3D Multi-Object Tracking with Differentiable Pose Estimation

Dominik Schmauser
Technical University of Munich
dominik.schmauser@tum.de

Zeju Qiu
Technical University of Munich
zeju.qiu@tum.de

Norman Müller
Technical University of Munich
norman.mueller@tum.de

Matthias Nießner
Technical University of Munich
niessner@tum.de

Abstract

We propose a novel approach for joint 3D multi-object tracking and reconstruction from RGB-D sequences in indoor environments. To this end, we detect and reconstruct objects in each frame while predicting dense correspondences mappings into a normalized object space. We leverage those correspondences to inform a graph neural network to solve for the optimal, temporally-consistent 7-DoF pose trajectories of all objects. The novelty of our method is two-fold: first, we propose a new graph-based approach for differentiable pose estimation over time to learn optimal pose trajectories; second, we present a joint formulation of reconstruction and pose estimation along the time axis for robust and geometrically consistent multi-object tracking. In order to validate our approach, we introduce a new synthetic dataset comprising 2381 unique indoor sequences with a total of 60k rendered RGB-D images for multi-object tracking with moving objects and camera positions derived from the synthetic 3D-FRONT dataset. We demonstrate that our method improves the accumulated MOTA score for all test sequences by 24.8% over existing state-of-the-art methods. In several ablations on synthetic and real-world sequences, we show that our graph-based, fully end-to-end-learnable approach yields a significant boost in tracking performance.

1 Introduction

Multi-object tracking (MOT) is a key component in many applications such as robot navigation, autonomous driving, or mixed reality. In the outdoor setting, we see significant progress, particularly in the context of LiDAR-based object tracking. In the indoor setting, however, reliable multi-object tracking remains in its infancy. Here, we naturally observe a high level of occlusion, large inter-class variety, and strong appearance changes that severely hamper tracking performance. In addition, we notice that in contrast to the autonomous driving or pedestrian tracking scenarios where large annotated tracking datasets exist, there is no equivalent available for indoor environments.

In the indoor setting, prior works hence often tackle this task by relying on strong 2D/3D detectors followed by an uncoupled data association step. For object matching, several frame-to-frame heuristics or learned-similarity or geometry-based approaches have been proposed. However, as those modules do not inform each other, this often leads to sub-optimal tracking performance.

We introduce a holistic approach for joint pose estimation, 3D reconstruction, and data association over time for reliable object pose tracking to address these shortcomings. We leverage differentiable pose estimation together with a graph neural network for object association in order to obtain temporally consistent 7-DoF object poses (3 rotations, 3 translations and 1 scale). By jointly learning

Preprint. Under review.

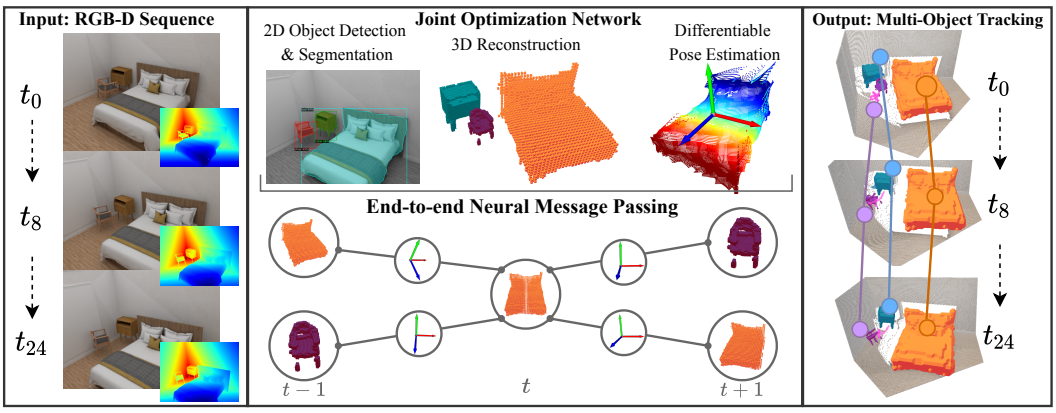

Figure 1: We investigate the task of 3D multi-object tracking using a novel synthetic indoor scene dataset. Our network leverages a 2D detection backbone with additional NOC [29] prediction and 3D reconstruction heads to predict per-object dense correspondences maps and 7-DoF pose parameters. We leverage those correspondences in our neural message passing based, fully end-to-end learnable network to model dependencies between objects over time for consistent multi-object tracking.

to estimate object shapes, we obtain additional feature priors that help to facilitate the association of rigidly-moving objects over time.

In order to train and evaluate MOT in the indoor setting, we introduce MOTFront, a new synthetic dataset consisting of 2381 unique indoor sequences with a total of 60k rendered RGB-D images together with corresponding instance semantics. For each sequence, we leverage scene layouts and 3D assets from 3D-FRONT to generate camera and object trajectories. Based on this data, we conducted a series of ablation studies which show that our holistic approach for differentiable pose estimation, 3D reconstruction and object association provides a significant improvement. Overall, our method improves the accumulated MOTA score by 24.8% over existing state-of-the-art.

In summary, our main contributions are as follows:

- A novel graph-based 3D tracking method with differentiable pose estimation for temporally-consistent object pose trajectories.
- A new formulation for joint object completion and pose estimation over time by inter-frame message passing for improved data association.
- A new synthetic dataset comprising extensive 2D and 3D annotations of indoor scenes with multiple moving objects and camera over time.

2 Related Work

2.1 RGB-D Object Tracking

With the wide availability of consumer-grade RGB-D sensors, several approaches have been proposed to understand dynamic environments by object tracking. Many SLAM-based systems [26, 21, 23, 16] perform instance segmentation, pose estimation, to achieve object tracking. Dynamicfusion [20] introduces the reconstruction and tracking of dynamic, non-rigid scenes in SLAM by decomposing a non-rigidly deforming scene into a rigid canonical space and include moving objects. In addition, several approaches have been dedicated to perform MOT in dynamic scenes. EM-Fusion [24] proposes a probabilistic expectation maximization formulation (EM) to conduct object-level SLAM for data association. Mask-Fusion [22] performs tracking based on optimizing the iterative closest point (ICP) error and photometric cost; however, Mask-Fusion relies on hand-crafted features for deciding non-static objects. MID-Fusion [37], introduced by Xu et al., uses an octree-based method to generate an object-level volumetric map and perform RGB-D camera tracking and object pose estimation with an ICP-algorithm.

In order to overcome reconstruction and object association issues, Müller et al. [19] propose a framework that jointly performs object completion and pose estimation, where objects are associated

in a frame-to-frame fashion levering hand-crafted heuristics for object assignment. In contrast, our method learns optimal object associations over time in an end-to-end fashion, using a neural message passing network while performing fully differentiable pose estimation and 3D reconstruction.

