K-Planes [10]	CoNeRF [20]	MK-Planes [10]	MK-Planes* [10]	CoGS [63]	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	PSNR	28.616	28.660	NaN	34.765	34.286	36.532	34.580	34.497	35.190
#seq002_Rs	SSIM	0.950	0.946	NaN	0.967	0.951	0.976	0.968	0.979	0.969
#seq002_Rs	LPIPS	0.096	0.112	NaN	0.074	0.217	0.036	0.074	0.051	0.070
#seq003_Ihlen	PSNR	26.720	28.255	33.551	35.217	34.700	34.758	34.753	36.816	35.323
#seq003_Ihlen	SSIM	0.940	0.944	0.946	0.964	0.953	0.966	0.966	0.980	0.966
#seq003_Ihlen	LPIPS	0.120	0.121	0.268	0.097	0.244	0.087	0.090	0.077	0.094
#seq004_Ihlen	PSNR	30.847	31.800	31.115	36.157	32.684	34.863	35.000	31.055	36.712
#seq004_Ihlen	SSIM	0.927	0.942	0.878	0.955	0.888	0.919	0.926	0.915	0.962
#seq004_Ihlen	LPIPS	0.104	0.102	0.389	0.085	0.366	0.145	0.135	0.209	0.072
#seq005_Beechwood	PSNR	27.183	27.295	30.699	31.944	32.549	33.195	33.098	33.664	33.623
#seq005_Beechwood	SSIM	0.930	0.937	0.906	0.944	0.927	0.961	0.959	0.978	0.962
#seq005_Beechwood	LPIPS	0.127	0.112	0.291	0.105	0.245	0.076	0.080	0.058	0.072
#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
#seq006_Beechwood	LPIPS	0.103	0.119	0.314	0.097	0.283	0.095	0.096	0.059	0.077
#seq007_Beechwood	PSNR	23.201	22.902	31.259	30.979	33.451	30.136	30.089	27.367	30.360
#seq007_Beechwood	SSIM	0.885	0.886	0.913	0.938	0.935	0.942	0.942	0.893	0.946
#seq007_Beechwood	LPIPS	0.220	0.219	0.289	0.140	0.229	0.120	0.121	0.219	0.107
#seq008_Benevolence	PSNR	25.750	25.574	32.691	31.914	34.319	30.926	30.916	33.795	33.393
#seq008_Benevolence	SSIM	0.943	0.940	0.945	0.948	0.960	0.941	0.941	0.980	0.970
#seq008_Benevolence	LPIPS	0.113	0.123	0.229	0.107	0.185	0.118	0.116	0.072	0.067
#seq009_Benevolence	PSNR	24.326	24.386	29.596	32.836	31.225	31.500	31.471	33.205	32.030
#seq009_Benevolence	SSIM	0.921	0.922	0.897	0.956	0.932	0.954	0.953	0.975	0.962
#seq009_Benevolence	LPIPS	0.124	0.128	0.327	0.090	0.248	0.088	0.090	0.074	0.071
#seq010_Merom	PSNR	22.927	22.765	28.985	30.120	31.092	29.461	29.396	30.254	30.029
#seq010_Merom	SSIM	0.917	0.925	0.939	0.960	0.957	0.960	0.959	0.974	0.966
#seq010_Merom	LPIPS	0.173	0.158	0.275	0.093	0.233	0.087	0.088	0.065	0.074
#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	26.856	27.074	NaN	35.185	33.065	32.248	32.209	37.284	33.367
#seq012_Pomaria	SSIM	0.936	0.943	NaN	0.972	0.954	0.966	0.966	0.985	0.969
#seq012_Pomaria	LPIPS	0.138	0.126	NaN	0.059	0.199	0.075	0.075	0.047	0.061
#seq013_Pomaria	PSNR	25.277	24.018	NaN	30.860	33.682	30.390	30.299	32.868	33.592
#seq013_Pomaria	SSIM	0.925	0.930	NaN	0.943	0.964	0.931	0.930	0.981	0.970
#seq013_Pomaria	LPIPS	0.154	0.161	NaN	0.123	0.166	0.162	0.164	0.045	0.056
#seq014_Wainscott	PSNR	26.011	25.966	NaN	32.517	29.580	30.511	30.504	31.885	31.197
#seq014_Wainscott	SSIM	0.927	0.924	NaN	0.955	0.925	0.951	0.951	0.969	0.952
#seq014_Wainscott	LPIPS	0.105	0.116	NaN	0.077	0.244	0.082	0.083	0.067	0.083
#seq015_Wainscott	PSNR	27.257	27.191	NaN	30.721	32.307	28.288	28.134	32.949	34.266
#seq015_Wainscott	SSIM	0.953	0.951	NaN	0.955	0.962	0.942	0.942	0.975	0.976
#seq015_Wainscott	LPIPS	0.080	0.092	NaN	0.083	0.202	0.110	0.108	0.078	0.050
#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
#seq016_Wainscott	LPIPS	0.175	0.194	0.327	0.089	0.260	0.086	0.087	0.066	0.083
#seq017_Benevolence	PSNR	26.364	26.367	27.533	29.833	30.349	29.254	26.565	28.701	31.645
#seq017_Benevolence	SSIM	0.927	0.920	0.897	0.937	0.923	0.933	0.887	0.970	0.948
#seq017_Benevolence	LPIPS	0.128	0.143	0.318	0.118	0.238	0.119	0.218	0.073	0.093
#seq018_Benevolence	PSNR	28.236	24.296	32.551	34.690	34.297	33.049	33.002	34.963	34.187
#seq018_Benevolence	SSIM	0.918	0.809	0.911	0.951	0.936	0.953	0.952	0.976	0.958
#seq018_Benevolence	LPIPS	0.145	0.342	0.293	0.093	0.248	0.090	0.091	0.114	0.081
#seq019_Rs	PSNR	20.059	20.854	33.119	34.462	34.598	33.679	33.653	25.947	35.223
#seq019_Rs	SSIM	0.794	0.808	0.950	0.956	0.963	0.963	0.962	0.879	0.969
#seq019_Rs	LPIPS	0.425	0.424	0.270	0.106	0.225	0.087	0.089	0.327	0.068
#seq020_Merom	PSNR	23.273	24.074	31.280	30.462	32.580	30.655	30.626	31.280	32.869
#seq020_Merom	SSIM	0.823	0.852	0.970	0.929	0.914	0.919	0.918	0.970	0.954
#seq020_Merom	LPIPS	0.306	0.259	0.086	0.140	0.276	0.139	0.142	0.086	0.078

