TP-AE: Temporally Primed 6D Object Pose Tracking with Auto-Encoders

Linfang Zheng[1,2], Aleš Leonardis[2], Tze Ho Elden Tse[2], Nora Horanyi[2], Hua Chen[1], Wei Zhang[1*], Hyung Jin Chang[2]

***Abstract*— Fast and accurate tracking of an object's motion is one of the key functionalities of a robotic system for achieving reliable interaction with the environment. This paper focuses on the instance-level six-dimensional (6D) pose tracking problem with a *symmetric* and *textureless* object under *occlusion*. We propose a Temporally Primed 6D pose tracking framework with Auto-Encoders (TP-AE) to tackle the pose tracking problem. The framework consists of a prediction step and a temporally primed pose estimation step. The prediction step aims to quickly and efficiently generate a guess on the object's real-time pose based on historical information about the target object's motion. Once the prior prediction is obtained, the temporally primed pose estimation step embeds the prior pose into the RGB-D input, and leverages auto-encoders to reconstruct the target object with higher quality under occlusion, thus improving the framework's performance. Extensive experiments show that the proposed 6D pose tracking method can accurately estimate the 6D pose of a symmetric and textureless object under occlusion, and significantly outperforms the state-of-the-art on T-LESS dataset while running in real-time at 26 FPS.**

I. Introduction

Thanks to the rapid development of reliable mechanical structures, highly efficient actuators and powerful algorithms, robotic systems have been deployed into various real-world applications such as mobile manipulation [1], legged systems [2], robotic manipulation [3], and so on. With the increasing need for interacting with the environment, accurate detection and tracking of a target object become a core functionality for modern robotic systems.

This paper focus on the instance-level six-dimensional (6D) pose tracking problem. Under various robotic application scenarios, the target objects to be manipulated or interacted are possibly symmetric and textureless. Furthermore, the target object may be occluded by the environment or other objects. In these situations, estimating the target object's pose becomes much more challenging. Unlike the conventional pose estimation problems based on single RGB(-D) data, pose tracking leverages historical information about the target object's movement to assist in obtaining the desired pose. Incorporating the historical information and considering the pose tracking problem allows for dealing with challenging scenarios involving *symmetric* and *textureless* objects under *occlusion*.

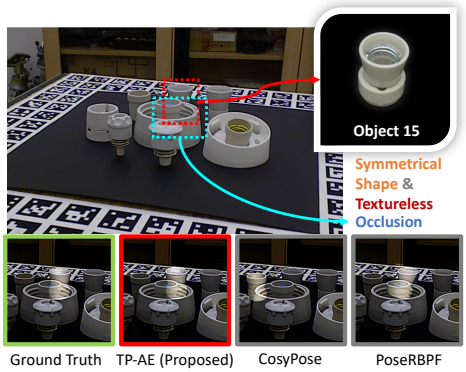

Fig. 1: **Performance of the proposed TP-AE framework.** The proposed framework outperforms state-of-the-art approaches such as PoseRBPF [4] and CosyPose [5] for symmetric and textureless objects under occulsion.

To solve the pose tracking problem, we develop a **T**emporally **P**rimed 6D object pose tracking framework with **A**uto-**E**ncoders (TP-AE). The proposed framework first predicts the 6D pose of the target object from a historical pose sequence and then uses the prediction to assist the visual-based pose estimation given the real-time RGB-D measurement. For the prediction step, we propose to use temporal pose information to encode the raw RGB-D image stream information. Once the prediction is generated, the temporally primed pose estimation step adjusts the predicted pose via a reconstructed pose reference generated by auto-encoders. By resorting to the auto-encoder based reconstruction, the proposed refinement scheme effectively handles *symmetric* and *textureless* objects under *occlusion*.

A. Related Works

6D Pose Estimation 6D object pose estimation problem that aims to estimate an object's 6D pose from a single image without temporal information has been extensively studied in the literature over several decades. Classical methods [6], [7] achieved high precision while requiring prohibitive hyper-parameter tuning when applied to new scenarios. Recently, deep learning-based methods have shown better generalization ability in challenging scenarios involving *symmetric* and *textureless* objects under *occlusion*. For example, [8], [9] aim to address rotational ambiguity. [10]–[12] focus on occlusion. [13], [14] consider textureless objects. More recently, various approaches have been proposed to deal with mixed challenges. For instance, [15], [16] handle symmetric objects under occlusion. Estimating poses for textureless

This work is supported in part by the National Natural Science Foundation of China under Grants 62073159 and 62003155, the Shenzhen Science and Technology Program under Grant JCYJ20200109141601708, the Science, Technology and Innovation Commission of Shenzhen Municipality under grant ZDSYS20200811143601004, and the Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00537, Visual common sense through self-supervised learning for restoration of invisible parts in images). (Corresponding author: Wei Zhang).

