2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610844

PoseFusion: Multi-Scale Keypoint Correspondence for Monocular Camera-to-Robot Pose Estimation in Robotic Manipulation

Xujun Han[1], Shaochen Wang[1], Xiucai Huang[2], and Zhen Kan[1]

Abstract— **Visual-based robot pose estimation is a fundamental challenge, involving the determination of the camera's pose with respect to a robot. Conventional methods for camera-to-robot pose calibration rely on fiducial markers to establish keypoint correspondences. However, these approaches exhibit significant variability in accuracy and robustness, particularly in 2D keypoint detection. In this work, we present an end-to-end pose estimation approach that achieves camera-to-robot calibration using monocular images and keypoint information. Our method employs a two-level nested U-shaped architecture, featuring a bottom-level residual U-block to extract richer contextual information from diverse receptive fields to enhance keypoint refinement. By incorporating the perspective-n-point (PnP) algorithm and leveraging 3D robot joint keypoints, we establish correspondence of 3D coordinate points between the robot's coordinate system and the camera's coordinate system, facilitating accurate pose estimation. Experimental evaluations encompass real-world and synthetic datasets, demonstrating competitive results across three distinct robot manipulators.**

I. Introduction

In unstructured and dynamically changing environments [1], the precise estimation of camera-to-robot pose stands as a paramount necessity. This estimation process serves to transform environmental information observed visually into the robot coordinate frame, thereby facilitating a spectrum of downstream tasks encompassing interaction, manipulation, and grasping. The crux of the challenge resides in translating measurements acquired in camera space into the domain of the robot's task space. Consequently, the determination of the camera-to-robot pose emerges as a fundamental quandary within the discipline of robot pose estimation.

While robotic arms of commercial-grade quality, exemplified by entities such as the Barrett WAM arm, Franka Emika Panda, or Kinova JACO, can attain a commendable degree of precision in joint state measurement through the judicious utilization of encoders, a different scenario unfolds when we consider cost-effective robotic arms often characterized by potential limitations in joint encoder resolution and backlash. In such instances, the proposition of harnessing computer vision techniques for the estimation of joint angles or end-effector poses emerges as an attractive and viable solution.

Traditional methodologies, such as the manual calibration of cameras employing fiducial markers[2], [3], present a series of substantial challenges. These approaches frequently necessitate offline calibration procedures, entailing the labor-intensive task of maneuvering the robot across diverse joint configurations. Furthermore, these methods make a static assumption regarding the relationship between the camera and the robot, rendering them susceptible to inaccuracies even in the presence of minor perturbations in the camera or robot position. The pervasive issue of self-occlusion further compounds these challenges, impacting the visibility of fiducial markers.

To surmount these formidable challenges, recent research endeavors have ventured into the realm of deep learning techniques for keypoint detection and camera-to-robot pose estimation [4]. Deep learning [5], [6] offers potent tools for enhancing robot vision capabilities [7], [8], enabling the localization of predefined keypoints within monocular images and the subsequent reconstruction of the robot's pose [9]. For rendering-based method [10], they employ the rendered images and the masks of an anchor part to estimate the robot's new pose. Subsequently, the iteration is completed by comparing with ground truth pose. Their method requires more time when estimating the initial robot pose. For keypoint-based method [11], they achieve the purpose of pose estimation through the marker-less detection of optimized keypoints. The inaccuracy of 2D keypoint detection and the success rate related to the general problem of perspective-n-point(PnP) algorithms still affect the practical performance of pose estimation [12].

In this work, our proposed approach intricately amalgamates multi-scale information extraction with the PnP transformation. To this end, we employ nested U-structures designed to capture contextual information spanning various scales. This strategic fusion culminates in heightened precision and enhanced robustness in keypoint detection. In terms of training data, our approach capitalizes on automatically generated synthetic images from [4], effectively bridging the divide between the real and simulated domains through domain randomization. Our system undergoes rigorous validation on industrially advanced robot arms, thereby substantiating its superiority in pose estimation accuracy when juxtaposed with traditional keypoint-based methods.

II. Related Work

6D Pose Estimation for Instance-Level Objects The estimation of 6D poses for instance-level objects has been a recurring challenge in the domain of computer vision. This task involves predicting the spatial configuration of objects within a defined reference frame and their corresponding

[1]Xujun Han, Shaochen Wang, and Zhen Kan are with Department of Automation, University of Science and Technology of China, Hefei, China.

[2]Xiucai Huang is with the School of Automation, Chongqing University, Chongqing, China.

