RoboKeyGen: Robot Pose and Joint Angles Estimation via Diffusion-based 3D Keypoint Generation

Yang Tian*, Jiyao Zhang*, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, Hao Dong

***Abstract*— Estimating robot pose and joint angles is significant in advanced robotics, enabling applications like robot collaboration and online hand-eye calibration. However, the introduction of unknown joint angles makes prediction more complex than simple robot pose estimation, due to its higher dimensionality. Previous methods either regress 3D keypoints directly or utilise a render&compare strategy. These approaches often falter in terms of performance or efficiency and grapple with the cross-camera gap problem. This paper presents a novel framework that bifurcates the high-dimensional prediction task into two manageable subtasks: 2D keypoints detection and lifting 2D keypoints to 3D. This separation promises enhanced performance without sacrificing the efficiency innate to keypoint-based techniques. A vital component of our method is the lifting of 2D keypoints to 3D keypoints. Common deterministic regression methods may falter when faced with uncertainties from 2D detection errors or self-occlusions. Leveraging the robust modeling potential of diffusion models, we reframe this issue as a conditional 3D keypoints generation task. To bolster cross-camera adaptability, we introduce the *Normalised Camera Coordinate Space (NCCS)*, ensuring alignment of estimated 2D keypoints across varying camera intrinsics. Experimental results demonstrate that the proposed method outperforms the state-of-the-art render&compare method and achieves higher inference speed. Furthermore, the tests accentuate our method's robust cross-camera generalisation capabilities. We intend to release both the dataset and code in https://sites.google.com/view/robokeygen/.**

I. INTRODUCTION

Estimating robot pose and joint angles is crucial in intelligent robotics with implications for multi-robot collaboration [1], online hand-eye calibration [2], and visual servoing [3] for close-loop control. Extensive research has been conducted on robot pose estimation, such as marker-based easy-handeye [4] and learning-based online calibration methods [5], [6], [7]. However, these approaches assume known joint angles, a condition not always met. In multi-robot collaborations, for instance, state data may be unshared, necessitating concurrent robot pose and joint angle estimation.

Contrasting robot pose estimation with known versus unknown joint angles, the latter reveals heightened complexity due to increased degrees of freedom (*e.g.*, from 6D to 13D for Franka). Existing methods can be divided into two categories: render&compare approaches [8] and keypoints-based methods [9]. RoboPose [8] extends render&compare [10], [11] strategies from rigid object pose estimation [12], [13] to robot pose and joint angles estimation. However, this method suffers from slow inference speed (1 FPS in single frame mode) due to the iterative rendering. Conversely, SPDH [9] introduces a Semi-Perspective Decoupled Heatmaps representation, which extends the well-known 2D heatmaps to the 3D domain. It enables direct prediction of the 3D coordinates of predefined keypoints on the robot arm from a depth input and has a higher inference speed (22FPS) compared to render&compare based methods. However, this approach exhibits a low accuracy and the proposed representation is theoretically limited by the presence of cross-camera generalisation issue. In general, the existing methods are faced with such limitations:

- The conflict between efficiency and performance.
- The cross-camera generalisation issue.

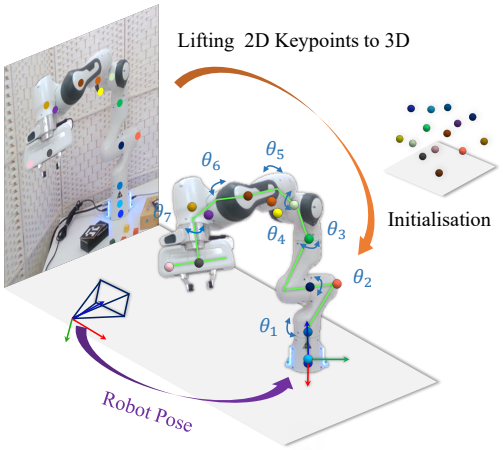

Fig. 1. **RoboKeyGen.** Given RGB images, we aim to estimate the robot pose and joint angles. We achieve this goal by decoupling it into two more tractable tasks: 2D keypoints detection and lifting 2D keypoints to 3D.

To address these challenges, we propose a novel framework named RoboKeyGen. The basic idea is illustrated in Fig. 1. Different from previous methods, we decouple this high-dimensional prediction task into two sub-tasks: 2D keypoints detection and lifting 2D keypoints to 3D. The former focuses on extracting the 2D keypoints from the appearance characteristics, while the latter concentrates on perspective transformation and the robot's structural information. This decoupling enables our method to improve performance while preserving the inherent efficiency of keypoints-based approaches. Specifically, our method first predicts the 2D projections of predefined keypoints. Then, we align these projections into a normalised camera coordinate space. Subsequently, we generate the 3D keypoints conditioned on the normalised 2D keypoints. These 3D keypoints are then utilised

Yang Tian, Jiyao Zhang and Hao Dong are with CFCS, School of CS, Peking University and National Key Laboratory for Multimedia Information Processing. Guowei Huang and Bin Wang are with Huawei. Ping Wang is with School of Software & Microelectronics and National Engineering Research Center for Software Engineering, Peking University. Jiangmiao Pang is with the Chinese University of Hong Kong.

* indicates equal contribution

Corresponding to hao.dong@pku.edu.cn

to regress joint angles. Finally, an off-the-shelf pose-fitting algorithm [14] is employed to estimate the robot pose.