[1] Department of Mechanical and Energy Engineering, Southern University of Science and Technology (SUSTech), China.
11956001@mail.sustech.edu.cn,
{chenh6,zhangw3}@sustech.edu.cn

[2] School of Computer Science, University of Birmingham, UK.
{lxz948,nxh840,txt994}@student.bham.ac.uk,
{a.leonadis,h.j.chang}@bham.ac.uk

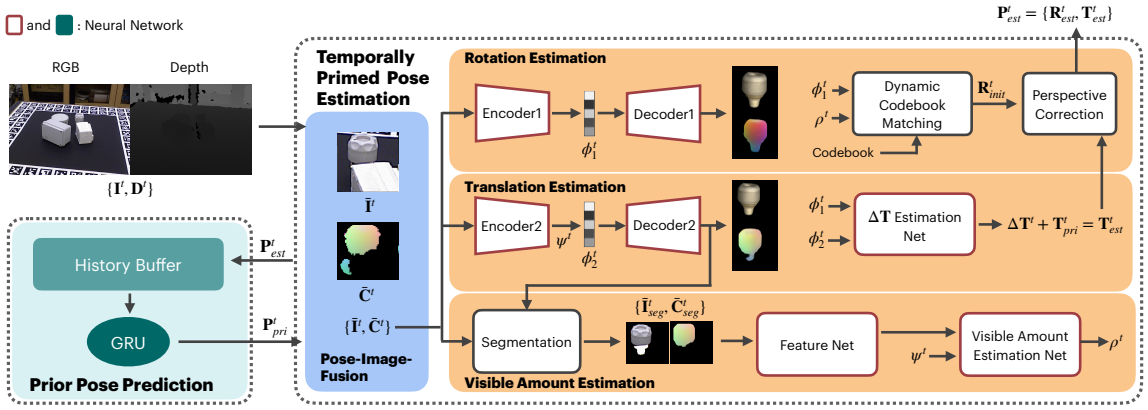

Fig. 2: **Overview of the proposed framework.** The framework contains two major units: the prior pose prediction unit and the temporally primed pose estimation unit. The prior pose prediction unit predicts a prior pose based on the historically estimated poses. Then the predicted pose and the input RGB-D image are fed into the temporally primed pose estimation unit to attain the final pose estimation.

objects under occlusion has been investigated in [5], [17]–[20]. Symmetric and textureless objects have been considered in [21]–[23]. However, none of them can address symmetry, textureless, and occlusion simultaneously.

6D Pose Tracking As a natural extension of classical pose estimation problem, pose tracking problem try to incorporate temporal information to achieve higher pose estimation accuracy, which offers opportunities to simultaneously address all three aforementioned challenges. Traditional pose tracking methods [24], [25] rely on hand-crafted likelihood functions, which is hard to generalize to new scenarios. Due to the lack of rich video datasets, the development of learning-based 6D pose tracking schemes remains limited. Along this direction, pioneering works such as [26]–[30] try to utilize the relationship between current image with the last image to aid estimating objects' pose. However, focusing on only two consecutive steps is restrictive in fully characterizing objects' motion. To this end, [4], [31] use Rao-Blackwellized Particle Filter [32] and Unscented Kalman Filter [33], respectively, to encode motion information that is subsequently used for object tracking. Nonetheless, the performance of [4] under occlusion remains unsatisfying due to the lack of utilization of temporal information in the object reconstruction phase.

Despite these recent advances on pose estimation and pose tracking, how to construct a reliable and efficient pose tracking framework for *symmetric* and *textureless* objects under *occlusion* remains a challenging problem.

B. Contributions

The main contributions of this paper are as follows. First, we propose, to the authors' best knowledge, the first neural-network based prior pose generation scheme. This scheme exploits the target object's pose history of any length to better predicts the object's future pose. By working with the pose information encoded in the complete movement of the object, the proposed scheme is computationally friendly and generates more accurate predictions in cases where the object moves with non-constant velocity. Second, we develop a novel temporally-primed pose estimation scheme consisting of a pose-image fusion and auto-encoder based pose estimation, which improves the learning performance of the residual pose. The pose-image fusion scheme helps with reconstructing the intact appearance and point cloud from only partially observed measurement. Combined with the latent codes and features available from the auto-encoders, the overall temporally-primed pose estimation scheme successfully handles *symmetric* and *textureless* objects under *occlusion*. Third, the overall framework achieves not only a state-of-the-art performance in standard 6D object pose estimation dataset benchmarks (especially for T-LESS dataset that contains numerous scenarios with symmetric and textureless objects under occlusion AR: 82.3 vs. 73), but also real-time speed (26 FPS).

C. Framework Overview

Fig. 2 shows an overview of the proposed framework. Generally speaking, the proposed framework contains a prior pose prediction unit and a temporally primed pose estimation unit. At each time step, the prior pose prediction unit first generates a predicted 6D pose based on the historical pose estimations of the target object. Then, such a predicted 6D pose together with the real-time measured RGB-D data is fed into the temporally primed pose estimation unit to further adjust the predicted pose to obtain the final result.

To achieve fast prediction and account for the lack of large video datasets, the proposed framework takes the historically estimated 6D pose sequence as input to a GRU-based neural network to generate the prediction. Once the prediction is obtained, a pose-image-fusion module merges the predicted pose and the input RGB-D image to generate a RGB-Cloud pair. Then, the RGB-Cloud pair is fed into three branches to estimate the object's rotation, translation, and visible amount, respectively.

II. Prior Pose Prediction

As one of the major differences from standard pose estimation, pose tracking strategies have access to historical information about the movement of the target object, which has a strong implication on determining the object's current pose. The first important question to be answered is how to extract such key information encoded in the historical data.

