DOI: 10.1111/cgf.14911

Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 5
e14911 pp. 1 - 14

Factored Neural Representation for Scene Understanding

Yu-Shiang Wong[1] Niloy J. Mitra[1,2]

[1]University College London [2]Adobe Research

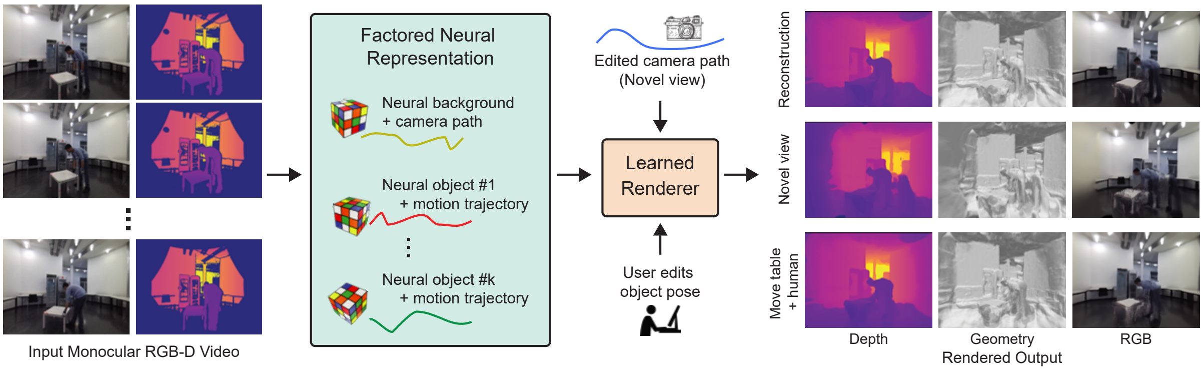
Input Monocular RGB-D Video
Factored Neural Representation
Neural background + camera path
Neural object #1 + motion trajectory
Neural object #k + motion trajectory
Edited camera path (Novel view)
Learned Renderer
User edits object pose
Reconstruction
Novel view
Move table + human
Depth
Geometry
RGB
Rendered Output

Figure 1: *We present an algorithm that directly factorizes raw RGB-D monocular video (sequence from [BXP*22] in this example) to produce object-level neural representation with motion trajectories and nonrigid deformation information. The decoupling subsequently enables different object manipulation and novel view synthesis applications to produce authored videos. We do not use object templates or motion prior but instead use an end-to-end optimization to enable the factorization. Note that the human subject deforms and moves in this sequence.*

Abstract

A long-standing goal in scene understanding is to obtain interpretable and editable representations that can be directly constructed from a raw monocular RGB-D video, without requiring specialized hardware setup or priors. The problem is significantly more challenging in the presence of multiple moving and/or deforming objects. Traditional methods have approached the setup with a mix of simplifications, scene priors, pretrained templates, or known deformation models. The advent of neural representations, especially neural implicit representations and radiance fields, opens the possibility of end-to-end optimization to collectively capture geometry, appearance, and object motion. However, current approaches produce global scene encoding, assume multiview capture with limited or no motion in the scenes, and do not facilitate easy manipulation beyond novel view synthesis. In this work, we introduce a factored neural scene representation that can directly be learned from a monocular RGB-D video to produce object-level neural presentations with an explicit encoding of object movement (e.g., rigid trajectory) and/or deformations (e.g., nonrigid movement). We evaluate ours against a set of neural approaches on both synthetic and real data to demonstrate that the representation is efficient, interpretable, and editable (e.g., change object trajectory). Code and data are available at: http://geometry.cs.ucl.ac.uk/projects/2023/factorednerf/.

CCS Concepts

• ***Computing methodologies*** → ***Reconstruction; Volumetric models;*** *Tracking;*

1. Introduction

Scene understanding from video capture has a long history in content creation. It subsequently enables editing by replaying the content from novel viewpoints and allowing object-level modifications. The task is particularly challenging in the dynamic context of moving and deforming objects when observed through a moving (monocular) camera. Traditional approaches make simplifications by assuming the scene to be static [CL96], or requiring access to a variety of priors in the form of object templates [CC13, RPMR13], deformable object models [BV99, ASK*05, LMR*15], or simultaneous localization and mapping [NLD11, IKH*11]. Additional complexity arising from unknown objects' appearance is ignored.

Neural Radiance Field (NeRF) [MST*20], a new volumetric neural representation, provided a breakthrough in terms of producing highly photorealistic (static) representation, simultaneously capturing geometry and appearance from only a set of posed images. A substantial body of work has rapidly emerged to extend the formulation to dynamic settings [LSZ*22, DZY*21, PCPMMN21, LNSW21, TTG*21, XHKK21, GSKH21], work with localized representations for real-time inference [LGZL*20, RPLG21, YLT*21, LSS*21, SSC22, KRWM22b, FYW*22, WZL*22], support fast training [DLZR22, SSC22, KRWM22b, FYW*22, LCM*22], and investigate applications in the context of generative models [KRWM22a]. However, such representations often lack interpretability, require multiview input, fail to provide scene understanding, and do not provide object-level factorization or enable object-level scene manipulation.

We introduce *factored neural representation*. This object-level scene representation supports interpretability and editability while capturing geometric and appearance details under object movement and viewpoint changes. Our approach does not require any object template, deformation prior, or pretraining object NeRFs. Starting from an RGB-D monocular video of a dynamic scene, we demonstrate how such a factored neural representation can be robustly extracted via joint optimization by leveraging off-the-shelf image-space segmentation and tracking information. Factorization is provided through object-level neural representations and object trajectory and/or deformations.

Technically, we formulate a global optimization to simultaneously build and track per-object neural representations along with a background model while solving for object trajectories and camera path. Further, we model deformable bodies (e.g., a moving human) by adapting the learned neural representation over time. Our proposed representation combines the advantages of object-centric representations and motion tracking, thereby allowing per object manipulation, without having to pay the overhead of separately building object priors or requiring 3D supervision, and naturally integrates information from a monocular input over time across the neural representations to recover from occlusion. For example, Figure 1 shows a factored representation obtained by our method by operating on a monocular RGB-D sequence [BXP*22] of 60 frames, along with some edits.

We evaluate on both synthetic and real scenes. We compare to competing methods and show that ours can produce better object representations and camera/object trajectories. Note that prior methods often focus on only rigid motion and separated optimization [WLNM21, MWM*21], assume access to geometric priors [MWM*21], a single non-rigid object with local motion [PSH*21, CFF*22], a global representation [LNSW21, TTG*21, XHKK21], foreground-background separation and novel-view rendering without geometric reconstruction [GSKH21, YLSL21, WZT*22, SCL*23], or static scenes with an implicit representation [ZPL*22, SLOD21, YPN*22]. We relax many of these restrictions and demonstrate that our factorized representation naturally enables edits involving object-level manipulations. In summary, we introduce a *neural factored scene representation* and develop an end-to-end algorithm involving a joint optimization formulation to factorize monocular RGB-D videos directly.

2. Related Work

Scene reconstruction using traditional methods. Aggregating raw scans while simultaneously estimating and accounting for underlying camera motion is an established way of acquiring large-scale geometry of rigid scenes (e.g., KinectFusion [IKH*11], VoxelHash [NZIS13]). This paradigm has been extended for dynamic scenes by simultaneously segmenting and tracking multiple (rigid) objects (e.g., CoFusion [RA17], MaskFusion [RBA18], MidFusion [XLT*19], EmFusion [SS19], RigidFusion [WLNM21]) or, decoupling the handling of objects and human motion (e.g., MixedFusion [ZX17]). These methods explicitly track and represent geometry, without or with textured colors, do not support joint optimization, and need special handling for multiple objects.

Neural implicit representation. In the context of object representation, the recent introduction of the neural implicit representation [PFS*19, MON*19, CZ19] has resulted in an explosion of works to overfit a single object or to encode object collections. Researchers have proposed improvements to better capture high-frequency details [TSM*20, MGB*21, SMB*20], and investigated hybrid implicit representations like point-based [EGO*20, CYAE*20, LWL*22, ZNW22], surface-based [GCS*20, CLI*20, MAG*22], or grid-based methods [TLY*21, CAPM20, PNM*20, KRWM22b] to achieve better trade-offs among inference speed, memory footprint, and locality of representations. These works couple geometry and appearance captures but largely focus on static, individual objects and do not model changing (object) configurations.

Neural representations through image guidance. In the context of joint material and geometry representation, differential rendering directly optimizes neural implicit representations using only RGB images for supervision. This is achieved by either ray tracing or volumetric rendering based approaches. Ray tracing accounts for explicit surface intersection and calculates gradient on the surface using implicit differentiation [NMOG20, YKM*20], max pooling [LSCL19], or unfolding sphere tracing [KJJ*21, LZP*20, JJHZ20]. However, for objects with complex topologies, these methods suffer from hard-to-propagate local gradients. In contrast, volumetric rendering [Max95, HMR19], leading to Neural Radiance Fields (NeRF) [MST*20], integrates density and color samples along rays by modeling a radiance field and employs a coarse-to-fine sampling scheme to focus on surface density, without explicitly distilling the underlying geometry. When converted from the density field, the learned implicit geometry is usually noisy and inaccurate. Again these approaches focus on isolated objects.