	Depth L1 Error ↓			
scene	CoNeRF	MK-Planes*	CoGS	LiveScene
#seq04	0.029	0.037	0.381	**0.018**
#seq08	**0.018**	0.301	0.655	0.025
#seq14	0.042	0.103	0.810	**0.039**
#seq15	**0.019**	0.568	0.690	0.021
#seq17	✗	0.282	0.706	**0.019**
#seq19	✗	0.136	0.689	**0.034**
#avg	0.027	0.238	0.655	**0.026**
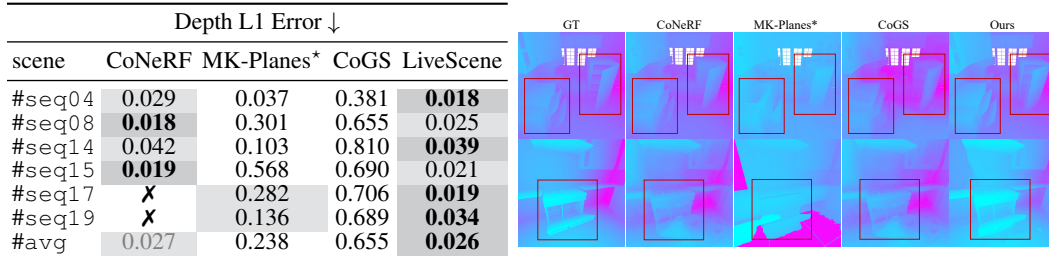
Figure 12: **Structure Reconstruction Performance on OmniSim Dataset.** Our method surpasses most previous works on chosen subsets.

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.

Language Grounding Comparison . We assess the language grounding performance on OmniSim dataset using mIOU metric. Figure. 13 suggests that our method obtains the highest mIOU score, with an average of 86.86. In contrast, traditional methods like LERF [23] encounter difficulties in locating objects precisely, with an average mIOU of 21.74. Meanwhile, 2D methods like SAM [25] 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.

More Detailed Rendering Comparison We provide more detailed visual comparisons, including RGB, depth, and language grounding on the OmniSim and InterReal datasets in Figure. 14, Figure. 15,

15

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	NeRF [38]	Instant-NGP [39]	HyperNeR [42]F	K-Planes [10]	CoNeRF [20]	CoGS [20]	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	20.093	20.028	24.433	26.232	26.917	30.513	29.706
#seq02_transformer	SSIM	0.736	0.778	0.635	0.763	0.732	0.938	0.899
#seq02_transformer	LPIPS	0.210	0.196	0.477	0.223	0.357	0.062	0.069
#seq03_door	PSNR	20.001	20.652	27.144	29.278	29.850	31.998	32.709
#seq03_door	SSIM	0.785	0.831	0.878	0.920	0.922	0.962	0.960
#seq03_door	LPIPS	0.250	0.250	0.316	0.101	0.231	0.071	0.044
#seq04_dog	PSNR	20.044	20.206	25.691	30.350	28.567	32.455	32.519
#seq04_dog	SSIM	0.723	0.819	0.730	0.894	0.815	0.950	0.943
#seq04_dog	LPIPS	0.196	0.178	0.435	0.107	0.324	0.074	0.049
#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
#seq05_sit	LPIPS	0.178	0.236	0.543	0.207	0.463	0.232	0.084
#seq06_stand	PSNR	23.109	24.483	24.833	27.285	26.159	31.442	29.400
#seq06_stand	SSIM	0.643	0.699	0.574	0.736	0.627	0.919	0.868
#seq06_stand	LPIPS	0.123	0.260	0.538	0.237	0.470	0.104	0.089
#seq07_flower	PSNR	21.150	21.813	25.334	26.545	26.854	28.435	28.208
#seq07_flower	SSIM	0.721	0.747	0.712	0.759	0.748	0.893	0.844
#seq07_flower	LPIPS	0.302	0.319	0.489	0.321	0.425	0.165	0.188
#seq08_office	PSNR	21.187	21.474	25.188	26.309	26.040	27.510	28.663
#seq08_office	SSIM	0.735	0.743	0.714	0.754	0.720	0.897	0.848
#seq08_office	LPIPS	0.371	0.358	0.545	0.341	0.520	0.138	0.181