An intuitive approach to incorporate temporal data is to train a neural network that directly maps the historical

image streams to the current pose. Such an approach requires substantial data, which is not practically feasible due to the lack of available datasets. Alternatively, we use historical pose trajectory to represent the object motion without extra data collection effort. Since the pose trajectories of different objects can be shared, one can use the pose trajectories of any object across different datasets to train a prior pose prediction network and then apply it to predict the pose of an unseen object. Moreover, using historical pose sequence for pose prediction runs faster than using an image stream, which is essential for object tracking tasks.

Our prior pose prediction unit consists of a buffer and a prior pose prediction net. The buffer stores the previously estimated poses of the target object. Let l be the size of the buffer and let $\mathbf{P}_{est}^t = \{\mathbf{R}^t, \mathbf{T}^t\}$ with $\mathbf{R}^t \in SO(3)$ and $\mathbf{T}^t \in \mathbb{R}^3$ be the estimated pose of the target object at time t, the sequence of estimated poses is denoted by $\{\mathbf{P}_{est}^i\}_{i=t-l-1}^{t-1}$. Given this sequence, the prior pose prediction net uses a GRU network [34] to regress the prior pose $\mathbf{P}_{pri}^t$.

By using the pose trajectories for prior pose prediction, we can train our prediction net more robustly using additional random data augmentation on the pose trajectories.

Remark 1: During inference, the initial prior pose is generated by the existing single-image 6D pose estimation approach, as there is no historical poses for prediction.

A. Network Architecture and Loss Function

We use a single GRU layer connected to two dense layers to predict the prior pose. During implementation, a 6D parameterization of the rotation space $SO(3)$ proposed by [35] is adopted to respect the continuity of the rotation space. Consequently, the input to the neural network is a vector in $R^{l\times 9}$ with l being the size of the buffer. The output is a vector in R^9 parameterizing the 6D pose of the target object, including a 3D translation vector and a 6D parameterization of the orientation state.

The total loss contains translation loss and rotation loss. We adopt the ℓ_2 norm as the translation loss function and calculate the rotation loss following [35]. Then, the pose prediction loss is:

$$\mathcal{L}oss_{pri} = \mathcal{L}oss_{pri,\mathbf{R}} + \beta \mathcal{L}oss_{pri,\mathbf{T}} \tag{1}$$

where β is a hyperparameter weighting the importance of the translation loss relative to the rotation loss.

III. Temporally Primed Pose Estimation

This module visually estimates an object's pose assisted by the prior pose. It first fuses the prior pose with the real-time RGB-D data to generate an RGB-Cloud pair, then feed the pair to three modules to estimate the object's rotation, translation, and visibility. To robustify the pose estimation network against occlusion, we leverage object reconstruction in both the rotation and the translation estimation module.

Reconstruction networks like auto-encoders can extract the low-dimensional representation of objects, which are commonly called latent codes. Roughly speaking, auto-encoder based strategies first extract the latent code of an object by supervising the reconstruction procedure, in which the decoder needs to recover the object using the latent code provided by the encoder. Then, the latent code is used for pose estimation to boost the inference speed without needing a decoder. For such method, acquiring high-quality latent codes under occlusion is crucial for pose estimation accuracy.

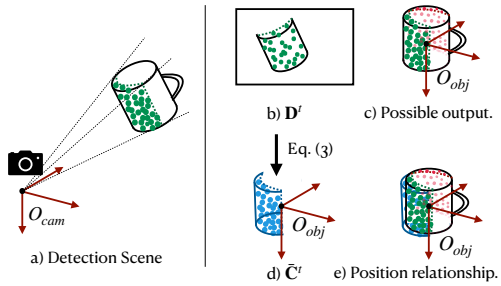

Fig. 3: **Illustration of prior pose embedding on $\mathbf{C}^t$.** The recovered point cloud in $\mathbf{D}^t$ can be mapped to either the green surface (GT) or the red surface in c) when no prior information is provided. After the transformation using the prior pose, the transformed point cloud in $\bar{\mathbf{C}}^t$ (see d) is closer to the GT point cloud distribution (see e), thus helping the network to recover the pose and smoothing the tracking procedure.

In this sequel, we first discuss how the prediction and input RGB-D data can be expressed in a unified way to support object reconstruction, then develop the proposed auto-encoder based strategy that particularly addresses the *symmetry*, *textureless* and *occlusion* challenges.

A. Pose-Image-Fusion

Given the prior pose prediction $\mathbf{P}_{pri}^t$ and the real-time RGB-D image $(\mathbf{I}^t, \mathbf{D}^t)$ at a generic time t, with $\mathbf{I}^t$ being the RGB data and $\mathbf{D}^t$ being the depth data, we first need to provide a unified way of representing the information encoded therein. For this purpose, we propose to use a cropped RGB-Cloud image pair $(\bar{\mathbf{I}}^t, \bar{\mathbf{C}}^t)$ which carries visually sensible prior information for reconstruction network. The procedure of obtaining the cropped RGB-Cloud image pair from the given inputs $(\mathbf{P}_{pri}^t, \mathbf{I}^t, \mathbf{D}^t)$ consists of three main steps.

First, we backproject the depth image using the camera's intrinsic parameters to recover its point cloud image $\mathbf{C}^t$, in which every pixel stores the recovered 3D coordinate of the corresponding pixel in the depth image. Working with the point cloud image helps the network to leverage the geometric structure of the object.

