IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, May 2024.

MOTPose: Multi-object 6D Pose Estimation for Dynamic Video Sequences using Attention-based Temporal Fusion

Arul Selvam Periyasamy and Sven Behnke

***Abstract*— Cluttered bin-picking environments are challenging for pose estimation models. Despite the impressive progress enabled by deep learning, single-view RGB pose estimation models perform poorly in cluttered dynamic environments. Imbuing the rich temporal information contained in the video of scenes has the potential to enhance models' ability to deal with the adverse effects of occlusion and the dynamic nature of the environments. Moreover, joint object detection and pose estimation models are better suited to leverage the co-dependent nature of the tasks for improving the accuracy of both tasks. To this end, we propose attention-based temporal fusion for multi-object 6D pose estimation that accumulates information across multiple frames of a video sequence. Our MOTPose method takes a sequence of images as input and performs joint object detection and pose estimation for all objects in one forward pass. It learns to aggregate both object embeddings and object parameters over multiple time steps using cross-attention-based fusion modules. We evaluate our method on the physically-realistic cluttered bin-picking dataset SynPick and the YCB-Video dataset and demonstrate improved pose estimation accuracy as well as better object detection accuracy.**

I. Introduction

Object detection is the task of localizing instances of object categories in images—typically by predicting bounding box parameters. 6D pose estimation aims at predicting the position and orientation of objects in the sensor coordinate system. Both tasks are essential for many autonomous robots and a prerequisite for object manipulation.

Although single-view pose estimation models have made significant progress in recent years, they face difficulties in cluttered environments [1] hampered by occlusions, reflective surfaces, transparency, and other challenges. One way to address these challenges is to utilize a sequence of images of the scene instead of a single image. In a video sequence, image features and object attributes evolve smoothly over time. Models can benefit from imbuing image features and predictions from the previous frames while processing the current frame. Also, enforcing temporal consistency of the image features and pose predictions from consecutive frames can facilitate efficient learning and better accuracy. Despite the apparent advantages of temporal processing, the popularity of single-view pose estimation methods can be attributed to the complexity, computation, and memory overhead of video pose estimation methods. Furthermore, CNN-based models for video processing often utilize 3D convolutions, which need more parameters and are slow compared to their 2D counterparts.

All authors are with the Autonomous Intelligent Systems group, Computer Science Institute VI – Intelligent Systems and Robotics – and the Center for Robotics and the Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, Germany; periyasa@ais.uni-bonn.de

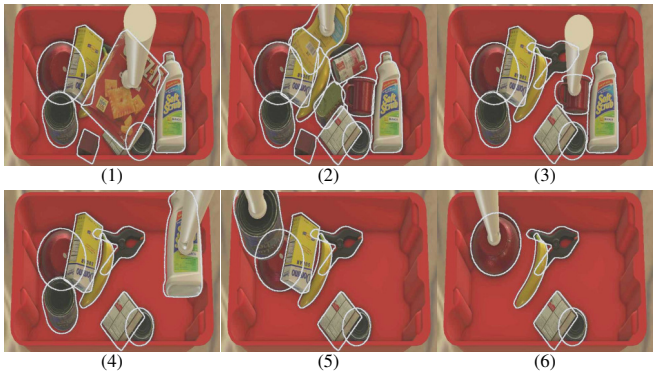


Fig. 1. Multi-object pose predictions for a cluttered bin-picking scene from the SynPick dataset (Untargeted-pick, Sequence 38). MOTPose jointly detects and estimates 6D pose for all objects in the scene in a single step using a vision-transformer model by fusing temporal information across multiple frames. Predicted object poses are visualized by contours.

Lately, the multi-head attention-based transformer architecture, which was initially proposed for natural language processing tasks, has shown tremendous capabilities in modeling long-term dependencies in many domains like audio, image, video, etc. [2]–[6]. Vision transformer architectures also enable single-stage models that jointly perform object detection and pose estimation for all objects in the scene in one forward pass [7], [8]. This ability is handy when dealing with highly cluttered bin-picking scenarios (see Fig. 1). In this work, we propose a vision transformer model for multi-object 6D pose estimation from monocular video sequences. The core component of the proposed MOTPose method is a cross-attention-based temporal fusion mechanism that fuses features from multiple past frames while processing the current frame. We use the stacked object embeddings from the past time steps as key and value in the cross-attention computation while the object embeddings from the current time step serve as query. To counter the permutation-invariant nature of the attention mechanism in the temporal fusion modules, we utilize relative frame encoding (RFE).

Our contributions include:

- a multi-object pose estimation model for dynamic video sequences,
- a method for cross-attention-based temporal fusion of object embeddings and object-specific outputs over multiple frames,
- SynPick-Ext, an extended version of the physically-realistic dataset SynPick consisting of 300 additional video sequences for each action split, and

- quantitative evaluation of the joint object detection and pose estimation task on SynPick, and competitive results on YCB-Video while being lighter and faster than other methods.