		mIOU ↑		
	setting	SAM [25]	LERF [23]	Ours
OmniSim	#easy	61.58	23.60	**86.94**
OmniSim	#medium	55.13	19.40	**86.32**
OmniSim	#challenging	63.86	19.87	**90.41**
OmniSim	#avg	59.11	21.74	**86.86**
InterReal	#medium	93.27	27.63	84.37
InterReal	#challenging	91.50	34.39	**91.90**
InterReal	#avg	92.82	29.32	86.26
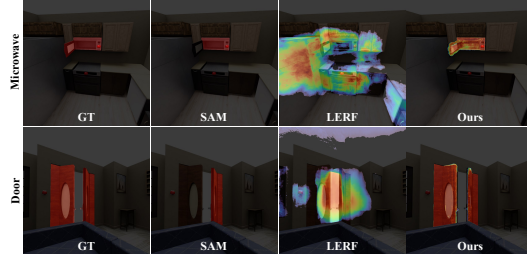
Figure 13: **Language Grounding Performance on OmniSim Dataset.** left): Our method gains the highest mIOU score. right): LiveScene's grounding exhibits clearer boundaries than other methods.

Figure. 16, and Figure. 17, 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 [23], our method can generate more accurate language grounding results, which is crucial for interactive object manipulation tasks, demonstrated in Figure. 17.

10.1 More Ablation Studies

4D Deformable Feature Visualization. We provide additional interaction feature visualization of x-z, y-z, and z-k in Figure. 18(a) to illustrate latent feature distribution. It can be seen that the features are clustered around the spatial coordinates of interactive objects, corresponding to the local deformable fields in Sec 3.2 of the manuscript. Figure. 18(b) validates the performance of LiveScene in scenarios with up to 10 complex interactive objects. Notably, our method demonstrates robustness in rendering quality, which does not degrade significantly as the object number increases. The number of objects is not a major limiting and our method is still feasible as long as the dataset provides mask and control variable labels. In contrast, the occlusion and topological complexity between objects do affect the reconstruction results, which will be discussed in the limitations section. In Figure. 18(c), we demonstrate the fine-grained control capability of LiveScene on a refrigerator and cabinet dataset without part-based labels. Our method can control a part of the object even though there are no individual part-based interaction variable labels. However, the effect is not entirely satisfactory, due to the lack of labels and CLIP's limited understanding of spatial relationships.