Thanks to the significant advancements in 2D robot keypoints detection [5], we focus more on addressing the challenge of lifting these 2D keypoints to 3D and cross-camera generalisation. Direct regression proves suboptimal due to its failure to model the uncertainty brought by 2D keypoints detection errors. Instead, modeling the conditional distribution of 3D keypoints is more reasonable. Leveraging the robust distribution modeling of diffusion-based models [15], [16], [13], we employ a diffusion model conditioned on the estimated 2D keypoints to generate 3D keypoints. For cross-camera generalisation, considering the diverse camera intrinsic parameters has distinct projection transformations, we introduce the *normalised camera coordinate space (NCCS)* for 2D keypoint alignment, effectively addressing the issue of cross-camera generalisation.

We provide a pipeline incorporating simulated training data and real-world datasets from two depth cameras for evaluation. Comparative analyses reveal our model's superiority over RoboPose [8] in performance and speed metrics, further underscoring its robustness in cross-camera generalisation.

II. RELATED WORKS

A. Learning-based Robot Pose and Joint Angles Estimation

1) Robot pose estimation with known joint angles: Recent advances in deep learning offer innovative methods for robot pose recovery. Dream [5] uses a convolutional network to regress 2D heatmaps and compute poses through a *Perspective-n-Point (PnP) RANSAC* solver [17]. SGTA-Pose [6] integrates temporal information to address self-occlusion in pose estimation. Meanwhile, CtRNet [7] employs a self-supervision framework, narrowing the sim-to-real gap effectively. Notably, these methods depend on immediate joint angles feedback, thus limiting their applicability.

2) Robot pose and joint angles estimation: When joint angles are unknown, methods fall into two main categories: render&compare, and 3D keypoint detection. RoboPose [8] offers a render&compare framework for pose and joint angles using a single RGB image but is limited by a 1 FPS single-frame inference speed due to rendering. SPDH [9], a depth-based approach, extends 2D to 3D heatmap pose estimation but faces a cross-camera challenge. Our approach, in contrast, combines the speed of keypoint methods with a novel conditional 3D keypoints generation, addressing the cross-camera gap more effectively than SPDH [9].

B. Diffusion Models

Diffusion models have gained significant attention in generative modeling. Some works have focused on theoretical aspects, such as training Noise Conditional Score Networks (SMLD) with denoising score matching objectives [18], [19], others introduced Denoising Diffusion Probabilistic Models (DDPM) that employ forward and reverse Markov chains [20], [21]. To provide a comprehensive understanding of these models, Song [22] presented a unified perspective that incorporates and explains the previously mentioned approaches. Some studies also explored various applications of diffusion models, including medical imaging [23], point cloud generation [15], object rearrangement [16], [24], object pose estimation [25], and human pose estimation [26]. Inspired by these advancements, we propose a novel diffusion-based framework for robot pose and joint angles estimation, specifically focusing on lifting 2D keypoints detection to conditional 3D keypoints generation. To the best of our knowledge, our method is the first exploration for learning the robot arm's structure via diffusion models.

III. METHOD

Task description. Given a live stream of RGB images $\{I\}$, we aim to estimate the Robot Pose $\{\Gamma = (R,T) \in SE(3)\}$ and joint angles $\{\theta \in \mathbb{R}^n\}$ (where n denotes the amounts of joint angles). Here we assume the forward kinematics and CAD models of the robot arm and camera intrinsics are known.

Overview. We decouple the original high-dimensional task into two more tractable, low-dimensional sub-tasks: **2D keypoints detection** and **lifting 2D keypoints to 3D**. We first predict 2D projections of predefined keypoints c from RGB images I. Then we align these estimated keypoints c into the form $\tilde{c}$ in *Normalised Camera Coordinate Space (NCCS)*. Further, a diffusion model Φ_ζ is employed to model the distribution $(P_{data}(X^{cam}|\tilde{c}))$ of 3D keypoints X^{cam} in camera space conditioned on normalised 2D keypoints $\tilde{c}$. Finally, we utilise a light regression network to predict joint angles θ and recover 3D keypoints X^{rob} in robot space. We restore the robot pose via pose fitting.

A. 2D Keypoints Detection and Canonicalisation

We firstly detect 2D keypoints c from RGB images. Then, considering that the distribution of X^{cam} conditioned on 2D keypoints projections c changes as camera intrinsics change, we align c into normalised camera coordinate space $\tilde{c}$ to ensure a unique and well-defined distribution $P_{data}(X^{cam}|\tilde{c})$.

1) 2D Keypoints Detection: We detect predefined 2D keypoints $c \in \mathbb{R}^{N\times 2}$ from the current RGB frame I and the last estimated 2D keypoints, where N denotes the amounts of keypoints. Specifically, to enable the 2D detection network Ψ_ω focus on extracting features from the pure robot arm and avoid disturbance from background textures, we first adopt the real-time semantic segmentation network PIDNet-L [27] M to segment the robot arm. Moreover, considering 2D keypoints between consecutive frames change slightly, we then project the estimated last frame's 2D projections into positional embedding priors $\mathscr{F}$ through sinusoidal transformations [28] and shallow MLPs as suggested in [22]. Finally, given the RGB image I, segmentation mask and positional embedding $\mathscr{F}$ as input, an encoder-decoder detection network Ψ_ω [29], [30] is employed to predict the 2D keypoints c of the current frame.