II. Related Work

1) Monocular Pose Estimation: Object pose estimation from RGB images has been a long-standing problem in computer vision. The traditional methods before the advent of deep learning include template-based methods [9], [10] and feature-based methods [11]–[13]. Modern deep-learning-based approaches include direct methods that regress the 6D pose parameters given the input RGB image [14]–[17], keypoint-based methods that predict the pixel coordinates of 3D keypoints first and then use the *perspective-n-points* (P*n*P) algorithm to recover 6D pose [18]–[22], and refinement-based methods. The latter iteratively refine an initial pose estimate using either the *render-and-compare* framework [23]–[27] or optical flow [28], [29]. Most monocular pose estimation methods are multi-staged. The standard pipeline involves object detection and/or semantic segmentation, target object crop extraction, and pose estimation from the extracted crop. To enable end-to-end trainable multi-staged models, specialized operations like *non-maximum suppression* (NMS), *region-of-interest pooling* (ROI), or *anchor boxes* are employed. Notable single-stage methods include [20], [30], [31]. Our proposed MOTPose method also incorporates single-stage design elements in its architecture.

2) Pose Estimation as Set Prediction: In recent years, vision transformer architectures, that formulate computer vision tasks like object detection, instance segmentation, and pose estimation as a set prediction problem, are achieving impressive results. Carion *et al.* [32] introduced DETR, the pioneering work in this new class of methods. Several methods extended DETR for multi-object pose estimation [7], [8], [33]. Following these methods, the proposed MOTPose model formulates multi-object pose estimation from video sequences as a set prediction problem.

3) 6D Pose Tracking: Many of the early works for 6D pose tracking were based on particle filtering [34]–[36], but the performance of particle filters heavily depends on the accuracy of the observation model. Deng *et al.* [37] introduced PoseRBPF utilizing a CNN-based observation model in the particle filtering framework. Wen *et al.* [38] introduced *se*(3)-TrackNet, which achieved state-of-the-art-results in object pose tracking from RGB-D images. In contrast to *se*(3)-TrackNet, our MOTPose method only needs RGB input and can estimate 6D pose for all objects in the input images in one stage.

4) Multi-Object Tracking: Multi-object tracking aims at tracking 2D bounding boxes of the target instances in a given video sequence. The task is often challenging, due to the presence of multiple instances of the same category. To address the problem of matching detections and tracked objects, sophisticated matching strategies were proposed [39]–[41]. In this work, we focus mainly on improving pose estimation accuracy by fusing information over multiple time steps. Thus, instead of focusing on the tracking metrics, we emphasize the standard pose estimation metrics—ADD-S and ADD(-S)—discussed in Section IV-B.

5) Tracking-by-Attention in DETR-Like Models: Recently, Meinhardt *et al.* [42] proposed TrackFormer, by introducing the *tracking-by-attention* framework in a DETR-like architecture. Their key idea is to use object embeddings from time step t as object queries in time step t+1. Propagating object embeddings over multiple time steps enables tracking the object over a long video sequence. State-of-the-art methods for multi-object tracking utilizing the tracking-by-attention framework include MOTR [43] and TransTrack [44]. The main downside of such methods is that the number of object queries in a time step is dynamic, which makes efficient vectorized implementation harder and results in a slow training process. In contrast to the *tracking-by-attention* framework, in our model, we fuse a fixed set of object embeddings and object-specific outputs from multiple time steps using cross-attention-based modules.

III. Method

A. Multi-Object Pose Estimation as Set Prediction

Following YOLOPose [7], we formulate multi-object pose estimation as a set prediction problem. YOLOPose exploits the permutation-invariant nature of the attention mechanism to generate a set of tuples—each consisting of class probabilities, 2D bounding box, 3D bounding box, position and orientation parameters. The 3D bounding box parameters are represented using the interpolated bounding box (IBB) keypoints [45]. YOLOPose employs a ResNet backbone for feature extraction (CNN). Positional encoding compensates for the loss of spatial information in the permutation-invariant attention computation. Combined image features and positional encodings are provided to the encoder module, which uses the multi-head self-attention mechanism to generate encoder feature embeddings. In the decoder, the cross-attention mechanism is employed between the encoder feature embeddings and a set of N learned embeddings called object queries to generate N object embeddings, which are then processed by feed-forward prediction networks (FFPNs) to generate class probabilities, 2D bounding box, and IBB keypoints in parallel. The IBB keypoints are then processed by a subsequent FFPN to estimate the translation and rotation parameters. Since the cardinality of the predicted set is fixed, the model is trained to predict Ø classes after detecting all the target objects present in the image. By associating predictions and ground truth objects using a bipartite matching algorithm [46], YOLOPose is trained end-to-end.