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

16
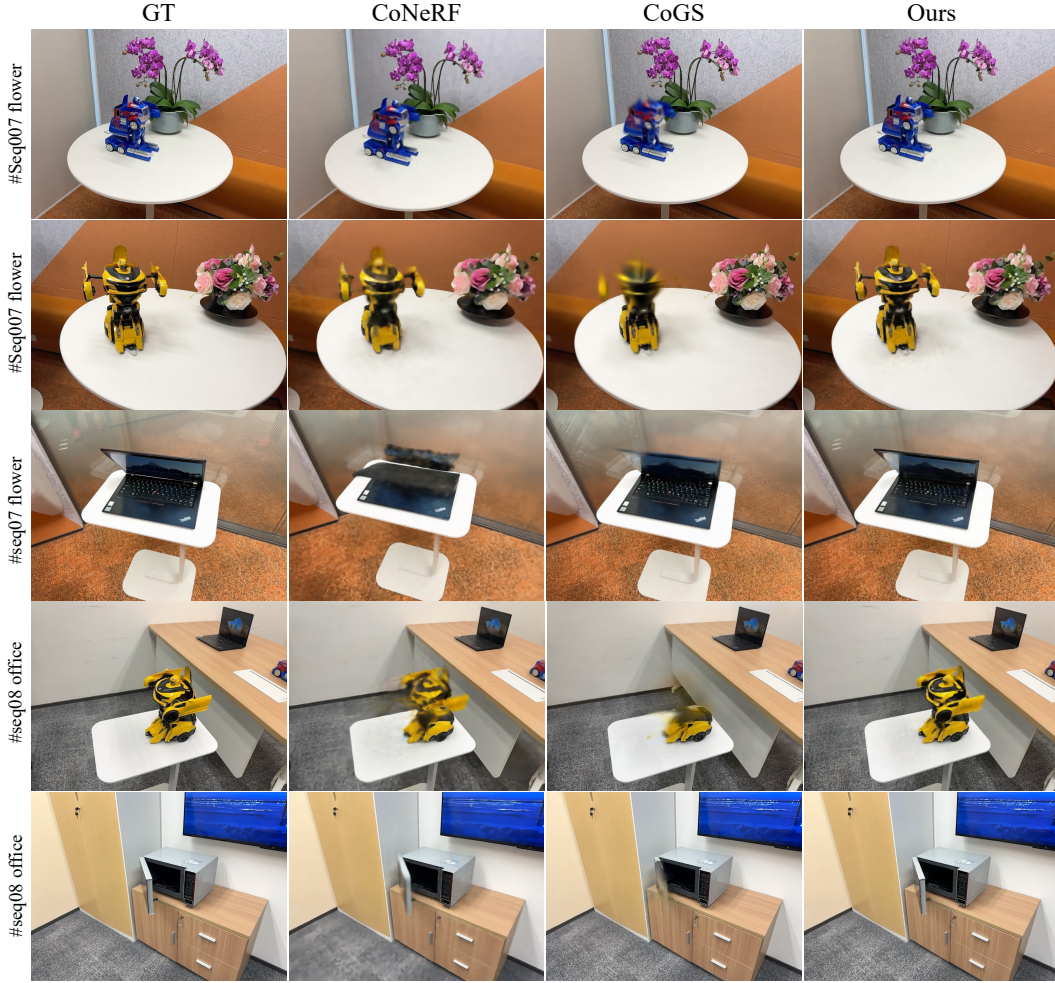
Figure 14: **View Synthesis Visualization on InterReal Dataset**. We compare our method with SOTA methods on RGB rendering across real scenes. LiveScene obtained more detailed and accurate representations of the objects. While other methods fail to capture the object's shape and cause significant artifacts.

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. 19.

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. 20.

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. 21.

17
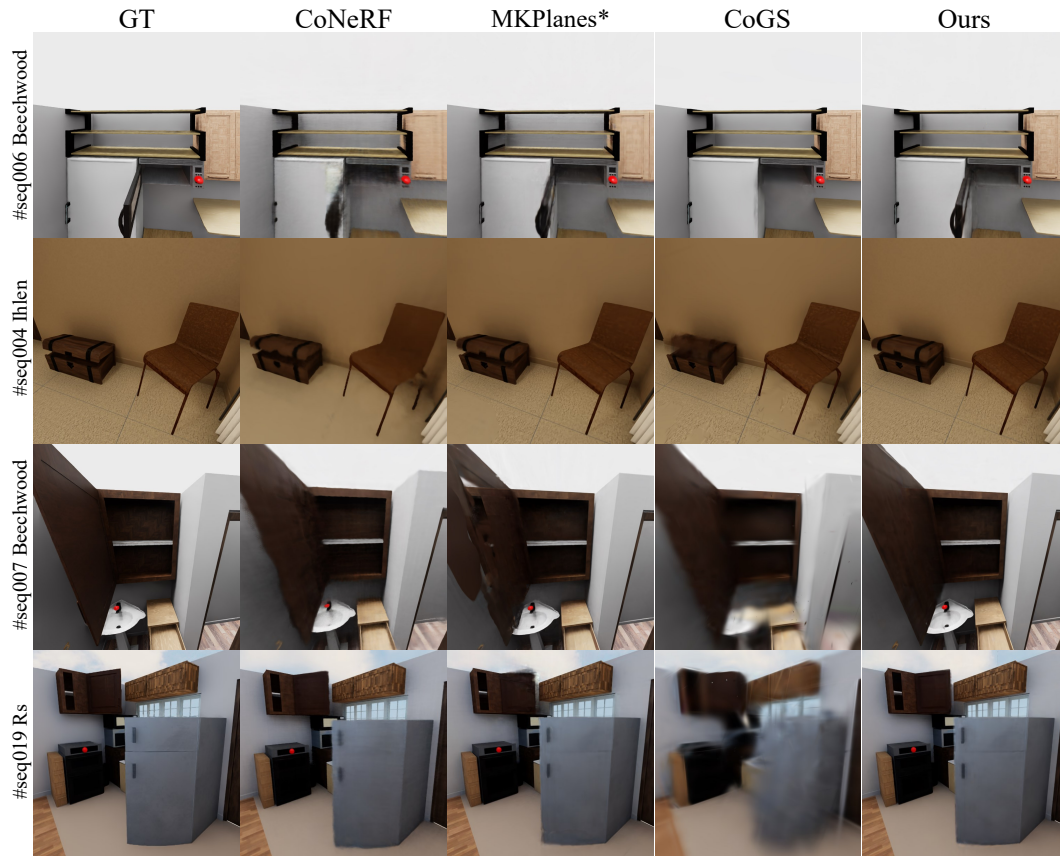
Figure 15: **View Synthesis Visualization on InterReal Dataset**. compared with the other methods, LiveScene reconstructs clear and accurate object shapes and textures.