This work was supported in part by National Key R&D Program of China under Grant 2022YFB4701400/4701403

Computer-Aided Design (CAD) representations. Conventional approaches have traditionally tackled this challenge by inferring 2D-3D correspondences associated with various features. For example, point cloud data alignment with CAD models has been utilized for template matching [13]. Recent advancements [14], [15], [16] have shifted towards detecting geometry-guided features within images to establish 2D-3D correspondences. Subsequently, the Perspective-n-Point (PnP) solver [17] is applied to estimate the poses of objects. In the context of our research, the presence of joint constraints and the inherent complexity arising from the numerous degrees of freedom in robot arm movements introduce a significantly heightened level of intricacy into the task of pose estimation, surpassing the challenges posed by instance-level objects.

Vision-Based Robot Arm Pose Estimation When it comes to estimating the pose of a robot arm using visual information, there are two primary approaches: those that rely on 2D images and those that utilize 3D sensors. In 2D image-based approaches, three major methodologies have emerged: marker-based, rendering-based, and keypoint-based methods. Marker-based approaches involve the detection of markers pre-positioned along the robot's kinematic chain within 2D images [18], [19]. These methods then calculate the coordinates of these markers in the robot's base frame using forward kinematics and provided joint configurations. The robot's pose is subsequently estimated through the resolution of an optimization problem [20], [21]. This approach offers accuracy but is dependent on the visibility and precise detection of markers. In the rendering-based approach, as demonstrated by Lu et al. [22], the robot's joint states are initially estimated and then transformed into silhouette images using differentiable rendering techniques. These resulting images are compared with the mask of the input image for self-supervision, enhancing the effectiveness of pose estimation. Labbe et al. [10] iteratively refine rendered images through comparison with ground truth, optimizing the camera-to-robot pose. However, this method is often time-consuming, especially during the initial pose estimation phase. In the keypoint-based method, Lu et al. [11] introduced an optimization algorithm aimed at identifying optimal 2D candidate keypoints on the bimanual robot and surgical robot. Tian et al. [23] integrates the robot's joint configuration and a temporal attention module to fuse keypoint features across frames within an image sequence. Then a PnP solver is employed to compute the camera-to-robot pose. In 3D sensor-based approaches, such as RGB-D cameras and LiDAR, these sensors offer distinctive advantages and face specific challenges. RGB-D cameras, due to their capacity to directly capture depth maps [24], offer advantages but are sensitive to light conditions, making them less effective in strongly illuminated environments [25]. Bohg et al. [26] utilized depth maps to classify each pixel as either part of the robot or the background. A voting scheme was employed to predict the robot's joint states. On the other hand, LiDAR sensors provide accurate three-dimensional information and excel in outdoor environments. However, their cost can be a limiting factor for many applications, especially high-quality LiDARs.

Multi-Scale Image Processing. Extraction of multi-scale features [27] is a fundamental and consequential endeavor. Achieving the precise extraction of 2D features from RGB images necessitates the consideration of a diverse and comprehensive set of features. To address this complex task, researchers have explored innovative techniques and approaches. Li et al. [28] devised a solution by harnessing feature pyramid attention blocks, which effectively capture features spanning a spectrum of scales. In a parallel endeavor, Zhang et al. [29] introduced a specialized module integrated within the backbone network. This module was meticulously designed to extract both global and local information concurrently, paving the way for the acquisition of multi-scale features. Wang et al. [30], [31] effectively bridge information in distinct image patches using self-attention and capture these multi-scale features through the encoder architecture within their transformer-based visual grasp detection framework. Qin et al. [32] embarked on their quest by employing a densely supervised encoder-decoder network, enriched with a residual optimization module. This holistic architecture was devised to predict and refine feature maps, contributing significantly to the process of boundary perception. In summary, within the domain of multi-scale image processing, researchers have ventured into distinct yet interconnected avenues, each contributing its own set of advancements and insights. These pioneering efforts encompass feature pyramid attention blocks, specialized network modules, densely supervised architectures, and innovative contextual information extraction techniques.

III. Problem Formulation

Given an inertial reference frame R for the robot manipulator and the camera coordinate system C, the estimated transformation matrix between the camera coordinate system C and the robot coordinate system R is denoted by $\tilde{T}_C^R \in SE(3)$. When provided with RGB images, the predetermined positions of 3D keypoints in the robot's coordinate space, and the camera's intrinsic parameters, the goal of this work is to develop an end-to-end pose estimation approach for the transformation matrix $\tilde{T}_C^R$.

