Robot Structure Prior Guided Temporal Attention for Camera-to-Robot Pose Estimation from Image Sequence

Yang Tian* Jiyao Zhang* Zekai Yin* Hao Dong†
CFCS, Peking University

Abstract

In this work, we tackle the problem of online camera-to-robot pose estimation from single-view successive frames of an image sequence, a crucial task for robots to interact with the world. The primary obstacles of this task are the robot's self-occlusions and the ambiguity of single-view images. This work demonstrates, for the first time, the effectiveness of temporal information and the robot structure prior in addressing these challenges. Given the successive frames and the robot joint configuration, our method learns to accurately regress the 2D coordinates of the predefined robot's keypoints (e.g. joints). With the camera intrinsic and robotic joints status known, we get the camera-to-robot pose using a Perspective-n-point (PnP) solver. We further improve the camera-to-robot pose iteratively using the robot structure prior. To train the whole pipeline, we build a large-scale synthetic dataset generated with domain randomisation to bridge the sim-to-real gap. The extensive experiments on synthetic and real-world datasets and the downstream robotic grasping task demonstrate that our method achieves new state-of-the-art performances and outperforms traditional hand-eye calibration algorithms in real-time (36 FPS). Code and data are available at the project page: https://sites.google.com/view/sgtapose.

1. Introduction

Camera-to-robot pose estimation is a crucial task in determining the rigid transformation between the camera space and robot base space in terms of rotation and translation. Accurate estimation of this transformation enables robots to perform downstream tasks autonomously, such as grasping, manipulation, and interaction. Classic camera-to-robot estimation approaches, *e.g.* [11, 14, 33], typically involve attaching augmented reality (AR) tags as markers to the end-effector and directly solving a homogeneous matrix equation to calculate the transformation. However, these approaches have critical drawbacks. Capturing multiple joint configurations and corresponding images is always troublesome, and these methods cannot be used online. These flaws become greatly amplified when downstream tasks require frequent camera position adjustment.

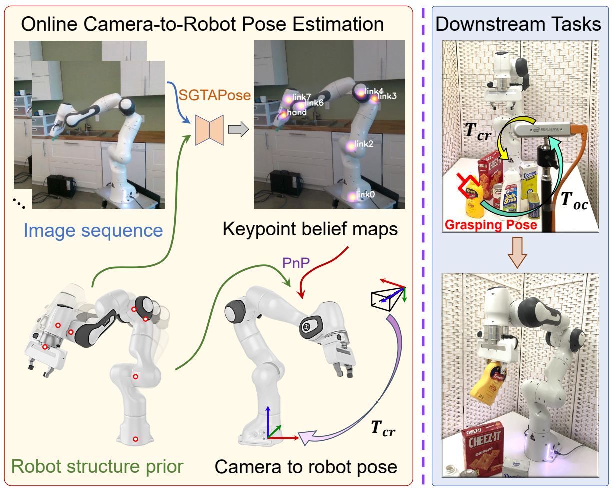

Figure 1. **Overview of the proposed SGTAPose.** Given a temporal sequence of RGB frames and known robot structure priors, our method estimates the 2D keypoints (*e.g.*, joints) of the robot and performs real-time estimation of the camera-to-robot pose by combining a *Perspective-n-point* (PnP) solver(left). This real-time camera-to-robot pose estimation approach can be utilised for various downstream tasks, such as robotic grasping(right).

To mitigate this limitation of classic offline hand-eye calibration, some recent works [19, 20] introduce vision-based methods to estimate the camera-to-robot pose from a single image, opening the possibility of online hand-eye calibration. Such approaches significantly grant mobile and itinerant autonomous systems the ability to interact with other robots using only visual information in unstructured environments, especially in collaborative robotics [21].

Most existing learning-based camera-to-robot pose estimation works [19, 21, 26, 30] focus on single-frame estimation. However, due to the ambiguity of the single-view image, these methods do not perform well when the robotic arm is self-occluded. Since the camera-to-robot pose is likely invariant during a video sequence and the keypoints are moving continually, one way to tackle this problem is

*: Equal contributions. †: Corresponding author

to introduce temporal information. However, a crucial technical challenge of estimating camera-to-robot pose temporally is how to fuse temporal information efficiently. To this end, as shown in Fig. 1, we propose Structure Prior Guided Temporal Attention for Camera-to-Robot Pose estimation (SGTAPose) from successive frames of an image sequence. First, we proposed robot structure priors guided feature alignment approach to align the temporal features in two successive frames. Moreover, we apply a multi-head-cross-attention module to enhance the fusion of features in sequential images. Then, after a decoder layer, we solve an initial camera-to-robot pose from the 2D projections of detected keypoints and their 3D positions via a PnP solver. We lastly reuse the structure priors as an explicit constraint to acquire a refined camera-to-robot pose.

By harnessing the temporal information and the robot structure priors, our proposed method gains significant performance improvement in the accuracy of camera-to-robot pose estimation and is more robust to robot self-occlusion. We have surpassed previous online camera calibration approaches in synthetic and real-world datasets and show a strong dominance in minimising calibration error compared with traditional hand-eye calibration, where our method could reach the level of 5mm calibration errors via multi-frame PnP solving. Finally, to test our method's capability in real-world experiments, we directly apply our predicted pose to help implement grasping tasks. We have achieved a fast prediction speed (36FPS) and a high grasping success rate. Our contributions are summarised as follows:

- For the first time, we demonstrate the remarkable performance of camera-to-robot pose estimation from successive frames of a single-view image sequence.
- We propose a temporal cross-attention strategy absorbing robot structure priors to efficiently fuse successive frames' features to estimate camera-to-robot pose.
- We demonstrate our method's capability of implementing downstream online grasping tasks in the real world with high accuracy and stability, even beyond the performance of classical hand-eye calibration.

