Fiducial Exoskeletons: Image-Centric Robot State Estimation

Anonymous Authors

Abstract— **We present a simple yet effective approach to robot state estimation that eliminates the need for expensive, high-precision actuators. Conventional methods rely on forward kinematics (FK) and precise motor encoders, but these degrade in accuracy due to backlash, thermal drift, and mismatched simulation–sensor dynamics. To improve both accuracy (especially on lower-cost hardware), we reformulate state estimation as a visual pose-estimation problem. Our method estimates the 6-DoF pose of each robot link from a single RGB image and performs a global joint optimization—optionally warm-started with encoder readings—to recover the full robot configuration. This formulation supports accurate 3D tasks such as joint-angle inference, camera pose estimation, and robot calibration directly from monocular images. To enable robust visual pose estimation of each link, we introduce the *fiducial exoskeleton*: a 3D-printed frame with fiducial markers, mounted on each link. The known marker-link relationship allows precise per-link pose estimation without sequential calibration or specialized equipment. Experiments on state-estimation and robotic control tasks demonstrate substantial accuracy gains over standard FK-based methods, even on such low-cost hardware. We will release our code and hardware designs to encourage algorithm–hardware co-design and reproducibility in the robotics community.**

I. Introduction

Accurate 3D state estimation and precise 3D control are fundamental components of robotics. The dominant paradigm for both is to use forward kinematics (FK), which integrates joint angles down the kinematic chain. However, the accuracy using FK depends on highly accurate motors and accumulates noise along the kinematic chain otherwise. Consequently, any pipelines requiring accurate 3D control and state estimation are typically demonstrated on highly expensive robot arms (usually over $10K) [1], [2], [3], and are difficult to deploy on lower-cost robots, which often have larger backlash and less precise encoder readings.

Other components of the 3D robotics stack are similarly restrictive and cumbersome. Camera-robot pose estimation requires an iterative approach which is tedious and importantly depends on highly accurate kinematic state estimation, similarly making it difficult for lower-cost hardware. Robot calibration is likewise clunky, either depending on operators to match an approximate a "target pose" iteratively move to the minimum and maximum joint angles, or requires complex hardware with homing sequences [4], [5]. These requirements render the 3D robotics control stack slow, brittle, and inaccurate for lower-cost motors.

To address these hardware and state-estimation challenges, we opt to use *vision* for robot state estimation, which has exciting potential as it observes a "holistic viewpoint" over the full robot state, instead of just integrating per-motor observations. In this work, we make vision a first-class signal for robot state estimation and control. We reformulate the entire 3D estimation and control stack around 6D pose estimation of each link from an RGB image and a simple optimization to recover joint angles. From the same set of poses, we also recover the robot to camera pose directly as well as robot calibration. We further make control more precise by using state-based refinement: after naively moving to the target state, we estimate the robot state from vision, compute the delta to the target state, and finally move the robot with the delta direction.

To facilitate per-link pose estimation, we introduce the *fiducial exoskeleton*: a lightweight 3D-printed mount with a fiducial marker [6], [7] for each link, providing an unambiguous marker-to-link transformation. The fiducial exoskeleton enables us to perform simple 6D pose estimation of each link from a single RGB image and without iterative calibration procedures which depend on the internal kinematics of the robot.

In summary, we introduce the following contributions:

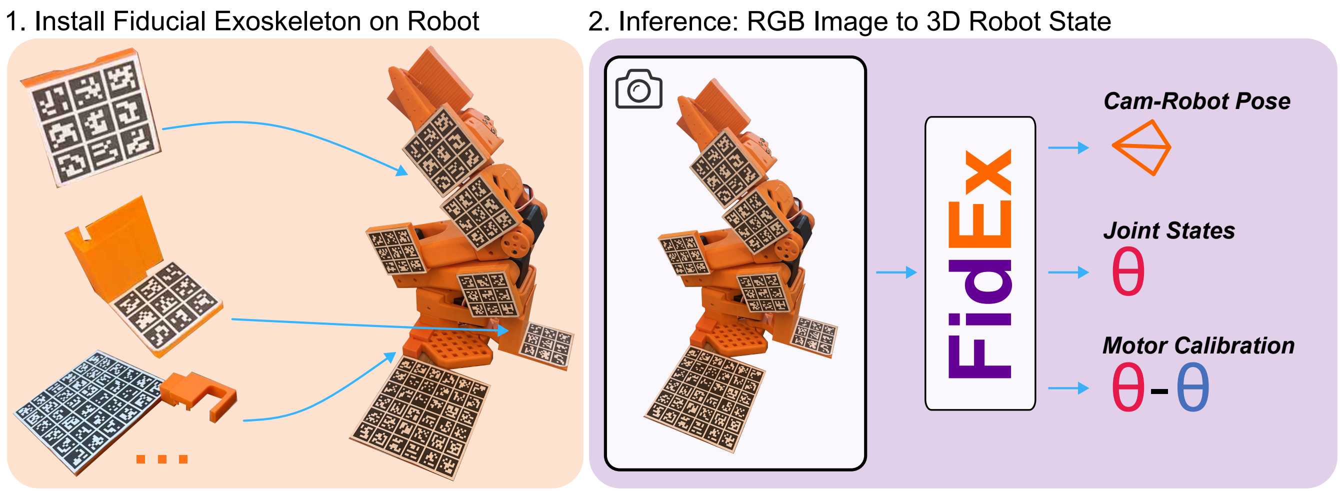
- A vision-centric reformulation of robot state estimation and control based on 6D link pose estimates, enabling recovery of joint angles, camera extrinsics, and robot calibration, from a single image and with significantly increased precision.
- A practical mechanism – the fiducial exoskeleton – which simplifies per-link pose estimation.
- On a low-cost 6-DoF arm, our method reduces end-effector state estimation and control error by $\sim$75% and $\sim$45%, respectively, over traditional forward-kinematics based estimation.