18
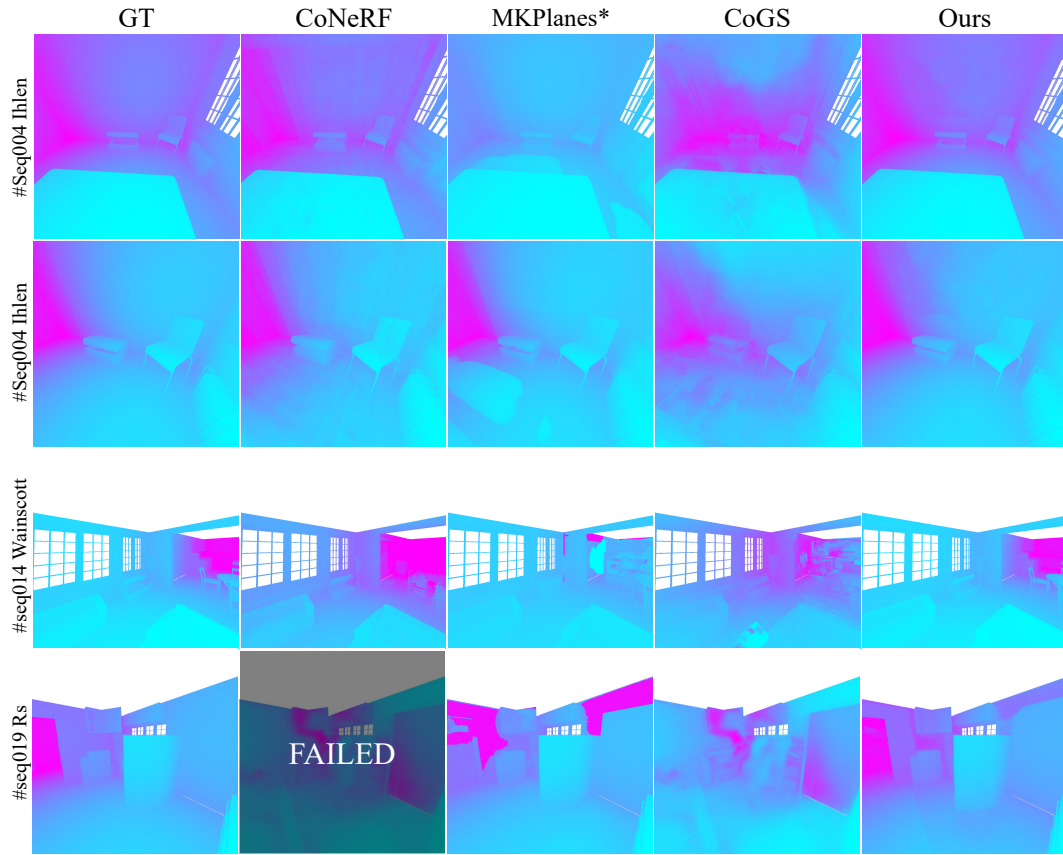
Figure 16: Illustration of the depth map comparison on the OmniSim datasets. Our method can generate more accurate depth maps than other methods, demonstrating the effectiveness of interactive scene reconstruction. In contrast, other methods either fail to capture the object's shape or cause significant artifacts.