2. Related Works

Instance-level 6D Object Pose Estimation. Given RGB images and the robot's CAD model, the camera-to-robot pose is solely determined by the 6D pose of the robot base. Therefore, our objective is highly correlated with the instance-level 6D rigid object pose estimation [17, 45]. Its goal is to infer an object's 6D pose given a reference frame by assuming the exact 3D CAD model is available. Traditional methods, including iterative closest point (ICP) [2], perform template matching by aligning CAD models with the observed pointclouds. Some recent works [15,25,29,32] regard the pose estimation as a regression or a classification task. 2D parameter representation or geometry-guided features will be predicted [8, 24, 31], and then improved PnP solvers [7, 36] are used to estimate poses. Although these works are closely related to ours, the manipulator is an articulated object with several degrees of freedom and potential entangling parts, whose pose is tougher to estimate.

2D Center-based Object Detection and Tracking. Our approach predicts camera-to-robot pose via keypoint estimation, which shares similar goals in 2D center-based object detection and tracking. The center-based method [9, 44] has been an emerging anchor-free object detection method in recent years, which models an object as a single point via keypoint estimation and regresses other object properties such as bounding boxes, 3D locations, or poses [37, 38, 42]. Some works [6, 27, 39, 43] also extend these center-based models to tracking tasks such as multi-category tracking and pose tracking, applying a detection model to a pair of images and detections from the previous frame. These center-based methods [16, 22, 46] have succeeded in simplicity and speed. However, the methods mentioned [40, 43, 44] mainly adopt simple concatenations to fuse temporal information and thus overlook accurate pixel-wise correspondence, where we propose a robot structure guided feature alignment module and a temporal cross-attention module to produce better feature fusion.

Robot Arm Pose Estimation. Recently, many learning-based camera-to-robot pose estimation methods have been proposed, which can be divided into three types. The first type falls into keypoint-based methods via a single RGB image. For example, DREAM [21] designs a CNN-based pipeline to regress 2D keypoints, construct 2D-3D correspondence and recover the camera-to-robot pose via a PnP-solver. Lately, [26] seeks to find the optimal 2D keypoint candidates for better acquiring pose estimations. The second type falls into rendering-based methods given robots' 3D CAD models [19, 47]. Robopose [19] optimises the camera-to-robot pose by iteratively rendering images and comparing them with ground truth. This method requires a long time (1s) to predict an excellent initial pose and fails in dynamic scenarios. The last type falls into depth-based methods [3, 30], which rely highly on depth sensors' precision and lack high accuracy in real-world experiments.

3. Method

We will first introduce the problem statement in Section 3.1. Then we will explain three modules, **Structure Prior Guided Feature Alignment**, **Temporal Cross Attention Enhanced Fusion**, and **Pose Refiner** in our approach. **Structure Prior Guided Feature Alignment** aims to align corresponding features between frames and is discussed in Section 3.2. **Temporal Cross Attention Enhanced Fusion** targets fusing temporal information and is discussed in Section 3.3. Finally, we describe **Pose Refiner**

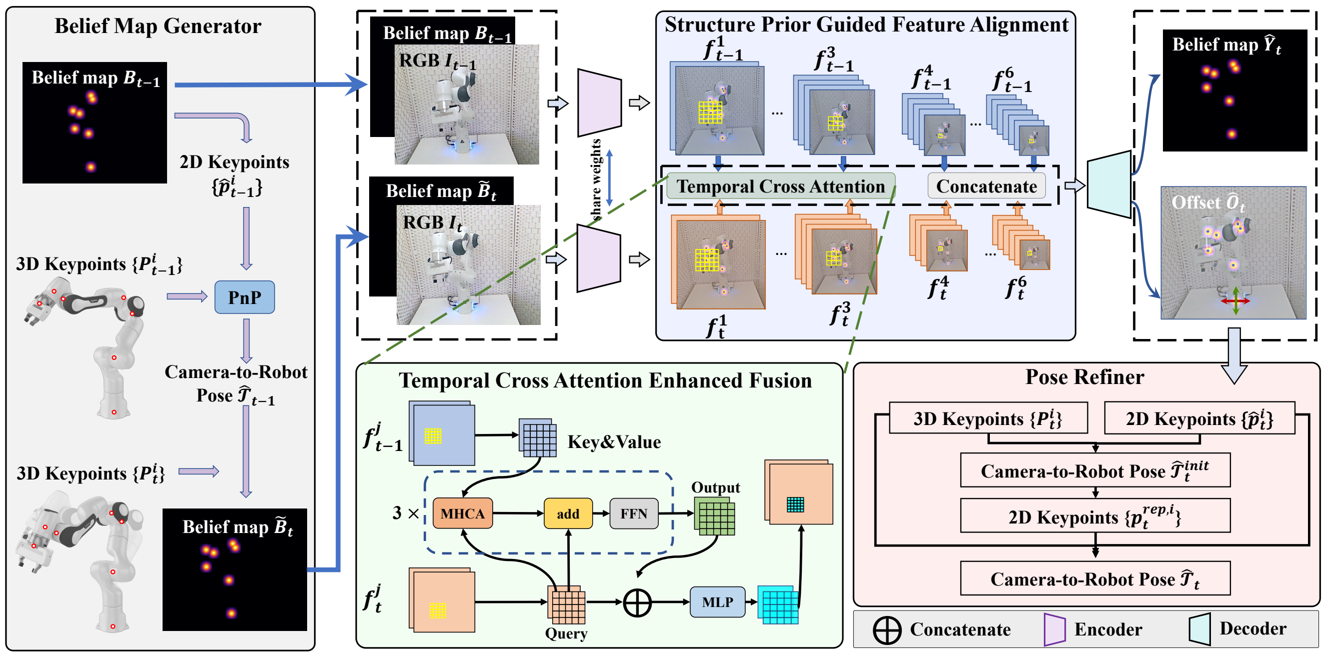

Figure 2. **Pipeline of the proposed SGTAPose.** (a) **Structure Prior Guided Feature Alignment** We use a **Belief Map Generator** to acquire belief maps as the structure prior. Given 2D/3D keypoints' locations $\{\hat{p}^i_{t-1}\}$ and $\{P^i_{t-1}\}$ from the previous frame I_{t-1}, an initial camera-to-robot pose is calculated and used to project current keypoints' 3D locations $\{P^i_t\}$ to a reprojected belief map $\tilde{B}_t$. The paired inputs are sent to a shared encoder and yield multi-scale features f^j_{t-1} and $f^j_t, j \in [6]$. (b) **Temporal Cross Attention Enhanced Fusion** For $j \in \{1,2,3\}$, we implement temporal cross attention in the regions around the detected keypoints in B_{t-1} and $\tilde{B}_t$ to efficiently fuse features from successive frames. (c) **Pose Refiner.** We acquire the detected keypoints' 2D locations and solve an initial camera-to-robot pose $\hat{\mathcal{T}}^{init}_t$. Then we gain a refined camera-to-robot pose $\hat{\mathcal{T}}_t$ with structure priors via a weighted Levenberg-Marquardt (LM) solver.

in Section 3.4 to refine the predicted pose better.