IV. Method

A. Approach Overview

To estimate the camera-to-robot transformation matrix $\tilde{T}_C^R$ from single-frame images, our approach consists of a two-stage process, i.e., the keypoint detection stage and the 2D-3D pose estimation stage, as visualized in Fig. 1. The proposed approach takes as input the RGB images and the 3D keypoints in the robot's coordinate space and outputs $\tilde{T}_C^R$. Specifically, we employ an encoder-decoder network [33] with a nested U-structure as the keypoint detector. This network generates belief maps for each keypoint. Leveraging the peaks of these belief maps, along with the camera intrinsic parameters and 3D keypoints, the transformation

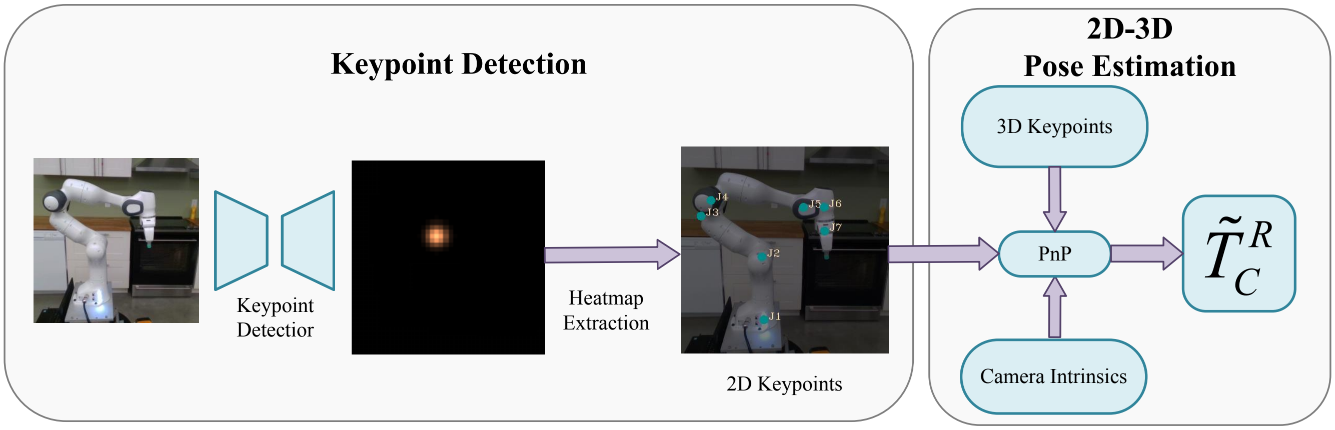

Fig. 1. The overview of our pose estimation framework. The network with nested U-structure outputs belief map, used for extracting 2D keypoints. These keypoints, along with camera intrinsic and 3D keypoints, collectively serve as inputs for the PnP algorithm, to estimate transformation matrix $\tilde{T}_C^R$.

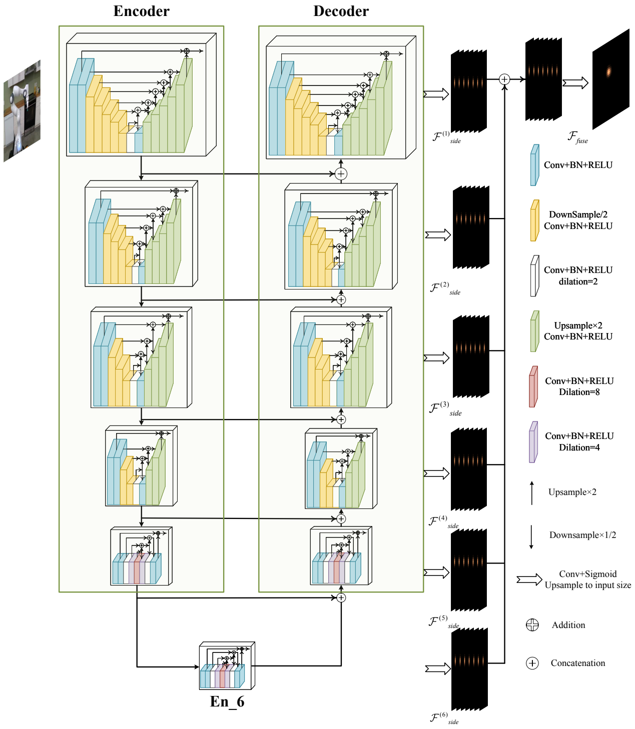
Encoder
Decoder
En_6
$\mathcal{F}^{(1)}_{side}$
$\mathcal{F}^{(2)}_{side}$
$\mathcal{F}^{(3)}_{side}$
$\mathcal{F}^{(4)}_{side}$
$\mathcal{F}^{(5)}_{side}$
$\mathcal{F}^{(6)}_{side}$
$\mathcal{F}_{fuse}$
Conv+BN+RELU
DownSample/2 Conv+BN+RELU
Conv+BN+RELU dilation=2
Upsample×2 Conv+BN+RELU
Conv+BN+RELU Dilation=8
Conv+BN+RELU Dilation=4
Upsample×2
Downsample×1/2
Conv+Sigmoid Upsample to input size
Addition
Concatenation

Fig. 2. Illustration of our network for belief map generating.

matrix $\tilde{T}_C^R$ is estimated using the Perspective-n-Point (PnP) solver [17].