B. MOTPose Architecture

The architecture of the proposed MOTPose model is shown in Fig. 2. We base the single-frame processing of MOTPose on the YOLOPose model. The transformer-based encoder-decoder modules generate object embeddings of cardinality N from CNN-computed image features that are augmented with positional encoding. FFPNs process the

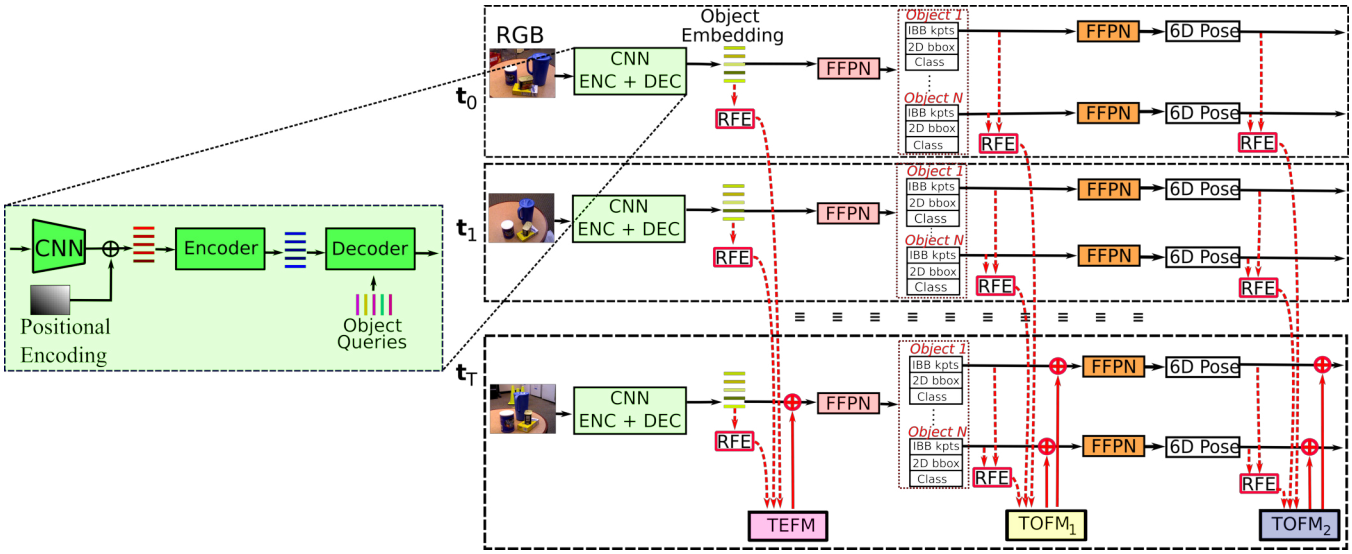


Fig. 2. MOTPose architecture. Positional Encoding: pixel coordinates represented using sine and cosine functions of different frequencies. Object Queries: learned embeddings that are trained jointly with the model and remain fixed during inference (Sec. III-A). FFPN: Feed Forward Prediction Network. TEFM: Temporal Embedding Fusion Module (Sec. III-B.1, Fig 3). TOFM: Temporal Object Fusion Module (Sec. III-B.2). $\oplus$: Element-wise addition. $\oplus$: Residual connection. The dashed red lines represent temporal connections. All modules that share a color also share weights. At each time step, object embeddings are generated using a CNN backbone and transformer-based encoder-decoder modules. The image features from the backbone are augmented with positional encoding. The object embeddings are processed in parallel using FFPNs to generate class probability, bounding box, and 6D pose parameters. At time step t_T, the object embeddings of different time steps are fused using TEFM. Similarly, object-specific predictions like the keypoints and the 6D pose parameters of different time steps are fused using TOFM. While fusing object embeddings and object-specific outputs from different time steps, Relative Frame Encoding (RFE) is added element-wise to uniquely identify the respective time step.

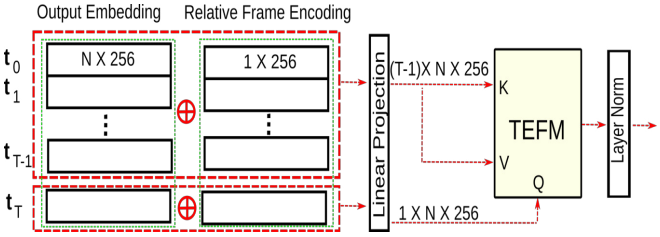


Fig. 3. Temporal Embedding Fusion Module (TEFM). $\oplus$: Concatenation operation. The object embeddings at each time step of shape N$\times$256 are added element-wise with relative frame encoding (RFE). The resulting vectors for time steps $t_0 - t_{T-1}$ are stacked to form *key* as well as *value* for the cross-attention operation in TEFM, whereas the embedding at time step T acts as *query*.

object embeddings to generate object-specific outputs. The object embeddings and the object-specific outputs from the past time steps provide rich temporal information that can be leveraged while processing the current frame. To this end, we fuse object embeddings and object-specific outputs from multiple past time steps using the Temporal Embedding Fusion Module (TEFM, Sec. III-B.1) and the Temporal Object Fusion Module (TOFM, Sec. III-B.2), respectively, before generating outputs for the current time step. To enable the fusion of embeddings and object parameters over multiple time steps using the permutation-invariant attention mechanism, we utilize *relative frame encoding* (RFE), which encodes the number of time steps relative to the current frame using 1D sinusoidal functions.