3.1. Problem Statement

Our problem is defined as follows: Given a live stream of RGB images $\{I_t\}_{t\geq 0}$ containing a manipulator along with its instant camera-to-robot pose $\{\mathcal{T}_t\}_{t\geq 0} = \{(R_t, T_t)\}_{t\geq 0} \in SE(3)$, the transformation from the camera space to the robot base space, our objective is to track the camera-to-robot pose in an online manner. In other words, at timestep t, provided images I_{t-1} and I_t, predefined keypoints' 3D positions $\{P^i_t\}$ in robot space (i denotes the index of each keypoint), camera intrinsics and the estimated pose $\hat{\mathcal{T}}_{t-1}$, we predict the rotation matrix R_t and translation T_t in $\hat{\mathcal{T}}_t$.

3.2. Structure Prior Guided Feature Alignment

Inspired by the fact that the joints' states and pose of the robot change slightly between successive two frames, we believe the estimated previous pose $\hat{\mathcal{T}}_{t-1}$ will work as solid structure priors for guiding network learning. In this way, our first step is to design a **Belief Map Generator** to produce a reprojection belief map. Suppose we have acquired the previous estimated pose $\hat{\mathcal{T}}_{t-1}$, we reproject the instant keypoints' 3D positions $\{P^i_t\}$ into $\{\tilde{p}^i_t\}$, and visualise them into a single-channel belief map $\tilde{B}_t \in [0,1]^{H\times W\times 1}$ (See in the left column of Figure 2) . Based on $\tilde{B}_t$, the network cares more about the residuals of $\tilde{B}_t$ to its ground truth.

Similarly, we also prepare the previous belief map B_{t-1} for guiding the network to align the previous frame's features. To be more specific, during the training process, B_{t-1} is produced by augmenting ground truth 2D keypoints. While during inference, we utilise 2D keypoints $\{\hat{p}^i_{t-1}\}$ estimated from previous network output belief map $\hat{Y}_{t-1} \in \mathbb{R}^{\frac{H}{R}\times\frac{W}{R}\times c}$ and $\hat{O}_{t-1} \in \mathbb{R}^{\frac{H}{R}\times\frac{W}{R}\times 2}$ where R and c denotes the downsampling ratio and amounts of keypoints.

Then, we send the pairs (I_{t-1}, B_{t-1}) and $(I_t, \tilde{B}_t)$ to a shared backbone and obtain two lists composed of 6 multi-scale features L_{t-1} and L_t. We denote L (for simplicity, we omit the subscript t) as $[f^1, \ldots, f^6]$ where $f^j \in \mathbb{R}^{c_j\times h_j\times w_j}$ and $\frac{c_{j+1}}{c_j} = \frac{h_j}{h_{j+1}} = \frac{w_j}{w_{j+1}} = 2$. $\{\hat{p}^i_{t-1}\}$ and $\{\tilde{p}^i_t\}$ are rescaled to match the f's size accordingly and treated as center proposals. Features extracted from the neighbourhoods of center proposals in f_{t-1} and f_t are regarded as aligned since they roughly entail the same keypoint's contextual information. The aligned features should be fused carefully, illustrated in the next section.

3.3. Temporal Cross Attention Enhanced Fusion

To carefully integrate the multi-scale aligned features in L_{t-1} and L_t, we adopt different strategies considering the size of f^j For $f^m, m \in \{1,2,3\}$, they have much higher resolutions and fine-grained features, which are crucial for detecting small-sized keypoints. Therefore, we propose a temporal cross-attention module to fuse the features at the neighbourhood of center proposals. In comparison, for $f^n, n \in \{4,5,6\}$, they have lower resolutions and a broader receptive field, so each pixel-level feature contains more contextual and temporal information. We thus directly concatenate the features at the center proposals of f^n_{t-1} and f^n_t and process them into original sizes $\mathbb{R}^{c \times c_n}$ via a shallow Multilayer Perceptron (MLP). The newly processed features will instantly replace the counterpart in f^n_t.

For the first three feature maps $\{f^m, m \in \{1,2,3\}\}$, we treat $\{\hat{p}^i_{t-1}\}$ and $\{\tilde{p}^i_t\}$ as center proposals respectively, and rescale them to match the size of f^m. Having measured the motion amplitude of the manipulator in finishing downstream tasks (e.g., Grasping), we confine the center proposals to a square area with window size d_m. We take f^m_{t-1} at the $d_m \times d_m$ window area around scaled $\{\hat{p}^i_{t-1}\}$ as query embeddings and f^m_t at the same size of area around $\{\tilde{p}^i_t\}$ as keys and values. After 3 vanilla Tranformer multi-head cross-attention layers [35], we concatenate the output features $Q^m_{t,d^2_m c} \in \mathbb{R}^{d^2_m c \times c_m}$ with $f^m_{t,d^2_m c}$ at the same $d^2_m c$ locations along the c_m-dimension, and send them through a shallow MLP to get $\hat{f}^m_{t,d^2_m c}$. We directly replace $f^m_{t,d^2_m c}$ with $\hat{f}^m_{t,d^2_m c}$ and pass all the six processed multi-scale features to the decoder layer and receive the output head.