2) 2D Keypoints Canonicalisation: For a given robot arm with available forward kinematics and predefined keypoints, we can easily find such an awkward property of $P_{data}(X^{cam}|c)$: For a common projection c, cameras with different intrinsics yield diverse 3D Ground Truth (GT) keypoints, which makes the distribution $P_{data}(X|c)$ poorly-defined. To eliminate this issue, we project c into a normalised camera coordinate space (NCCS) $\tilde{c}$. Specifically, with known camera intrinsics

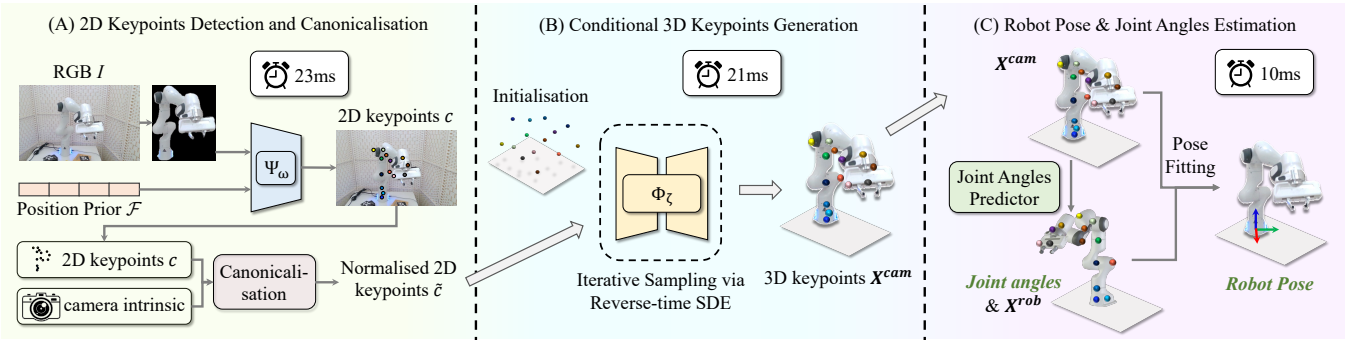

Fig. 2. **The inference pipeline of RoboKeyGen.** (A) Combined with the RGB image I, predicted segmentation mask and positional embedding prior $\mathcal{F}$, we firstly predict 2D keypoints c through the detection network Ψ_ω. (B) Conditioning on 2D detections, we generate 3D X^{cam} via the score network Φ_ζ. (C) Finally, we predict joint angles from X^{cam} and recover X^{rob} based on URDF files. We do pose fitting between X^{cam} and X^{rob} to acquire the robot pose.

$\{f_x, f_y, c_x, c_y\}$, for $i-$th 2D keypoint $c^i = (u^i, v^i) \subset c$, we transform c^i into $\tilde{c}^i = (\frac{u^i - c_x}{f_x}, \frac{v^i - c_y}{f_y})$. According to the pinhole camera model (which is followed by most cameras in robotics), this transformation equals $\tilde{c}^i = (\frac{x^i}{z^i}, \frac{y^i}{z^i})$, where $(x^i, y^i, z^i) \subset X^{cam}$ is the i-th keypoint's coordinates in camera space. Now we consider the new joint distribution $\tilde{\mathscr{D}} = \{(\tilde{c}, X^{cam}) = (\{(\frac{x^i}{z^i}, \frac{y^i}{z^i})\}_{i=1}^N, \{(x^i, y^i, z^i)\}_{i=1}^N) \sim P_{data}(\tilde{c}, X^{cam})\}$. We observe the condition $\tilde{c}$ in $P_{data}(X^{cam}|\tilde{c})$ is decoupled from camera intrinsics since it owns a normalised form regarding only coordinates in camera space. In such situations, learning the new conditional distribution $P_{data}(X^{cam}|\tilde{c})$ is essentially ensuring the z-coordinates for each keypoint. In other words, our method only requires to concentrate on the robot arm's structure with no disturbance from camera intrinsics.

B. Conditional 3D Keypoints Generation via Diffusion Model

This section will illustrate how to sample the predefined 3D keypoints X^{cam} conditioned on the normalised 2D keypoints $\tilde{c}$ in a generative modeling paradigm. Here we denote $X \in \mathbb{R}^{N\times 3}$ as 3D keypoints in camera space (X^{cam} in Fig. 2) for simplicity. We assume the 2D-3D keypoints pair in each image is sampled from an implicit joint distribution $\mathscr{D} = \{(\tilde{c}, X) \sim P_{data}(\tilde{c}, X)\}$, and our objective is to model $P_{data}(X|\tilde{c})$.

1) Learning the score function Φ_ζ: We adopt a score-based diffusion model to model $P_{data}(X|\tilde{c})$. Specifically, we take Variance Preserving (VP) Stochastic Differential Equation (SDE) proposed in [22] to construct a continuous time-dependent diffusion process $\{X(t)\}_{t=0}^T$. $X(0)$ originates from $P_{data}(X|c)$ and $X(T)$ comes from the diffused prior distribution p_T. As t increases, $\{X(t)\}_{t=0}^T$ is given by :

$$dX = -\frac{1}{2}\beta(t)Xdt + \sqrt{\beta(t)}dw \tag{1}$$

where $\beta(t) = \beta(0) + t(\beta(1) - \beta(0))$. $\beta(0)$, $\beta(1)$ and T are set as 0.1, 20.0 and 1.0 respectively.