2.2 Multi-object tracking with graphs and transformers

A common approach for MOT is the tracking-by-detection paradigm [3, 33, 38]. Here, objects are usually first localized in each frame by an object detector, followed by associations of proposals in adjacent frames to generate the tracking results. For this, Kalman Filters [32] or similarity measures [35, 1] association metric together with the Hungarian algorithm [15] are leveraged for track association.

In recent years, several graph-based approaches have been introduced to perform data association. Brasó et al. [4] propose a fully differentiable framework based on a message passing network modeling temporal dependencies to perform MOT for outdoor scenes. Yu et al. [39] additionally model spatial context with a second graph network for the spatial domain and incorporate a self-attention mechanism in both graphs for improved context learning. Novel transformer-based MOT frameworks perform multi-object tracking in a frame-to-frame fashion employing the concept of autoregressive track queries [40, 17]. These works perform tracking in a 2D space while ignoring the 3D spatial relations between objects and their 3D geometry. We instead utilize 3D pose and geometry features to model scene configurations over time. For graph-based 3D tracking, Wang et al. [31] propose a framework that jointly optimizes object detection and data association. Additionally, GNN3DMOT [34] introduces a graph neural network using 2D and 3D features for the MOT domain. They show that spatial and temporal interaction of the 2D and 3D object features can improve the tracking performance. Our method has similar motivation as prior graph-based MOT methods, but our differentiable optimization for object poses and geometry facilitates end-to-end learning, thus improving overall tracking performance.

2.3 MOT datasets

Recently, many new datasets have been proposed to facilitate research in the MOT domain, primarily for outdoor applications [28, 13, 18, 5, 25, 10]. MOTS [28] which is based on the outdoor dataset KITTI [13], is the first dataset including annotations for multi-object tracking and additional pixel-level instance mask annotations. The MOTChallenge [18] is a popular benchmark providing multiple datasets for 2D MOT with annotated pedestrians in crowded outdoor scenes. However, these datasets do not contain any 3D annotations, which makes tracking in the 3D domain infeasible. Other popular datasets such as KITTI [13], nuScenes [5] and Waymo Open [25] contain 3D bounding box annotations, but lack pose and instance segmentation labels. The recently published MOTSynth dataset [10] has diverse 3D annotations, yet does not provide the 3D geometry of objects. While existing datasets have been created for outdoor MOT, to the best of our knowledge, a publicly available dataset for indoor 3D MOT currently does not exist. Hence, we believe our new dataset MOTFront, providing complete 2D and 3D annotations will help to drive forward future research in the domain of indoor 3D MOT.

3 MOTFront: Synthetic indoor MOT dataset

Dataset Overview We propose a dynamic indoor MOT dataset MOTFront based on the 3D-FRONT dataset [12]. 3D-FRONT is a large-scale, comprehensive repository containing 18797 rooms diversely furnished by 3D furniture objects. We use Blenderproc [9], a procedural pipeline based on the open-source platform Blender, to generate photo-realistic renderings of the 3D-FRONT scenes. To the best of our knowledge, there is currently no publicly available dataset with extensive 2D and 3D annotations that depicts dynamically moving objects with a moving camera in indoor scenes.

MOTFront provides photo-realistic RGB-D images with their corresponding instance segmentation masks, class labels, 2D & 3D bounding boxes, 3D geometry (voxel grids), 3D poses (NOCs maps) and camera parameters. Our dataset comprises $2,381$ sequences with a total of 60k images. Each sequence contains 25 frames and depicts different types of rooms with up to 5 moving objects belonging to 7 distinct object categories: chair, table, sofa, bed, TV stand, wine cooler, nightstand. The dataset statistics can be found in Table 1. Our dataset comprises diverse object shapes with

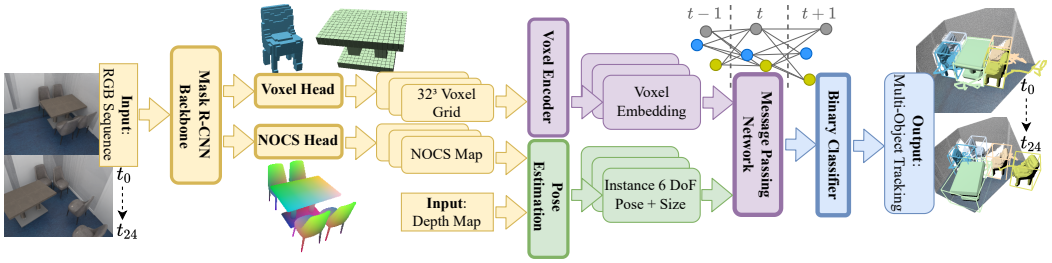

Figure 2: Network architecture overview: a Mask R-CNN backbone detects object instances from an RGB sequence, additionally predicting a 32^3 voxel grid and a NOCs map for each object which is aligned by a pose estimation module using supplementary depth sequence data represented as a depth point cloud, outputting a 7-DoF pose. A 3D convolutional network encodes predicted voxel grids into initial node embeddings. For a consistent object tracking we employ a graph neural network with a consecutive binary classifier, predicting active and non-active graph connections, which is constructed from geometry and pose embeddings and jointly optimizes reconstruction, pose, and object association.

textured backgrounds, unlike data from [8] which is restricted by white floors and walls. We synthetically generate physically plausible camera and object motion to create the desired image sequence with realistically moving objects and moving camera over time.

Data Generation Dynamic data is generated automatically, without any manual labeling or human supervision, and instead relying on sampling techniques and score-based evaluation metrics to guarantee the generation of plausible data sequences. Several approaches [6, 30] propose to map object instances to a Normalized Object Coordinate Space (NOCs) to infer object poses by predicting NOCs maps. Furthermore, we generate ground truth voxel grids in a 32^3 resolution for 3D reconstruction.

Table 1: Dataset statistics of our synthetic indoor scene dataset MOTFront

Num of chairs	4,210
Num of tables	1,820
Num of sofas	2,161
Num of beds	449
Num of tv stands	216
Num of wine coolers	31
Num of nightstands	140
Dataset size (num scenes)	2,381
Avg. num obj. per scene	3.8

We randomly choose rooms that contain at least 5 objects and generate one sequence per room. We randomly choose at least 3 objects (N_{obj}) and sample their location and orientation along with that of the camera 25 times to generate one tracking sequence. The detailed object and camera sampling can be seen in Algorithm 1. The 3D position is denoted as $\boldsymbol{x} \in \mathbb{R}^3$ and the Euler orientation as $\boldsymbol{\theta} \in \mathbb{R}^3$. We first perform object position sampling: we weight the sampling direction with a factor $\sigma(n, d(\boldsymbol{x}, \boldsymbol{x'}))$ (see eq. 1), similar to the repulsive potential [7], which is dependent on the distance from current object $\boldsymbol{x}$ to nearest obstacle $\boldsymbol{x}'$. This will guide the objects from each other and all obstacles away when within some predefined threshold d^* to achieve larger motions. Additionally, we use Bezier interpolation to smooth the object trajectories. For camera pose sampling, we uniformly sample camera orientation angles between a predefined range ϕ. Camera position sampling is also weighted with a factor ${}^i\epsilon$, which gradually increases but is bounded to give the camera more freedom during the sampling. Admissible camera poses should surpass a minimum interest score threshold which is computed as the weighted sum of visible objects in that view. Views capturing moving objects receive a high interest score to guarantee meaningful renderings with enough objects. We restrict the number of maximal tries N_{max} for camera and object sampling to 500.