Then, we extract the potentially useful area by cropping the input RGB image and the recovered point cloud data with an enlarged object's bounding box (Bbox) centering at the predicted translation $\mathbf{T}_{pri}^t$. To improve the robustness of the overall scheme, a Bbox scaling factor δ depending on the object's diameter d is introduced as follows:

$$\delta = \max\{2\sqrt{3}\epsilon_T/d, s_{\min}\} \tag{2}$$

where ϵ_T is the maximum allowable displacement between the ground truth (GT) translation and the prediction and s_{min} is a pre-specified minimum scaling factor. This cropped RGB image used as $\bar{\mathbf{I}}^t$.

The point cloud image crops, $\tilde{\mathbf{C}}^t$, is then transformed into the frame defined by the predicted pose $\mathbf{P}_{pri}^t$ and then normalized with a scaling factor associated with the Bbox:

$$\bar{\mathbf{C}}^t = \mathbf{R}_{pri}^T(\tilde{C}^t - \mathbf{T}_{pri}^t)/|0.5\delta|. \tag{3}$$

In essence, the above operations transform the input RGB-D image to the RGB-Cloud pair $(\bar{\mathbf{I}}^t, \bar{\mathbf{C}}^t)$. By doing so, the object's position in the RGB-Cloud pair indicates the distance from $\mathbf{T}^t_{pri}$ to GT position. The point-wise fusion in (3) makes the transformed point cloud robust to occlusion, as the prior pose can be inferred even when part of the point cloud is invisible. Fig. 3 provides a visual explanation about the motivation of embedding the prior information to the point cloud. The RGB-Cloud pair, which contains appearance and transformed geometric information, are ideal for object reconstruction networks to recover textureless objects robustly under occlusion.

B. Auto-Encoder and Matching based Rotation Estimation

Once the RGB-Cloud pair is generated, we exploit the auto-encoder based strategy to address the main challenges in estimating the object's pose. We first generate the latent code of the object to achieve the robustness to occlusion, then get the rotation by a dynamic codebook matching method. This method naturally handles textureless and symmetric objects, as the code matching compares the intact object's feature among different rotations rather than local features.

1) Latent code extraction and codebook generation: We train an auto-encoder to extract the latent code that encodes the object's rotation while invariant to translation. To do so, we generate the target RGB-Cloud pair in which the object is in the center while maintaining its orientation (See the output of Decoder1 in Fig. 2) by fusing the GT pose with the GT image. The GT image shows the intact object against a black background with constant lighting at the GT pose. After the training, a codebook is generated by collecting the latent codes of the object in different rotations.

2) Dynamic latent code matching: Typically, a code matching procedure compares the cosine similarity between the input latent code with the codebook and then obtains the rotation using the highest similarity score [21], [23]. However, this might fail when the latent code is of poor quality (*e.g.* for almost invisible objects). Therefore, we enhance the occlusion robustness of the matching phase by inducing the historical information into this phase and using the object's visible amount to balance currently observed information and historical information. Specifically, during inference, we generate two cosine similarity score lists by comparing the codebook with the latent code of the input RGB-Cloud pair and that of the historical RGB-Cloud pair. The historical RGB-Cloud pair is obtained by fusing the prior pose with a synthetically generated RGB-D image in which the target object is viewed from the prior pose perspective. Denoting the first score list as $\mathbf{S}^t_{obs}$ and the second score list as $\mathbf{S}^t_{his}$, the final score is constructed as follows

$$\mathbf{S}^t = \begin{cases} \mathbf{S}^t_{obs}, & \rho^t > \sigma \\ \frac{\rho^t}{\sigma} \cdot \mathbf{S}^t_{obs} + \frac{\sigma - \rho^t}{\sigma} \cdot \mathbf{S}^t_{his}, & \rho^t \leq \sigma \end{cases} \tag{4}$$

where σ is a threshold to indicate whether the dynamic adjustment is used. We then get an initial rotation estimation $\mathbf{R}^t_{init}$ using the highest score from the final score list $\mathbf{S}^t$. To account for the rotation ambiguity caused by translation as mentioned in [36], [37], we correct $\mathbf{R}^t_{init}$ by first finding a rotation transformation $\Delta(\mathbf{R}^t)$ that aligns the direction of the estimated translation $\mathbf{T}^t_{est}$ to camera's z-axis, then getting the final rotation estimation by $\mathbf{R}^t_{est} = \Delta(\mathbf{R}^t)^{-1}\hat{\mathbf{R}}^t_{init}$.

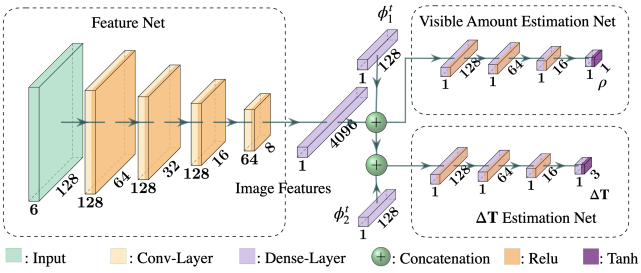

Fig. 4: The architecture of Feature Net, the Visible Amount Estimation Net and the $\Delta\mathbf{T}$ Estimation Net.