During Training, we aim to estimate the *score function* of perturbed conditional distribution $\nabla_X \log p_t(X|\tilde{c})$ for all t:

$$p_t(X(t)|\tilde{c}) = \int p_{0t}(X(t)|X(0)) \cdot p_0(X(0)|\tilde{c}) dX(0) \tag{2}$$

where p_{0t} is the transition kernel and $p_0(X(0)|\tilde{c})$ is exactly $P_{data}(X|\tilde{c})$. $\nabla_X \log p_t(X|\tilde{c})$ can be estimated by training a score network $\Phi_\zeta : \mathbb{R}^{3\times N} \times \mathbb{R} \times \mathbb{R}^{2\times N} \to \mathbb{R}^{3\times N}$ via:

$$\begin{aligned}\mathscr{L}(\zeta) = \mathbb{E}_{t\sim\mathscr{U}(\varepsilon,1)}\{\lambda(t)\mathbb{E}_{\tilde{c},X(0)\sim p_{data}(\tilde{c},X)}\mathbb{E}_{X(t)\sim p_{0t}(X(t)|X(0))} \\ [\|\Phi_\zeta(X(t),t|\tilde{c}) - \nabla_{X(t)}\log p_{0t}(X(t)|X(0))\|_2^2]\}\end{aligned} \tag{3}$$

where ε is 0.0001 and $\lambda(t)$ is set as $\beta(t)$ suggested in [31]. The choice of VP SDE brings a closed form of p_{0t} as follows:

$$\mathscr{N}(X(t); X(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(s)ds}) \tag{4}$$

It is ensured that the optimal solution to Eq. 3, denoted by $\Phi_{\zeta^*}(X, t|\tilde{c})$ equals $\nabla_X \log p_t(X|\tilde{c})$ according to [22].

2) Sampling via the DDIM [33] sampler: After training, we can sample K groups of 3D Keypoints' candidates $\{X_j\}_{j=1}^K$ via diffusion samplers.

To speed up the inference phase, we select a fast DDIM sampler [33]. We iteratively generate $X(\tau_{t-1})$ from $X(\tau_t)$ via the following equation:

$$\begin{aligned}X(\tau_{t-1}) = \sqrt{\overline{\alpha}_{\tau_{t-1}}}\left(\frac{X(\tau_t) - \sqrt{1-\overline{\alpha}_{\tau_t}}\varepsilon_\zeta(X(\tau_t), \tau_t|\tilde{c})}{\sqrt{\overline{\alpha}_{\tau_t}}}\right) \\ + \sqrt{1 - \overline{\alpha}_{\tau_{t-1}} - \sigma_{\tau_t}^2} \cdot \varepsilon_\zeta(X(\tau_t), \tau_t|\tilde{c}) + \sigma_{\tau_t}\varepsilon_{\tau_t}\end{aligned} \tag{5}$$

where $\{\tau_i\}_{i=1}^m$ is the sampling timesteps.

$\overline{\alpha}_{\tau_t}$, $\{b_{\tau_i}\}_{i=1}^m$ and σ_{τ_t} remain the same notation and computation in [20]. $\varepsilon_\zeta(X(\tau_t), \tau_t|\tilde{c})$ is the noise function and can be computed as $(-\sqrt{1 - e^{-\int_0^{\tau_t}\beta(s)ds}}\Phi_\zeta(X(\tau_t), \tau_t|\tilde{c}))$. In our implementation, we set K as 10 and output the average value of $\{X(\tau_1)_j\}_{j=1}^K$.

C. Robot Pose and Joint Angles Estimation

To further recover the robot's configuration, we target at estimating the robot's joint angles. Intuitively, we can connect the estimated 3D keypoints X^{cam} sequentially and regard them as a skeleton. To estimate the joint angles from a skeleton, we only need to care about the positional relationship between adjacent "bones". Hence, we train a simple MLP to directly regress joint angles θ from X^{cam}. With available robot's forward kinematics and joint angles, we can recover the whole robot's configuration and compute X^{rob} according to the URDF file. Finally, we take a robust strategy using differentiable outliers estimation introduced in [14] to implement the pose fitting between X^{cam} and X^{rob}.

TABLE I

QUANTITATIVE COMPARISON WITH BASELINES. ↑ MEANS HIGHER IS BETTER, AND ↓ MEANS LOWER IS BETTER. ✓/× DENOTE WHETHER JOINT ANGLES ARE KNOWN. *Ours (single-frame)* AND *Ours (online)* DENOTE INITIALIZATION FROM GAUSSIAN NOISE AND THE PREDICTION OF THE LAST FRAME, RESPECTIVELY. WE ALSO REPLACE THE BACKBONE IN [9] WITH RESNET-101[32] AS ANOTHER BASELINE *SPDH-RESNET (Ours)*. FOR A FAIR COMPARISON, WE TRAIN ALL THE METHODS LISTED ABOVE ON SIMRGBD-FRANKA AND REPORT ADD AND AUC ACROSS TWO DATASETS.