$$\sigma(n, d(\boldsymbol{x}, \boldsymbol{x'})) = \begin{cases} \frac{1}{2}\sigma_0 \left(\frac{1}{d(\boldsymbol{x}, \boldsymbol{x'})} - \frac{1}{d^*} \right)^2 & \text{if } d(\boldsymbol{x}, \boldsymbol{x}') < d^* \text{ and } n < N_{max} \\ 1, & \text{else} \end{cases} \tag{1}$$

4 Method Overview

From an input RGB-D sequence, our method predicts multi-object tracking of the objects observed in the sequence. Our network architecture (Fig. 2) consists of a Mask R-CNN backbone [14], a 3D

reconstruction network, a pose fitting pipeline and a neural message passing network for a subsequent multi-object tracking. The network backbone takes as input the RGB sequence and performs 2D detection and instance segmentation for each image. Detected objects are then associated using input depth data by a neural message passing network in an end-to-end fashion, which enables joint optimization of object pose and 3D geometry for temporal consistent trajectories.

4.1 2D Object Detection & 3D Reconstruction

We extend a Mask R-CNN backbone [14] with two additional heads: a *voxel head* and a *NOCs head*. The voxel head takes as input the predicted instance image patch I_{RGB} from the box & segmentation head and conducts both 3D object reconstruction and shape completion, outputting a 32^3 per object voxel grid O. The NOCs head processes the instance image patch I_{RGB} in parallel and outputs a Normalized Object Coordinate space (NOCs) patch $I_{\text{NOCs}} \in \mathbb{R}^{3 \times w_{\text{box}} \times h_{\text{box}}}$ [29] with w_{box} and h_{box} as the bounding box dimensions, containing pose information of the predicted object. The voxel and NOCs heads follow a decoder structure and take the same RoI (region of interest) feature embedding, computed by a RoI align operation, to predict voxel grid O and NOCs patch I_{NOCs}, respectively. Our voxel head initially reshapes the RoI embedding $\boldsymbol{e}_{\text{RoI}} \in \mathbb{R}^{14 \times 14}$ into a $n_{\text{channels}} \times 4^3$ embedding and reconstructs a 32^3 voxel grid O using a series of transposed 3D convolution layers with added 3D batch normalization. The NOCs head comprises multiple transposed 2D convolution layers with added 2D batch normalization, predicting from the RoI embedding $\boldsymbol{e}_{\text{RoI}}$ a $28 \times 28 \times 3$ NOCs map which is resized to the respective predicted bounding box size using a RoI align operation into a NOCs patch I_{NOCs}. During inference, we filter object detections utilizing non-maximum suppression between 2D bounding box predictions, as well as two thresholds κ = 0.35 and ν = 0.35, discarding object detections with 2D bounding boxes which have a lower 2D IoU with any ground truth bounding box than κ and objects with a lower objectness score than ν.

Algorithm 1 MOT sampling

Require: $N_{\text{obj}} \geq 3$
Initialize $\{{}^0\boldsymbol{T}_{obj}^k\}_{k=1}^K$, ${}^0\boldsymbol{T}_{cam}$, ${}^0\epsilon$
for $n \leftarrow 1$ to N_{max} **do**
  Draw $\boldsymbol{x}_{\text{obj}} \sim U([-\sigma, \sigma]^3)$
  Draw $\boldsymbol{\theta}_{\text{obj}} \sim U([-\phi_{\text{obj}}, \phi_{\text{obj}}]^3)$
  if collision free **then**
    ${}^{i+1}\boldsymbol{T}_{obj}^k \leftarrow {}^i\boldsymbol{T}_{obj}^k \cdot \boldsymbol{T}(\boldsymbol{x}_{\text{obj}}, \boldsymbol{\theta}_{\text{obj}})$
    for $i \leftarrow 1$ to N_{max} **do**
      ${}^i\epsilon \leftarrow {}^0\epsilon \cdot [1 + \log(i+1)]$
      Draw $\boldsymbol{x}_{\text{cam}} \sim U([-{}^i\epsilon, {}^i\epsilon]^3)$
      Draw $\boldsymbol{\theta}_{\text{cam}} \sim U([-\phi_{\text{cam}}, \phi_{\text{cam}}]^3)$
      if interest score **then**
        ${}^{i+1}\boldsymbol{T}_{cam} \leftarrow {}^i\boldsymbol{T}_{cam} \cdot \boldsymbol{T}(\boldsymbol{x}_{\text{cam}}, \boldsymbol{\theta}_{\text{cam}})$

4.2 Differentiable Pose Estimation

A pose estimation module utilizes the predicted NOCs patch I_{NOCs}, the depth map patch I_{Depth} corresponding to the detected object, and camera intrinsics to infer frame-wise size c^*, location $\boldsymbol{t}^*$, and rotation $\boldsymbol{R}^*$ in a camera coordinate space for each object. Therefore, we backproject depth patch I_{Depth} and NOCs patch I_{NOCs} to point clouds $\boldsymbol{P_o}$ and $\boldsymbol{P_n}$. We perform statistical outlier removal on both point clouds, based on the distance to neighboring points, using a RANSAC outlier removal algorithm [11]. This enables removing erroneous residuals which could potentially have a negative effect on the alignment. Finally, the Umeyama algorithm [27] is employed to find the optimal 7-dimensional rigid transformation to align both cleaned point clouds:

$$c^*, \boldsymbol{R}^*, \boldsymbol{t}^* := \underset{c \in \mathbb{R}^+, \boldsymbol{R} \in SO_3, \boldsymbol{t} \in \mathbb{R}^3}{\operatorname{argmin}} \|\boldsymbol{P_o} - (c\boldsymbol{R} \cdot \boldsymbol{P_n} + \boldsymbol{t})\|. \tag{2}$$

4.3 Neural Message Passing and Tracking

We define a bidirectional graph neural network connecting object detections in consecutive frames within a window of 5 neighboring frames. A window size of 5 was selected since it enables a large receptive field for each graph node in the temporal domain to derive better features by having temporal context between frames. We initialize each graph node from the predicted object geometry and edge embeddings with encoded relative pose features. After a number of message passing steps $n_{\text{mp}} = 4$, each edge embedding comprises information from neighboring nodes ensuring a temporal-context understanding. The final edge embeddings are classified by a binary classifier into active and non-active graph connections to predict unique tracklets. We assign ground truth instance

ids to object detections by finding the maximum 3D IoU between a predicted bounding box with all possible ground truth bounding boxes in a frame and selecting its respective instance id. We discard graph connections between object detections which have a 3D IoU lower than a threshold $\tau = 0.05$ for all ground truth bounding boxes. Object pairs with matching instance ids are assigned as positive (active) training pairs and objects pairs with distinct instance ids are assigned as negative (non-active) training pairs.