C. Auto-Encoder based Translation Estimation

The estimation of translation can also leverage the latent code of an auto-encoder to achieve robustness to occlusion. Specifically, we train an auto-encoder (Auto-Encoder 2 in the proposed framework Fig. 2) to reconstruct the intact object while preserving its location in the RGB-Cloud pair. The reconstruction target is generated by fusing the prior pose with the GT image. Motivated by the observation that the object's location in the RGB-Cloud pair provides information about the translation difference between the prior prediction $\mathbf{T}^t_{pri}$ and the ground truth $\mathbf{T}^t_{GT}$, we concatenate the latent code of rotation auto-encoder and translation auto-encoder to an estimation network that generates the desired adjustment $\Delta\mathbf{T}^t$. Then, the resulting translation estimation is simply given by $\mathbf{T}^t_{est} = \Delta\mathbf{T}^t + \mathbf{T}^t_{pri}$.

D. Visible Amount Estimation

As it is important whether the target object is still being tracked correctly, this module estimates the object's visibility. We define the visible amount ρ^t as the ratio of visible and total object pixels. As shown in Fig. 2, we use the feature of the intact object (the output of Encoder2) and the occluded object (the output of Feature Net) as the input of the Visible Amount Estimation Net to regress ρ^t. During testing, ρ is used to i) trigger re-initialization of tracking when it is lower than a pre-defined level and ii) enhance rotation estimation under severe occlusion (See Eq.(4)).

E. Network Architecture and Loss Function

The network architecture of the used auto-encoders is the same as AAE [21], except that the channel number of input and output images are set to 6. The structure of the Feature Net, the Visible Amount Estimation Net, and the $\Delta\mathbf{T}$ Estimation Net is shown in Fig. 4. The loss function for training the temporally primed pose estimation unit includes three terms: reconstruction loss $\mathcal{L}oss_{rec}$, translation loss $\mathcal{L}oss_{\Delta\mathbf{T}}$, and the visible amount loss $\mathcal{L}oss_{\rho}$.

The object reconstruction loss is the region weighted sum of the pixel-wise losses between the reconstructed and the target RGB-Cloud pair. We divide the pixels into three regions, the matched object region, the matched background region, and the mismatch region. Denoting the set of pixels

belonging to the object in the target crops and the reconstructed crops as V and $\hat{V}$, respectively, the object matching region is $V \cap \hat{V}$, the mismatch region is $(V - \hat{V}) \cup (\hat{V} - V)$, and all other pixels belong to the background region. By denoting the ℓ_2 loss of the i^{th} pixel as $\mathcal{L}oss_{px,i}$, the object reconstruction loss is:

$$\mathcal{L}oss_{rec} = \Sigma_{i \in \mathbf{I}} (\gamma \cdot \mathcal{L}oss_{px,i}) \quad (5)$$

where $\mathbf{I}$ is the input crops and $\gamma \in \{\gamma_1, \gamma_2, \gamma_3\}$ is the region based weight, in which $\gamma_1, \gamma_2, \gamma_3$ is used for the mismatched region, the matched object region, and matched background region, respectively. By setting $\gamma_1 > \gamma_2 > \gamma_3$, the auto-encoder is guided to focus on aligning silhouette.

We use ℓ_2 loss for translation loss $\mathcal{L}oss_{\Delta \mathbf{T}}$ and visible amount loss $\mathcal{L}oss_{\rho}$. The total estimation loss $\mathcal{L}oss_{est}$ is:

$$\mathcal{L}oss_{est} = \lambda_1 \mathcal{L}oss_{rec} + \lambda_2 \mathcal{L}oss_{\rho} + \lambda_3 \mathcal{L}oss_{\Delta \mathbf{T}}. \quad (6)$$

Empirically, the parameters are chosen as $\beta = 0.1$, $\lambda_1 = \lambda_3 = 1$, $\lambda_2 = 0.5$, $\gamma_1 = 3$, $\gamma_2 = 2$ and $\gamma_3 = 1$ to achieve balance between all losses.

IV. Experiments

We provide the implementation details and the experiment results of the propose framework in this section.

Baseline methods: The result of PoseRBPF is available from [4]. The single-image single-object (siso) result of CosyPose is available from its official website [38]. We use the RGB version of CosyPose since its average recall (AR_{vsd}) performance is lower when applying the ICP refinement according to the BOP challenge results [39]. The rest of the results are from corresponding papers.

Datasets: We use T-LESS [40] and YCB-V [41] to evaluate our framework, since other existing tracking datasets are either limited in size [42], not accurately labeled [43], or unsuitable for our problem setting, such as [27], [44].

1) T-LESS is widely used for pose estimation and is the best-fit to our problem setting. It contains 10K test images and 39K training images, in which all 30 industrial objects are symmetric and textureless. The testing scenarios include various occlusion levels from non-occlusion to fully occlusion. We use VSD metric [45] for evaluation. The recall accuracy AR_vsd is reported at $err_{vsd} < 0.3$ with a tolerance $\tau = 20mm$ and $> 10\%$ object visibility.

2) YCB-V contains 92 RGB-D videos (12 for testing) with 130K real images and 80K synthetic images. It provides 21 daily objects with various shapes and texture levels. We use the ADD-S [13] as the metric, where a pose is regarded as correct if the average distance of the model points to the nearest estimated points is less than 10% of the model diameter. Following PoseCNN [41], we report the area under the accuracy-threshold curve (AUC) for pose evaluation.