19
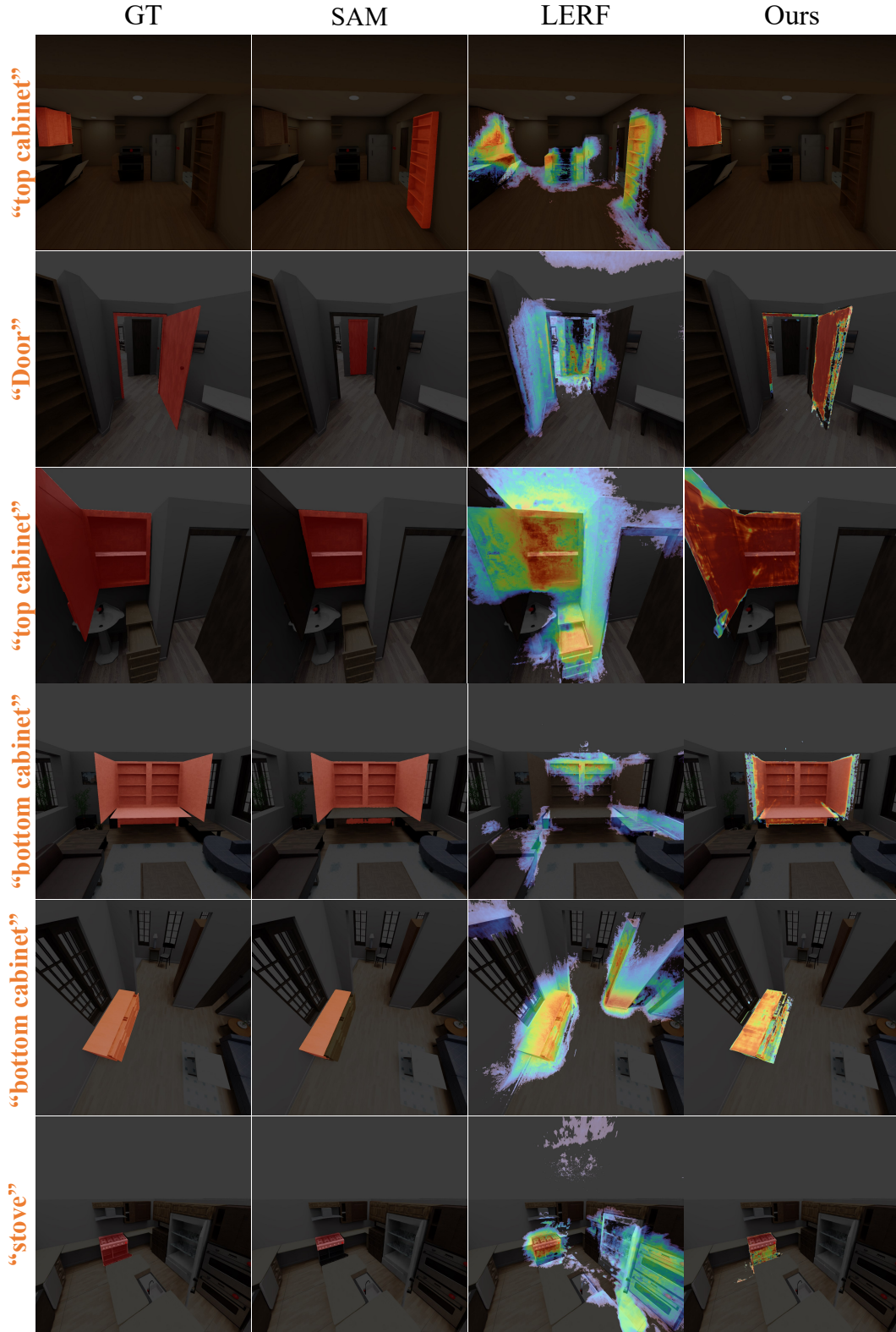
Figure 17: Illustration of the language grounding comparison on the OmniSim datasets. Compared to LeRF, our method can locate more accurate interactive objects, overcoming the obvious inconsistency problem in interactions, while maintaining accurate boundaries. In contrast, LeRF suffers from a diffusion phenomenon in object localization due to changes in object topology structure.