Pose Embedding. An edge of our message passing network consists of a relative pose embedding $\boldsymbol{e}_{ij} \in \mathbb{R}^{12}$ between two connected nodes n_i and n_j. The initial edge feature is computed by an MLP $\mathcal{N}_{\text{edge-enc}}$, encoding a concatenated feature vector with relative translation $\boldsymbol{t} = (x, y, z) \in \mathbb{R}^3$, relative rotation as euler representation $\boldsymbol{R} = (\alpha, \beta, \gamma) \in \mathbb{R}^3$, relative scale c and relative time step s.

$$\boldsymbol{e}_{ij} = \mathcal{N}_{\text{edge-enc}}([\boldsymbol{t}_j - \boldsymbol{t}_i, \boldsymbol{R}_j - \boldsymbol{R}_i, \log(\frac{c_j}{c_i}), s_j - s_i]) \tag{3}$$

Geometry Embedding. A node feature of our message passing network consists of a shape embedding $\boldsymbol{a}_i \in \mathbb{R}^{16}$, encoded by a 3D convolutional network $\mathcal{N}_{\text{net-conv3D}}$ from the predicted 32^3 object voxel grid O outputted by the voxel head. The voxel encoder network employs a series of 3D convolutions, followed by a flatten operation with two consecutive affine layers and leaky ReLU as non-linearities.

4.4 Training and Inference

We train our end-to-end approach on a single RTX A4000 with a batch size of 4. We first independently train our object detection, 3D reconstruction and pose estimation pipeline for 60 epochs with a learning rate of 0.008, Adam optimizer and L2-regularization of 0.0005 to ensure stable object detections with accurate geometry and pose predictions. Additionally, we pre-train the tracking pipeline for 40 epochs with a learning rate of 0.001, Adam optimizer, and a L2-regularization of 0.001, using fixed pose and geometry features. Finally, our network is trained in an end-to-end fashion for 20 more epochs to jointly optimize object detections, 3D reconstructions, 7-DoF poses, and neural message passing to achieve the best performance.

We guide the model to extract per-frame object information by the loss $\mathcal{L}_{\text{obj}}$ which we define as a weighted sum of the detection loss $\mathcal{L}_{\text{det}}$ proposed by Mask R-CNN [14], a reconstruction loss $\mathcal{L}_{\text{rec}}$ and a NOCs loss $\mathcal{L}_{\text{noc}}$ for correspondence matching. The reconstruction loss $\mathcal{L}_{\text{rec}}$ is defined by a balanced binary cross-entropy loss (BCE), balancing occupied O_{occ} and non-occupied voxels O_{free} for a larger weighting of occupied areas utilizing an object dependent weighting w_{occ}. The NOCs loss $\mathcal{L}_{\text{noc}}$ is a symmetrical smooth-L1 loss which considers object symmetries for the predicted object class $pred_{cls}$ *table* by choosing the minimal loss between ground truth NOCs patch $I^{\text{gt}}_{\text{NOCs}}$ and predicted NOCs patch I^{i}_{NOCs} for all possible target rotations $i \in (0°, 180°)$.

$$\mathcal{L}_{\text{noc}} = \begin{cases} \min_{i=[0°,180°]} L1_{\text{smooth}}(I^{i}_{\text{NOCs}}, I^{\text{gt}}_{\text{NOCs}}) & \text{if } pred_{cls} == table \\ L1_{\text{smooth}}(I_{\text{NOCs}}, I^{\text{gt}}_{\text{NOCs}}) & \text{else} \end{cases}$$

$$\mathcal{L}_{\text{rec}} = w_{\text{occ}} \cdot \text{BCE}(O_{\text{occ}}, O^{t}_{\text{occ}}) + \text{BCE}(O_{\text{free}}, O^{t}_{\text{free}})$$

$$\mathcal{L}_{\text{obj}} = \mathcal{L}_{\text{det}} + 3 \cdot \mathcal{L}_{\text{noc}} + 0.75 \cdot \mathcal{L}_{\text{rec}}$$

For our tracking pipeline, we employ a binary cross-entropy loss with a weighting factor w_{act} to account for the high imbalance between active graph connections $\boldsymbol{e}_{\text{act}}$ (GT associations) and inactive connections $\boldsymbol{e}_{\text{non-act}}$. For a final multi-object tracking across a sequence, we associate object detections by connecting active edges of the graph to trajectories. Nodes with no prior connections create a new trajectory if their instance id does not already exist in any other trajectory. In each time step, we extend tracklets according to the predicted associations. In case of non-unique assignment, we select the closest detection in terms of center distance.

$$\mathcal{L}_{\text{track}} = w_{\text{act}} \cdot \text{BCE}(\boldsymbol{e}_{\text{act}}, \boldsymbol{e}^{t}_{\text{act}}) + \text{BCE}(\boldsymbol{e}_{\text{non-act}}, \boldsymbol{e}^{t}_{\text{non-act}})$$

Evaluation metrics. To evaluate our dynamic object tracking, we adopt the Multiple Object Tracking Accuracy metric (MOTA) [2]:

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t gt_t}. \tag{4}$$

Table 2: Evaluation of MOTA, F1, Precision, and Recall on our MOTFront dataset. We see that our end-to-end learnable approach outperforms a F2F-MaskRCNN baseline and current SOTA (SbO) [19]. Moreover, ablations show that joint optimization of the 7-DoF pose, 3D geometry and object associations over time via a message passing network improves tracking performance.

	m↓	fp↓	mme ↓	F1 ↑	Precison ↑	Recall ↑	MOTA(%)↑
F2F-MaskRCNN	13794	6107	645	0.721	0.795	0.662	46.2
SbO [19]	12949	6400	802	0.724	0.777	0.677	46.7
Ours (no geometry)	10240	1749	58	0.824	0.928	0.747	68.5
Ours (no joint opt.)	10025	1820	59	0.828	0.926	0.752	68.8
Ours (no graph)	11068	**1423**	**47**	0.824	**0.940**	0.734	67.2
Ours	**8984**	1873	58	**0.841**	0.927	**0.770**	**71.5**

where m_t, fp_t, mme_t, gt_t are number of missed targets, false positives, identity switches and ground truth objects at time t. A match is considered positive if its L2 distance to the ground truth center is less than 40cm. We report the accumulated MOTA over all test sequences.

5 Results

5.1 Quantitative Results

We compare our method against the current SOTA approach for indoor MOT, Seeing behind objects (SbO) [19], which performs object detection directly in 3D and tracking via heuristic-based frame-wise matching. Additionally, we evaluate against a baseline F2F-MaskRCNN approach which conducts a frame-to-frame tracking based on point cloud matching by ICP with 2D detections from the same pretrained Mask R-CNN backbone as ours. Table 2 depicts quantitative results on our MOTFront test set of 398 sequences, evaluating F1, Precision, Recall and MOTA with the number of misses, false positives and mismatches. Our graph-based, end-to-end-learnable approach uses geometry and relative pose features between connected objects, achieving the best performance on all evaluation metrics, outperforming baselines in overall MOTA by 24.8% and F1-score by 0.117. We refer to the supplemental material for additional class-specific tracking evaluations.