Neural scene representation. In scene analysis, combining volumetric rendering with an implicit representation [MST*20, WLL*21, OPG21] has led to a series of works revisiting traditional scene representations. For example, methods have been proposed for the 3D reconstruction and scene editing tasks, including indoor scene reconstruction [AMBG*22, YPN*22], structure from motion [MBRS*21], simultaneous localization and mapping [SLOD21, ZPL*22], bundle adjustment [AMBG*22, Cla22, LMTL21], multiview stereo [WLR*21], scene reconstruction using ellipsoid proxies [ZKF*23], surface meshing and interactive editing [GKE*22, JKK*23], distilling segmentation priors to extract instances using a single neural network [KMS22, WCY23]. Most of these works, however, focus on static scenes. In Section 5, we present several comparisons with IMAP [SLOD21] and NICESLAM [ZPL*22] that perform implicit scene representation with simultaneous tracking but focus on global scene representations with static objects.

Modeling dynamic objects. In order to obtain NeRF representations for deforming objects, parametric and non-parametric template models have been exploited. When parametric models are available (e.g., for human bodies), the underlying parametric template is utilized to create a part-based NeRF representation [PZX*21, CZK*21, GTZN21, LHR*21, NSLH21, WCS*22], i.e., each part having a corresponding NeRF encoding, to create dynamic avatar models with pose and shape control. For non-parametric models, dynamic NeRF has been proposed by solving for a template representation and capturing dynamic appearance by reindexing into a base (i.e., canonical model) NeRF representation [PCPMMN21, XAS21, PSB*21, PSH*21, FYW*22, WZL*22, CFF*22, LNSW21, GSKH21]. Such representations are then used to model rigidly moving objects assuming access to static pretrained NeRF [YCFB*21], predict object-space normalized coordinates for 6 DOF extraction and tracking [LYS*22], perform point-to-SDF tracking [UFK*22], or predict surface correspondences [HHM*22]. These methods focus on objects in isolation.

Modeling dynamic scenes. Recent works have trained global object NeRF from monocular input [LNSW21, GSKH21], capture dynamic effects by overfitting to a global 4D space-time volume [XHKK21, CJ23, FKMW*23], and explicitly capture human interactions [JJS*22, SGF*22]. Researchers have investigated the effect of segmentation, tracking, and NeRF modeling tasks in other efforts. Notable examples include monocular with foreground and background decomposition [MBRS*21, YLSL21, WZT*22, SCL*23], modeling rigid objects with a planar background model [OMT*21, KGY*22], egocentric video segmentation [TLV21], neural fusion fields [TLLV22]. We also develop a dynamic NeRF representation that can be extracted, without requiring pre-training and parametric templates, simultaneously with reconstruction for each object instead of only supporting novel-view rendering with a single dynamic element. Further, we aggregate information across views to recover from occlusion. Once trained, our factored representation can be viewed from novel camera paths and used to make changes to object trajectories and placements.

3. Image Formation Model

Before introducing the optimization formulation in Section 4, we present our image formation model to produce a rendered image from a factored neural representation.

Volume rendering. To render an image I from a given camera setup Π, volume rendering [Max95, MST*20] maps each image pixel to form a camera ray $\mathbf{r}$. Points are sampled on each such ray and sorted based on their depth values to produce a rendered color $C(\mathbf{r})$ as the integration of the sampled point colors $\{\mathbf{c_i}\}$ weighted by the corresponding point density $\{\sigma_i\}$ and (accumulated) transmittance $\{T_i\}$. Note that samples along a ray $\mathbf{r} := (\mathbf{o}, \mathbf{d})$, going through point $\mathbf{o}$ along a unit direction $\mathbf{d}$, are parameterized as $\mathbf{p}(s_i) := \mathbf{o} + s_i\mathbf{d}$ for increasing scalar depth samples $s_i \in \mathbb{R}^+$. Using the samples, we discretize the continuous formulation using the quadrature approximation as:

$$\begin{aligned} C(\mathbf{r}) &:= \sum_i T_i \alpha_i \mathbf{c}_i \\ \alpha_i &:= 1 - \exp(-\sigma_i \delta_i) \\ T_i &:= \prod_j^{i-1} (1 - \alpha_i) \end{aligned} \tag{1}$$

where δ_i is the depth distance between two adjacent samples and σ_i is the predicted point density. Recall that point opacity α_i represents the opacity of the point position $\mathbf{p}(s_i)$, while the transmittance T_i indicates the cumulative transmittance before a ray hits the i-th sample point. Looping over all the image pixels $\{uv\}$, we obtain $I := \mathcal{R}(\Pi, f_\theta, \{\mathbf{r}_{uv}\})$, where the function f_θ, typically modeled by an MLP [MST*20], can be probed to produce density and color samples as $f_\theta(\mathbf{p}(s_i), \mathbf{d}) := (\sigma_i, \mathbf{c}_i)$. Typically, only the color values are view dependent.

Volume rendering with implicit surface. Implicit surface representation, such as occupancy or signed distance fields, can also be used with volume rendering [WLL*21, OPG21, YGKL21] and provides an inductive bias for modeling surface geometry. We found this more suitable for object-level factored representation as we can easily regularize the optimization to encode object surfaces instead of producing volumetric clouds. Here, we employ the signed distance field formulation proposed by Wang *et al.* [WLL*21] and convert the signed distance value ψ to the density values by assigning non-zero values near the zero level set of the modeled surface geometry:

$$\alpha_j \leftarrow max\left(\frac{\Phi(\psi_j) - \Phi(\psi_{j+1})}{\Phi(\psi_j)}, 0\right), \tag{2}$$

where we use a shorthand $\psi_j := \psi(\mathbf{p}(s_j))$ for the j-th sample and Φ is the sigmoid function. Here, we represent the rendering function as $I := \mathcal{R}(\Pi, f_\theta, \psi, \{\mathbf{r}_{uv}\})$, where the function f_θ again can be probed

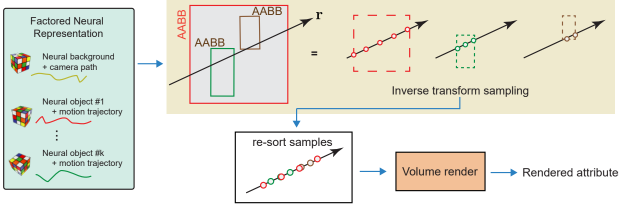
Factored Neural Representation
Neural background + camera path
Neural object #1 + motion trajectory
Neural object #k + motion trajectory
AABB
AABB
AABB
r
=
Inverse transform sampling
re-sort samples
Volume render
Rendered attributes

Figure 2: ***Rendering neural factored representation.*** *Given a factored representation $\mathcal{F}\{(f^i, \psi^i, B_i, \mathbf{T}_i)_{i=0}^k\}$ and any query ray $\mathbf{r}$ from the current camera, we first intersect each objects' bounding box B_i to obtain a sampling range and then compute a uniform sampling for each of the intervals. For each such sample $\mathbf{p}$, we lookup feature attributes by re-indexing using local coordinate $\mathbf{T}_i^{-1}\mathbf{p}$, resort the samples across the different objects based on (sample) depth values, and then volume render to get a rendered attribute. Background is modeled as the 0-th object. See Section 3 for details. For objects with active nonrigid flag, we also invoke the corresponding deformation block (see Section 4 and Figure 5). The neural representations and the volume rendering functions are jointly trained.*

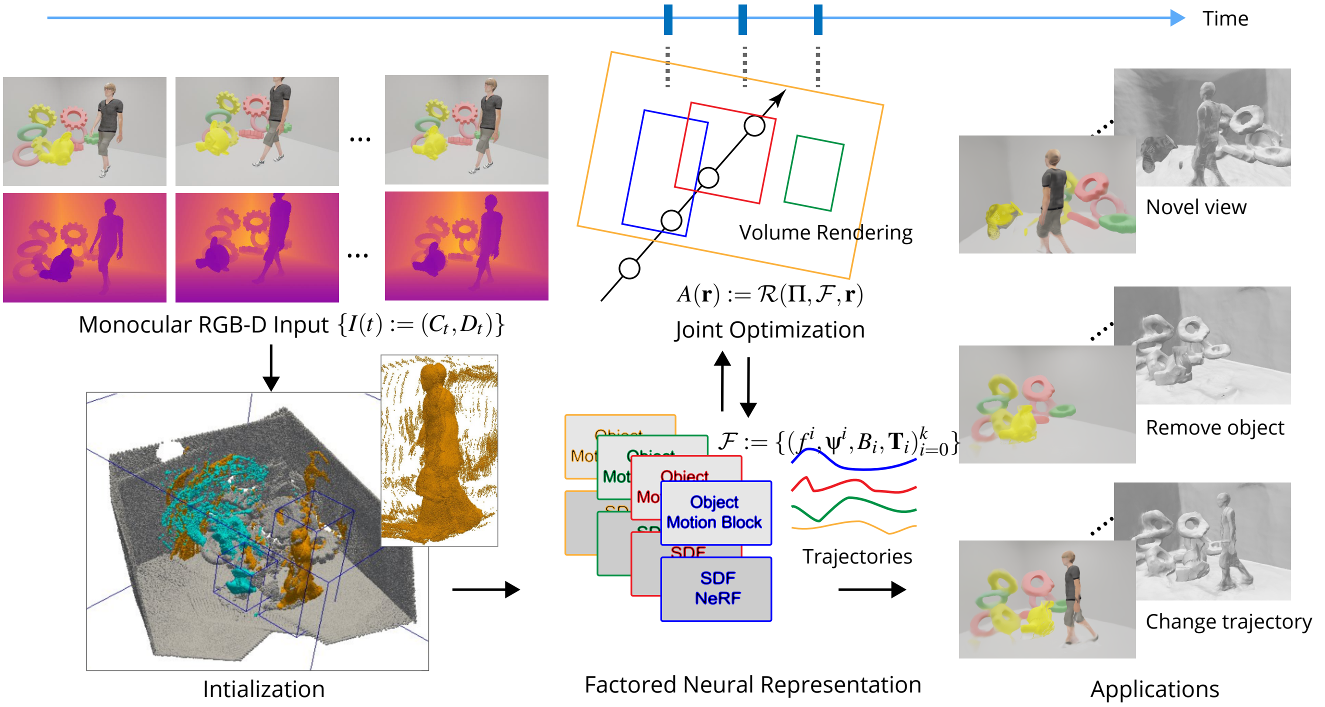
Time
Monocular RGB-D Input $\{I(t) := (C_t, D_t)\}$
Volume Rendering
$A(\mathbf{r}) := \mathcal{R}(\Pi, \mathcal{F}, \mathbf{r})$
Joint Optimization
Novel view
Remove object
Change trajectory
$\mathcal{F} := \{(f^i, \psi^i, B_i, \mathbf{T}_i)_{i=0}^k\}$
Object Motion Block
SDF NeRF
Trajectories
Intialization
Factored Neural Representation
Applications