B. Network Architecture

The network architecture is designed to capture multi-scale features, a crucial element for improving 2D keypoint detection. It takes RGB images of dimensions $400 \times 400 \times 3$ as input and produces 2D belief maps of size $400 \times 400 \times n$ for each keypoint, where n is the number of robot joints. These belief maps encode the likelihood of keypoints being projected onto specific pixels, with keypoints' 2D projection coordinates identified by locating the peak values within these maps.

The network architecture, as depicted in Fig. 2, is based on a nested U-structure and consists of six encoder stages and five decoder stages, incorporating a belief map fusion module [33]. Here's a breakdown of its components:

Encoder Component: This segment comprises six residual U-blocks (RSU $-$ L), with L denoting the number of layers in the encoder as shown in Fig. 3. Each RSU-L block contains three components. 1) Local Feature Extraction: an initial convolutional layer for local feature extraction, transforming the feature map into an intermediate map $\mathcal{F}_1(\mathbf{x})$ with the same output channel as the entire RSU block. 2) Multi-scale Contextual Information: a U-shaped encoder-decoder structure that employs a cascade of downsampling and convolution stages to extract contextual information across various scales. This information is encoded into a high-resolution feature map $\mathcal{U}(\mathcal{F}_1(\mathbf{x}))$ through a progressive upsampling structure. 3) Fusion of Local and Contextual Features: a fusion process that combines the intermediate feature map $\mathcal{F}_1(\mathbf{x})$ containing local features with the feature map $\mathcal{U}(\mathcal{F}_1(\mathbf{x}))$ containing multi-scale contextual features: $\mathcal{F}_1(x)+\mathcal{U}(F_1(x))$. This fusion enhances the network's ability to capture diverse information at different scales.

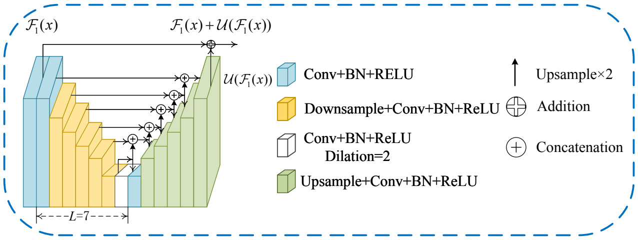

Fig. 3. RSU-L block

Decoder Stages: The decoder stages in our network architecture are responsible for reconstructing high-resolution belief maps from the abstract and multi-scale representations learned by the encoder. These stages play a crucial role in the final output of the network. In each decoder stage, the following steps occur. 1) Feature Concatenation: the decoder takes inputs from the corresponding encoder stage and the upsampled feature maps from the previous decoder stage. These inputs are concatenated to ensure that the network has access to the detailed information from the encoder and the context learned in earlier stages. 2) Upsampling: the concatenated features are then upsampled to match the dimensions of the belief maps generated by the encoder. This upsampling process helps in recovering the spatial details lost during the downsampling stages of the encoder. 3) Convolution and Activation: the upsampled features pass through a convolutional layer, followed by an activation function. This step refines the features and prepares them for generating the belief maps. The decoder stages work in a cascaded manner, progressively refining and expanding the features to generate the final belief maps for each keypoint.

Belief Map Fusion Module: This module connects the decoder stages and the final encoder stage. It employs a 3×3 convolution layer and a sigmoid function to generate six side belief maps $\mathcal{F}_{side}$ from stage En-6 (i.e., the bottom of Fig. 2) and the decoder. These side belief maps are upsampled to match the input image's size and are then fused through concatenation. Subsequently, they pass through a 1×1 convolution layer and a sigmoid function to produce the final belief map $\mathcal{F}_{fuse}$ for each keypoint. For training, an L2 loss function is applied, with the target being the ground truth belief map generated using a σ of 2 pixels for Gaussian filter to smooth the peaks.

This architecture allows us to capture multi-scale features effectively, enhancing the precision and robustness of 2D keypoint detection.