3.4. Pose Refiner

We design a pose refiner to mitigate the influence of outlier keypoints with significant reprojection errors when computing the camera-to-robot pose. Since the initial pose solved by *Perspective-n-Point* (PnP) algorithms might be inaccurate sometimes due to outliers [12], we correct such bias by solving reweighted PnP problems. Utilising predicted projections $\{\hat{p}^i_t\}$ and known $\{P^i_t\}$, we obtain an initial camera-to-robot pose $\hat{\mathcal{T}}^{init}_t$ via a PnP-RANSAC solver [23]. Next, we project $\{P^i_t\}$ via $\hat{\mathcal{T}}^{init}_t$ to 2D coordinates $\{p^{rep,i}_t\}$. We set the weights $\omega^i_t = \exp(-5 \times \|\hat{p}^i_t - p^{rep,i}_t\|^2)$ based on practical experience and optimise the following equation via an LM solver [28].

$$\arg\min_{R_t,T_t} \frac{1}{2} \sum_{i=1}^{c} \|\omega^i_t(\pi(R_t P^i_t + T_t) - \hat{p}^i_t)\|^2 \quad (1)$$

where $\pi(\cdot)$ is the projection function, R_t and T_t is the rotation and translation in the camera-to-robot pose $\hat{\mathcal{T}}_t$ .The reweighted optimisation objective focuses more on the "influence" of comparatively precise predictions, thus mitigating the impact of keypoints with large reprojection errors.

3.5. Implementation details

Loss function. Our network output involves a predicted pixel-level belief map $\{\hat{Y}_t\} \in [0,1]^{\frac{H}{R} \times \frac{W}{R} \times c}$ and subpixel-level local offsets $\{\hat{O}_t\} \in [0,1]^{\frac{H}{R} \times \frac{W}{R} \times 2}$. We design two loss functions L_B and L_{off} for $\{\hat{Y}_t\}$ and $\{\hat{O}_t\}$ respectively. For keypoints' ground truth 2D locations $\{p^i_t\}$, we scale them into a low-resolution equivalence $p^i_{low,t} = \lfloor \frac{P^i_t}{R} \rfloor$. We draw each keypoint in a single-channel feature map with a Gaussian Kernel $K(x,y) = \exp(-\frac{(x-p^i_{low,t,x})^2+(y-p^i_{low,t,y})^2}{8})$ and shape the ground truth belief map $Y_t \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times c}$. The L_B becomes:

$$L_B = \|Y_t - \hat{Y}_t\|^2_{L_2} \quad (2)$$

where $R = 4$ in our network. We further follow [43] to correct the error induced by the output stride. The offsets$\{\hat{O}_t\}$ are trained via smooth L_1 loss and only supervised in locations p^i_{low}:

$$L_{off} = \|\hat{O}_{t_{p^i_{low,t}}} - (\frac{p^i_t}{R} - p^i_{low,t})\| \quad (3)$$

The overall training objective is designed as follows:

$$L = \lambda_B L_B + \lambda_{off} L_{off} \quad (4)$$

where λ_B = 1.0 and $\lambda_{off} = 0.01$ in implementation.

Training details. During the training time, we preprocess the input image I_{t-1}, I_t into the size of $\mathbb{R}^{480 \times 480 \times 3}$ via affine transformation and normalisation with mean $[0.5, 0.5, 0.5]$ and standard deviation $[0.5, 0.5, 0.5]$. To further improve our model's robustness, we apply $\mathcal{N}(0, 1.5I)$ noises as well as randomly drop with probability 0.2 to the ground truth keypoints in B_{t-1}. Our backbone is based on Deep Layer Aggregation [41] and trained for 20 epochs with batch size 16, Adam optimiser [18] with momentum 0.9 and 180k synthetic training images. The learning rate warms up to 1.25e-4 from 0 during the first 3,000 iterations and drops to 0 during the rest of the iterations linearly.

Inference details. During inference, we are given a long-horizon video split into consecutive frames. We use the first frame as I_0 and I_1, and blank images as the initial belief maps B_0 and B_1 to perform inference. For each timestep $t > 1$, we select the keypoints' 2D locations p^i_{low} with the largest confidence score for each belief map in $\hat{Y}_{t-1} \in \mathbb{R}^{\frac{H}{R} \times \frac{W}{R} \times c}$. We then determine the accurate locations by adding $\hat{O}_{t_{p^i_{low,t}}}$ to $p^i_{low,t}$. Finally, we rescale these low-resolution keypoints' locations to match the raw image's size via inverse affine transformation and obtain $\{\hat{p}^i_t\}$.

4. Experiments

4.1. Datasets

Our datasets involve one self-generated synthetic training set (Panda Syn Training), one self-generated synthetic

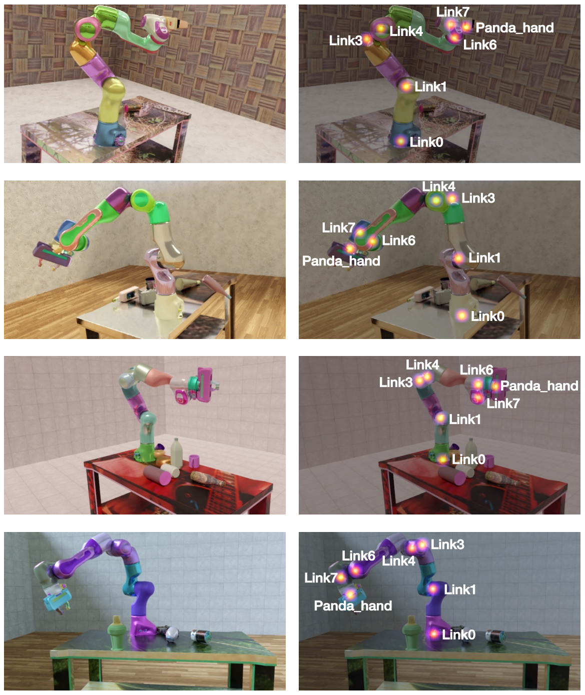

Figure 3. Sample images from Panda Syn Training (first column) and annotations (second column). We set every part a random colourful tint to improve the diversity of our datasets and apply several domain randomisations to shorten the sim-to-real gap.

testing set (Panda Syn Testing), and three real-world testing sets (Panda 3CAM-RS, Panda 3CAM-AK, and Panda Orb) provided by [21]. Since the training set proposed in [21] doesn't support temporal images, we thus generate Panda Syn Training/Testing. The three public real-world sets are generated by an externally mounted camera filming a Franka Emika Panda manipulator according to [21]. Panda 3CAM-AK is collected by a Microsoft Azure Kinect camera, and Panda 3CAM-RS/Orb are collected by Intel RealSense D415. Panda 3CAM-AK and Panda 3CAM-RS are captured from a single fixed view and involve about 6k images respectively, while Panda Orb is captured from 27 different views and involves approximately 32k images.