Figure 3: ***Method overview.*** *Starting from a monocular RGB-D sequence $\{I(t)\}$, we extract a* ***factored neural representation*** *$\mathcal{F}$ that contains separate neural models for the background and each of the moving objects along with their trajectories. For any object tagged as* `nonrigid`*, we also optimize a corresponding deformation block (e.g., human). First, in an initialization phase, we assume access to keyframe annotation (segmentation and AABBs) over time, propagate the annotation to neighboring frames via dense visual tracking and optical flow, and estimate object trajectories. Then, we propose a joint optimization formulation to perform end-to-end optimization using a customized neural volume rendering block. The factored representation enables various applications involving novel view synthesis and object manipulations. Please refer to the supplemental showing reconstruction quality and applications.*

to produce only view-dependent color samples $f_\theta(\mathbf{p}(s_i), \Pi) := \mathbf{c}_i$ and ψ represents the learned SDF function.

Attributes rendering. By replacing point color $\mathbf{c}_i$ with any other attribute a_i, such as depth [XHKK21, ZPL*22] or semantic labels [ZSM*21], volumetric rendering can be generalized to render depth or semantic segmentation, respectively. Specifically, for any attribute a_i and ray $\mathbf{r}$, we simply compute an attribute as $A(\mathbf{r}) := \sum_i T_i \alpha_i a_i$.

Volume rendering with factored neural representation. Our proposed factored representation $\mathcal{F} := \{(f^i, \psi^i, B_i, \mathbf{T}_i)_{i=0}^k\}$ for a background model f^0 and the foreground objects $\{f^i, i \in [1,k]\}$, which can be probed to output density and color attributes. Each model, the background or any foreground object, can be probed to output color attributes with corresponding AABB (axis aligned bounding boxes) $\{B_i\}$, transformations $\{\mathbf{T}_i\}$ to map the AABB local coordinates to the global coordinate system, and implicit SDF functions $\{\psi^i\}$ to produce density samples. We now define the rendering function $I := \mathcal{R}(\Pi, \mathcal{F}, \{\mathbf{r}_{uv}\})$ using our factored representation $\mathcal{F}$, with background and superscripts $i \in [1,k]$ denoting the k foreground objects. Figure 2 illustrates the process. For each ray $\mathbf{r}$, for each intersected model, computed using its AABB B_i, we obtain SDF density values using uniform samples and perform inverse transform sampling to generate 128 samples per ray. For the background ($i = 0$) or foreground ($i \in [1,k]$) samples, we obtain $f^i(\mathbf{T}_i^{-1}\mathbf{p}^i(s_j), \Pi) := \mathbf{c}^i_j$ and density σ^i_j using Equation 2 using ψ^i using the remapped samples $\mathbf{T}_i^{-1}\mathbf{p}^i(s_j)$, expressed in the local coordinate systems of the objects. We collect the samples across the background and all the intersecting objects, sort the samples based on their depth values, and volume render the colors/attributes as described earlier (see Equation 1).

4. Algorithm

As input, we take in RGB-D frames, denoted by $\{I(t) := (C_t, D_t)\}$ with color C_t and depth D_t frames at time t, of scenes with one or more moving objects, where objects can be moving rigidly or nonrigidly (e.g., humans). We assume access to keyframe annotation over time, containing instance segmentation, axis-aligned bounding boxes (AABBs), and `rigid` or `nonrigid` flags. In an initialization step, this information is used to extract initial camera and object trajectories and instance masks over time in the camera space. As output, we produce a factored neural representation $\mathcal{F}$ of the scene, where for each object we produce a neural representation along with its estimated object trajectory, and for a nonrigid object also an associated deformation function. In Section 5, we use these inferred factored representations to directly render novel view synthesis or perform object-level manipulations.

To obtain such a factored representation, we have to address several challenges. First, the extracted segmentation information from the RGB-D frames is imperfect; hence, any information or supervision (e.g., segmentation loss) derived from them leads to error accumulation. Second, we must recover from artifacts in initial pose estimation, especially in scenes with insufficient textures to guide the camera calibration stage. Auto-focus, color correction, and error accumulation in real captures pose further challenges. Third, since we only use monocular input, the input provides partial information in the presence of occlusion, both in shape and appearance. Without priors, we have to recover from the missing information by fusing information across the (available) frames. Finally, we allow objects to exhibit nonrigid motion (e.g., human walking) and have to factorize object deformation from object motion. In the following, we present how to set up a joint optimization, with suitable initialization and regularizers, involving object tracking, neural representations, and volume rendering to solve these challenges.

Initialization. We use an off-the-shelf visual tracker [WZB*19] with keyframe annotation, including instance segmentation and AABBs, to propagate the keyframe segmentation across the frames. To get an initial registration, we run an optical flow network [TD20] to find initial correspondences and solve for frame-to-frame rigid alignment using iterated closest point (ICP) approach. The registration information across frames provides object trajectory $\{\mathbf{T}_i(t)\}$ estimates.

Joint optimization. We now introduce the main loss terms to capture reconstruction quality and additional regularizers to get a desired factored representation.

Reconstruction loss: We render color and depth images using current (multi-object) neural factored representation as described in Section 3. Note that the object trajectories $\mathbf{T}_i$ are indexed by frame times, i.e., $\mathbf{T}_i(t)$. We compare the sampled color C and depth D attributes in a set of minibatch samples P against the estimated attributes using the L1 reconstruction loss, i.e.,

$$L_{\text{color}}(\mathcal{F}) := \sum_{(\mathbf{r},t)\in P} \|C_t(\mathbf{r}) - R_C(\Pi, \mathcal{F}(t), \mathbf{r})\| \,/\, |P| \text{ and}$$
$$L_{\text{depth}}(\mathcal{F}) := \sum_{(\mathbf{r},t)\in P} |D_t(\mathbf{r}) - R_D(\Pi, \mathcal{F}(t), \mathbf{r})| / |P|. \tag{3}$$

We render the current background and foreground neural objects to produce RGB and depth attributes and sum them up over the individual frames.

Free-space loss: One approach to check the factorization quality is to compare the predicted object segmentation, computed using the current re-projection of objects' transmission, against the input segmentation. However, this approach leads to poor results as segmentation estimates are noisy. Instead, we focus on the complement space and define a free-space loss (cf., [XHKK21]) to penalize density values in regions indicated to be free according to the raw depth information. For any point sampled from any of the objects, we want identify free-space samples using depth $D(\mathbf{r})$. Specifically, we constrain the integrated weights of each free-space sample $\mathbf{p} \in P_{\text{free}}$, before reaching the object point (i.e., zero-isosurface of ψ^i), to be zero using L1 loss. We found this loss to be better than a cross-entropy segmentation loss in the joint-training setting. Specifically,

$$L_{\text{free}}(\mathcal{F}) := \sum_{\mathbf{p}\in P_{\text{free}}} |\, T_{\mathbf{p}} \alpha_{\mathbf{p}} \,| / |P_{\text{free}}|$$
$$\text{where} \quad P_{\text{free}} = \{\mathbf{p}(s,\mathbf{r}) | s < D_t(\mathbf{r})\}\,. \tag{4}$$

Non-rigid deformation: In order to handle non-rigid objects, we additionally incorporate a deformation block, for objects marked with flag `nonrigid`. Specifically, we adopt a state-of-the-art bijective deformation network proposed by Cai *et al.* [CFF*22], which consists of three sub-networks, each predicting a low-dimensional deformation. Given an input 3D point, each sub-network selects one axis, predicts a 1D displacement, and infers a 2D translation and rotation for the other axes. These sub-networks are sequentially invoked in the XYZ axis order. Note that this block gets directly optimized via the reconstruction loss and is *not* supervised with ground truth deformation.