5.2 Qualitative Results

We further compare our approach qualitatively against the baselines on our synthetic indoor scene dataset MOTFront in Figure 4 as well as on real-world office sequences from [19] in Figure 3. Our approach is able to estimate accurate trajectories and shape reconstructions, even for heavily occluded objects such as the chairs in Figure 4. As the baselines are not optimized for temporal consistency, we observe more tracking failures in comparison to our method. Moreover, our approach yields higher reconstruction quality compared to [19], which does not optimize geometry over time. F2F Mask-RCNN reconstructs and tracks solely based on segmented object surfaces, often producing inaccurate geometry and pose estimates. Additionally, our method achieves more accurate reconstructions and more precise pose estimation over time on real-world sequences, as shown in Figure 3.

5.3 Ablations

What is the effect of graph-based 3D tracking with differentiable pose estimation? When comparing our association step with a graph neural network (Ours) against an approach with a L2-distance heuristic (no graph), we obtain an increase in MOTA score by 4.3% and a reduction of misses by 18.9% (refer Table 2). This confirms that our end-to-end trained, graph-based network can better learn consistent pose trajectories over time, enabling reasoning for trajectory matching considering an enlarged receptive field over multiple frames in contrast to a frame-by-frame heuristic.

Does end-to-end joint object completion and pose estimation over time improve tracking? We further analyze the effect of excluding geometry features in our tracking pipeline (no geometry). This results in a notable decrease in MOTA score by 3% and recall by 2.3% (refer Table 2). We conclude that joint reconstruction and completion of the object geometry enables a more robust and geometrically more stable tracking. In particular, the effects of frequently occurring object

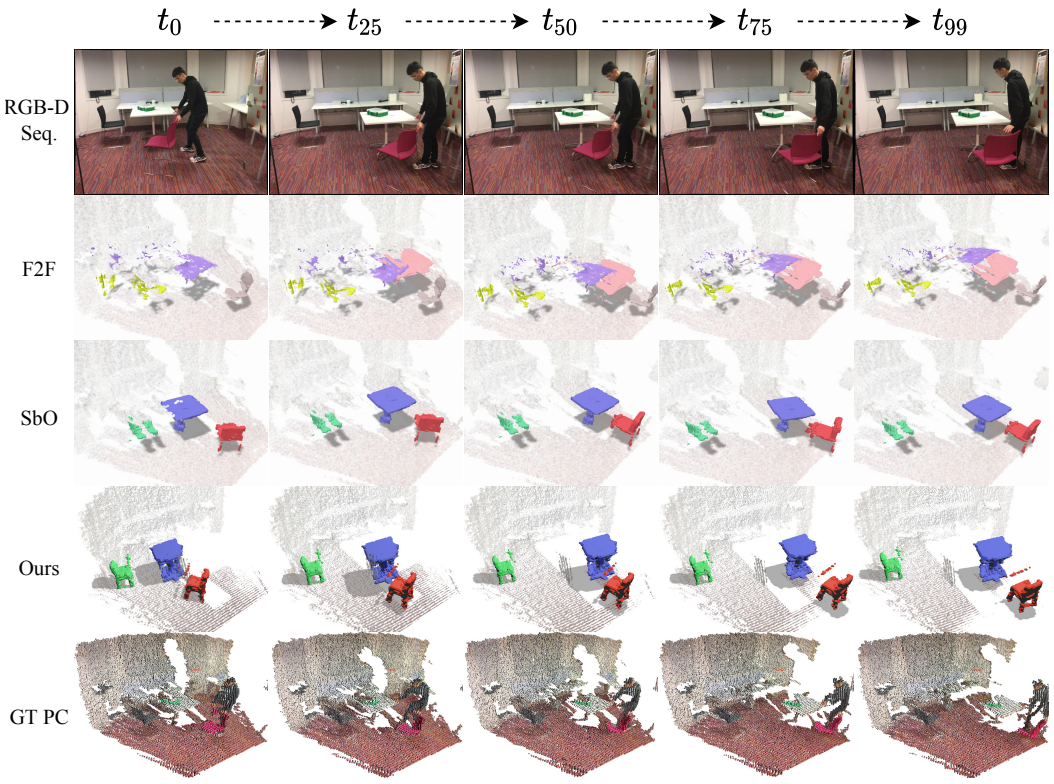

Figure 3: Qualitative comparison with SOTA method Seeing behind objects [19] and a F2F-MaskRCNN baseline for a scene from a real-world office dataset.

occlusions can be alleviated, leading to a lower number of misses in trajectories. We further analyze the effect of training our pipeline end-to-end versus separate optimization (no joint opt.): Our jointly optimized approach improves MOTA by 2.7% and achieves more reliable object detections and pose predictions. This shows that updating the object-level feature extraction steps based on the data association improves the final tracking.

5.4 Limitations

While our approach presents a promising step to robust multi-object tracking, several limitations remain. As our reconstruction is limited by the dense voxel grid resolution, fine-scale details cannot be captured. Additionally, one could consider an appearance-based object representation that also models textures. This could further improve data association and lead to even more consistent object tracking by also optimizing for the appearance over time.

5.5 Societal impact

This work proposes a method for multi-object tracking in indoor scenes. It can benefit XR applications and service robots to enable a better understanding of the dynamic environment. By joint reconstruction of the moving objects, it can enable 3D navigation and interaction with the tracked objects (like grasping, obstacle avoidance or digital replication of indoor scenes). For real-world applications, it requires careful consideration in terms of personal data privacy and potential bias towards certain object instances introduced by the training data.

6 Conclusion

We have introduced a new method for 3D multi-object tracking in RGB-D indoor scenes. By employing a graph-based, end-to-end-learnable network with differentiable pose estimation and joint reconstruction, our method can predict robust object trajectories over time. Experiments demonstrate