Utilising the open source tool Blender [1], we construct large-scale synthetic sets Panda Syn Training/Testing (Figure 3). We generate frames in a video at FPS 30 with a fixed scene and a moving manipulator. Several domain randomisations have been used to shorten the sim-to-real gap and will be detailedly explained in supplementary materials. The synthetic dataset consists of temporal RGB images, 2D/3D predefined keypoints' locations, and part poses. Overall, Panda Syn Training involves approximately 60k videos which contain 3 successive frames per video (180k images), and Panda Syn Testing involves 347 videos which contain 30 successive frames per video (10k images).

4.2. Baselines and Evaluations

Baselines. We compare our framework with previous online keypoint-based camera-to-robot pose estimation approaches and center-based object detection methods via single or multi frames. Besides, we also compare our framework with traditional offline hand-eye calibration. **Dream** [21]: A pioneering approach for estimating the camera-to-robot pose from a single frame by direct 2D keypoints regression and PnP-RANSAC solving. **CenterNet** [44]: A single-frame object detection approach that models an object as a single point. We adapt CenterNet to estimate 2D keypoints and reserve the heatmap loss and offset loss [44] during training. **CenterTrack** [43]: A multi-frame object detection and tracking approach. We adapt CenterTrack using similar strategies in CenterNet to detect keypoints.
Metrics. During the inference, we evaluate both 2D and 3D metrics across all datasets. **PCK**: The L2 errors between the 2D projections of predicted keypoints and ground truth. Only keypoints that exist within the frame will be considered. **ADD**: The average Euclidean norm between 3D locations of keypoints and corresponding transformed versions, which directly measures the pose estimation accuracy. For PCK and ADD, we compute the area under the curve (AUC) lower than a fixed threshold (12 pixels and 6cm, respectively) and their median values.

4.3. Results and Analysis

As can be seen in Table 1, our method has outperformed all other baselines in PCK and ADD metrics across datasets. Specifically, our median PCK, an important measure that reflects the 2D accuracy, is superior to others, indicating that we have achieved better overall precision in predicting the 2D projections. We also surpass all other baselines in the AUC of ADD, and so is the median. ADD is a more direct indicator of whether the estimated pose is accurate, and we have outperformed the best of others in Panda 3CAM-AK and Panda Orb at 4.9% and 7.8%. Further, our method can reach 9.77mm and 18.12mm in median ADD in Panda 3CAM-RS and Panda Orb, comparable to the accuracy of the traditional hand-eye calibration empirically.

Compared with single-frame methods (CenterNet [44], Dream [21]), our method absorbs temporal information and shows greater robustness to self-occlusion. The analysis of robustness to self-occlusion is discussed in Section 4.6. Meanwhile, compared with the tracking-based method CenterTrack [43], our method owns the robot structure prior guiding feature alignment and temporal cross attention, which we believe facilitates our model's superiority.

4.4. Compare with Classic Hand-Eye Calibration

We further design an experiment to compare our approach with traditional hand-eye calibration (HEC) methods, implemented via the easy_handeye ROS package [34].

Dataset	Real Data	# Images	# 6D Poses	Method	PCK		ADD	
					AUC↑	Median@pix↓	AUC↑	Median@mm↓
Panda 3CAM-RS [21]	✓	5944	1	CenterNet [44]	67.38	3.51	59.26	21.25
				CenterTrack [43]	68.85	3.59	58.24	23.77
				Dream [21]	64.82	3.90	58.60	24.57
				Ours	**75.68**	**2.68**	**79.89**	**9.77**
Panda 3CAM-AK [21]	✓	6394	1	CenterNet [44]	52.38	4.90	34.07	37.56
				CenterTrack [43]	58.26	4.45	43.10	32.83
				Dream [21]	52.28	4.83	44.55	33.68
				Ours	**62.75**	**3.19**	**49.42**	**29.61**
Panda Orb [21]	✓	32315	27	CenterNet [44]	60.11	3.47	50.59	24.22
				CenterTrack [43]	61.03	3.73	47.67	25.13
				Dream [21]	57.44	3.73	52.56	22.53
				Ours	**63.28**	**3.46**	**60.30**	**18.12**
Panda Syn Testing	×	10410	347	CenterNet [44]	92.60	0.89	85.97	5.79
				CenterTrack [43]	91.86	0.63	85.01	6.18
				Dream [21]	80.79	0.85	79.05	7.13
				Ours	**94.36**	**0.44**	**89.62**	**4.11**

Table 1. **Quantitative comparison with keypoint-based methods.** ↑ means higher is better, ↓ means lower is better. The ✓ and × in Real Data denote whether the dataset is real-world. # Images and # 6D Poses denote the total amounts of images and 6D poses in the dataset respectively. For a fair comparison, we train all the methods listed in the above table on Panda Syn Training dataset and report the results regarding PCK and ADD across four testing datasets. Obviously, our method is taking the lead in all metrics upon all datasets.

We attach an Aruco Fiducial Marker [13] to the Franka Emika Panda's gripper and place an external RealSense D415 to take photos. The manipulator is commanded to move to $L = 20$ positions along a predefined trajectory and stays at each position for 1 second. To assess the accuracy, we consider choosing l positions from the set of L positions and feed all the detection results from the l positions to a single PnP solver. Specifically, we choose C_L^l combinations of l images and if $C_L^l > 2500$, we randomly sample 2500 combinations. As shown in Fig 4, our approach and traditional HEC solve a more accurate pose with increasing frames. However, our approach outperforms HEC in both median and mean ADD, where we can reach the level of 5 mm while HEC reaches 15 mm at last. Moreover, our approach is more stable than HEC with frames increasing, which can be inferred from the region surrounded by the minimum and maximum ADD.