Surface regularizers: In order to regularize our network to output a canonical model, we employ auxiliary losses to constrain our geometry models to be actual surfaces by penalizing the implicit functions ψ^i to (i) be a true signed distance field (i.e., using Eikonal loss) ; (ii) requiring the surface points (i.e., points within $\pm\varepsilon$ of the zero level set of the SDFs denoted by $\Omega_\varepsilon(\psi^i)$) to have normals in the direction of normals $\mathbf{n}(\mathbf{x})$ estimated from the input RGB-D [GKOM18]; and (iii) surface points to have zero implicit values. These auxiliary losses does not slow down the optimization since they can be directly calculated without performing volumetric rendering. Putting them together we get,

$$L_{\text{surface}}(\mathcal{F}) := \frac{1}{(k+1)} \sum_{i\in[0,k]} \Bigg[\sum_{\mathbf{x}\in P_{B_i}} |\|\nabla\psi^i(\mathbf{x})\|_2 - 1| / |P_{B_i}|$$
$$+ \sum_{\mathbf{x}\in P_{\Omega_i}} |1 - < \nabla\psi^i(\mathbf{x}), \mathbf{n}(\mathbf{x}) > | / |P_{\Omega_i}|$$
$$+ \sum_{\mathbf{x}\in P_{\Omega_i}} |\psi^i| / |P_{\Omega_i}| \Bigg], \tag{5}$$

where P_{B_i} and P_{Ω_i} denote the randomly sampled spatial points and surface samples in the object bounding box B_i, respectively. Finally, we arrive at the full optimization problem as,

$$\min_{\mathcal{F}} L_{\text{total}}(\mathcal{F}) := L_{\text{color}}(\mathcal{F}) + \lambda_1 L_{\text{depth}}(\mathcal{F})$$
$$+ \lambda_2 L_{\text{free}}(\mathcal{F}) + \lambda_3 L_{\text{surface}}(\mathcal{F}), \tag{6}$$

We use $\lambda_1 = 0.1$, $\lambda_2 = 1.0$, and $\lambda_3 = 0.1$ in our experiments where $\lambda_1 < 1$ due to noisy depth input. Recall that the factored representation $\mathcal{F} := \{(f^i, \psi^i, B_i, \mathbf{T}_i)_{i=0}^k\}$ maintains a specialized model for the background and each of the object trajectories $\mathbf{T}_i(t)$ being time dependent.

5. Evaluation

We evaluated Factored Neural Representations on a variety of synthetic and real scenes, in the presence of rigid and nonrigid objects. In each case, we start with only RGB-D sequences, without access to any object template.

Dataset. We tested on two types of datasets, synthetic and real. As *synthetic dataset*, we propose a new dataset using public available CAD models [CFG*15, GBB*22] and render RGB-D sequences using Blender [GBB*22, Com18] with simulated sensor noises [HWMD14]. To inject motion, we manually edit camera motion, rigid object motion, and combine non-rigid motion from the DeformingThings4D [LTT*21] dataset. As representative examples, we present three sequences, SYN-SCENE A, B, and C, each spanning for 90-100 frames and containing multiple dynamic objects. For these synthetic sequences, we have access to ground truth data (e.g., object trajectory, object segmentation, deformation model). *This new dataset will be made publicly available on publication*.

As *real dataset*, we use the BEHAVE [BXP*22] dataset, which provides human object interaction RGB-D videos with keyframe annotation. We crop and evaluate the first non-occlusion sequence in each scene to avoid the object re-identification issue. Figure 4 shows some representative frames.

Architecture. Figure 5 shows our network architecture. For the geometry network, we use an SDF field with geometric initialization [GYH*20], weighted normalization [SK16], Softplus activations, and a skip-connection MLP. The input coordinates and view directions are lifted to a high dimensional space using positional encoding [MBRS*21]. For rigid objects, we use $\mathbb{SE}3$ representation, i.e., a quaternion and a translation vector. For non-rigid objects, we use bijective deformation blocks [CFF*22] with Softplus activations. For the color MLP network, we use ReLU activation.

Model size and implementation details. We report the model size of our methods and comparisons. IMAP uses 0.9MB (FG/BG); NICESLAM uses 76MB (FG) and 135MB (BG) with 32^3+64^3 grid resolutions for foreground and 32^3+80^3 for background. In contrast, our model takes 5.7MB for the whole scene. We train all methods using our training framework on a single Nvidia RTX 3090 GPU. We do not use input depth to guide ray sampling for any of the methods as we observed that this reduces models' generalization ability. Instead, at each training iteration, we perform inverse transform sampling and sample 256 rays with 128 points per ray.

Comparison. We compare our approach against different competing alternatives. Existing monocular approaches can be categorized as either employing an MLP (e.g., IMAP [SLOD21]), or using multi-resolution feature grids (e.g., NICESLAM [ZPL*22]). Since these competing methods do not support joint optimizing multiple objects, we *additionally provide* the segmentation and poses generated from our initialization step (see Section 4) and *manually* run them multiple times to reconstruct background and dynamic objects. Note that we modified the ray sample step of IMAP and NICESLAM to accept object segmentation input, and we use L1 segmentation loss when training foreground models. For both IMAP and NICESLAM, we employ the open-source network implementation [ZPL*22] in our training framework instead of their multi-threads SLAM framework, which contains several optimizations (e.g., view-purging) for real-time applications. Note that IMAP, with provided background and object segmentation information, can be seen as an upper bound for performance of a method like RigidFusion [WLNM21].

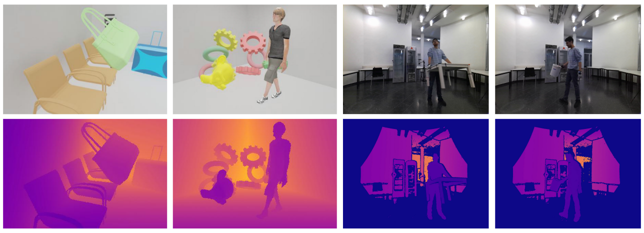

Figure 4: ***Dataset.*** *We test on a mix of dynamic datasets, including synthetic scans (SYN-SCENE A, B, and C) and real RGB-D monocular captures from BEHAVE [BXP*22] (tablesquare-move, trashbin, yogaball-play, chairblack-lift). Here we show some representative frames, RGB (top) and depth (bottom).*

Evaluation metrics. We compare different methods across a range of metrics. We evaluate novel view *rendering quality* using PSNR, SSIM, and L1 for reconstruction quality in Table 1 and Table 2. We also qualitatively evaluate resynthesis quality under the authoring of updated object trajectory as well for addition or deletion of objects from the factored scenes in Figure 8.

Qualitative evaluation. In Figure 6, 7, and 8, we qualitatively compare our method against alternative approaches (IMAP and NICESLAM). Note that although the comparison approaches jointly learn for scene geometry and appearance, they assume the scenes to be static. In other words, these methods provide only partial factorization into scene models and camera trajectories. Thus, we run them multiple times with the same segmentation and pose initialization to reconstruct background and dynamic objects. Please check the supplemental webpage for result comparisons.

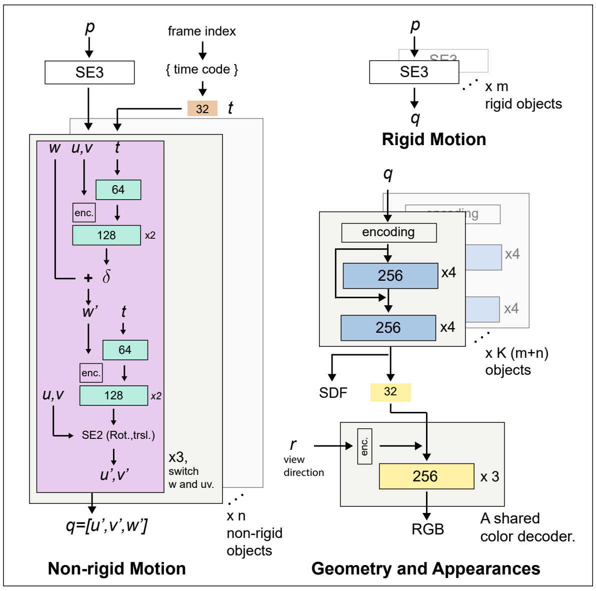

Figure 5: ***Our network architecture.*** *We use sine positional encoding as NeRF [MBRS*21]. The number of rigid and non-rigid motion blocks depends on the objects' motion labels. We employ a bijective deformation block [CFF*22] for each non-rigid object. Unlike [PSH*21, CFF*22], we do not predict ambient coordinates in the non-rigid motion block.*

Table 1: ***Reconstruction error on our synthetic dataset.*** *(Top/Bottom) Quantitative color/depth novel view rendering results on validation cameras. Ours largely produces better reconstruction, validating that our joint optimization captures better scene geometry.*

Color Reconstruction (PSNR ↑ / SSIM ↑)											
	SYN-Scene-A			SYN-Scene-B				SYN-Scene-C			
	Full	BG	FG1	Full	BG	FG1	FG2	Full	BG	FG1	FG2
IMAP	20.32/0.79	21.02/0.86	15.02/0.88	17.98/0.78	21.87/0.90	18.78/0.93	-	16.56/0.76	25.73/0.93	18.48/0.90	-
NICESLAM	22.48/0.85	23.16/0.90	14.34/0.88	18.89/0.80	26.38/0.91	18.11/0.93	-	15.45/0.78	23.20/0.91	17.90/0.92	-
Ours	24.38/0.86	24.91/0.90	19.31/0.93	22.77/0.82	27.01/0.92	20.71/0.95	16.75/0.88	20.04/0.80	27.15/0.93	20.97/0.95	15.23/0.84
Depth Reconstruction (PSNR ↑ / L1 ↓)											
	SYN-Scene-A			SYN-Scene-B				SYN-Scene-C			
	Full	BG	FG	Full	BG	FG	FG2	Full	BG	FG	FG2
IMAP	17.87/0.67	22.96/0.45	14.99/0.27	16.49/0.96	20.23/0.66	16.74/0.10	-	16.43/1.63	21.22/1.27	16.74/0.19	-
NICESLAM	16.23/0.62	20.84/0.38	13.90/0.32	12.64/1.16	14.82/0.79	16.03/0.12	-	17.50/0.84	26.32/0.39	16.68/0.19	-
Ours	23.19/0.26	26.70/0.18	18.59/0.11	21.81/0.33	26.67/0.20	18.75/0.07	14.96/0.19	23.60/0.45	31.28/0.27	19.78/0.11	13.73/0.50