A. Implementation Details

We conduct all the experiments using one NVIDIA RTX 2080Ti GPU and an Intel i9-CPU@3.30GHZ. During training, Adam optimizer is adopted with 15000 training steps and a batch size of 64. Similar to AAE, we use domain randomization methods for training images. Image backgrounds are augmented by the images from [46]. We set ϵ_T to 28.8mm and s_{min} to 1.3 for pose-image fusion, and the RGB-Cloud pair is scaled to 128×128 before being fed into the encoders. Pose trajectories for training the prior pose prediction net are augmented with rotation shift, translation shift, noise addition, random drop and permutation. The buffer size l is 10. During inference, we initialize the prior pose using CosyPose (1-view version). Same as PoseRBPF, we take the one with the highest confidence score from the pose hypotheses as the initial pose. Re-initialization is triggered when: 1) $\rho < 0.2$, and 2) the rotation tracking fails with the same threshold as PoseRBPF [4], which is 0.6 for latent code comparison. The viewpoint number of the codebook is the same as PoseRBPF (184464). We set σ of dynamic codebook matching to 0.5 for T-LESS dataset. Since the pose label of YCB-V is not very accurate, σ is set to 1 as a compensate.

TABLE I: Ablation study results (AR: Average Recall).

Item	Comparison	AR_{vsd}
Full net	Train data: Mixed	**82.3**
	Train data: Syn. only	77.2
[AS-1]	RGB (AAE +Retina + ICP)	57.1
	PIEM: RGB-D w/o Eq. 3 (syn.)	60.2
	PIEM: RGB-D with Eq. 3 (syn.)	77.2
[AS-2]	LSTM (mixed)	81.4
	GRU (mixed)	**82.3**
[AS-3]	W/o perspective correction (mixed)	80.6
[AS-5]	No dynamic codebook matching (mixed.)	81.6
[AS-6]	Refine CosyPose [5] (siso, acc: 63.8%)	74.3

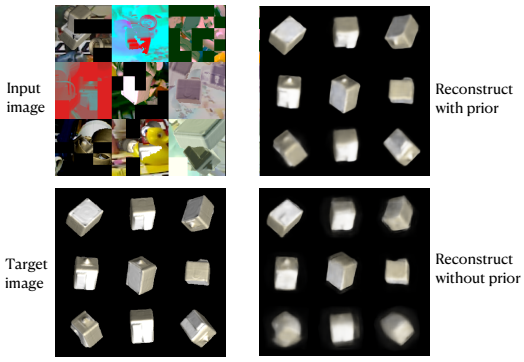

Fig. 5: **Reconstructed images from Decoder1 when without and with the Eq.** (3) **in the pose-image-fusion module.** This illustrates that the image reconstruction quality increases when applied prior pose fusion.

B. Ablation Study

We conducted an intensive ablation study using T-LESS dataset to validate our framework design. Full evaluation results are shown in TABLE. I.

[AS-1] Pose-image-fusion: We evaluate the pose-image fusion module by comparing the performance of the auto-encoders trained with i) RGB-only synthetic (syn.) images, ii) RGB-D synthetic images without (w/o) Eq. (3), and iii) RGB-D synthetic images with Eq. (3). For the first one, we referenced the result of AAE. The increased pose estimation accuracy in TABLE. I and the improved image reconstruction quality under occlusion shown in Fig. 5 both confirm the the effectiveness and motivation for pose-image fusion.

[AS-2] Prior pose prediction: To investigate the suitable network for pose prediction, we compared the LSTM [47] and GRU network. As the motion pattern is similar between the training set and test set of both T-LESS and YCB-

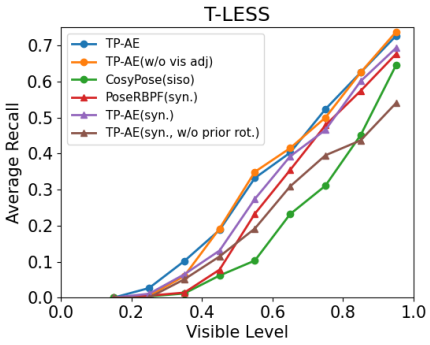

Fig. 6: Pose accuracy under a range of occlusion level.

TABLE II: Performance on T-LESS test set (Primesense)

Type	Method	AR_{vsd}	Speed
RGBD	**TP-AE** (mixed)	**82.3**	**26**
	TP-AE (syn)	77.2	**26**
	AAE [21] + (ICP)	57.1	2
	PoseRBPF [4]	72.9	10
RGB	AAE [21]	18.4	5.9
	PoseRBPF [4]	41.5	**11.5**
	CosyPose(siso)	**63.8**	-
	Pix2Pose [50]	29.5	0.6
	PFRL [51] + AAE	51.53	4.2
D	StablePose [20]	73	2.5
Rot. Track	**TP-AE** (mixed)	**93.4**	26
	TP-AE (syn)	92.7	26
	AAE(GT BBox)	72.8	-
	PoseRBPF(GT BBox)	85.3	-
GT re-init	**TP-AE** (mixed)	**84**	26
	TP-AE (syn.)	79.8	26

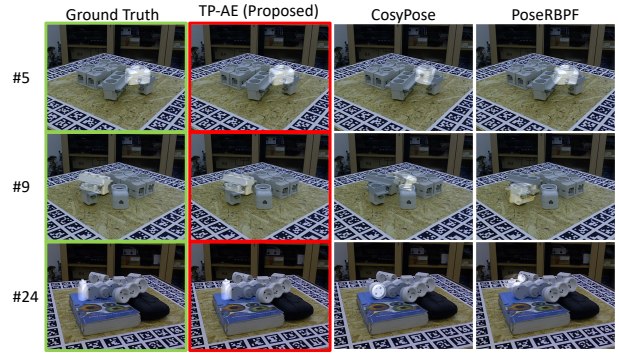

Fig. 7: **Qualitative results of Decoder1 on the T-LESS dataset.** The target objects (5,9,24) and their pose are highlighted with a white overlay on the input image. The results are displayed from left to right as GT, TP-AE, CosyPose, and PoseRBPF respectively.