C. 2D - 3D Perspective Transformation

In our pursuit of accurate camera-to-robot pose estimation, we employ a systematic approach underpinned by geometric transformations and the Perspective-n-Point (PnP) algorithm. The fundamental objective is to determine the pose of a robot arm relative to the camera's coordinate system, enabling seamless interaction between the robot and its environment through visual data by

$$P_C = \tilde{T}_C^R P_R, \tag{1}$$

where P_R and P_C represent the positions of a set of 3D keypoints on the robot joints within the robot frame and the camera frame, respectively.

Our method commences with the P_R. The crux of the problem lies in establishing the transformation matrix $\tilde{T}_C^R$, which characterizes the relationship between these 3D points and their corresponding 2D projections on the camera's image plane. Solving this transformation is relative to the intrinsic properties of the camera, encapsulated within the intrinsic matrix K.

The transformation equation is expressed as:

$$\alpha \cdot i = K \cdot \tilde{T}_C^R \cdot P_R \tag{2}$$

where α represents the scale factor, i denotes the 2D image coordinates.

To resolve for the transformation matrix $\tilde{T}_C^R$, we leverage the Perspective-n-Point (PnP) algorithm. This algorithm employs the camera's intrinsic K, the 2D positions of keypoints i, and the known 3D coordinates of these keypoints on the robot P_R. The 2D positions of keypoints are derived from the output belief maps, which serve as heatmaps indicating the probable locations of these keypoints in the image. To obtain precise 2D coordinates, we apply a weighted average calculation to the values surrounding the heatmap's highest points, ensuring accuracy in keypoint localization.

In summary, our camera-to-robot pose estimation methodology involves the determination of the transformation matrix ($\tilde{T}_C^R$). This transformation is facilitated by the PnP algorithm. This comprehensive approach empowers the robot to discern its precise spatial orientation within its environment, thereby enabling it to execute tasks based on visual information.

V. Experients and results

In this section, we present a comprehensive evaluation of our approach on a diverse set of real-world datasets. We compare our method with state-of-the-art algorithms in the domain of camera-to-robot pose estimation tasks. Additionally, we conduct an ablation study involving three different robot arms to validate the effectiveness of our architecture.

A. Datasets, Metrics, and Baselines

Dataset: For the training datasets, we leverage the Panda, Kuka, and Baxter datasets provided by [4]. These datasets, generated through domain randomization, offer a wide range of realistically synthesized robot arm images across various scenarios. For testing purposes, we evaluate our method on five distinct datasets involving different robot manipulators: Panda 3cam-AK, Panda 3cam-RS, Panda ORB, Kuka Synth Test, and Baxter Synth Test [4].

Metrics: We employ both 2D and 3D metrics to assess our approach's performance. For 2D metrics, we use the Percentage of Correct Keypoints (PCK) [34], which quantifies the L2 error between ground truth and predicted keypoints. For 3D metrics, we utilize the Average Distance (ADD) [35], calculated as the average L2 distance between $\tilde{T}_C^R P_R$ (the estimated 3D keypoints using the transformation matrix) and $\bar{P}_C$ (the ground truth 3D keypoints). The ADD is computed as follows:

$$ADD = \frac{1}{n} \sum_{i=1}^{n} \left\| \tilde{T}_C^R P_R - \bar{P}_C \right\|_2 \tag{3}$$

For both PCK and ADD metrics, we compute the Area Under the Curve (AUC) as the percentage below a specific threshold. Additionally, we consider the median across all keypoints in both cases.

Compared baselines: The following three baselines are considered. RobotStructure [23] specializes in estimating camera-to-robot pose from single-view successive frames. Dream [4] represents an advanced approach for estimating camera-to-robot pose using single-frame images. CenterNet

[36] focuses on achieving single-frame object detection by modeling an object as a singular keypoint.