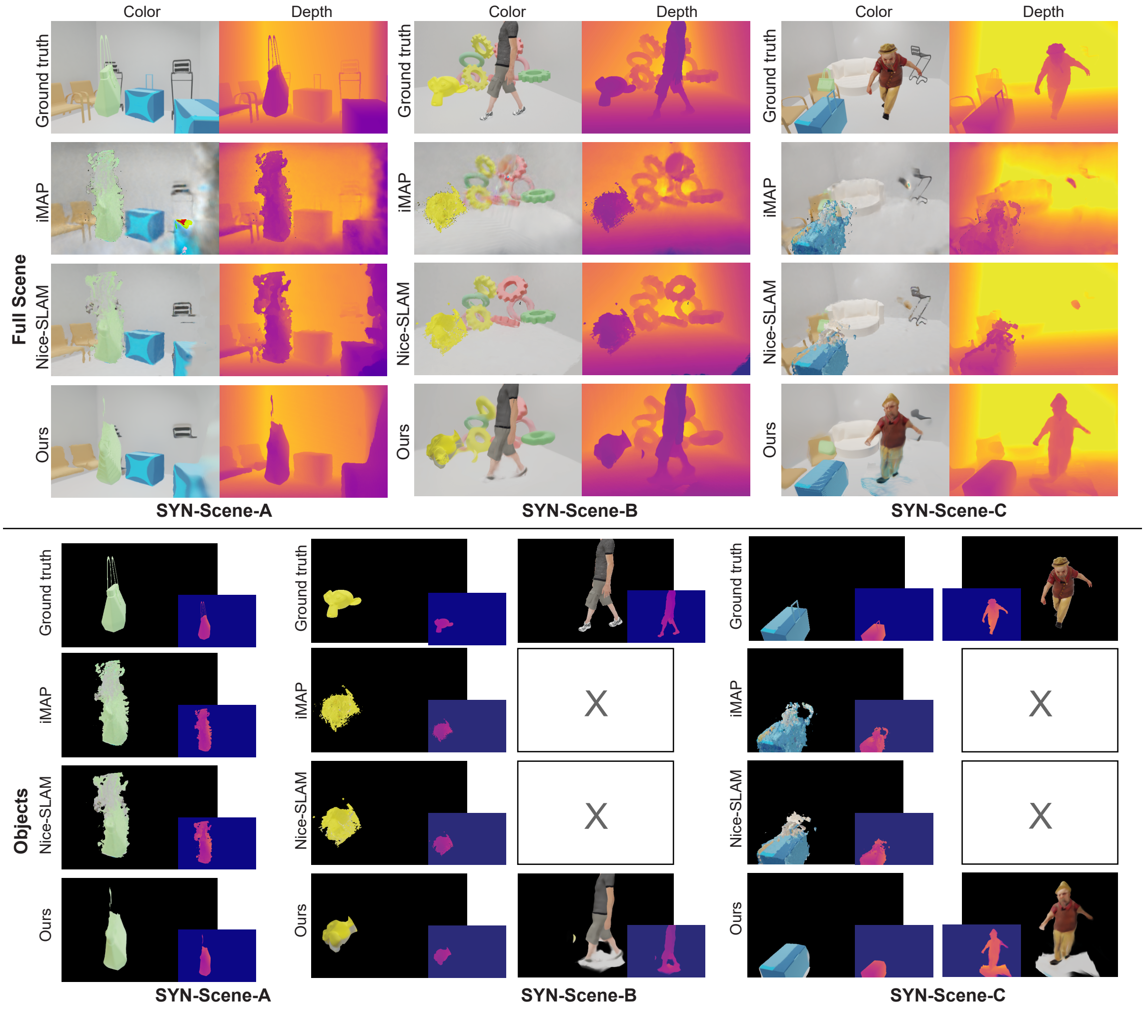

Figure 6: ***Comparisons of scene and object reconstruction on our synthetic dataset.*** *Visually comparing ours against* IMAP *[SLOD21] and* NICESLAM *[ZPL*22] on our synthetic sequences using the validation cameras. See Table 1 for quantitative evaluation. Note that the other methods fail to produce any reconstruction for the nonrigidly moving human. Further, our results are higher in quality and capture finer geometric (e.g., the handle of the green bag) and appearance details (e.g., shading on the yellow monkey face). Please note that Scene C shows a challenging validation frame where the human undergoes a strong deformation. Handling it requires more regularization to force the network to learn the non-rigid motion.*

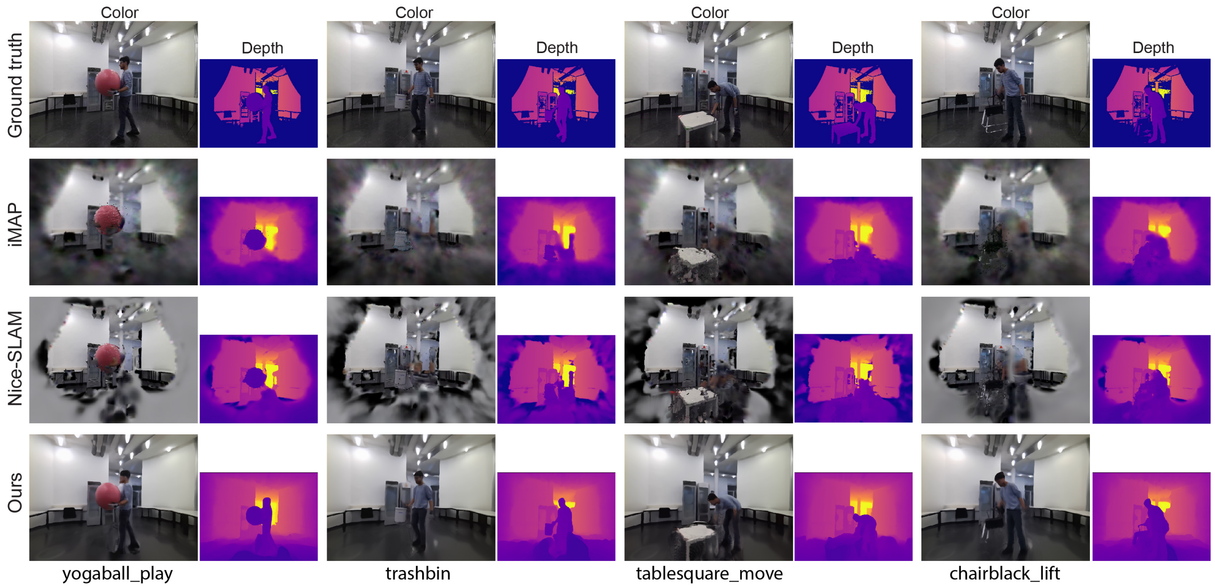

Figure 7: ***Comparisons of scene reconstruction on the* BEHAVE *dataset.*** *Visually comparing our results against* IMAP *[SLOD21] and* NICESLAM *[ZPL*22] on the* BEHAVE *sequences. Notably, our method generalizes better when the scene contains large missing depth areas, showing the learned geometry model is constrained well (see the wall in the training views). See Table 2 for quantitative evaluation.*

Table 2: ***Reconstruction error on* BEHAVE.** *We report total scene reconstruction errors using the training camera k0 due to the lack of validation views and per-frame annotation. Our method consistently produces better reconstruction quality benefiting from the proposed joint optimization and the deformation module. See Figure 7 for qualitative evaluation.*

Color Reconstruction (PSNR ↑ / SSIM ↑)				
	tablesquare_move	trashbin	yogaball_play	chairblack_lift
IMAP	14.68 / 0.66	13.46 / 0.65	12.42 / 0.64	13.28 / 0.64
NICESLAM	11.35 / 0.54	11.71 / 0.55	12.16 / 0.60	13.30 / 0.63
Ours	26.48 / 0.85	27.75 / 0.87	28.03 / 0.87	26.71 / 0.85

Depth Reconstruction (PSNR ↑ / L1 ↓)				
	tablesquare_move	trashbin	yogaball_play	chairblack_lift
IMAP	24.84 / 0.31	21.67 / 0.49	22.25 / 0.43	26.21 / 0.28
NICESLAM	25.48 / 0.23	21.89 / 0.44	22.62 / 0.38	27.00 / 0.24
Ours	30.06 / 0.14	30.07 / 0.15	30.29 / 0.13	30.39 / 0.14

Our method produces better quality on both synthetic and real-world scans, both appearance and geometry. Figure 6 demonstrates our joint optimization scheme improves the object segmentation leading to clearer geometry. Figure 7 shows the comparison of scene reconstruction on the BEHAVE sequences with large missing depth areas. Our method generalizes better than the comparison. In Figure 8, we also present the extracted object motion trajectories in $\mathbb{R}^3$ as recovered by our initialization step. For the synthetic example, we add groundtruth trajectories (gray colored) for comparison. Note that since we do not perform any loop closure, the trajectory estimates degrade over a longer distance due to error accumulation.

Quantitative evaluation. We present a quantitative comparison in Table 1 for reconstruction quality using the validation cameras, separately for RGB and depth channels. Notably, our method consistently outputs better reconstruction than others (IMAP and NICESLAM), indicating that our sampling scheme extracts a proper factorization and hence avoids overfitting to training views. In the absence of ground truth and validation views, we cannot run quantitative evaluation for real sequences (Figure 7).

Ablation study. In Table 3 and Figure 10, we conduct an ablation study using our synthetic dataset. While the commonly employed segmentation loss [WLL*21, YGKL21, CFF*22] can constrain the object shape through the rendered mask (weights of each sampled ray), it blocks the foreground reconstruction in joint optimization. The surface regularizers can stabilize the geometry models and improve both color and depth reconstruction. Our final setting (with surface regularizers and freespace loss) has the best full-scene reconstruction quality. The segmentation loss fights with thee reconstruction loss in the joint training setting, and the implicit networks fail to learn object surface. Therefore, we replace the segmentation loss with freespace loss allowing the network to optimize all objects and learn correct object geometry.