TABLE III: AUC Performance on the YCB-V dataset.

Type	RGB-D Method	ADD-S	Speed
Tracking	**TP-AE**(mixed)	93.8	26
	TP-AE(syn)	92.5	26
	PoseRBPF(200 particles)	93.3	5
	DeepIM [30] + PoseCNN [41] (4 it)	94.0	6
	MaskUKF [31]	94.2	52.6
	$se(3)$-TrackNet [28]	**95.52**	**90.0**
Estimation	PoseCNN (ICP) [41]	93.0	< 0.1
	PVN3D [11] (w/o refinement)	95.5	5
	Densefusion [19] (w/o refinement)	91.2	20
	G2L-Net [52] (w/o refinement)	92.4	21
	FFB6D [18]	**96.6**	13.3

V, training directly on the training set does not reflect the effectiveness of the proposed module. We thus use the pose trajectories extracted from other datasets [1] and test the trained model on YCB-V and T-LESS. Note that this will make the task harder as the model needs to overcome the domain gap between the training and testing data. TABLE. I shows that both GRU and LSTM can deal well with the domain gap, and the GRU performs better than LSTM by 0.9%.

[AS-3] Perspective correction: The mean recall dropped by 1.7% when no perspective correction was carried out.

[AS-4] Pose accuracy distribution under occlusion: Fig. 6 shows the pose estimation accuracy under a varying level of occlusion. We compared our method with PoseRBPF and CosyPose the on T-LESS test set of the BOP challenge. As shown in the figure, our proposed method outperforms them across all occlusion levels. Moreover, the improvements is more significant when the visibility is under 0.5.

[AS-5] No dynamic codebook matching: TABLE. I shows that the performance drops by 0.5% when only using simple codebook matching (use S^t_{obs} directly). The influence of dynamic codebook matching on different occlusion levels is shown in Fig. 6, which indicates the dynamic matching is effective on all the occlusion levels when the model is trained only on synthetic data, but in case of mixed data is more effective when the visible amount is lower than 0.5.

[AS-6] Refinement: In addition to the object tracking task, we were also interested in how our proposed method could be used to refine the pose estimation method by taking the estimated pose of other approaches as the prior pose. We report a 10.5% increase on CosyPose when using our approach as pose refinement without any iteration.

[1]The pose trajectories are extracted from the test set of OPT [44], YCB-M [48], TUD-L [39], and HO-3D [49]. Note that the size of the test set is much smaller than the training set, we thus use several datasets.

C. Comparison with State-of-the-Art Methods

Results on T-LESS dataset: We show the evaluation results on TLESS in TABLE. II. We included results from training on synthetic (syn.) data for a fair comparison with AAE and PoseRBPF. For rotation tracking comparison, we used the GT to provide 2D bounding boxes for AAE and PoseRBPF. We also reported results when using the GT poses for initialization. It is clear that our framework demonstrates better performances among its competitors on the T-LESS dataset. Qualitative results are shown in Fig. 7.

Results on YCB-V Dataset: We compare our results with state-of-the-art methods in TABLE. III. We only use 20% of the images of the YCB-V training set for training and got comparable results with other approaches.

V. Conclusion

In this paper, we proposed a novel TP-AE object pose tracking framework that can robustly handle symmetric and textureless objects under occlusion. Our method outperforms the state-of-the-art, while also running in real-time (26 FPS). We successfully demonstrated that embedding temporal information in our proposed framework can increase the pose estimation accuracy by a large margin. We also demonstrated the generalizability of our prediction network and its robustness under disturbances. In addition, we reported a thorough analysis on the effectiveness of perspective correction. As a future work, the proposed method could achieve an improved performance when combined with a refinement process.

REFERENCES

[1] L. P. Kaelbling and T. Lozano-Pérez, "Unifying perception, estimation and action for mobile manipulation via belief space planning," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2952–2959.

[2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," 09 2018.

[3] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3665–3671.

[4] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao-Blackwellized Particle Filter for6D Object Pose Estimation," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

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

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 1150–1157 vol.2.

[7] 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*, vol. 66, pp. 231–259, 03 2006.

[8] F. Manhardt, D. Arroyo, C. Rupprecht, B. Busam, N. Navab, and F. Tombari, "Explaining the Ambiguity of Object Detection and 6D Pose from Visual Data," in *IEEE International Conference on Computer Vision (ICCV)*, 12 2018.

[9] G. Pitteri, M. Ramamonjisoa, S. Ilic, and V. Lepetit, "On Object Symmetries and 6D Pose Estimation from Images," in *International Conference on 3D Vision (3DV)*, 2019, pp. 614–622.

[10] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[11] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D Pose Object Detector and Refiner," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[13] 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.