1) Temporal Embedding Fusion Module (TEFM): At each time step, the decoder generates object embeddings of shape $N\times 256$, where N is the cardinality of the object set to be predicted. TEFM, shown in Fig. 3, fuses object embeddings from multiple time steps to extract valuable temporal information. First, relative frame encoding is concatenated with the object embeddings, and then the resulting embeddings are projected back to 256 dimensions using linear layers. The stacked embeddings until $T-1$ time steps form *key* and *value* for the cross-attention operation in TEFM, whereas the embedding from the time step T is used as *query*. This allows the object embeddings from time step T to interact with object embeddings from all previous time steps. The key-query similarity is reflected in the resulting attention weights. These attention weights are used to weigh the *value* vectors, which in our case are the object embeddings from all previous time steps. After applying layer normalization, the output of TEFM is added element-wise to the object embeddings of time step T, representing a residual connection.

2) Temporal Object Fusion Module (TOFM): In addition to fusing embeddings using TEFM, we employ two TOFM modules to fuse object-specific outputs. The design of TOFM is similar to that of TEFM, except for the usage of additional linear projection layers at the beginning and the end. The object embeddings are of shape $N\times 256$, whereas the shape of the predictions is $N\times P$, which depends on the prediction generated; three in the case of translation prediction, six in the case of rotation prediction, and 32 in the case of keypoints. We use a linear layer to project the predictions to a 256-dimensional vector and supplement them with RFEs. After computing cross-attention, we project the resulting embeddings back to $N\times P$. TOFM_1 is used for fusing keypoints and TOFM_2 is used for fusing pose parameters.

C. Matching

We use the bipartite matching algorithm [7], [32], [46] to associate predicted and ground-truth objects. Despite jointly estimating 2D bounding box, class probabilities, key points, and pose parameters, similar to [7], [47], we use only the bounding box and the class probability components in the matching cost function. This is based on the empirical observation that a combination of the bounding box and the class probability components alone is enough to ensure an optimal match between the ground-truth and the predicted sets.

D. Loss Function

The *Hungarian loss* used to train MOTPose is a weighted combination of five components:

1) Class Probability Loss: We use the standard negative log-likelihood (NLL) loss to train the classification branch of the model. To deal with the class imbalance due to the Ø class appearing disproportionately often, we weigh it down by a factor of 0.1.

2) Bounding Box Loss: To train the bound box prediction branch of our model, we employ a linear combination of the generalized IOU [48] and the ℓ_1-loss.

3) Keypoint Loss: We use a weighted combination of the ℓ_1-loss and the cross-ratio consistency loss [7], [45] to train the keypoint estimation branch.

4) Pose Loss: We decouple the pose loss into a translation and a rotation component. For translation, we employ the ℓ_2-loss. For rotation, we use the symmetry-aware ShapeMatch-loss proposed by Xiang *et al.* [14].

5) Temporal Consistency Loss: We enforce temporal consistency using the ℓ_2-loss between the object embeddings of consecutive time steps. Embeddings evolve smoothly over frames and any big changes are undesirable. Thus, the ℓ_2-loss, which penalizes bigger differences significantly more than smaller differences, is a natural choice.

IV. Evaluation

A. Datasets

1) YCB-Video: We use the challenging YCB-Video dataset [14] to benchmark the performance of our model against other state-of-the-art methods. The dataset consists of 92 (80 training and 12 testing) moving-camera video sequences of static scenes with multiple objects. High-resolution 3D models of all 21 objects are provided with the dataset. Following Li *et al.* [23] and Deng *et al.* [37], we use all the frames in the test split for evaluation. Additionally, we utilize the synthetic dataset provided by Xiang et al. [14] to train our model.

2) SynPick: SynPick [49] is a physically-realistic synthetic dataset of dynamic bin-picking scenes that contain a chaotic pile of the same 21 YCB-Video objects in a tote. It consists of simulations of three different bin-picking actions: move, targeted pick, and untargeted pick. For each action, SynPick provides 300 video sequences: 240 for training and 60 for testing. Moreover, the dataset generator is publicly available[1] making it easy to generate additional data, if needed. In contrast to the commonly used object pose estimation datasets [1], [10], [50], which consist of static tabletop scenes with a relatively low degree of occlusion, SynPick is highly cluttered and the gripper movements generate complex object interactions. Moreover, the objects in the SynPick dataset appear in a wide range of pose configurations and multiple instances of the same object are present in the scenes. Thus, SynPick is an ideal dataset for evaluating the proposed MOTPose model.

TABLE I

Quantitative results on the SynPick dataset.