Model size and reconstruction quality. We conduct another ablation study to examine the effect of model size using our synthetic dataset. We adjust the hidden dimension size and set up three models: small, medium, and big, with 370K, 1381K, 5342K parameters, respectively. We trained all models for 100K iterations and observed that the reconstruction quality increased linearly when more parameters were used. The result color PSNR values are 22.9 (small), 23.1 (medium), and 23.18 (big); and the depth L1 errors are 0.35 (small), 0.33 (medium), and 0.32 (big).

Applications. We demonstrate three different editing modes in Figure 8: (i) novel view synthesis by changing the extracted camera

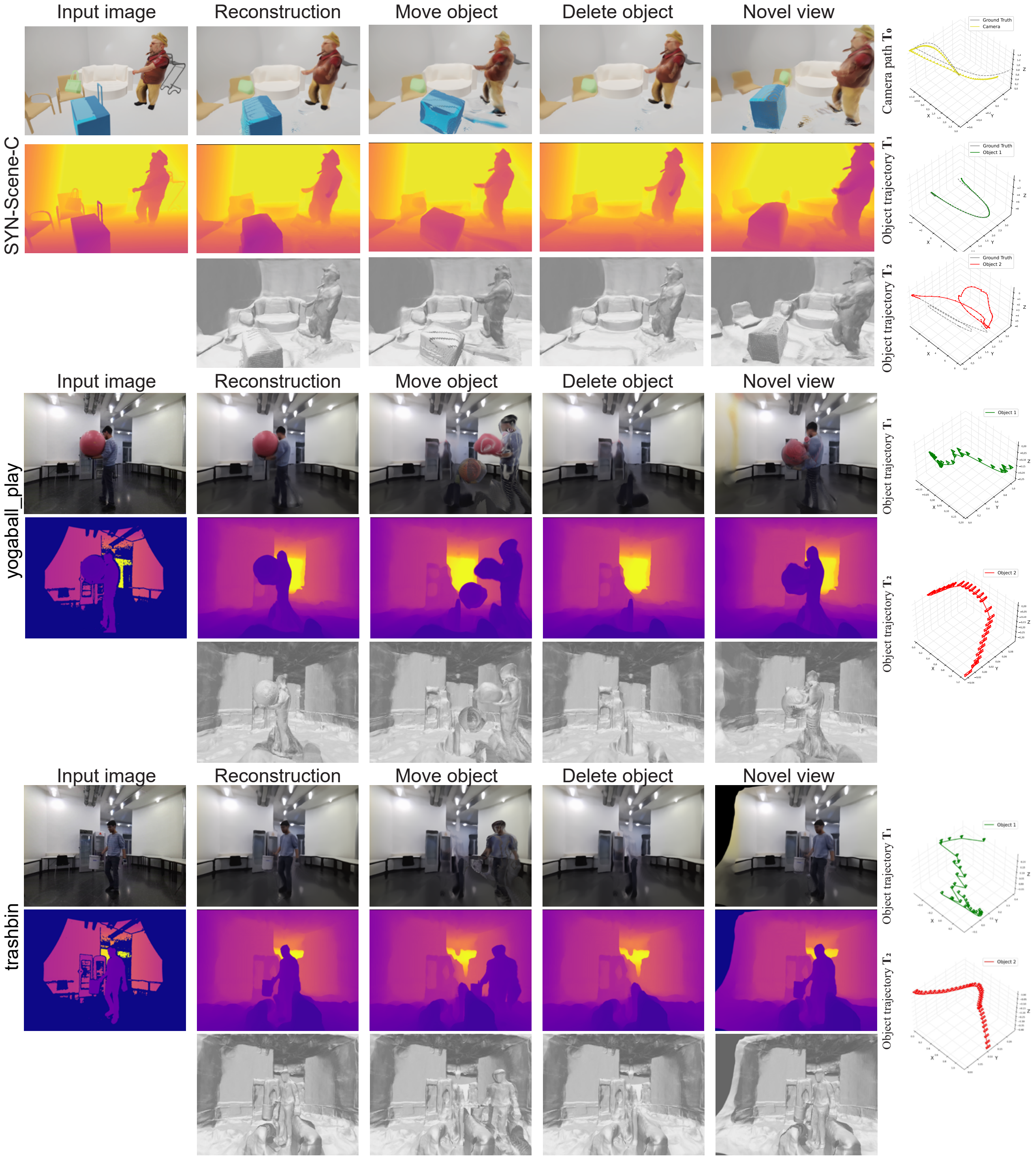

Figure 8: ***Reconstruction and Applications.*** *Reconstruction quality and enabled applications on two synthetic scenes. Please refer to the supplemental videos. Here we show frames for the output RGB, depth, and underlying recovered geometries (extracted by running Marching cubes on the estimated implicit representations ψ^i). We also show the recovered trajectories, along with corresponding ground truth trajectories. Recall that the 0-th object being the background, and $\{\mathbf{T}_0(t)\}$ represents the camera path. Any stationary object gets reconstructed in the background layer in our factorization. We observed some artifacts caused by unseen geometry (e.g., move objects examples in the second and third rows) and ambiguous decomposition (e.g., the blue box in the first row), because we only have access to monocular and partially occluded input.*

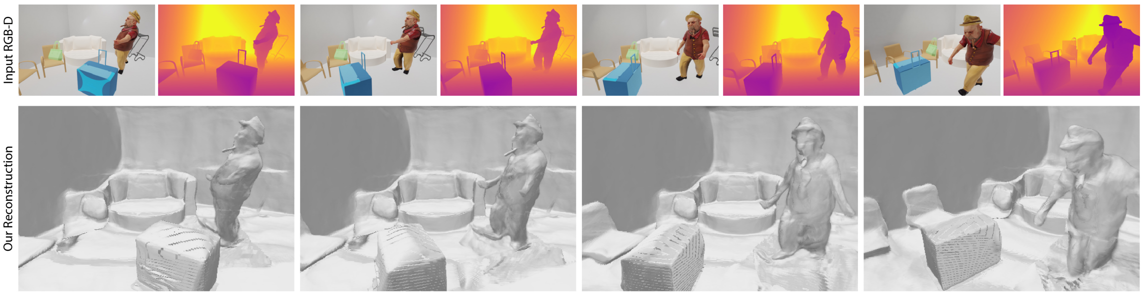

Figure 9: ***Dynamic scene reconstruction.*** *We demonstrate our full scene reconstruction exhibiting non-rigid object deformation across different time indexes. Note how ours can recover plausible movement of the movement of the limbs across time.*

Table 3: ***Ablation study on our synthetic dataset.*** *We evaluate total scene reconstruction errors using the validation cameras on our synthetic dataset. Segment. Loss: supervise the rendered masks (weights of each sampled ray) using the input segmentation [WLL*21, YGKL21, CFF*22]. Recon. Loss: color and depth reconstruction loss. Surface Reg. and Freespace Loss: the surface regularizer and the loss described in Section 4. We observed that the commonly employed segmentation loss is unsuitable for supervising multiple implicit networks (iv) and causes a large performance drop. Also, the freespace loss is not performed well in single object setting (v) due to the lack of background information. See Figure 10 for the visualization. Our setting (vi) performs well and is able to handle imperfect segmentation input.*

Ablation Settings						Scene Reconstruction	
	Recon. Loss	Segment. Loss	Surface Reg.	Freespace Loss	Joint Training	Color (PSNR↑/ SSIM↑)	Depth (PSNR↑/ L1↓)
i	✓					18.59 / 0.81	16.54 / 0.65
ii	✓	✓				18.87 / 0.82	17.31 / 0.57
iii	✓	✓	✓			20.08 / 0.79	18.34 / 0.49
iv	✓	✓	✓		✓	14.47 / 0.74	8.48 / 3.07
v	✓		✓	✓		17.85 / 0.76	14.85 / 0.85
vi	✓		✓	✓	✓	22.78 / 0.84	20.99 / 0.42

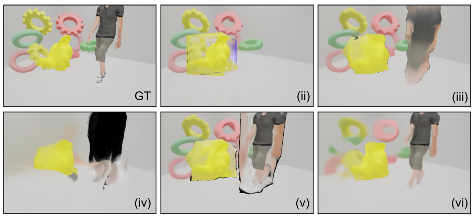

Figure 10: ***Ablation study and reconstruction results.*** *The number indicates the setting in Table 3. Surface regularizers (iii) enforce the network to learn geometry. The segmentation loss in joint setting (iv) performed poorly due to the conflicted signals between each network, which may require per-object rendering during training.*

trajectory; (ii) object level manipulation by changing one or more object trajectories; (iii) deleting objects by removing them from the factored representations. Note that the scene-specific learned renders are held fixed during any of the edits. While we only train with monocular input, our model can still support editing and output reasonable reconstruction. These edit modes are be applied separately or in parallel, and test the quality of the scene understanding (i.e., factorization) by revealing unseen object parts and configurations. These editing operations are non-trivial because our model is supervised using monocular input containing large motion. Removing the artifacts in Figure 8 will be interesting future work.

6. Conclusion

We have presented *factored neural representation* along with a joint optimization formulation that allows to separate a monocular RGB-D video into object level encodings, without requiring access to additional shape or motion priors. We demonstrated how to directly obtain object level coupled geometry and appearance encoding, along with object trajectories and deformations. The factorized representation directly supports novel view synthesis along with authoring edits on object trajectories. Our work has limitations that we want to address in future works, as discussed next.

Joint camera and object tracking. In our current implementation, we do not optimize the camera obtained during the initialization phase. It would be interesting to jointly finetune the initial estimates, possibly by loop closing and locally linearizing the transformation estimates to simplify the resultant optimization.