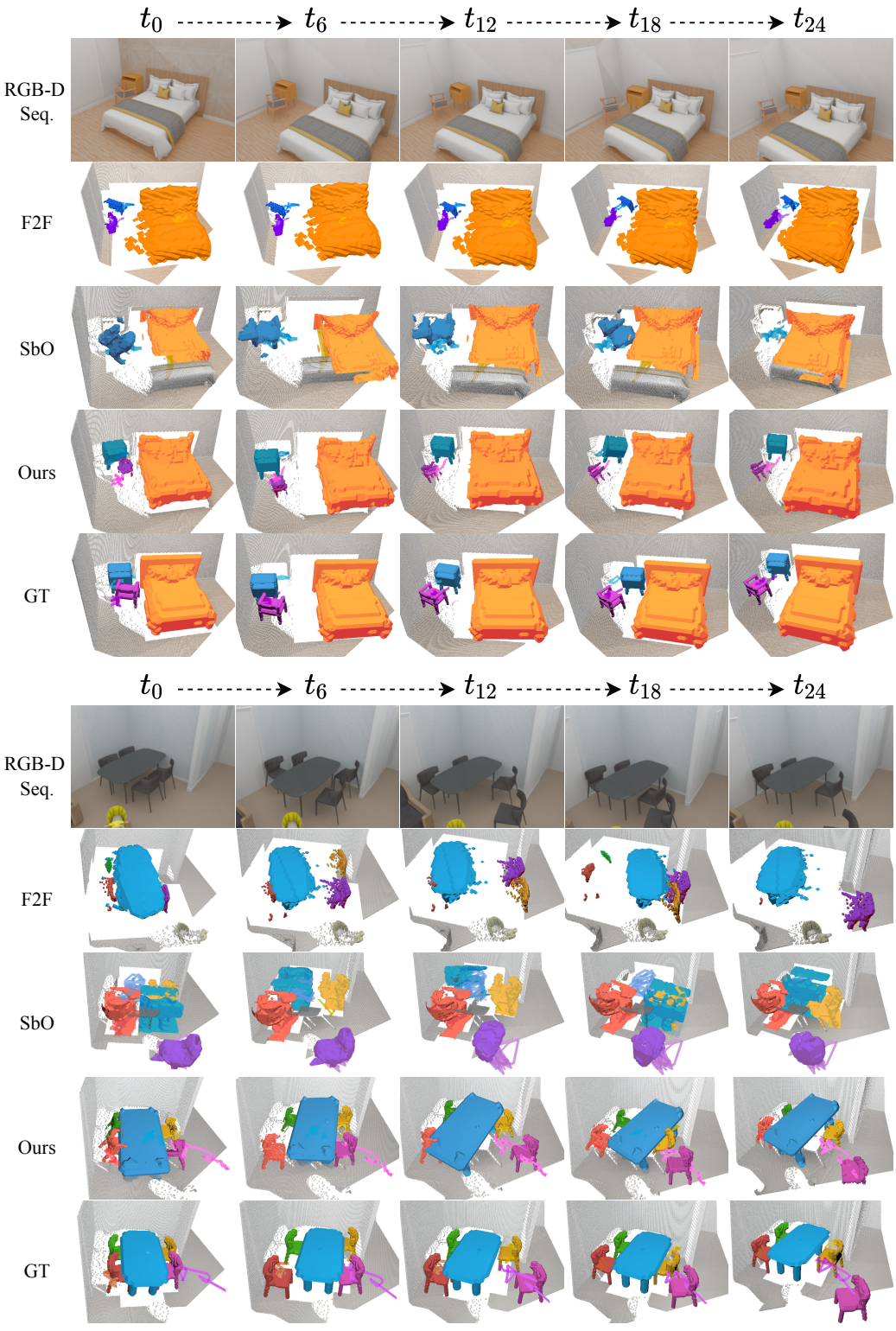

Figure 4: Qualitative comparison with Seeing behind objects (SbO) [19] and a F2F-MaskRCNN baseline on our synthetic MOTFront dataset. By jointly optimizing features over time, our approach predicts more consistent tracks for objects and achieves improved geometry completion and pose estimation accuracy. Object color encodings and line segments show instance id and estimated trajectories, respectively.

a 24.8% improvement in MOTA score over existing SOTA alternatives. In a series of ablations, we conclude that by learning to optimize object poses and shapes over time, our method achieves temporally and spatially plausible trajectories. To train and evaluate our holistic approach, we introduce a novel synthetic MOT dataset MOTFront, with extensive 2D & 3D annotations, which we hope will facilitate MOT research in the indoor setting. Overall, we believe our method is an important stepping stone for tracking and reconstruction of indoor environments.

7 Acknowledgements

This project is funded by the TUM Institute of Advanced Studies (TUM-IAS), the ERC Starting Grant Scan2CAD (804724), and the German Research Foundation (DFG) Grant Making Machine Learning on Static and Dynamic 3D Data Practical.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation, 2020.

[7] Howie Choset, Kevin M Lynch, Seth Hutchinson, George A Kantor, and Wolfram Burgard. *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

[8] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[10] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10849–10859, 2021.

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[12] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[15] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[16] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 international conference on 3D vision (3DV)*, pages 32–41. IEEE, 2018.

[17] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[18] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[19] Norman Muller, Yu-Shiang Wong, Niloy J Mitra, Angela Dai, and Matthias Nießner. 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*, pages 6071–6080, 2021.

[20] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[21] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017.

[22] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018.

[23] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[24] Michael Strecke and Jorg Stuckler. Em-fusion: Dynamic object-level slam with probabilistic data association. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2019.

[25] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[26] Keisuke Tateno, Federico Tombari, and Nassir Navab. When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2295–2302. IEEE, 2016.

[27] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[28] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019.

[29] He Wang, Srinath Sridhar, Jingwei Huang andbernardin2008evaluating Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. *CoRR*, abs/1901.02970, 2019.

[30] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.

[31] Yongxin Wang, Xinshuo Weng, and Kris Kitani. Joint detection and multi-object tracking with graph neural networks. *CoRR*, abs/2006.13164, 2020.

[32] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[33] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *CoRR*, abs/1907.03961, 2019.

[34] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris Kitani. GNN3DMOT: graph neural network for 3d multi-object tracking with multi-feature learning. *CoRR*, abs/2006.07327, 2020.

[35] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.

[36] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[37] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.

[38] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[39] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.

[40] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. *arXiv preprint arXiv:2105.03247*, 2021.

Appendix

In this appendix, we provide further details about our proposed method, additional quantitative and qualitative results and further information on our dataset MOTFront.

A Details on MOTFront

Our dataset MOTFront is created based on assets and scene layouts of the 3D-Front dataset [12]. The dataset is available at: `http://tiny.cc/MOTFront`.