	MOTPose without Temporal Fusion				MOTPose with Temporal Fusion			
Obj. ID†	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S @0.1d	AUC of ADD(-S) @0.1d	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S @0.1d	AUC of ADD(-S) @0.1d
1	88.8	72.2	86.1	53.4	88.5	79.1	86.8	61.2
2	90.7	82.5	89.5	76.2	91.4	84.2	90.2	78.4
3	80.8	74.4	79.1	63.9	81.6	76.2	80.2	69.5
4	72.5	64.1	70.1	43.9	73.5	68.0	71.2	45.1
5	80.3	72.2	78.8	62.3	80.2	74.8	78.9	67.9
6	81.1	64.1	68.1	19.1	81.8	75.1	72.2	25.6
7	69.9	63.4	66.3	48.2	70.9	65.7	68.4	48.3
8	65.8	60.3	60.6	40.9	67.4	62.1	63.3	32.0
9	84.3	76.1	80.6	56.4	85.1	78.9	82.5	56.5
10	78.0	70.5	73.9	56.9	80.3	73.9	77.9	64.9
11	92.8	84.7	92.2	79.4	93.1	85.8	92.4	81.8
12	85.7	76.9	85.2	71.1	87.0	80.7	86.4	76.9
13*	89.0	89.0	83.2	83.2	89.5	89.5	85.9	85.9
14	84.9	74.8	80.8	49.0	85.9	78.5	82.5	45.6
15	90.5	83.7	89.9	75.2	92.9	87.2	92.3	83.0
16*	90.0	90.0	88.8	88.8	90.0	90.0	88.9	88.9
17	72.0	65.0	65.1	49.7	75.9	69.5	71.1	55.2
18	68.1	62.4	61.7	36.6	66.9	61.9	60.4	36.2
19*	76.0	76.0	73.6	73.6	79.0	79.0	77.5	77.5
20*	80.5	80.5	75.7	75.7	83.6	83.6	81.8	81.8
21*	75.9	75.9	72.2	72.2	76.3	76.3	69.7	69.7
Mean	80.8	74.2	77.2	60.8	**82.0**	**77.1**	**79.1**	**63.4**

* Symmetric objects.
† Object ID in the standard order of YCB-Video.

TABLE II

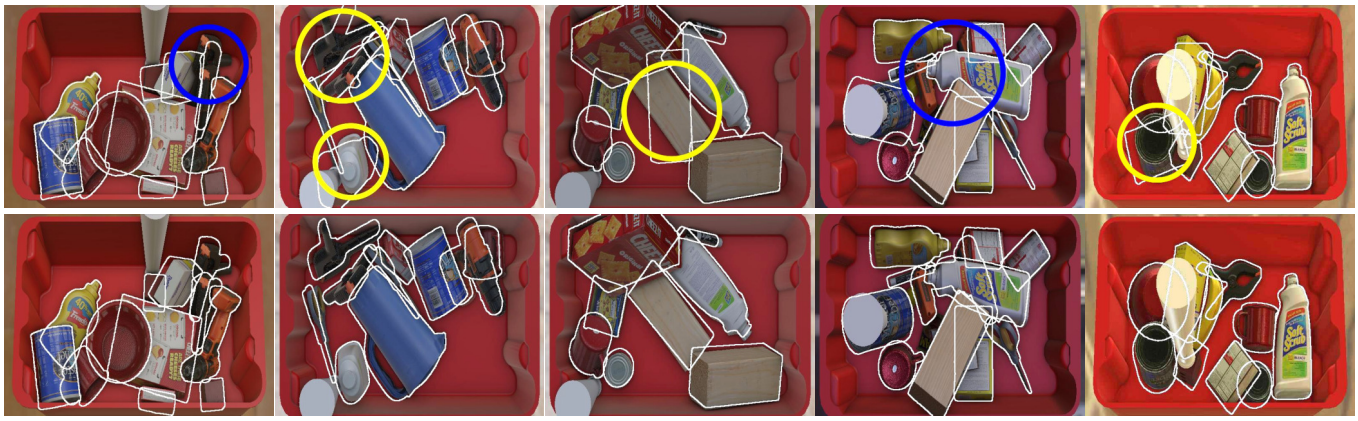
Cardinality Error on SynPick Splits [$\times 10^{-2}$].

Method	Move	Targeted pick	Untargeted pick	All
W/o temporal fusion	3.26	1.64	0.48	2.06
With temporal fusion	**0.62**	**0.52**	**0.44**	**0.53**

B. Metrics

We report the area under the curve (AUC) of the ADD and ADD-S metrics at an accuracy threshold of 0.1m for non-symmetric and symmetric objects, respectively [14]. The ADD metric is the average ℓ_2 distance between the subsampled mesh points in the ground truth and the predicted pose, whereas the symmetry-aware ADD-S metric is the average distance between the closest subsampled mesh points in the ground truth and the predicted pose. The ADD(-S) metric combines both ADD and ADD-S into one metric by utilizing ADD for objects without symmetry and ADD-S for objects exhibiting symmetry.