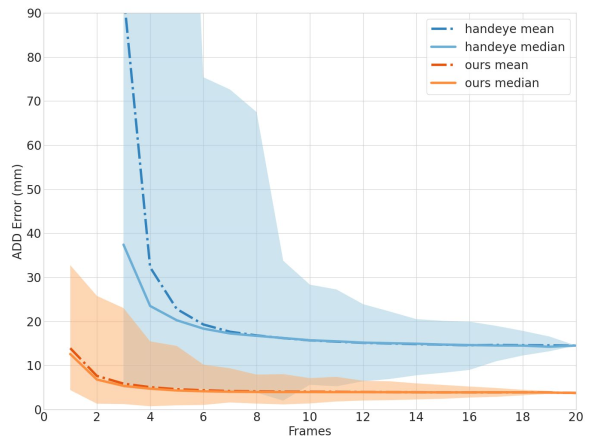

Figure 4. Comparison between classic hand-eye calibration and our approach. Results regarding ADD are calculated with increasing amounts of used frames. Mean ADD (dashed lines), median ADD (solid lines), and values between minimum and maximum (shaded areas) have been demonstrated.

4.5. Ablation Study

We conduct the ablation studies to investigate : 1) The function of the robot structure prior for guiding feature alignment. 2) The necessity of using a cross attention module for enhancing the fusion of temporal aligned features between successive frames. 3) The effectiveness of introducing the pose refiner module for updating a more accurate pose. 4) The influence of different window sizes adopted during temporal cross attention. To this end, we construct four ablated versions of our model and test their performances on the most diverse real dataset Panda Orb [21].

We report the results in Table 2. For the version without any module, we send B_{t-1}, I_{t-1}, I_t as input with a shared encoder, directly concatenating the multi-scale features. With the addition of **Structure Prior Guided Feature Alignment**, we add one more input $\tilde{B}_t$, and the performance in both PCK and ADD improves greatly. This fact reveals that the reprojection belief map $\tilde{B}_t$ provides a confined area rather than the whole image for the network to focus on, which is easier to detect keypoints. Secondly, with the addition of **Temporal Cross Attention Enhanced Fusion**, we replace the direct concatenation with cross attention. The performance in ADD improves higher than that in PCK, which shows that the temporal cross attention module facilitates the precision of 3D keypoint predictions.

SGF	TCA	PRF	PCK		ADD	
			AUC↑	Median@pix↓	AUC↑	Median@mm↓
			49.78	4.69	37.93	34.48
✓			62.86	3.49	55.83	21.04
✓	✓		**63.28**	**3.46**	58.86	19.17
✓	✓	✓	**63.28**	**3.46**	**60.30**	**18.12**

Table 2. Ablation studies with different modules. SGF, TCA, and PRF denote the **Structure Prior Guided Feature Alignment**, **Temporal Cross Attention Enhanced Fusion**, and **Pose Refiner** modules, respectively. ✓ denotes the corresponding module that has been used. Results show that all the proposed modules benefit camera-to-robot pose estimation.

Thirdly, the final introduction of **Pose Refiner** witnesses an improvement of 1.5% in the AUC of ADD and 1mm in median ADD. This proves that the reweighted optimisation objective (Equation 1) succeeds in filtering out the disturbance of outliers and recomputing a better solution.

As can be shown in Table 3, we change the window size during Temporal Cross Attention. We analyse the average movement amplitude of a manipulator between two successive frames at FPS 30 and determine the window sizes of the first three multi-scale features. We ultimately select the window size $[13, 7, 3]$ to ensure most of the keypoints' movement between two frames can be detected within the window. Results show that the larger or smaller window size will degrade the model's performance in both AUC of ADD and PCK. We believe the reason for the degradation is that a larger window size will contain more redundant information, while a smaller window size might miss some important information during fast movement tracking.

Window Size $[d_1, d_2, d_3]$	PCK		ADD	
	AUC↑	Median@pix↓	AUC↑	Median@mm↓
$[7, 3, 1]$	61.29	**3.41**	57.11	18.21
$[13, 7, 3]$	**63.28**	3.46	**60.30**	**18.12**
$[17, 9, 5]$	63.01	3.53	56.54	21.45

Table 3. Results of our ablation study with different window sizes during **Temporal Cross Attention**. The $[13, 7, 3]$ window size performs best among all choices on Panda Orb [21].

4.6. Robustness to Self-Occlusion Scenarios

Method	PCK		ADD	
	AUC↑	Median@pix↓	AUC↑	Median@mm↓
Dream [21]	49.02/59.46	4.13/3.67	43.87/54.53	30.23/21.28
CenterNet [44]	52.86/61.77	4.00/3.39	39.05/53.21	33.73/22.62
CenterTrack [43]	54.43/62.54	4.31/3.62	30.37/51.58	42.96/22.52
Ours	**60.02/64.03**	**3.95/3.37**	**59.37/60.51**	**20.18/17.69**

Table 4. Quantitative results of the self-occlusion experiment. The Left and right sides of '/' are the results of severe and no self-occlusion respectively. Results show our method performs robustly when encountering self-occlusion while baselines drop greatly.

We perform an additional experiment to show our model's robustness to self-occlusion. We divide Panda Orb [21] into 89 videos, including 31 videos with severe self-occlusion (5971 images) and 58 videos with less or no self-occlusion (26344 images). Results in Table 4 show baselines drop greatly when encountering occlusion, while ours decreases slightly. Also, from Figure 5, we can see that baselines detect occluded keypoints with significant deviation or even fail. In comparison, our model shows more precise predictions. The main reason for our model's robustness is that we adopt temporal information fusion and utilise structure priors efficiently rather than estimating pose from a single frame or concatenating temporal features crudely.

4.7. Robot Grasping Experiments

In this section, we construct real-world robotic grasping experiments to demonstrate the performance of our method.