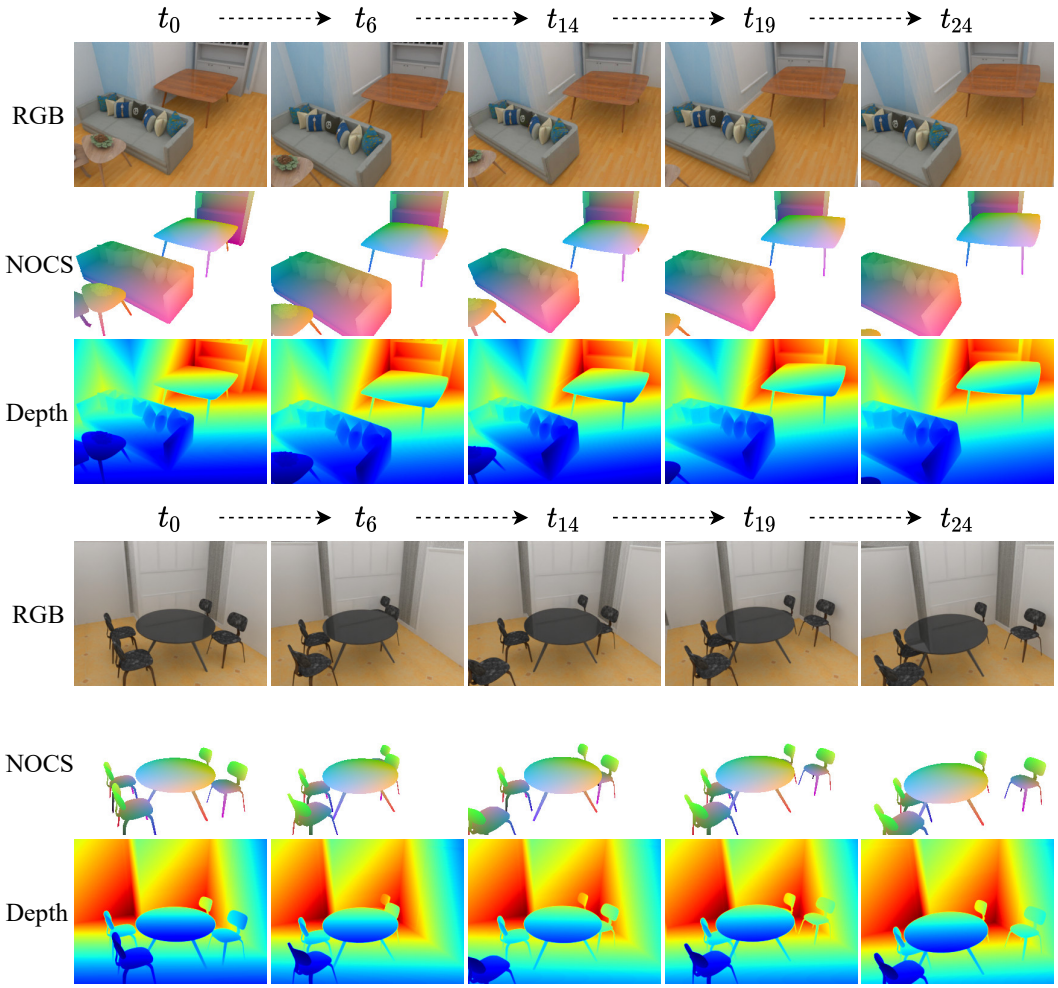

Figure 5: Additional sample sequences of our dataset MOTFront. The top row shows RGB images, the center row shows NOCs map images and the bottom row depicts the according depth map.

In Figure 5, two sample dataset sequences are depicted showing objects from different categories and distinct appearances. In general, our MOTFront dataset is very versatile, depicting different types of rooms (dining room, living room, bedroom) decorated with unique types of furniture. Our object and camera pose sampling algorithm generates dynamically moving objects and cameras. Some objects are heavily occluded, e.g. both chairs in the bottom dataset sequence, underlying the difficulty of 3D indoor tracking. For each scene, our dataset provides an *annotation file* with 2D and 3D annotations, 25 *RGB images*, 25 *depth map renderings*, 25 *NOCs map renderings* and the ground truth 3D geometry for each object.

- *Annotation file (per scene)*: Contains GT segmentation mask, class label, object location, rotation, scale, 2D & 3D bounding boxes
- *RGB images*: Photo-realistic renderings of an indoor scene
- *Depth map*: Distance from the image plane
- *NOCS images*: Object orientation
- *Voxelized objects*: 3D object occupancy masks

B Additional quantitative results

Table 3 shows a class-wise comparison of 3D IoU scores between Seeing behind objects [19] and our approach. We compute the IoU score between the completed ground truth voxel grid and the predicted voxel grid for each object. By informing our shape feature extraction steps from the resulting data associations over time, our end-to-end learnable approach yields more fine-grained reconstructions in comparison to [19] which is reflected in higher IoU scores across all object classes and an increased overall 3D IoU score by 22.7%.

Table 3: Class-wise comparison of 3D IoU scores between Seeing behind objects [19] and our method, calculated from completed ground truth and predicted object voxel grid.

3D IoU(%)	chair	table	sofa	bed	tv-stand	cooler	night-stand	overall
Seeing behind objects [19]	35.3	22.2	30.1	22.1	23.6	23.0	17.2	29.0
Ours	**53.3**	**39.2**	**60.1**	**47.5**	**39.8**	**40.1**	**29.0**	**51.7**

Furthermore, we depict overall and class-wise MOTA scores in comparison with our baselines in Table 3. Our graph-based end-to-end learnable method achieves the highest MOTA scores for 3 out of the 7 object classes. Interestingly, our method performs best in the classes chair and table, the two largest classes with the most occurrences and the highest number of different shapes. Especially, for the most challenging object class chair which often occurs in scenes with multiple chairs located very close to each other, e.g. as shown in Figure 5, our method has an increased MOTA score by 19.6% in comparison to Seeing behind objects [19].

Table 4: Classwise MOTA score evaluation on our dataset MOTFront.

MOTA(%)↑	chair	table	sofa	bed	tv-stand	cooler	night stand	overall
F2F-MaskRCNN	37.7	34.9	41.3	63.3	53.6	48.9	62.6	46.2
Seeing behind objects [19]	56.8	33.3	39.5	18.9	52.1	42.2	36.2	46.7
Ours (no pose)	57.6	52.6	55.3	72.7	54.3	50.6	28.8	58.3
Ours (no geometry)	71.8	58.3	59.3	77.7	56.9	**52.3**	49.8	68.5
Ours (no joint opt.)	72.4	**58.4**	59.3	**77.8**	57.3	**52.3**	**50.1**	68.8
Ours (no graph)	68.4	56.4	**63.8**	73.2	56.9	47.7	47.4	67.2
Ours	**75.4**	**58.4**	62.1	74.3	**61.1**	39.2	50.0	**71.5**

C Additional qualitative results

In Figure 6, we further show qualitative results depicting a living room scene with 3 moving objects. Our approach is able to predict detailed object shapes and robust pose trajectories for the whole sequence while F2F-MaskRCNN and Seeing behind objects have inconsistent object poses, missing detections and inaccurate 3D reconstructions.