B. Implementation Details

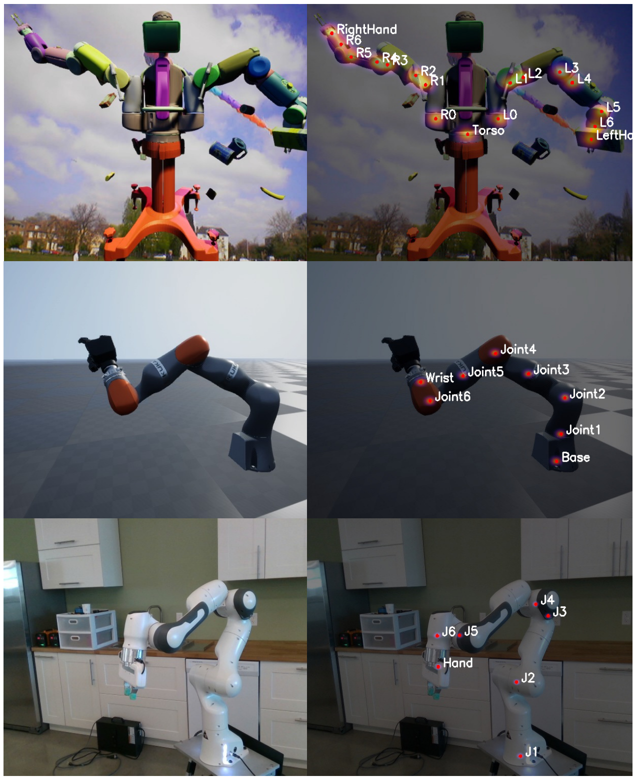

Fig. 4. Belief maps (right column) overlayed on the original images(left column) for Rethink Baxter, Kuka LBR iiwa, Franka Emika Panda from up to down.

The entire system is implemented using PyTorch. Seven keypoints are strategically placed at the joints of the Franka Emika Panda robot arm. The model is trained for 100 epochs with a learning rate set to 1e-4. Network parameter optimization is conducted using the Adam optimizer [37] with a momentum of 0.9. The training dataset comprises approximately 100,000 synthesized images. We select the weights that yield optimal performance during validation. To minimize the discrepancies between ground truth and predictions, we utilize the L2 loss function.

C. Robot Pose Estimation on Real-world Datasets

We present the evaluation of our algorithm on real-world datasets and provide qualitative results for pose estimation, as illustrated in Fig. 4. We compare our approach with baseline methods, including RobotStructure [23], Dream [4], and CenterNet [36]. Table I presents the comparative analysis of the PCK and ADD metrics, along with their respective AUC values and medians, across three different real-world datasets. Baseline results are primarily sourced from RobotStructure [23], where CenterNet [36] is employed for 2D keypoint estimation in this context.

In terms of the 2D metric PCK, our approach exhibits a significant advantage over the baseline across various datasets. Notably, on the Panda 3cam-AK dataset, our method outperforms the second-best model by 7.24% in PCK and reduces the median error by 0.63. This demonstrates the adaptability of our model, trained on synthetic data, to perform well under various camera intrinsics. On the Panda ORB dataset, which includes multiple camera viewpoints, our method demonstrates a notable improvement of 6.23 points in PCK compared to state-of-the-art approaches.

To assess the precision of pose estimation through the 3D metric ADD, which provides a more intuitive representation, our method achieves superior performance on both Panda 3cam-AK and Panda ORB, with respective improvements of 11.84% and 9.8% over alternative approaches. Furthermore, the median of our ADD metric achieves 22.05mm and 12.76mm on the Panda 3cam-AK and Panda ORB datasets, respectively.

Compared to previous methodologies, our approach allows for greater network depth, enabling higher resolutions. Additionally, we extract multi-level deep features through a straightforward architectural design, contributing significantly to the superiority of our approach.

D. Ablation Study

In this section, we conduct comprehensive experiments aimed at exploring two key aspects:

1. The necessity of introducing a nested U-shaped architecture and utilizing RSU (Residual and U-shaped) blocks for the extraction of intra-stage multi-scale features. 2. The capacity of our framework to generalize across a diverse set of robot manipulators.

To validate the effectiveness of our model design, we replace the network backbone of our architecture with different alternatives in the encoder section. We experiment with two popular architectures, VGG and ResNet, and output resolutions at full (F), half (H), or quarter (Q) in the decoder section, resulting in three distinct designs: vgg-Q, vgg-F, and resnet-H [4].

Furthermore, we assess the generalizability of our approach by conducting experiments on three different robotic manipulators: Franka Emika Panda, Kuka LBR iiwa, and Rethink Baxter. To ensure fairness and consistency, all methods are trained and tested on the same datasets specific to each robot.

As depicted in Fig. 5, our architecture consistently outperforms other models in terms of performance. Notably, our approach exhibits remarkable optimization, particularly at lower error thresholds. This achievement can be attributed to our capability for feature extraction and integration across multiple scales, leading to more accurate predictions at lower permissible error levels.