Experimental Protocol. We perform two experiments using the Franka Emika Panda robot. One of the experiments focuses on robot grasping in a static environment, and another on grasping in a dynamic environment. To ensure a fair comparison, GraspNet [10] is used to estimate the robot grasping pose in all experiments, while different methods are employed for camera-to-robot pose estimation. In the experiments, all learning-based camera-to-robot pose estimation methods use 30 frames to estimate the camera-to-robot pose, and all the objects are selected from YCB [4] dataset. For hand-eye calibration, in the static experiment, we use easy_handeye [34] to acquire the pose. In the dynamic experiment, hand-eye calibration is incapable of online calibration, so their results are none.

In the static experiment, we conduct six scenes, and each scene includes 4-7 randomly chosen objects. During the completion of a grasping task in one scene, the camera remains stationary. After finishing grasping in a particular scene, we move the camera to another position for the next scene. In the dynamic experiment, we not only change the camera pose when switching between different scenes but also adjust the camera pose after completing the grasping of each object during the execution of a grasping task in the same scene, which is a tougher setting than the static one.

Metrics. To evaluate the performance accurately, we follow the grasping metrics from SuctionNet [5]. We adopt R_{grasp}, the ratio of the number of successful grasps to the number total grasps, and R_{object}, the ratio of the number of successfully cleared objects to the number of totals. Moreover, if three consecutive grasping attempts fail in a scene, we consider the experiment for that scene terminates.

Method	R_{grasp} ↑	R_{object} ↑
easy_handeye [34]	28/52 = 53.8%	28/32 = 87.5%
DREAM [21]	26/52 = 50.0%	26/32 = 81.2%
CenterTrack [43]	26/60 = 43.3%	26/32 = 81.2%
CenterNet [44]	29/51 = 56.9%	29/32 = 90.6%
Ours	**32/48 = 66.7%**	**32/32 = 100%**

Table 5. Quantitative comparison of the performance of different methods applied to robot grasping tasks in the static experiment.

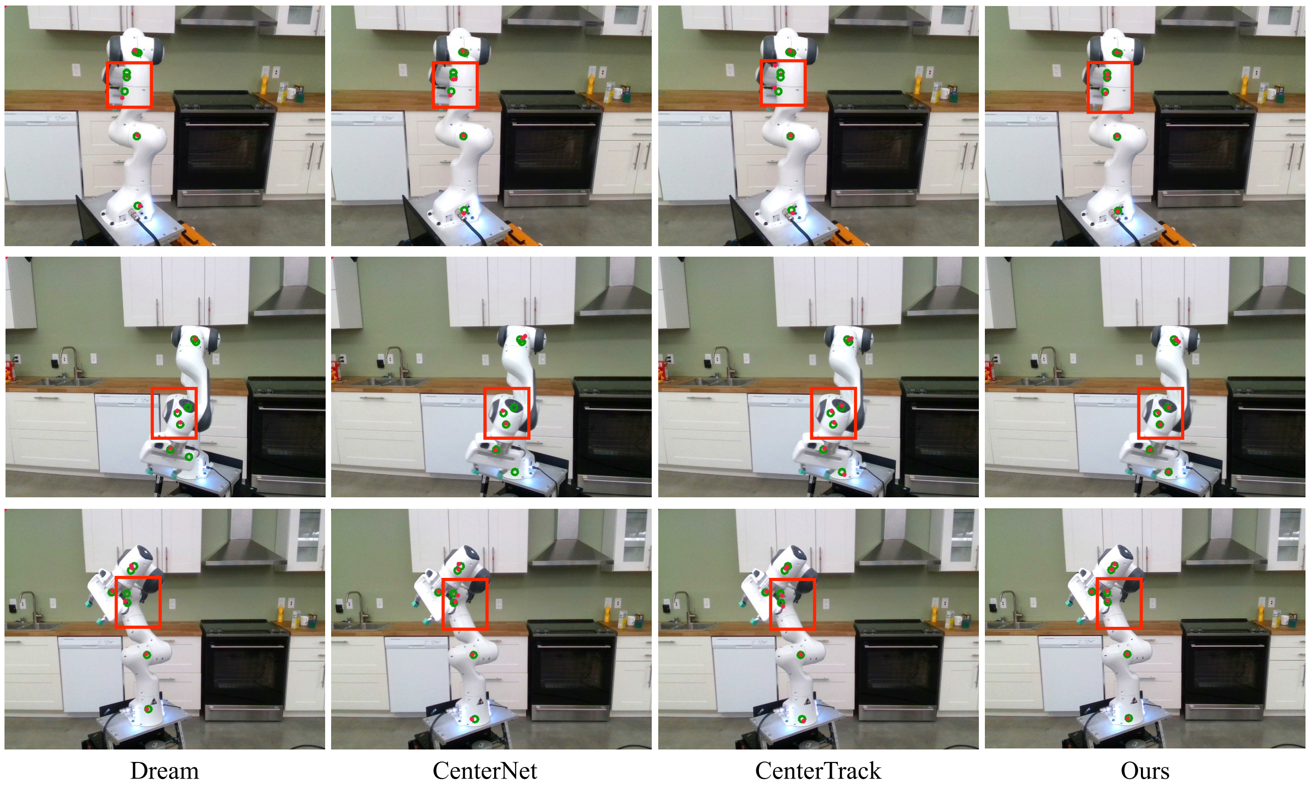

Figure 5. Comparison with keypoint-based methods in severe self-occlusion scenarios. For a fair comparison, we retrain all the baselines on Panda Syn Training and show visualisations when encountering severe self-occlusion on the largest dataset Panda Orb [21]. The green circles and red points denote the ground truth and estimated keypoints, respectively. Red boxes highlight occluded regions where our method performs much better than all baselines. This vividly demonstrates that our method is more robust to self-occlusion.