20

a) Feature visualization in compact 4d planes   **c)** Object part interaction
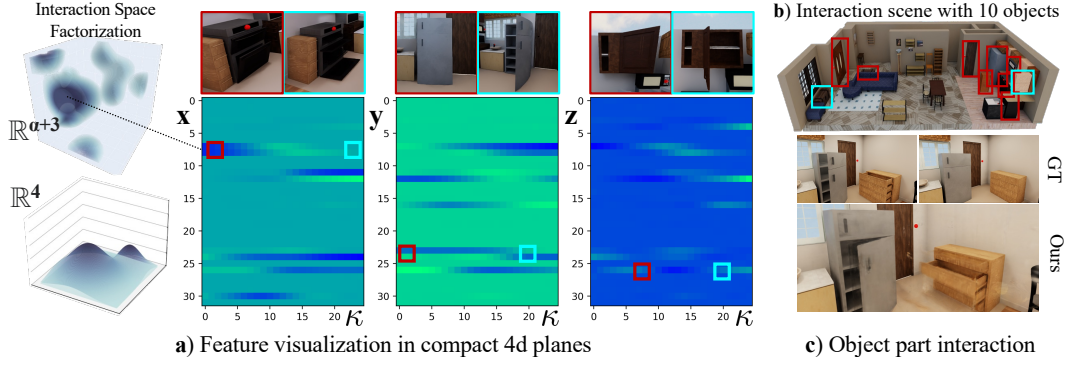
Figure 18: **a)** Visualization of x-κ, y-κ and z-κ latent feature planes. **b)** Sce ne with more objects. **c)** Part-level interaction.
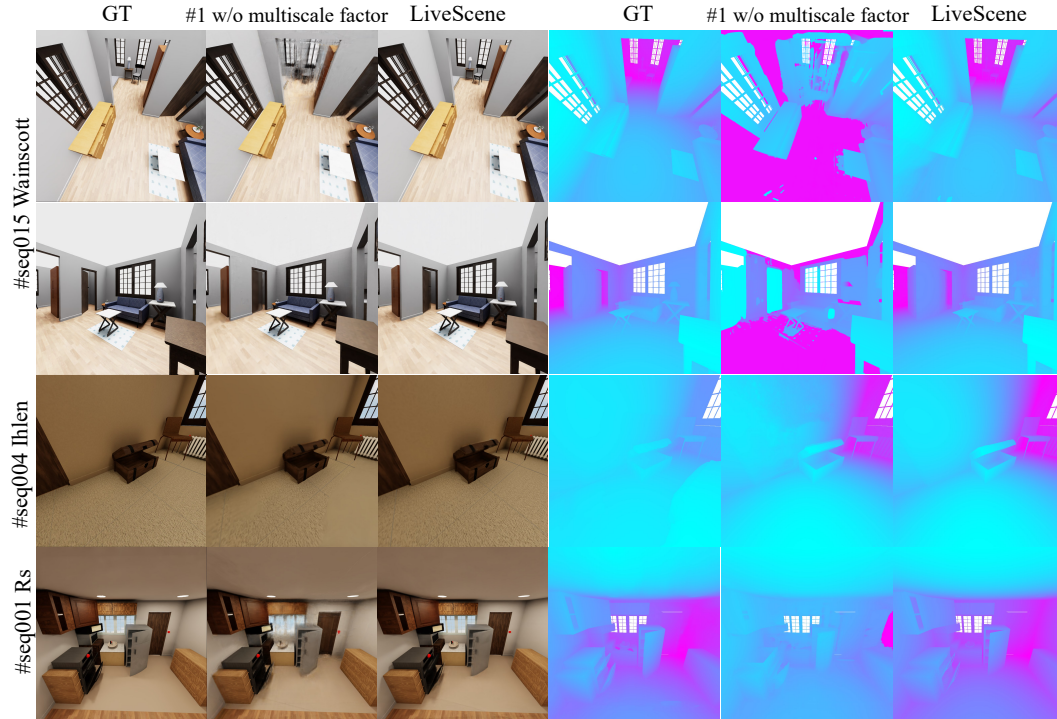
Figure 19: More ablation of multi-scale factorization on the OmniSim dataset. We compare the performance of LiveScene with w/o multiscale factor. The results show that the multi-scale factorization can improve the model's performance by capturing the object's detailed structure and texture. However, the model without multi-scale factorization performs poorly in depth rendering, illustrating the improvements of multi-scale factorization in scene geometric modeling.

21
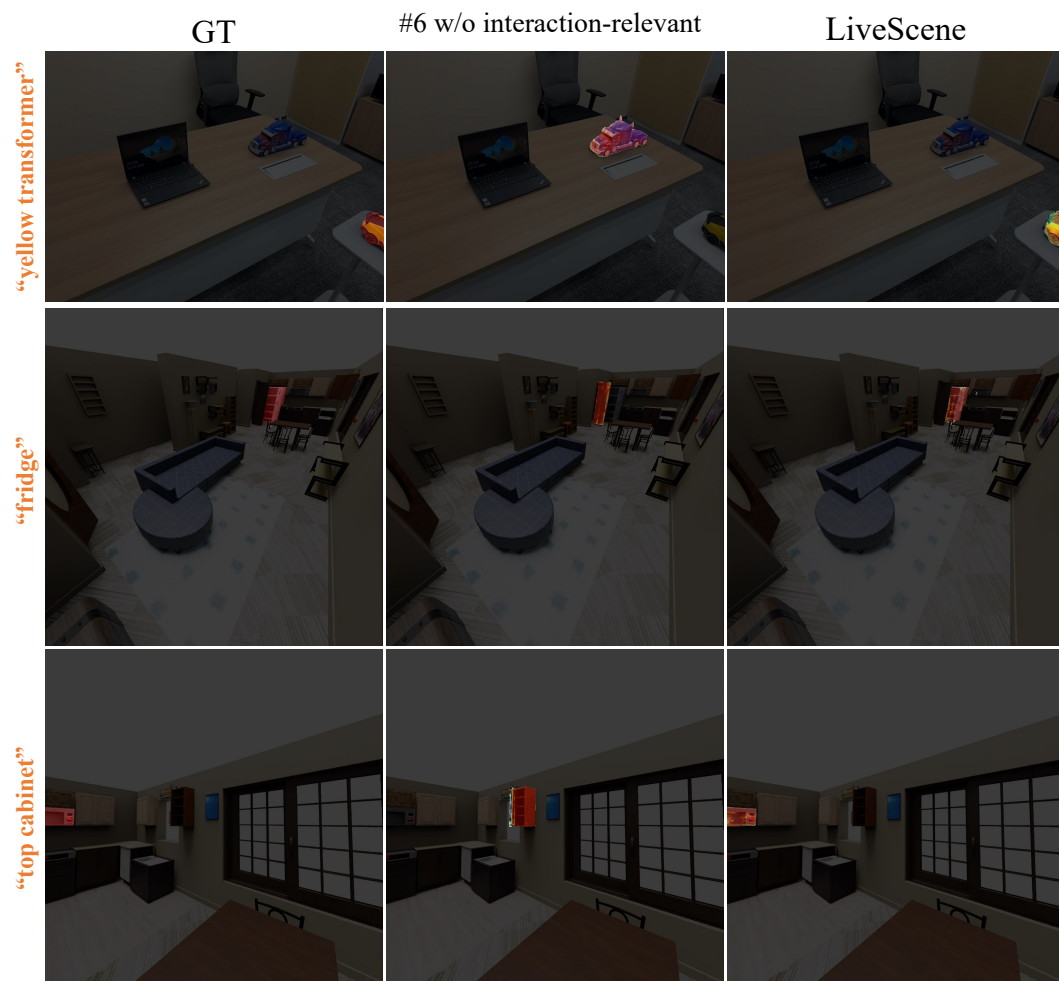
Figure 20: More ablation of interactive object modeling on the OmniSim dataset. We compare the performance of LiveScene with w/o interaction-aware language embedding. The results show that the interaction-aware language embedding can effectively improve the model's performance when encouraging significant 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.