Significantly, on the Rethink Baxter manipulator, over 90% of 3D distance errors are confined to less than 20 mm, and 87.4% of reprojection errors do not exceed 20 pixels. Notably, the performance on the Kuka LBR iiwa surpasses that on other robot manipulators. For the Franka Emika Panda, we observe a notable increase of 8.2% in the AUC of ADD and an 8.2% increase in the AUC of PCK. Moreover,

TABLE I
PCK and ADD

Method	PCK						ADD					
	Panda 3cam-AK		Panda 3cam-RS		Panda ORB		Panda 3cam-AK		Panda 3cam-RS		Panda ORB	
	AUC↑	Median@pix↓	AUC↑	Median@pix↓	AUC↑	Median@pix↓	AUC↑	Median@mm↓	AUC↑	Median@mm↓	AUC↑	Median@mm↓
CenterNet[36]	52.38	4.90	67.38	3.51	60.11	3.47	34.07	37.56	59.26	21.25	50.59	24.22
Dream[4]	52.28	4.83	64.82	3.90	57.44	3.73	44.55	33.68	58.60	24.57	52.56	22.53
RobotStructure[23]	62.75	3.19	75.68	2.68	63.28	3.46	49.42	29.61	**79.89**	**9.77**	60.30	18.12
Ours	**69.99**	**2.56**	**77.71**	**2.36**	**69.51**	**2.87**	**61.26**	**22.05**	75.82	12.45	**70.16**	**12.76**

↑ indicates higher is preferable, while ↓ indicates lower is preferable.The AUC for PCK and ADD are captured at thresholds of 12 pixels and 6 cm, respectively.

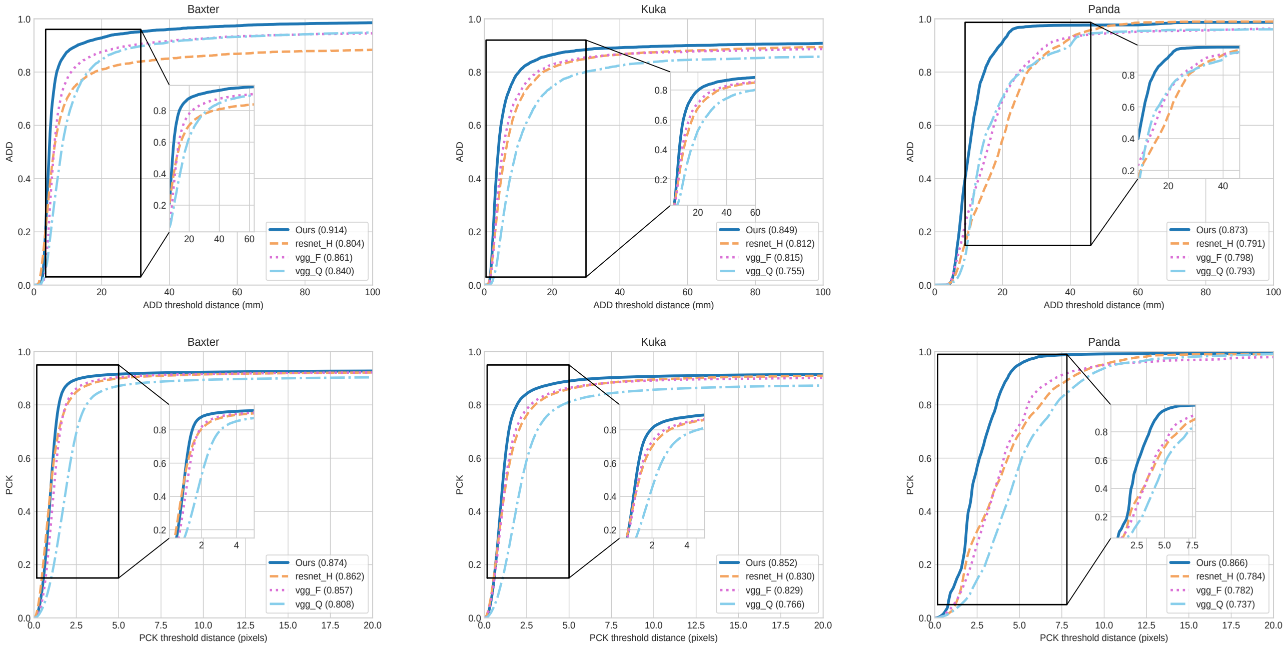

Fig. 5. The results of ADD (top) and PCK (bottom) metrics on diverse backbones and robot manipulators(Rethink Baxter, Kuka LBR iiwa, Franka Emika Panda from left to right), with the numbers in parentheses indicating their respective area under the curve (AUC) values.

it is noteworthy that over 95% of the samples exhibit errors less than 40 mm when the threshold is set at 40 mm.

In conclusion, our nested architecture shows promise in achieving more favorable results in the robot pose estimation task compared to other backbone-based designs.