Method	RealSense-Franka			AzureKinect-Franka			FPS
	AUC@0.1m↑	Median(m)↓	Mean(m)↓	AUC@0.1m↑	Median(m)↓	Mean(m)↓	
RoboPose (single-frame) [8]	29.01	0.081	0.116	32.00	0.069	0.106	1
RoboPose (online) [8]	31.78	0.073	0.105	39.83	0.047	0.083	16
SPDH-HRNet [9]	3.39	1.366	1.297	0.00	0.643	0.669	10
SPDH-SH [9]	17.24	0.251	0.272	0.00	0.844	0.854	**22**
SPDH-RESNET [9] (Ours)	19.46	0.090	0.135	0.00	0.793	0.789	18
Ours (single-frame)	67.00	0.027	0.035	60.72	0.030	0.049	12
Ours (online)	**72.93**	**0.022**	**0.028**	**63.33**	**0.028**	**0.045**	18

D. Implementation Details

To train the segmentation and detection network, we remain the same augmentations, loss functions and training strategies as suggested in [27], [29]. To train the score network Φ_ζ, we modify a vanilla fully connected network in [26] as the backbone. We optimise the object in Eq. 3 for 2000 epochs with a batch size of 4096, learning rate 0.0002 and Adam optimiser. To train the joint angle regression network, we design a shallow feedforward network. We train the network for 720 epochs with a batch size of 3600 via AdamW optimiser with initial learning rate 0.01 dropping by 0.1 at epoch 150, 300, 450. See more details when code is released.

IV. EXPERIMENTS AND RESULTS

A. Datasets, Baselines and Metrics

1) Datasets: Since the public dataset in DREAM [5] doesn't provide temporal images for training and lacks depth images, which are required for SPDH [9], we propose three new datasets: a simulated training set, **SimRGBD-Franka**, and two real-world testing sets, **RealSense-Franka** and **AzureKinect-Franka** captured with different depth cameras.

SimRGBD-Franka: Following in [6] and [34], we create this large-scale simulated dataset with Blender [35]. It comprises 4k videos, each with 3 consecutive frames, providing RGB images, robot pose, joint angles, masks, IR images, actual depth images, and simulated noisy depth images.

RealSense-Franka and **AzureKinect-Franka**: Captured using external cameras (Realsense D415 and Microsoft Azure Kinect), these datasets showcase the Franka Emika Panda robot in motion. RealSense-Franka comprises 4 videos (3931 images), while AzureKinect-Franka has 5 videos (5576 images). Each video starts with a stationary camera that eventually moves. Regarding annotation, we firstly use COLMAP [36] to calibrate the camera extrinsics. Then, the initial frame in each video segment is manually annotated for robot pose and joint angles. Finally, leveraging the calibrated camera extrinsics, we automatically get the annotations for the entire video segment. Both datasets include RGB images, robot pose, joint angles, and depth images.

2) Baselines: We compare our approach with previous methods in both unknown and known joint angles scenarios. **Unknown Joint Angles: RoboPose** [8]: A state-of-the-art (SOTA) method that employs render&compare to deduce joint angles and robot pose. **SPDH** [9]: A direct method that derives 3D robot pose from a single depth map using semi-perspective decoupled heatmaps. **Known Joint Angles: Dream** [5]: An innovative technique that infers robot pose from a single frame via 2D heatmap regression and PnP-RANSAC solving. **SGTAPose** [6]: A pioneering approach that leverages temporal information for robot pose estimation. **CtRNet** [7]: A pioneering approach that introduces a self-supervised strategy for online camera-to-robot calibration.

3) Metrics: We evaluate 3D metrics across all datasets. **ADD**: The average Euclidean norm between 3D keypoints and their transformed versions, measuring the pose estimation accuracy. We compute the area under the curve (**AUC**) lower than a fixed threshold (10cm), median and mean values.

B. Comparison with Baselines

Table I showcases a notable performance enhancement of our method compared to state-of-the-art (SOTA) techniques. In single-frame scenarios, our approach surpasses the current SOTA, RoboPose [8], by 37.99% and 28.72% in AUC for RealSense-Franka and AzureKinect-Franka, respectively. Moreover, our inference speed increases from 1FPS to 12 FPS in the single-frame mode. This is due to the iterative rendering process involved in RoboPose, which is highly time-consuming. In online scenarios, our method consistently outperforms RoboPose. Besides, the visualisation results in Figure 3 also support our method's superiority, where white boxes highlight our better predictions than RoboPose's [8]. Our faster inference speed is attributed to the efficient DDIM sampler [33] and our online sampling strategy, both of which significantly reduce sampling steps. Compared to the depth-based SPDH [9], our method, despite a marginally slower inference speed, exhibits considerable advantages, particularly in AzureKinect-Franka. This is attributed to the theoretical limitations of the cross-camera generalisation issue. We will discuss this in Sec IV-C The commendable results of our approach can be credited to our decoupling scheme, allowing each module to specialise in a simpler sub-task.