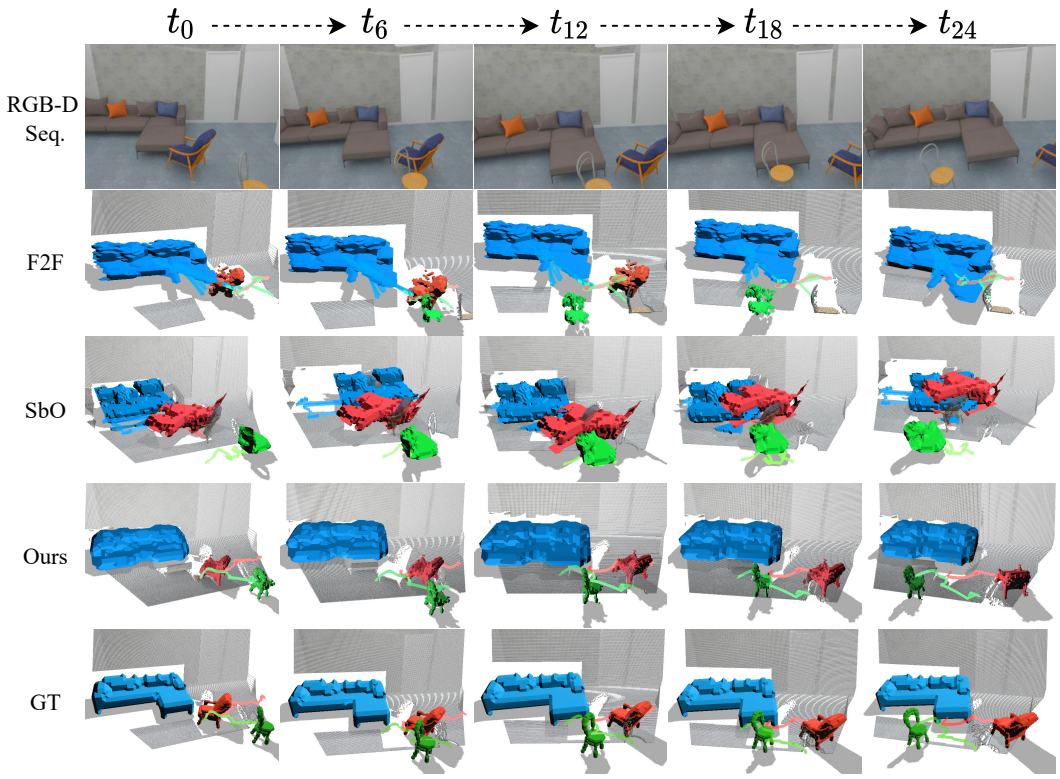

Figure 6: Further qualitative comparison with Seeing behind objects (SbO) [19] and a F2F-MaskRCNN baseline on our synthetic MOTFront dataset. Object color encodings and line segments show instance id and estimated trajectories, respectively.

D Implementation details

D.1 Network Details

In Figure 7, we detail the configuration of each network component. Besides the default Mask-RCNN backbone, we employ two more network heads NOCsHead and VoxelHead, and 4 additional networks: Voxel Encoder, Graph Network, Edge Encoder, and Edge Classifier. We provide the layer parameters as (input channels, output channels, kernel size, stride) and (input features, output features).

D.2 Pose estimation

We employ two outlier removal steps to clean the predicted NOCs point cloud $\boldsymbol{P_n}$ and back-projected depth point cloud $\boldsymbol{P_o}$: first, a statistical outlier removal algorithm considers for each point a number of neighboring points $n_{\text{nbr}} = 20$, calculates the average distance for a given point and removes all points which are exceeding a standard deviation threshold $std_{\text{nbr}} = 2$. Second, the RANSAC outlier removal algorithm [11] selects an optimal set of corresponding points from the source and target point cloud derived from a minimal distance score utilizing linear least squares regression.

As a final step in our pose estimation pipeline, we transform the differentially optimized transformation matrix ${}^{\text{cam}}\boldsymbol{T}_{\text{pose}} \in (c^*, \boldsymbol{t}^*, \boldsymbol{R}^*)$ from camera frame by multiplication with camera extrinsics into a uniform world coordinate space ${}^{\text{world}}\boldsymbol{T}_{\text{pose}}$ to ensure comparability between input frames of a sequence.

D.3 Neural Message Passing

Information of connected nodes is propagated by a series of message passing steps which is divided into two updates. First, a node to edge update is performed by a MLP taking as input the previous

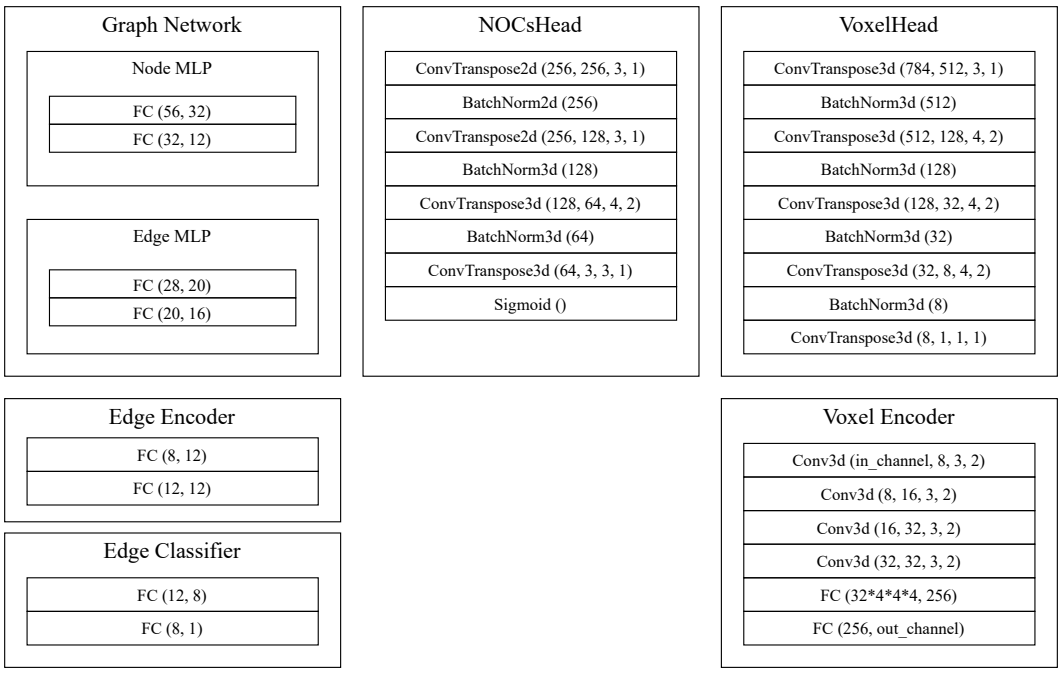

Figure 7: Network architecture specification.

edge embedding and the two attached nodes. At a second step an edge to node update is conducted by a second MLP. Each MLP input is defined by a message which aggregates all neighboring edge embeddings by the permutation invariant *mean* operation and the previous node embedding. After a number of message passing steps $n_{\text{mp}} = 4$ each edge embedding comprises information from neighboring nodes ensuring a temporal-context understanding.

E Code and Data

Our dataset MOTFront is created based on the 3D-Front dataset [12]. We utilize code from the official implementations from Detectron2 [36] to train the Mask R-CNN backbone [14] and BlenderProc [9] to render images.

F Licenses

The 3D-Front dataset is distributed under the CC BY-NC-SA 4.0 license, we release MOTFront under the same licence and note that all rights remain with the owners of 3D-Front. DLR-RM/BlenderProc is licensed under the GNU General Public license v3.0. Detectron2 is released under the Apache 2.0 license.