[1] https://github.com/AIS-Bonn/synpick



(a) (b) (c) (d) (e)

Fig. 4. Qualitative results on SynPick. 6D pose predictions are visualized by object contours. Top: Predictions from the model without temporal fusion. Bottom: Predictions from the model with temporal fusion. Temporal fusion facilitates better pose prediction as well as object detection accuracies. The blue circles highlight failed object detections and the yellow circles highlight erroneous pose predictions.

TABLE III

FALSE NEGATIVE DETECTIONS ON SYNPICK SPLITS [$\times 10^{-2}$].

Method	Move	Targeted pick	Untargeted pick	All
W/o temporal fusion	2.79	1.39	1.36	1.79
With temporal fusion	**0.57**	**0.44**	**0.44**	**0.48**

C. Implementation Details

Following [32], [47], we choose the cardinality of the predicted set N proportional to the maximum number of objects in an image in the respective datasets: 30 for SynPick and 20 for YCB-Video. In Section III-D, the bounding box components are weighted using factors 2 and 5, and the keypoint components are weighted with factors 10 and 1. The pose component and the temporal consistency component are weighted down using factors 0.05 and 0.1, respectively. The encoder and decoder modules consist of six layers each. All the embeddings used in our model are of dimension 256. We train our model for 150 epochs using the AdamW optimizer with a learning rate of 1×10^{-4} and early stopping. We set the number of time steps T to eight in the temporal fusion modules and use a batch size of 32 (four groups of eight consecutive images).

D. Results on SynPick

Formulating multi-object pose estimation as a set prediction problem enables joint object detection and pose estimation of all objects in the scene. However, it compounds the size of the dataset required to train transformer models. Thus, to complement the existing 240 videos for training, we generate additional 300 video sequences for each action split. We call this extended version SynPick-Ext and make it publicly available[2]. We downsample the image resolution to 640×480. SynPick consists of objects piled up in a tote and in many cases, objects are completely occluded. To exclude heavily occluded objects, we use a minimum visibility threshold of 30% in our evaluation. In Fig. 4, we present pose estimates generated by our model with and without temporal fusion. Both models generate predictions of admissible quality. However, the model without temporal fusion suffers from failed object detections (Fig. 4(a), (d)), and isolated highly erroneous pose predictions (Fig. 4(b), (c), (e)). Temporal fusion helps in alleviating these shortcomings.

In Table I, we report quantitative results of our model. MOTPose achieves impressive AUC of ADD-S and AUC of ADD(-S) scores of 82.0 and 77.1, respectively, which is an improvement of 1.2 and 2.9 compared to the model without temporal fusion. Additionally, we also report the AUC metrics with a threshold of 10% of the object diameter (AUC@0.1d). This metric takes the object size into account better. In terms of AUC of ADD-S and ADD(-S)@0.1d, temporal fusion boosts the accuracy by 1.9 and 2.6 points, respectively.

Furthermore, to understand the impact of temporal fusion on object detection, we analyze the cardinality error and the bounding box accuracy metrics. The cardinality error is the difference between elements in the ground-truth and predicted sets. Formally, given the ground-truth set $\mathcal{Y}$ and the predicted set $\hat{\mathcal{Y}}$, the cardinality error (CE) is defined as:

$$\text{CE} = \frac{|(\mathcal{Y} - \hat{\mathcal{Y}}) \cup (\hat{\mathcal{Y}} - \mathcal{Y})|}{|\mathcal{Y}|}. \tag{1}$$

In Table II, we report the cardinality error of our model on different splits of the SynPick dataset. Over the complete test set, the cardinality error of the model without temporal fusion is 0.021, whereas it is only 0.005 for the model with temporal fusion. The difference is more evident in the *Move* split, which is more challenging than the other two splits.

Although CE reflects the set prediction ability of a model, in real-world bin-picking systems, the identity of the objects present in the bin might be known a priori [51], [52]. Thus, in this *informed detection* scenario, false positives can be easily mitigated, whereas false negatives (FN), i.e., $|(\mathcal{Y} - \hat{\mathcal{Y}})|/|\mathcal{Y}|$ are detrimental. In Table III, we report the false negatives of object detection. Over the entire test set, the model without temporal fusion has a FN rate of 0.018; with temporal fusion,

[2] https://www.ais.uni-bonn.de/videos/tempose

TABLE IV

BOUNDING BOX PREDICTION ACCURACY.

Method	AP†	AP@ [IoU=0.50]	AP@ [IoU=0.75]	AR†
W/o temporal fusion	0.756	0.872	0.853	0.789
With temporal fusion	**0.779**	**0.876**	**0.858**	**0.811**

† @[IoU=0.50:0.95]

the FN rate drops to 0.005.

To compare the bounding box detection accuracy, we analyze the average precision and recall metrics defined by COCO evaluation protocol[3]. In Table IV, we report the AP@[IoU=0.50:0.95], AP@[IoU=0.50], AP@[IoU=0.75], and AR@[IoU=0.50:0.95] metrics of the models with and without temporal fusion. Across all the reported metrics, temporal fusion yields consistent improvements.