C. Cross-Camera Generalisation Analysis

In practice, an ideal online calibration tool should adapt to different cameras. In this section, we evaluate our method's cross-camera generalisation capacity against the keypoint-based approach SPDH [9] in the context of unknown joint angles. To ensure a fair comparison, we create three synthetic datasets emulating distinct camera fields of view (FOV):

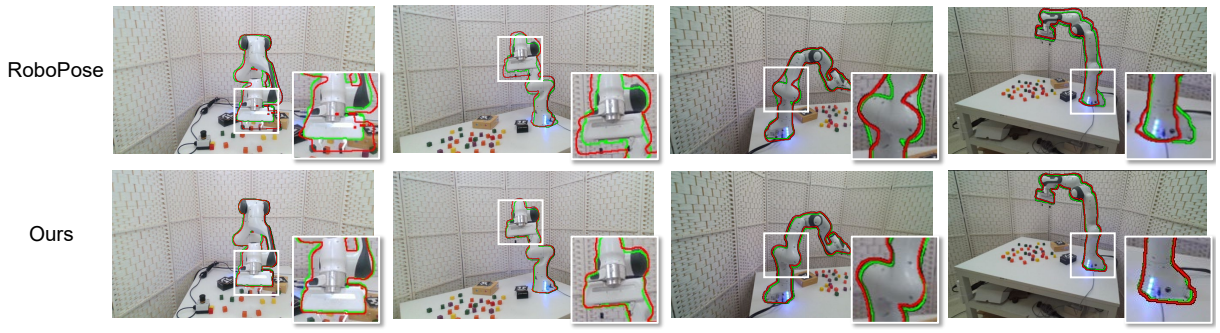

Fig. 3. **Visualisation results on real-world datasets**. Green edges are ground truth while red edges are rendered via estimated robot pose and joint angles. White boxes highlight regions where ours (online) performs better than RoboPose (online) [8].

TABLE II

QUALITATIVE RESULTS OF THE CROSS-CAMERA EXPERIMENT. RESULTS SHOW THAT OUR METHOD PERFORMS ROBUSTLY ACROSS DIFFERENT CAMERAS WHILE SPDH [9] FLUCTUATES DRAMATICALLY.

Method	AUC@0.1m	Median(m)	Mean(m)
SimXBox360Kinect (FOV@62.73)			
SPDH-HRNET [9]	60.80	0.034	0.058
SPDH-SH [9]	71.20	0.027	0.029
SPDH-RESNET [9] (Ours)	69.84	0.029	0.030
Ours (online)	**77.47**	**0.017**	**0.025**
SimSense (FOV@70.21)			
SPDH-HRNET [9]	7.88	1.269	1.084
SPDH-SH [9]	61.37	0.038	0.039
SPDH-RESNET [9] (Ours)	58.98	0.040	0.041
Ours (online)	**76.53**	**0.018**	**0.026**
SimAzureKinect (FOV@93.01)			
SPDH-HRNET [9]	0.00	2.097	2.064
SPDH-SH [9]	0.04	0.413	0.449
SPDH-RESNET [9] (Ours)	0.00	0.211	0.233
Ours (online)	**69.01**	**0.022**	**0.040**

SimXBox360Kinect(FOV@62.73), SimSense(FOV@70.21), and SimAzureKinect(FOV@93.01), keeping other elements, such as robot pose, robot joint angles, and background, consistent. Table II reveals a notable performance decline for SPDH across varying cameras, while our method remains stable. This observation is reinforced by results from the real datasets **RealSense-Franka** and **AzureKinect-Franka** in Table I. The primary reason for this substantial difference is that SPDH relies on XYZ-maps as inputs and employs convolutional networks as backbones. Consequently, when the topological structures of XYZ-maps are transformed due to changes in camera intrinsics, the translation invariance property of convolutional networks leads to misguided predictions in SPDH's UZ map. Our method's resilience is attributed to our task decoupling. Specifically, for the conditional 3D keypoints generation, we employ normalised camera coordinates $\tilde{c}$, unaffected by camera intrinsic alterations. Additionally, prior studies [5] have already proved the satisfying cross-camera generalisation capacity of 2D keypoints detection.

D. Ablaton Studies

1) Conditional generation vs. regression: Here we evaluate the efficacy of our conditional 3D generation module against direct regression. Utilising the framework by Martinez et al. [37], we adapt it as a regression baseline to transform 2D keypoints into 3D. Table III contrasts our method with this baseline, underscoring a significant enhancement with our method. This improvement can be credited to the generative models' advanced nonlinear modeling capabilities and their resilience to noise disturbances frequently observed in 2D keypoints detection, such as missing or noisy keypoints.

TABLE III

ABLATION BETWEEN GENERATION AND REGRESSION.

Method	2D	AUC@0.1m	Median(m)	Mean(m)
RealSense-Franka				
Regression	GT	37.48	0.060	0.067
Regression	Prediction	30.84	0.064	0.084
Generation	GT	**83.51**	**0.015**	**0.016**
Generation	Prediction	72.93	0.022	0.028
AzureKinect-Franka				
Regression	GT	43.28	0.049	0.067
Regression	Prediction	37.66	0.058	0.076
Generation	GT	**82.33**	**0.016**	**0.018**
Generation	Prediction	63.33	0.028	0.046

TABLE IV

IMPORTANCE OF CONDITIONING ON NORMALISED CAMERA COORDINATES TO GENERATE 3D KEYPOINTS ACCURATELY.