Inter object interactions and shading. In this paper, we do not model object-object or object-background effects. For example, we do not explicitly model shadows [WZT*22], reflections [GKB*22], transparency [IAKG20], or object interactions arising from human affordance considerations. In the future, it would be a possibility to model these in the volume rendering step.

Better architecture. At present, we modeled object functions of the form f_θ simply using MLPs. More recent alternatives and localized versions like hashing [MESK22] or direct functions (e.g., Relu-Fields [KRWM22b]) can be alternatively explored. However, the challenge would then be to effectively integrate information across multiple frames to model deformations, possibly by dynamically reindexing the local grid-based representations.

Shape priors. As our method does not rely on any object or motion priors, it cannot recover from significant occlusions. We plan to regularize the problem by incorporating data priors, and possibly reducing the dimensions of the variables by working in a learned latent space. However, even deciding which representation to use to anchor such a learned shape space for arbitrary objects still remains an open research topic.

Acknowledgement This work was supported by the Marie Skłodowska-Curie grant number 956585, the UCL AI Centre, and gifts from Adobe.

References

[AMBG*22] Azinović D., Martin-Brualla R., Goldman D. B., Niessner M., Thies J.: Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 6290–6301. 2

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (2005), 408–416. 1

[BV99] Blanz V., Vetter T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 187–194. 1

[BXP*22] Bhatnagar B. L., Xie X., Petrov I. A., Sminchisescu C., Theobalt C., Pons-Moll G.: Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 15935–15946. 1, 2, 6

[CAPM20] Chibane J., Alldieck T., Pons-Moll G.: Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 6970–6981. 2

[CC13] Choi C., Christensen H. I.: Rgb-d object tracking: A particle filter approach on gpu. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2013), IEEE, pp. 1084–1091. 1

[CFF*22] Cai H., Feng W., Feng X., Wang Y., Zhang J.: Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *arXiv preprint arXiv:2206.15258* (2022). 2, 3, 5, 6, 8, 10

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 6

[CJ23] Cao A., Johnson J.: Hexplane: A fast representation for dynamic scenes. *arXiv preprint arXiv:2301.09632* (2023). 3

[CL96] Curless B., Levoy M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 303–312. 1

[Cla22] Clark R.: Volumetric bundle adjustment for online photorealistic scene capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 6124–6132. 2

[CLI*20] Chabra R., Lenssen J. E., Ilg E., Schmidt T., Straub J., Lovegrove S., Newcombe R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision* (2020), Springer, pp. 608–625. 2

[Com18] Community B. O.: *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org. 6

[CYAE*20] Cai R., Yang G., Averbuch-Elor H., Hao Z., Belongie S., Snavely N., Hariharan B.: Learning gradient fields for shape generation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III* (Berlin, Heidelberg, 2020), Springer-Verlag, p. 364–381. 2

[CZ19] Chen Z., Zhang H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5939–5948. 2

[CZK*21] Chen J., Zhang Y., Kang D., Zhe X., Bao L., Jia X., Lu H.: Animatable neural radiance fields from monocular rgb videos. *arXiv preprint arXiv:2106.13629* (2021). 3

[DLZR22] Deng K., Liu A., Zhu J.-Y., Ramanan D.: Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 12882–12891. 2

[DZY*21] Du Y., Zhang Y., Yu H.-X., Tenenbaum J. B., Wu J.: Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), pp. 14304–14314. 2

[EGO*20] Erler P., Guerrero P., Ohrhallinger S., Mitra N. J., Wimmer M.: Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision* (2020), Springer, pp. 108–124. 2

[FKMW*23] Fridovich-Keil S., Meanti G., Warburg F., Recht B., Kanazawa A.: K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241* (2023). 3

[FYW*22] Fang J., Yi T., Wang X., Xie L., Zhang X., Liu W., Niessner M., Tian Q.: Fast dynamic radiance fields with time-aware neural voxels. *arXiv preprint arXiv:2205.15285* (2022). 2, 3

[GBB*22] Greff K., Belletti F., Beyer L., Doersch C., Du Y., Duckworth D., Fleet D. J., Gnanapragasam D., Golemo F., Herrmann C., et al.: Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 3749–3761. 6

[GCS*20] Genova K., Cole F., Sud A., Sarna A., Funkhouser T.: Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 4857–4866. 2

[GKB*22] Guo Y.-C., Kang D., Bao L., He Y., Zhang S.-H.: Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 18409–18418. 10

[GKE*22] Garbin S. J., Kowalski M., Estellers V., Szymanowicz S., Rezaeifar S., Shen J., Johnson M., Valentin J.: Voltemorph: Realtime, controllable and generalisable animation of volumetric representations. *arXiv preprint arXiv:2208.00949* (2022). 2

[GKOM18] Guerrero P., Kleiman Y., Ovsjanikov M., Mitra N. J.: PCPNet: Learning local shape properties from raw point clouds. *Computer Graphics Forum 37*, 2 (2018), 75–85. 5

[GSKH21] Gao C., Saraf A., Kopf J., Huang J.-B.: Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5712–5721. 2, 3

[GTZN21] Gafni G., Thies J., Zollhofer M., Niessner M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 8649–8658. 3

[GYH*20] Gropp A., Yariv L., Haim N., Atzmon M., Lipman Y.: Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*. 2020, pp. 3569–3579. 6

[HHM*22] Huang L., Hodan T., Ma L., Zhang L., Tran L., Twigg C., Wu P.-C., Yuan J., Keskin C., Wang R.: Neural correspondence field for object pose estimation. *arXiv preprint arXiv:2208.00113* (2022). 3

[HMR19] Henzler P., Mitra N., Ritschel T.: Escaping plato's cave: 3d shape from adversarial rendering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 9983–9992. 2

[HWMD14] Handa A., Whelan T., McDonald J., Davison A. J.: A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE international conference on Robotics and automation (ICRA)* (2014), IEEE, pp. 1524–1531. 6

[IAKG20] Ichnowski* J., Avigal* Y., Kerr J., Goldberg K.: Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning (CoRL)* (2020). 10

[IKH*11] Izadi S., Kim D., Hilliges O., Molyneaux D., Newcombe R., Kohli P., Shotton J., Hodges S., Freeman D., Davison A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), pp. 559–568. 1, 2

[JJHZ20] Jiang Y., Ji D., Han Z., Zwicker M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 1251–1261. 2

[JJS*22] Jiang Y., Jiang S., Sun G., Su Z., Guo K., Wu M., Yu J., Xu L.: Neuralhofusion: Neural volumetric rendering under human-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 6155–6165. 3

[JKK*23] Jambon C., Kerbl B., Kopanas G., Diolatzis S., Drettakis G., Leimkühler T.: Nerfshop: Interactive editing of neural radiance fields. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 1 (2023). 2

[KGY*22] Kundu A., Genova K., Yin X., Fathi A., Pantofaru C., Guibas L. J., Tagliasacchi A., Dellaert F., Funkhouser T.: Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 12871–12881. 3

[KJJ*21] Kellnhofer P., Jebe L. C., Jones A., Spicer R., Pulli K., Wetzstein G.: Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 4287–4297. 2

[KMS22] Kobayashi S., Matsumoto E., Sitzmann V.: Decomposing NeRF for editing via feature field distillation. In *Advances in Neural Information Processing Systems* (2022), Oh A. H., Agarwal A., Belgrave D., Cho K., (Eds.). URL: https://openreview.net/forum?id=IJNDyqdRF0m. 2

[KRWM22a] Karnewar A., Ritschel T., Wang O., Mitra N.: 3inGAN: Learning a 3D generative model from images of a self-similar scene. In *Proc. 3D Vision (3DV)* (2022). 2

[KRWM22b] Karnewar A., Ritschel T., Wang O., Mitra N.: Relu fields: The little non-linearity that could. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. 2, 10

[LCM*22] Liu J.-W., Cao Y.-P., Mao W., Zhang W., Zhang D. J., Keppo J., Shan Y., Qie X., Shou M. Z.: Devrf: Fast deformable voxel radiance fields for dynamic scenes. *arXiv preprint arXiv:2205.15723* (2022). 2

[LGZL*20] Liu L., Gu J., Zaw Lin K., Chua T.-S., Theobalt C.: Neural sparse voxel fields. *Advances in Neural Information Processing Systems 33* (2020), 15651–15663. 2

[LHR*21] Liu L., Habermann M., Rudnev V., Sarkar K., Gu J., Theobalt C.: Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–16. 3

[LMR*15] Loper M., Mahmood N., Romero J., Pons-Moll G., Black M. J.: Smpl: A skinned multi-person linear model. *ACM Trans. Graph. 34*, 6 (nov 2015). 1

[LMTL21] Lin C.-H., Ma W.-C., Torralba A., Lucey S.: Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5741–5751. 2

[LNSW21] Li Z., Niklaus S., Snavely N., Wang O.: 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 (CVPR)* (2021), pp. 6498–6508. 2, 3

[LSCL19] Liu S., Saito S., Chen W., Li H.: Learning to infer implicit surfaces without 3d supervision. *Advances in Neural Information Processing Systems 32* (2019). 2

[LSS*21] Lombardi S., Simon T., Schwartz G., Zollhoefer M., Sheikh Y., Saragih J.: Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–13. 2

[LSZ*22] Li T., Slavcheva M., Zollhoefer M., Green S., Lassner C., Kim C., Schmidt T., Lovegrove S., Goesele M., Newcombe R., Lv Z.: Neural 3d video synthesis from multi-view video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 5511–5521. 2

[LTT*21] Li Y., Takehara H., Taketomi T., Zheng B., Niessner M.: 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12706–12716. 6

[LWL*22] Li T., Wen X., Liu Y.-S., Su H., Han Z.: Learning deep implicit functions for 3d shapes with dynamic code clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 12840–12850. 2