Method	R_{grasp} ↑	R_{object} ↑
easy_handeye [34]	-	-
DREAM [21]	15/38 = 39.5%	15/32 = 46.9%
CenterTrack [43]	24/51 = 47.1%	24/32 = 75.0%
CenterNet [44]	16/42 = 38.0%	16/32 = 50.0%
Ours	**30/51 = 58.8%**	**30/32 = 93.8%**

Table 6. Quantitative comparison of different methods' performance applied to robot grasping tasks in the dynamic experiment. The blank in the "easy_handeye" column is because this traditional calibration method cannot be performed online.

Results and Analysis. Table 5 and Table 6 show the results of applying different camera-to-robot pose estimation methods to the grasping experiments in static and dynamic environments, respectively. In the static experiment, our method achieves a 100% object grasping success rate, outperforming all other baselines, including traditional hand-eye calibration. In the dynamic experiment, since the camera pose changes after each grasping attempt in the same scene, this places higher demands on the robustness and speed of the camera-to-robot pose estimation. Other methods experience significant drops in grasping success rates, while our method maintains a very high success rate with only a slight decrease compared to the static experiment. These experiments demonstrate the accuracy and stability of our method for camera-to-robot pose estimation.

5. Conclusion and Future Work

In this paper, we study the camera-to-robot pose estimation using single-view successive frames from an image sequence. By leveraging the robot structure priors, we use a temporal attention mechanism to efficiently fuse keypoint features from different frames. Our method demonstrates significant improvements over synthetic and real-world datasets, strong dominance compared with traditional hand-eye calibration and high accuracy and stability in downstream grasping tasks. One limitation of our method is that although domain randomisation can narrow the sim-to-real gap to some extent, generalising to arbitrary scenes remains a significant challenge. To address this limitation, future work could explore domain adaptation between real and simulated scenes and fine-tuning in real-world settings.

Acknowledgements

This project was supported by the National Natural Science Foundation of China (No. 62136001). We would also like to thanks the lab mates for the helpful discussion.

References

[1] Blender. https://www.blender.org/. 5

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 2

[3] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3143–3150, 2014. 2

[4] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015. 7

[5] Hanwen Cao, Hao-Shu Fang, Wenhai Liu, and Cewu Lu. Suctionnet-1billion: A large-scale benchmark for suction grasping. *IEEE Robotics and Automation Letters*, 6(4):8718–8725, 2021. 7

[6] Mohamed Chaabane, Peter Zhang, J. Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *ArXiv*, abs/2102.02267, 2021. 2

[7] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2771–2780, 2022. 2

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 2

[9] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019. 2

[10] Haoshu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11441–11450, 2020. 7

[11] Irene Fassi and Giovanni Legnani. Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb. *Journal of Robotic Systems*, 22(9):497–506, 2005. 1

[12] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–508, 2014. 4

[13] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.*, 47:2280–2292, 2014. 6

[14] Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210, 1995. 1

[15] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3283–3292, 2021. 2

[16] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019. 2

[17] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538, 2017. 2

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 4

[19] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render & compare. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1654–1663, 2021. 1, 2

[20] Jens Lambrecht, Philipp Grosenick, and Marvin Meusel. Optimizing keypoint-based single-shot camera-to-robot pose estimation through shape segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13843–13849. IEEE, 2021. 1

[21] Timothy E. Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9426–9432, 2020. 1, 2, 5, 6, 7, 8

[22] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13903–13912, 2019. 2

[23] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1444–1450, 2012. 4

[24] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision*, 128:657–678, 2019. 2

[25] Jiehong Lin, Hongyang Li, Ke Chen, Jiangbo Lu, and Kui Jia. Sparse steerable convolutions: An efficient learning of se(3)-equivariant features for estimation and tracking of object poses in 3d space. In *NeurIPS*, 2021. 2

[26] Jingpei Lu, Florian Richter, and Michael C Yip. Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer. *IEEE Robotics and Automation Letters*, 7(2):4622–4629, 2022. 1, 2

[27] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8834–8844, 2021. 2

[28] M. Hashem Pesaran, Aman Ullah, and Takashi Yamagata. A bias-adjusted lm test of error cross-section independence. *Wiley-Blackwell: Econometrics Journal*, 2008. 4

[29] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3848–3856, 2017. 2

[30] Alessandro Simoni, Stefano Pini, Guido Borghi, and Roberto Vezzani. Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps. *IEEE Robotics and Automation Letters*, 7:11569–11576, 2022. 1, 2

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 2

[32] Chen Song, Jiaru Song, and Qi-Xing Huang. Hybridpose: 6d object pose estimation under hybrid representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 428–437, 2020. 2

[33] Klaus H Strobl and Gerd Hirzinger. Optimal hand-eye calibration. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 4647–4653. IEEE, 2006. 1

[34] Roger Y. Tsai and Reimar K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. Robotics Autom.*, 5:345–358, 1989. 5, 7, 8

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[36] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Pnp-detr: Towards efficient visual analysis with transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4641–4650, 2021. 2

[37] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. *ArXiv*, abs/2007.02846, 2020. 2

[38] Haoran Wei, Xin Chen, Lingxi Xie, and Qi Tian. Cornerformer: Purifying instances for corner-based detectors. In *European Conference on Computer Vision*, 2022. 2

[39] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12347–12356, 2021. 2

[40] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11779–11788, 2021. 2

[41] Fisher Yu, Dequan Wang, and Trevor Darrell. Deep layer aggregation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 4

[42] Mohsen Zand, Ali Etemad, and Michael A. Greenspan. Objectbox: From centers to boxes for anchor-free object detection. In *European Conference on Computer Vision*, 2022. 2

[43] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ArXiv*, abs/2004.01177, 2020. 2, 4, 5, 6, 7, 8

[44] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *ArXiv*, abs/1904.07850, 2019. 2, 5, 6, 7, 8

[45] Yingzhao Zhu, Man Li, Wensheng Yao, and Chunhua Chen. A review of 6d object pose estimation. *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 10:1647–1655, 2022. 2

[46] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111:257–276, 2019. 2

[47] Yiming Zuo, Weichao Qiu, Lingxi Xie, Fangwei Zhong, Yizhou Wang, and Alan Loddon Yuille. Craves: Controlling robotic arm with a vision-based economic system. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4209–4218, 2018. 2