E. Results on YCB-Video

In Table V, we report the quantitative comparison of our MOTPose model against state-of-the-art methods on the YCB-Video dataset. In our experiments, we fuse seven previous frames (T=8) in MOTPose. Since our model does not produce outputs for the initial $T-1$ frames in a video sequence, we report the accuracy scores excluding the initial frames. Temporal fusion enables considerable improvement in the MOTPose model: 0.9 and 1.3 accuracy points in terms of the AUC of ADD-S and AUC of ADD-(S) metric, respectively. Compared to DeepIM-Tracking [23], our method achieves a comparable AUC of ADD-S score and a slightly worse AUC of ADD-(S) score. DeepIM-Tracking formulates 6D pose tracking as pose refinement, i.e., pose prediction from the previous frame is used to initialize the render-and-compare pose refinement for the current step. To initialize the first frame, the authors used the ground-truth pose. While Castro and Kim [27] achieve a significantly better accuracy than MOTPose, they perform only pose refinement. In contrast, our method performs multi-object detection and pose estimation jointly. Moreover, MOTPose accuracy is comparable to the state-of-the-art multi-object pose estimation method of Periyasamy *et al.* [53] in terms of the AUC of ADD-(S) metric and only slightly worse in terms of the AUC of ADD-S metric. Note that the frame rates reported in Table V are observed on GPUs of different generations and the values are provided only for a relative comparison.

F. Ablation Study

To understand the contribution of the individual components to the overall performance of MOTPose, we investigated removing different components of the model and varying the number of time steps used in the fusion modules. In Table VI, we report the results of the ablation experiment on SynPick. Removing the TEFM module resulted in a big drop in the overall accuracy of the MOTPose model. In terms of the AUC of ADD(-S) metric, the MOTPose model without the TEFM module achieves a score of 74.9, compared to 77.1, while the AUC of ADD-S metric score drops by 0.6. Similarly, removing the TOFM module results in a drop of 0.9 AUC of ADD(-S) and 1.8 AUC of ADD-S accuracy scores. Moreover, in terms of the number of time steps used in the fusion modules, eight time steps resulted in the best performance overall.

[3]https://cocodataset.org/#detection-eval

TABLE V

RESULTS ON THE YCB-VIDEO DATASET.

Method	AUC of ADD-S	AUC of ADD(-S)	*fps*
CRT-6D [27]	-	**87.5**	30
Periyasamy *et al.* [53]	**92.0**	84.7	26
DeepIM-Tracking [23]	91.0	85.9	13
MOTPose w/o temporal fusion	90.3	83.2	59
MOTPose with temporal fusion	91.2	84.5	30

TABLE VI

ABLATION STUDY RESULTS ON THE SYNPICK DATASET.

Method	AUC of ADD-S	AUC of ADD(-S)
MOTPose	**82.0**	**77.1**
MOTPose without temporal fusion	80.8	74.2
MOTPose without TEFM	81.1	74.9
MOTPose without TOFM	81.4	75.3
MOTPose without SynPick-Ext	76.4	69.2
MOTPose [T=4]	80.9	76.4
MOTPose [T=8]	82.0	**77.1**
MOTPose [T=12]	**82.2**	76.7

G. Limitations

Our formulation of multi-object pose estimation as a set prediction problem limits the datasets available for training our model. Compared to 2D annotations, 6D pose annotations are significantly harder to obtain. Thus, many of the standard datasets for evaluating object pose estimation like Linemod-Occluded [50] and Linemod [10] provide pose annotations only for a partial number of objects per scene in the training dataset. While this is not a limitation for multi-stage methods that process the cropped version of the images for estimating the pose of target objects, our method needs 6D pose annotation for all objects in the scene, which can be prohibitively expensive to acquire in some scenarios.

V. CONCLUSION

We presented MOTPose, a multi-object pose estimation model for RGB video sequences. Employing the cross-attention-based TEFM and TOFM modules, the MOTPose model fuses object embeddings and object-specific outputs over multiple time steps, respectively. Aided by the temporal information, our model performs significantly better than the single-frame RGB model while being lighter and significantly faster than other pose tracking methods.

VI. ACKNOWLEDGMENT

This work has been funded by the German Ministry of Education and Research (BMBF), grant no. 01IS21080, project "Learn2Grasp: Learning Human-like Interactive Grasping based on Visual and Haptic Feedback".

REFERENCES

[1] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP challenge 2020 on 6D object localization," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 577–594.

[2] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, *et al.*, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 1, pp. 87–110, 2022.

[3] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[4] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Computing Survey*, vol. 54, no. 10s, 200:1–200:41, 2022.

[5] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "EfficientViT: Memory efficient vision transformer with cascaded group attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 420–14 430.

[6] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H. Shum, "Mask DINO: Towards a unified transformer-based framework for object detection and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 3041–3050.