[LYS*22] Li F., Yu H., Shugurov I., Busam B., Yang S., Ilic S.: Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. *arXiv preprint arXiv:2203.04802* (2022). 3

[LZP*20] Liu S., Zhang Y., Peng S., Shi B., Pollefeys M., Cui Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 2019–2028. 2

[MAG*22] Morreale L., Aigerman N., Guerrero P., Kim V. G., Mitra N. J.: Neural convolutional surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 19333–19342. 2

[Max95] Max N.: Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 99–108. 2, 3

[MBRS*21] Martin-Brualla R., Radwan N., Sajjadi M. S., Barron J. T., Dosovitskiy A., Duckworth D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 7210–7219. 2, 3, 6

[MESK22] Müller T., Evans A., Schied C., Keller A.: Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989* (2022). 10

[MGB*21] Mehta I., Gharbi M., Barnes C., Shechtman E., Ramamoorthi R., Chandraker M.: Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14214–14223. 2

[MON*19] Mescheder L., Oechsle M., Niemeyer M., Nowozin S., Geiger A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4460–4470. 2

[MST*20] Mildenhall B., Srinivasan P. P., Tancik M., Barron J. T., Ramamoorthi R., Ng R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision* (2020), Springer, pp. 405–421. 2, 3

[MWM*21] Muller N., Wong Y.-S., Mitra N. J., Dai A., Niessner M.: Seeing behind objects for 3d multi-object tracking in rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 6071–6080. 2

[NLD11] Newcombe R. A., Lovegrove S. J., Davison A. J.: Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision* (2011), IEEE, pp. 2320–2327. 1

[NMOG20] Niemeyer M., Mescheder L., Oechsle M., Geiger A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 3504–3515. 2

[NSLH21] NOGUCHI A., SUN X., LIN S., HARADA T.: Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), pp. 5762–5772. 3

[NZIS13] NIESSNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG) 32*, 6 (2013), 1–11. 2

[OMT*21] OST J., MANNAN F., THUEREY N., KNODT J., HEIDE F.: Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 2856–2865. 3

[OPG21] OECHSLE M., PENG S., GEIGER A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5589–5599. 2, 3

[PCPMMN21] PUMAROLA A., CORONA E., PONS-MOLL G., MORENO-NOGUER F.: D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 10318–10327. 2, 3

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 165–174. 2

[PNM*20] PENG S., NIEMEYER M., MESCHEDER L., POLLEFEYS M., GEIGER A.: Convolutional occupancy networks. In *European Conference on Computer Vision* (2020), Springer, pp. 523–540. 2

[PSB*21] PARK K., SINHA U., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., SEITZ S. M., MARTIN-BRUALLA R.: Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5865–5874. 3

[PSH*21] PARK K., SINHA U., HEDMAN P., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., MARTIN-BRUALLA R., SEITZ S. M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228* (2021). 2, 3, 6

[PZX*21] PENG S., ZHANG Y., XU Y., WANG Q., SHUAI Q., BAO H., ZHOU X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 9054–9063. 3

[RA17] RÜNZ M., AGAPITO L.: Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 4471–4478. 2

[RBA18] RÜNZ M., BUFFIER M., AGAPITO L.: Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2018), IEEE, pp. 10–20. 2

[RPLG21] REISER C., PENG S., LIAO Y., GEIGER A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14335–14345. 2

[RPMR13] REN C. Y., PRISACARIU V., MURRAY D., REID I.: Star3d: Simultaneous tracking and reconstruction of 3d objects using rgb-d data. In *2013 IEEE International Conference on Computer Vision (ICCV)* (2013), pp. 1561–1568. 1

[SCL*23] SONG L., CHEN A., LI Z., CHEN Z., CHEN L., YUAN J., XU Y., GEIGER A.: Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics* (2023). 2, 3

[SGF*22] SHUAI Q., GENG C., FANG Q., PENG S., SHEN W., ZHOU X., BAO H.: Novel view synthesis of human interactions from sparse multi-view videos. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–10. 3

[SK16] SALIMANS T., KINGMA D. P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems 29* (2016). 6

[SLOD21] SUCAR E., LIU S., ORTIZ J., DAVISON A. J.: imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 6229–6238. 2, 3, 6, 7, 8

[SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems 33* (2020), 7462–7473. 2

[SS19] STRECKE M., STUCKLER J.: Em-fusion: Dynamic object-level slam with probabilistic data association. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 5865–5874. 2

[SSC22] SUN C., SUN M., CHEN H.-T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 5459–5469. 2

[TD20] TEED Z., DENG J.: Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision* (2020), Springer, pp. 402–419. 5

[TLLV22] TSCHERNEZKI V., LAINA I., LARLUS D., VEDALDI A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494* (2022). 3

[TLV21] TSCHERNEZKI V., LARLUS D., VEDALDI A.: Neuraldiff: Segmenting 3d objects that move in egocentric videos. In *2021 International Conference on 3D Vision (3DV)* (2021), IEEE, pp. 910–919. 3

[TLY*21] TAKIKAWA T., LITALIEN J., YIN K., KREIS K., LOOP C., NOWROUZEZAHRAI D., JACOBSON A., MCGUIRE M., FIDLER S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 11358–11367. 2

[TSM*20] TANCIK M., SRINIVASAN P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHI R., BARRON J., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems 33* (2020), 7537–7547. 2

[TTG*21] TRETSCHK E., TEWARI A., GOLYANIK V., ZOLLHÖFER M., LASSNER C., THEOBALT C.: 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* (2021), pp. 12959–12970. 2

[UFK*22] UEDA I., FUKUHARA Y., KATAOKA H., AIZAWA H., SHISHIDO H., KITAHARA I.: Neural density-distance fields. *arXiv preprint arXiv:2207.14455* (2022). 3

[WCS*22] WENG C.-Y., CURLESS B., SRINIVASAN P. P., BARRON J. T., KEMELMACHER-SHLIZERMAN I.: Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 16210–16220. 3

[WCY23] WANG B., CHEN L., YANG B.: DM-neRF: 3d scene geometry decomposition and manipulation from 2d images. In *The Eleventh International Conference on Learning Representations* (2023). URL: `https://openreview.net/forum?id=C_PRLz8bEJx`. 2

[WLL*21] WANG P., LIU L., LIU Y., THEOBALT C., KOMURA T., WANG W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021). 2, 3, 8, 10

[WLNM21] WONG Y.-S., LI C., NIESSNER M., MITRA N. J.: Rigidfusion: Rgb-d scene reconstruction with rigidly-moving objects. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 511–522. 2, 6

[WLR*21] WEI Y., LIU S., RAO Y., ZHAO W., LU J., ZHOU J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5610–5619. 2

[WZB*19] WANG Q., ZHANG L., BERTINETTO L., HU W., TORR P. H.: Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 1328–1338. 5

[WZL*22] WANG L., ZHANG J., LIU X., ZHAO F., ZHANG Y., ZHANG Y., WU M., YU J., XU L.: Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 13524–13534. 2, 3

[WZT*22] WU T., ZHONG F., TAGLIASACCHI A., COLE F., OZTIRELI C.: D$\hat{2}$ nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *arXiv preprint arXiv:2205.15838* (2022). 2, 3, 10

[XAS21] XU H., ALLDIECK T., SMINCHISESCU C.: H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems 34* (2021), 14955–14966. 3

[XHKK21] XIAN W., HUANG J.-B., KOPF J., KIM C.: Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 9421–9431. 2, 3, 4, 5

[XLT*19] XU B., LI W., TZOUMANIKAS D., BLOESCH M., DAVISON A., LEUTENEGGER S.: Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)* (2019), IEEE, pp. 5231–5237. 2

[YCFB*21] YEN-CHEN L., FLORENCE P., BARRON J. T., RODRIGUEZ A., ISOLA P., LIN T.-Y.: inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021), IEEE, pp. 1323–1330. 3

[YGKL21] YARIV L., GU J., KASTEN Y., LIPMAN Y.: Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems 34* (2021), 4805–4815. 3, 8, 10

[YKM*20] YARIV L., KASTEN Y., MORAN D., GALUN M., ATZMON M., RONEN B., LIPMAN Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems 33* (2020), 2492–2502. 2

[YLSL21] YUAN W., LV Z., SCHMIDT T., LOVEGROVE S.: 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 (CVPR)* (2021), pp. 13144–13152. 2, 3

[YLT*21] YU A., LI R., TANCIK M., LI H., NG R., KANAZAWA A.: Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5752–5761. 2

[YPN*22] YU Z., PENG S., NIEMEYER M., SATTLER T., GEIGER A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665* (2022). 2

[ZKF*23] ZHANG X., KUNDU A., FUNKHOUSER T., GUIBAS L., SU H., GENOVA K.: Nerflets: Local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 8274–8284. 2

[ZNW22] ZHANG B., NIESSNER M., WONKA P.: 3dilg: Irregular latent grids for 3d generative modeling. *arXiv preprint arXiv:2205.13914* (2022). 2

[ZPL*22] ZHU Z., PENG S., LARSSON V., XU W., BAO H., CUI Z., OSWALD M. R., POLLEFEYS M.: Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 12786–12796. 2, 3, 4, 6, 7, 8

[ZSM*21] ZHI S., SUCAR E., MOUTON A., HAUGHTON I., LAIDLOW T., DAVISON A. J.: ilabel: Interactive neural scene labelling. *arXiv preprint arXiv:2111.14637* (2021). 4

[ZX17] ZHANG H., XU F.: Mixedfusion: Real-time reconstruction of an indoor scene with dynamic objects. *IEEE transactions on visualization and computer graphics 24*, 12 (2017), 3137–3146. 2