VI. Conclusions

In this study, we have introduced an end-to-end deep neural network framework for the challenging task of camera-to-robot pose estimation. Our approach employs a nested U-shaped network architecture, enabling the extraction of multi-scale features at various stages to accurately determine keypoint locations in single-frame images, ultimately leading to precise robot pose estimation. Significantly, our method achieves remarkable precision without relying on fiducial markers, demonstrating its practicality and effectiveness. The results highlight the superiority of our approach compared to other state-of-the-art algorithms in the realm of camera-to-robot pose estimation. The utilization of refined keypoints significantly enhances the performance of pose estimation for robot arms, showcasing the potential for real-world applications.

References

[1] Z. Zhou, S. Wang, Z. Chen, M. Cai, H. Wang, Z. Li, and Z. Kan, “Local observation based reactive temporal logic planning of human-robot systems,” *IEEE Transactions on Automation Science and Engineering*, 2023.

[2] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[3] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 590–596.

[4] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9426–9432.

[5] D. Li, S. Wang, K. Chen, and B. Li, "Contrastive inductive bias controlling networks for reinforcement learning," in *Asian Conference on Machine Learning*. PMLR, 2023, pp. 563–578.

[6] S. Wang, R. Yang, B. Li, and Z. Kan, "Structural parameter space exploration for reinforcement learning via a matrix variate distribution," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1025–1035, 2023.

[7] S. Wang, Z. Zhou, H. Wang, Z. Li, and Z. Kan, "Unsupervised representation learning for visual robotics grasping," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2022, pp. 57–62.

[8] S. Wang, W. Zhang, Z. Zhou, J. Cao, Z. Chen, K. Chen, B. Li, and Z. Kan, "What you see is what you grasp: User-friendly grasping guided by near-eye-tracking," in *IEEE International Conference on Development and Learning (ICDL)*. IEEE, 2023, pp. 194–199.

[9] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, "Craves: Controlling robotic arm with a vision-based economic system," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4214–4223.

[10] Y. Labbé, J. Carpentier, M. Aubry, and J. 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*, 2021, pp. 1654–1663.

[11] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[12] J. Lambrecht, P. Grosenick, and M. Meusel, "Optimizing keypoint-based single-shot camera-to-robot pose estimation through shape segmentation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 843–13 849.

[13] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[14] S. Iwase, X. Liu, R. Khirodkar, R. Yokota, and K. M. Kitani, "Repose: Fast 6d object pose refinement via deep texture rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3303–3312.

[15] C. Song, J. Song, and Q. Huang, "Hybridpose: 6d object pose estimation under hybrid representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 431–440.

[16] J. Lin, H. Li, K. Chen, J. Lu, and K. Jia, "Sparse steerable convolutions: An efficient learning of se (3)-equivariant features for estimation and tracking of object poses in 3d space," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 779–16 790, 2021.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, pp. 155–166, 2009.

[18] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb," *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[19] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407.

[20] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 67–74.

[21] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[22] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 296–21 306.

[23] Y. Tian, J. Zhang, Z. Yin, and H. Dong, "Robot structure prior guided temporal attention for camera-to-robot pose estimation from image sequence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8917–8926.

[24] K. Chen, S. Wang, B. Xia, D. Li, Z. Kan, and B. Li, "Tode-trans: Transparent object depth estimation with transformer," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4880–4886.

[25] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Factored pose estimation of articulated objects using efficient nonparametric belief propagation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7221–7227.

[26] J. Bohg, J. Romero, A. Herzog, and S. Schaal, "Robot arm pose estimation through pixel-wise part classification," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3143–3150.

[27] Y. Xia, S. Wang, and Z. Kan, "A nested u-structure for instrument segmentation in robotic surgery," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2023, pp. 994–999.

[28] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[29] L. Zhang, J. Zhang, Z. Lin, H. Lu, and Y. He, "Capsal: Leveraging captioning to boost semantics for salient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6024–6033.

[30] S. Wang, Z. Zhou, and Z. Kan, "When transformer meets robotic grasping: Exploits context for efficient grasp detection," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8170–8177, 2022.

[31] S. Wang, Z. Zhou, B. Li, Z. Li, and Z. Kan, "Multi-modal interaction with transformers: bridging robots and human with natural language," *Robotica*, vol. 42, no. 2, pp. 415–434, 2024.

[32] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7479–7489.

[33] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[34] J. Tremblay, T. To, A. Molchanov, S. Tyree, J. Kautz, and S. Birchfield, "Synthetically trained neural networks for learning human-readable plans from real-world demonstrations," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5659–5666.

[35] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[36] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.