[7] A. Amini, A. Selvam Periyasamy, and S. Behnke, "YOLOPose: Transformer-based multi-object 6D pose estimation using keypoint regression," in *17th International Conference on Intelligent Autonomous Systems (IAS)*, 2022, pp. 392–406.

[8] A. Amini, A. S. Periyasamy, and S. Behnke, "T6D-Direct: Transformers for multi-object 6D object pose estimation," in *DAGM German Conference on Pattern Recognition (GCPR)*, 2021.

[9] S. Hinterstoißer, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. V. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, pp. 876–888, 2012.

[10] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model-based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asian Conference on Computer Vision (ACCV)*, 2013, pp. 548–562.

[11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision (IJCV)*, vol. 66, pp. 231–259, 2006.

[12] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2011–2018, 2017.

[13] S. Tulsiani and J. Malik, "Viewpoints and keypoints," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1510–1519, 2014.

[14] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[15] A. S. Periyasamy, M. Schwarz, and S. Behnke, "Robust 6D object pose estimation in cluttered scenes using semantic segmentation and pose regression networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[16] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "SO-Pose: Exploiting self-occlusion for direct 6D pose estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 396–12 405.

[18] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3828–3836.

[19] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3385–3394.

[21] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DOF pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4561–4570.

[22] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6D object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2930–2939.

[23] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[24] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6D pose refinement in RGB," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 800–815.

[25] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2020.

[26] A. S. Periyasamy, M. Schwarz, and S. Behnke, "Refining 6D object pose predictions using abstract render-and-compare," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 739–746.

[27] P. Castro and T.-K. Kim, "CRT-6D: Fast 6D object pose estimation with cascaded refinement transformers," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 5746–5755.

[28] Y. Hai, R. Song, J. Li, and Y. Hu, "Shape-constraint recurrent flow for 6D object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4831–4840.

[29] Y. Hu, P. Fua, and M. Salzmann, "Perspective flow aggregation for data-limited 6D object pose estimation," in *European Conference on Computer Vision (ECCV)*, Springer, 2022, pp. 89–106.

[30] C. Capellen, M. Schwarz, and S. Behnke, "ConvPoseCNN: Dense convolutional 6D object pose estimation," in *15th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

[31] S. Thalhammer, M. Leitner, T. Patten, and M. Vincze, "PyraPose: Feature pyramids for fast and accurate object pose estimation under domain shift," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 909–13 915.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[33] T. G. Jantos, M. A. Hamdad, W. Granig, S. Weiss, and J. Steinbrener, "PoET: Pose estimation transformer for single-view, multi-object 6D pose estimation," in *Conference on Robot Learning (CoRL)*, PMLR, 2023, pp. 1060–1070.

[34] P. Azad, D. Münch, T. Asfour, and R. Dillmann, "6-DoF model-based tracking of arbitrarily shaped 3D objects," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5204–5209, 2011.

[35] K. Pauwels, L. Rubio, J. Díaz, and E. Ros, "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2347–2354, 2013.

[36] Y. Xiang, C. Song, R. Mottaghi, and S. Savarese, "Monocular multi-view object tracking with 3D aspect parts," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 220–235.

[37] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A rao-blackwellized particle filter for 6D object pose tracking," in *Robotics: Science and Systems (RSS)*, 2019.

[38] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se(3)-TrackNet: data-driven 6D pose tracking by calibrating image residuals in synthetic domains," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[39] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[40] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *15th European Conference on Computer Vision (ECCV)*, 2020.

[41] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense representations for multiple-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[42] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[43] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," in *17th European Conference on Computer Vision (ECCV)*, 2022.

[44] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, "TransTrack: Multiple-object tracking with transformer," *arXiv:2012.15460*, 2020.

[45] S. Li, Z. Yan, H. Li, and K.-T. Cheng, "Exploring intermediate representation for monocular vehicle pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1873–1883.

[46] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[47] A. S. Periyasamy, A. Amini, V. Tsaturyan, and S. Behnke, "YOLO-Pose V2: Understanding and improving transformer-based 6D, pose estimation," *Robotics and Autonomous Systems*, vol. 168, p. 104 490, 2023.

[48] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

[49] A. S. Periyasamy, M. Schwarz, and S. Behnke, "SynPick: A dataset for dynamic bin picking scene understanding," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 488–493.

[50] E. Brachmann, *6D Object Pose Estimation using 3D Object Coordinates [Data]*, version V1, 2020. [Online]. Available: `https://doi.org/10.11588/data/V4MUMX`.

[51] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3347–3354.

[52] M. Schwarz, A. Milan, C. Lenz, A. Munoz, A. S. Periyasamy, M. Schreiber, S. Schüller, and S. Behnke, "NimbRo picking: Versatile part handling for warehouse automation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3032–3039.

[53] A. S. Periyasamy, V. Tsaturyan, and S. Behnke, "Efficient multi-object pose estimation using multi-resolution deformable attention and query aggregation," *IEEE International Conference on Robotic Computing (IRC)*, 2023.