On a $100 robot arm (SO-100 [8]), our approach enables highly accurate state estimation compared to that from forward kinematics, as well as high-precision control.

II. Related Work

A. Classical Camera Pose Estimation

The robot-camera pose is an important component in many robotic pipelines, linking the image space to the robot 3D workspace. The standard way to obtain the camera pose is through hand-eye calibration [9], an iterative procedure where a robot operator attaches a fiducial marker on the end-effector and moves the robot through several kinematic states while capturing image observations. The pose is recovered via an $AX = XB$ optimization [10], [9]. Note that this iterative data collection and the $AX = XB$ optimization is required only because the relationship between the fiducial marker and the end-effector is unknown. This hand-eye calibration is iterative, considered tedious to set up, has to be performed


1. Install Fiducial Exoskeleton on Robot

2. Inference: RGB Image to 3D Robot State

Fig. 1: **FidEx Overview**

each time the robot or camera moves, and critically depends on accurate joint kinematics in order to provide correct motion pairs to the solver. Also note that this pipeline is not only tedious but difficult and inaccessible for low-cost robots with potentially noisy encoders.

In contrast, we simplify this design greatly with our fiducial exoskeleton by utilizing 3D-printed mounts where the relationship between the marker and the link is known, circumventing iterative data collection and the $AX = XB$ optimization.

B. Robot Keypoint Prediction for Camera Pose Estimation

Recent learning-based approaches aim to recover the camera pose from a single image, typically by predicting 2D keypoints of link centers [11], [12], [13], [14], [15], [16], [17] These approaches then solve a PnP optimization from the 2D keypoints to the 3D model obtained from the forward kinematics and known joint states. While reducing operator effort in avoiding the fiducial marker mounting and collection of multiple image observations, these methods similarly require highly accurate joint information, are only demonstrated on high-end robots ($>$ \$10k) with precise encoders, have questionable generalization, and have awkward processing of occluded links. In contrast, because our formulation predicts a full 6D pose for each link rather than sparse 2D locations, our method is able to not only recover the joint angles as well as the extrinsic calibration of external cameras with respect to the robot.

C. Robot Pose Estimation from Vision

Earlier work also explored markerless robot arm pose estimation with similar motivations of circumventing potentially unreliable kinematic state. These approaches aimed to recover joint angles but often involved complex pipelines, often leveraging depth maps and heuristic methods for link clustering and segmentation [18], [19], [20]. They also importantly do not recover the robot to camera pose. In contrast, our approach simplifies these designs greatly without using any training data, while offering not just joint recovery but also camera pose estimation.

More recently, differentiable rendering approaches such as *Dr. Robot* [21] attempt to estimate robot state by optimizing the parameters of a differentiable robot appearance model (e.g., Gaussian splatting [22], [23], [24]) to match input images. These methods are powerful for enabling image-space-to-robot gradients and have been proposed as a way to integrate predictions from large vision models into robot pipelines. However, they require accurate robot segmentation, are sensitive to lighting and appearance conditions, and optimize in the raw RGB space – making them slow, unconstrained, and often unstable. In contrast, our method leverages vision *not at the pixel level* but at the structured level of per-link 6D poses, yielding a much more constrained, low-dimensional, real-time, and accurate optimization for robot state estimation.

D. Latent Inverse Robot Dynamics from Videos

An orthogonal line of work aims to learn robot actions directly from video, often inferring *latent* actions to explain the observed visual trajectories [25], [26], [27], [28], [29], [30], [31], [32]. For instance, DreamGen proposes a multi-stage pipeline that fine-tunes a video generator with tele-operated video data to generate novel robot data, without leveraging explicit robot actions, and an inverse dynamics model or latent action model to label trajectories. Our method, in contrast, aims to recover instantaneous and explicit robot state. These video-based latent dynamics models are thus complementary to our method: they provide implicit priors for action generation and policies, whereas our method offers explicit and reliable state estimation, perhaps to be used in tandem.