22

Figure 21: 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.

23

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] 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.

[3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[5] 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. In *IEEE International Conference on Robotics and Automation*, pages 11509–11522. IEEE, 2023.

[6] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438, 2024.

[7] 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*, pages 1–9, 2022.

[8] Roya Firoozi, 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.

[9] 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.

[10] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.

[11] 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.

[12] Stephan J Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew A Johnson, and Julien Valentin. Voltemorph: Real-time, controllable and generalizable animation of volumetric representations. In *Computer Graphics Forum*, volume 43, page e15117. Wiley Online Library, 2024.

[13] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan. Interactive segmentation of radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4201–4211, 2023.

[14] Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15372–15383, 2023.

[15] 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.

[16] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26406–26416, 2024.

24

[17] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 10608–10615. IEEE, 2023.

[18] Krishna Murthy Jatavallabhula, Ali Kuwajerwala, Qiao Gu, Mohd. Omama, 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.

[19] 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.

[20] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022.

[21] 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.

[22] 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.

[23] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[24] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024.

[25] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[26] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022.

[27] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. *arXiV*, 2023.

[28] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4340–4350, 2023.

[29] 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.

[30] 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.

[31] 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.

[32] Bangyan Liao, Delin Qu, Yifei Xue, Huiqing Zhang, and Yizhen Lao. Revisiting rolling shutter bundle adjustment: Toward accurate and fast solution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4863–4871, June 2023.

[33] Jingbo Zhang3 Zhihao Liang4 Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024.

[34] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. 3d open-vocabulary segmentation with foundation models. *arXiv preprint arXiv:2305.14093*, 2(3):6, 2023.

25

[35] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[36] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[37] 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.

[38] 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.

[39] T. Müller, Alex Evans, Christoph Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022.

[40] M. Oquab, Timoth'ee Darcet, 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.

[41] 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.

[42] 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.

[43] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[44] Delin Qu, Chi Yan, Dong Wang, Jie Yin, Qizhi Chen, Dan Xu, Yiting Zhang, Bin Zhao, and Xuelong Li. Implicit event-rgbd neural slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19584–19594, June 2024.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[46] Nur Muhammad (Mahi) Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur D. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *ArXiv*, 2022.

[47] 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.

[48] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024.

[49] 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.

[50] Jonathan Steuer. Defining virtual reality: dimensions determining telepresence. 1992.

[51] 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.

[52] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *The Twelfth International Conference on Learning Representations*, 2024.

26

[53] Yiwen Tang, Ray Zhang, Jiaming Liu, Zoey Guo, Bin Zhao, Zhigang Wang, Peng Gao, Hongsheng Li, Dong Wang, and Xuelong Li. Any2point: Empowering any-modality large models for efficient 3d understanding. In *European Conference on Computer Vision*, pages 456–473. Springer, 2025.

[54] 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.

[55] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *International Conference on 3D Vision*, pages 443–453. IEEE, 2022.

[56] Feng Wang, Zilong Chen, Guokang Wang, Yafei Song, and Huaping Liu. Masked space-time hash encoding for efficient dynamic scene reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] 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.

[58] Zian Wang, Tianchang 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*, 42:1–15, 2023.

[59] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[60] 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.

[61] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, June 2024.

[62] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

[63] Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. Cogs: Controllable gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21624–21633, 2024.

[64] Heng Yu, Koichiro Niinuma, and László A Jeni. Confies: Controllable neural face avatars. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–8. IEEE, 2023.

[65] 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.

[66] 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. In *Computer Graphics Forum*, page e15062. Wiley Online Library, 2024.

[67] 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.

[68] Chengwei Zheng, Wenbin Lin, and Feng Xu. Editablenerf: Editing topologically varying neural radiance fields by key points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8327, 2023.

[69] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.

27

[70] 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.

28