Method	AUC@0.1m	Median(m)	Mean(m)
RealSense-Franka			
w/o NCCS (online)	0.00	0.413	0.434
Ours (online)	**72.93**	**0.022**	**0.028**
AzureKinect-Franka			
w/o NCCS (online)	0.00	0.397	0.412
Ours (online)	**63.33**	**0.028**	**0.046**

2) Normalised camera coordinate space (NCCS): Here we highlight the impact of normalised camera coordinates space. In Table IV, w/o NCCS indicates models trained solely on raw 2D keypoints. Notably, w/o NCCS exhibits a substantial error, approximating 40cm in both median and mean ADD. The reason behind this exceptionally poor performance is that the diffusion model not only needs to memorise the robot's structural information, but also excessively fits to the fixed training camera intrinsics and projection formula. Therefore, when exposed to a novel camera, *w/o NCCS* struggles to adjust to the altered intrinsics. Conversely, the integration of NCCS provides a standardised 2D representation, effectively mitigating disruptions from diverse intrinsics and ensuring

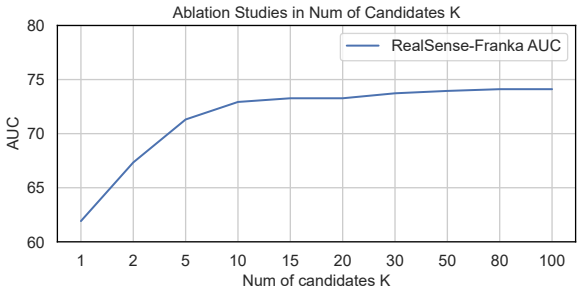

Fig. 4. Ablation on different number of 3D keypoints candidates K. We finally adopt $K = 10$ in implementation.

consistent performance across varying cameras.

TABLE V

ABLATION ON DIFFERENT SAMPLERS AND INITIALIZATIONS.

Sampler	Online	AUC@0.1m	Median(m)	Mean(m)	FPS
RealSense-Franka					
ODE	55	64.85	0.032	0.037	1
DDIM	55	67.00	0.027	0.035	12
ODE	✓	**73.94**	**0.021**	**0.027**	14
DDIM	✓	72.93	0.022	0.028	**18**
AzureKinect-Franka					
ODE	55	62.61	0.029	0.047	1
DDIM	55	60.72	0.030	0.049	12
ODE	✓	62.90	0.029	**0.044**	14
DDIM	✓	**63.33**	**0.028**	0.045	**18**

3) Samplers and initializations: We investigate the impact of different sampling solvers, specifically ODE [38] and DDIM [33], coupled with distinct initialization techniques on the sampling procedure. Table V demonstrates that the *DDIM* solver significantly reduces sampling time compared to the *ODE* solver, yet maintains comparable performance. Furthermore, *Online* initialization consistently outperforms the initialization from Gaussian noise in terms of both inference speed and performance, regardless of the type of solvers. This superiority of the *Online* initialisation can be attributed to its use of predictions from the last frame, which offers an initialisation proximate to the genuine distribution. Consequently, this enables shorter sampling steps and helps circumvent certain local optima. All the inference speeds were tested using a single V100 GPU.

4) Number of candidates: Figure 4 elucidates the impact of the number of 3D keypoints' candidates K during inference time. Regarding the AUC in RealSense Franka, the network's performance shows a great enhancement when K rises from 1 to 10. This can be explained that the augmented size of samples leads to a keypoints candidate set more closely aligned with the predicted distribution. Nonetheless, the enhancement becomes marginal as K extends to 100, likely due to the mean predictions nearing the upper limit of the sampling strategy. In view of the trade-off between performance and overhead, we adopt $K = 10$.

E. Additional comparison in settings with known joint angles.

While our primary emphasis is on estimating the robot pose and joint angles, our method demonstrates a marked advantage over prior methods, even when estimating robot pose with known joint angles. In this setting, we straightforwardly employ ground truth joint angles to reconstruct X^{rob} in Sec III-C, devoid of any specific additional design. As illustrated in Table VI, our method consistently outperforms others across all evaluation metrics. Notably, relative to SGTAPose [6], which integrates temporal information, our method exhibits a 20% enhancement in AUC. Moreover, against the CtRNet [7] that relies on additional real images for self-supervision, our method maintains superior performance with respect to ADD median and mean, achieving an average decrease of nearly 1 cm and 2.5 cm, respectively. It's pivotal to note that our method excels even in the absence of known joint angles, outpacing alternative methods that utilise them.

TABLE VI

ADDITIONAL COMPARISON WITH BASELINES IN SETTINGS WITH KNOWN JOINT ANGLES. "-" DENOTES ERRORS LARGER THAN 5M.

Method	AUC@0.1m	Median(m)	Mean(m)
RealSense-Franka			
Dream-VGG-Q [5]	27.48	0.080	0.244
Dream-VGG-F [5]	2.31	1.385	-
Dream-RESNET-H [5]	40.75	0.053	0.177
Dream-RESNET-F [5]	20.31	1.095	-
RoboPose (single-frame) [8]	44.21	0.050	0.062
RoboPose (online) [8]	44.18	0.050	0.062
SGTAPose [6]	52.00	0.036	1.370
CtRNet [7]	59.51	0.031	0.056
Ours (single-frame)	68.34	0.025	0.033
Ours (Online)	**74.76**	**0.020**	**0.026**
AzureKinect-Franka			
Dream-VGG-Q [5]	32.35	0.075	0.352
Dream-VGG-F [5]	0.37	1.471	-
Dream-RESNET-H [5]	51.05	0.038	0.133
Dream-RESNET-F [5]	38.60	0.053	-
RoboPose (single-frame) [8]	41.50	0.053	0.062
RoboPose (online) [8]	41.59	0.053	0.062
SGTAPose [6]	44.80	0.050	0.129
CtRNet [7]	55.22	0.035	0.062
Ours (single-frame)	63.74	0.027	0.045
Ours (Online)	**66.84**	**0.025**	**0.041**