[14] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively Trained Templates for 3D Object Detection: A Real Time Scalable Approach," in *IEEE International Conference on Computer Vision (ICCV)*, 12 2013, pp. 2048–2055.

[15] 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)*, Oct 2017.

[16] T. Hodan, D. Barath, and J. Matas, "Epos: Estimating 6d pose of objects with symmetries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] G. Gao, M. Lauri, X. Hu, J. Zhang, and S. Frintrop, "Cloudaae: Learning 6d object pose regression with on-line data synthesis on point clouds," *CoRR*, vol. abs/2103.01977, 2021. [Online]. Available: https://arxiv.org/abs/2103.01977

[18] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 3003–3013.

[19] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] Y. Shi, J. Huang, X. Xu, Y. Zhang, and K. Xu, "Stablepose: Learning 6d object poses from geometrically stable patches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 222–15 231.

[21] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[22] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras, and R. Triebel, "Multi-path learning for object pose estimation across domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Z. Li and X. Ji, "Pose-guided auto-encoder and feature-based refinement for 6-dof object pose regression," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8397–8403.

[24] C. Choi and H. Christensen, "RGB-D object tracking: A particle filter approach on GPU," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11 2013, pp. 1084–1091.

[25] T. Schmidt, R. Newcombe, and D. Fox, "DART: Dense Articulated Real-Time Tracking," in *Proceedings of Robotics: Science and Systems*, 07 2014.

[26] D. J. Tan, F. Tombari, S. Ilic, and N. Navab, "A Versatile Learning-Based 3D Temporal Tracker: Scalable, Robust, Online," in *IEEE International Conference on Computer Vision (ICCV)*, 12 2015, pp. 693–701.

[27] M. Garon and J.-F. Lalonde, "Deep 6-DOF Tracking," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, 03 2017.

[28] B. Wen, C. Mitash, B. Ren, and K. Bekris, "se(3)-TrackNet: Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 07 2020.

[29] I. Marougkas, P. Koutras, N. Kardaris, G. Retsinas, G. Chalvatzaki, and P. Maragos, "How to track your dragon: A multi-attentional framework for real-time RGB-D 6-dof object pose tracking," *CoRR*, vol. abs/2004.10335, 2020. [Online]. Available: https://arxiv.org/abs/2004.10335

[30] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 657–678, 2020. [Online]. Available: https://doi.org/10.1007/s11263-019-01250-9

[31] N. A. Piga, F. Bottarel, C. Fantacci, G. Vezzani, U. Pattacini, and L. Natale, "Maskukf: An instance segmentation aided unscented kalman filter for 6d object pose and velocity tracking," *Frontiers in Robotics and AI*, vol. 8, p. 38, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.594583

[32] K. Murphy and S. Russell, *Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks*. New York, NY: Springer New York, 2001, pp. 499–515. [Online]. Available: https://doi.org/10.1007/978-1-4757-3437-9_24

[33] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[34] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://www.aclweb.org/anthology/D14-1179

[35] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the Continuity of Rotation Representations in Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2019, pp. 5738–5746.

[36] A. Kundu, Y. Li, and J. Rehg, "3d-rcnn: Instance-level 3d object reconstruction via render-and-compare," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2018, pp. 3559–3568.

[37] M. Sundermeyer, Z. Marton, M. Durner, and R. Triebel, "Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection," *International Journal of Computer Vision (IJCV)*, vol. 128, 10 2019.

[38] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Ylabbe/cosypose: Code for "cosypose: Consistent multi-view multi-object 6d pose estimation", eccv 2020. pre-trained model ids: refiner-bop-tless-synt+real–881314 and detector-bop-tless-synt+real–452847," 2020. [Online]. Available: https://github.com/ylabbe/cosypose

[39] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D Object Pose Estimation: 15th European Conference, Munich, Germany, September 8-14, Proceedings, Part X," 2018, pp. 19–35.

[40] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[41] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *Robotics: Science and Systems (RSS)*, 06 2018.

[42] T. Fäulhammer, A. Aldoma, M. Zillich, and M. Vincze, "Temporal integration of feature correspondences for enhanced recognition in cluttered and dynamic environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3003–3009.

[43] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1817–1824.

[44] P. Wu, Y. Lee, H. Tseng, H. Ho, M. Yang, and S. Chien, "[poster] a benchmark dataset for 6dof object pose tracking," in *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, 2017, pp. 186–191.

[45] T. Hodaň, J. Matas, and Š. Obdržálek, "On Evaluation of 6D Object Pose Estimation," in *ECCV 2016 Workshops*, 2016, pp. 606–619.

[46] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 858–865.

[47] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] T. Grenzdörffer, M. Günther, and J. Hertzberg, "YCB-M: A Multi-Camera RGB-D Dataset for Object Recognition and 6DoF Pose Estimation," in *International Conference on Robotics and Automation (ICRA)*, May 2020.

[49] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "HOnnotate: A method for 3D Annotation of Hand and Object Poses," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[50] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation," in *IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[51] J. Shao, Y. Jiang, G. Wang, Z. Li, and X. Ji, "Pfrl: Pose-free reinforcement learning for 6d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[52] W. Chen, X. Jia, H. J. Chang, J. Duan, and A. Leonardis, "G2L-Net: Global to Local Network for Real-Time 6D Pose Estimation With Embedding Vector Features," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.