V. CONCLUSION AND DISCUSSION

In this paper, we tackle the challenges in robot pose and joint angles estimation, specifically the efficiency-performance trade-off and cross-camera generalisation. To this end, we propose a novel framework named RoboKeyGen, which decouples this task into 2D keypoints detection and lifting 2D keypoints to 3D. Our method achieves high performance while preserving the efficiency inherent in keypoints-based methods. Our diffusion-based, conditional 3D keypoints generation effectively manages uncertainties arising from errors in 2D keypoints detection. Moreover, incorporating *Normalised Camera Coordinate Space* (NCCS) handles cross-camera generalisation issue. Experimental results show the effectiveness of our approach over state-of-the-art methods, achieving better performance with higher inference speed and improved cross-camera generalisation. **Limitations and future works**: One of limitations of our method is efficiency. Although our method outperforms render&compare based methods in performance and inference speed (18 FPS), it doesn't meet real-time requirements in certain scenarios. Future research could further explore faster sampling techniques.

REFERENCES

[1] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems," *ACM Computing Surveys (CSUR)*, vol. 52, pp. 1 – 31, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:146012430

[2] T. Taunyazov, W. Sng, H. H. See, B. Z. H. Lim, J. Kuan, A. F. Ansari, B. C. K. Tee, and H. Soh, "Event-driven visual-tactile sensing and learning for robots," *ArXiv*, vol. abs/2009.07083, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220070303

[3] F. Chaumette, "Image moments : a general and useful set of features for visual servoing," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:6783563

[4] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. Robotics Autom.*, vol. 5, pp. 345–358, 1988. [Online]. Available: https://api.semanticscholar.org/CorpusID:30068970

[5] 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," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208202164

[6] Y. Tian, J. Zhang, Z. Yin, and H. Dong, "Robot structure prior guided temporal attention for camera-to-robot pose estimation from image sequence," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8917–8926, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260126044

[7] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21 296–21 306, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257232804

[8] Y. Labb'e, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1654–1663, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233296915

[9] A. Simoni, S. Pini, G. Borghi, and R. Vezzani, "Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps," *IEEE Robotics and Automation Letters*, vol. 7, pp. 11 569–11 576, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250311091

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

[11] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1941–1950.

[12] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[13] J. Zhang, M. Wu, and H. Dong, "Genpose: Generative category-level object pose estimation via diffusion models," *arXiv preprint arXiv:2306.10531*, 2023.

[14] W. Hua, Z. Zhou, J. Wu, H. Huang, Y. Wang, and R. Xiong, "Rede: End-to-end object 6d pose robust estimation using differentiable outliers elimination," *IEEE Robotics and Automation Letters*, vol. 6, pp. 2886–2893, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225070704

[15] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. J. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *European Conference on Computer Vision*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221139756

[16] M.-Y. Wu, F. Zhong, Y. Xia, and H. Dong, "Targf: Learning target gradient field for object rearrangement," *ArXiv*, vol. abs/2209.00853, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252070636

[17] X. Lu, "A review of solutions for perspective-n-point problem in camera pose estimation," *Journal of Physics: Conference Series*, vol. 1087, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:125876238

[18] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Neural Information Processing Systems*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:196470871

[19] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, pp. 1661–1674, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:5560643

[20] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *ArXiv*, vol. abs/2006.11239, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:219955663

[21] J. N. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *ArXiv*, vol. abs/1503.03585, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:14888175

[22] Y. Song, J. N. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *ArXiv*, vol. abs/2011.13456, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227209335

[23] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," *ArXiv*, vol. abs/2111.08005, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244130146

[24] M. Wu, Y. Wang, H. Dong *et al.*, "Example-based planning via dual gradient fields," 2022.

[25] J. Zhang, M.-Y. Wu, and H. Dong, "Genpose: Generative category-level object pose estimation via diffusion models," *ArXiv*, vol. abs/2306.10531, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259202743

[26] H. Ci, M.-Y. Wu, W. Zhu, X. Ma, H. Dong, F. Zhong, and Y. Wang, "Gfpose: Learning 3d human pose prior with gradient fields," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4800–4810, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254823445

[27] J. Xu, Z. Xiong, and S. Bhattacharyya, "Pidnet: A real-time semantic segmentation network inspired from pid controller," *ArXiv*, vol. abs/2206.02066, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249395578

[28] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:13756489

[29] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen, "Rtmpose: Real-time multi-person pose estimation based on mmpose," *ArXiv*, vol. abs/2303.07399, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257504954

[30] Y. Li, S. Yang, P. Liu, S. Zhang, Y. Wang, Z. Wang, W. Yang, and S. Xia, "Simcc: A simple coordinate classification perspective for human pose estimation," in *European Conference on Computer Vision*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:250280272

[31] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," in *Neural Information Processing Systems*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235352469

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *ArXiv*, vol. abs/2010.02502, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222140788

[34] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang, "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in *European Conference on Computer Vision*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251402966

[35] "Blender," https://www.blender.org/.

[36] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[37] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2640–2649.

[38] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, pp. 19–26, 1980. [Online]. Available: https://api.semanticscholar.org/CorpusID:122754533