III. Accurate State Estimation and Control of Robots using Fiducial Exoskeletons

Our aim is to estimate the full 3D state of the robot (including the robot base to camera pose as well as the joint

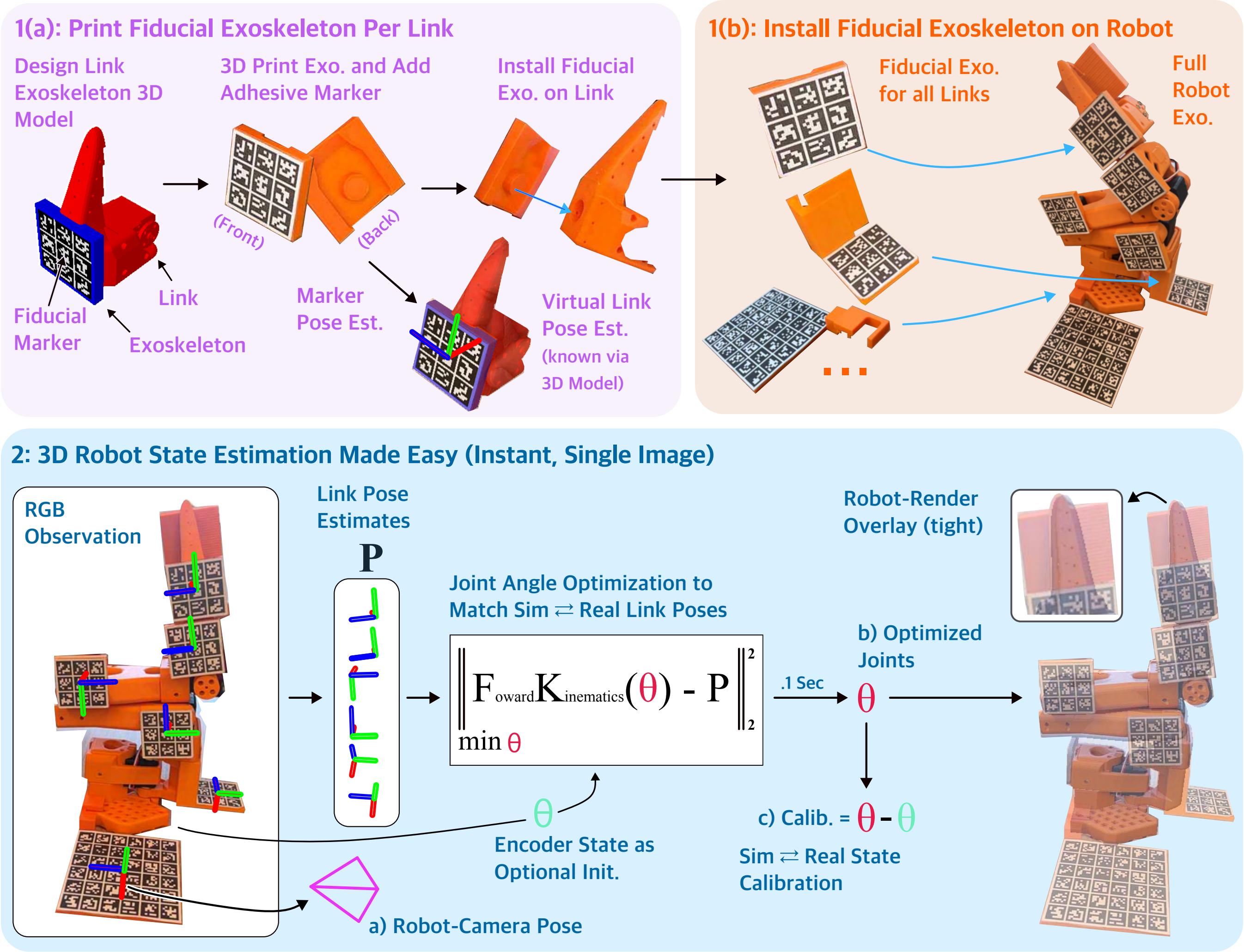

1(a): Print Fiducial Exoskeleton Per Link
Design Link Exoskeleton 3D Model
3D Print Exo. and Add Adhesive Marker
Install Fiducial Exo. on Link
Fiducial Marker
Link
Exoskeleton
(Front)
(Back)
Marker Pose Est.
Virtual Link Pose Est. (known via 3D Model)
1(b): Install Fiducial Exoskeleton on Robot
Fiducial Exo. for all Links
Full Robot Exo.
...
2: 3D Robot State Estimation Made Easy (Instant, Single Image)
RGB Observation
Link Pose Estimates
P
Joint Angle Optimization to Match Sim ⇄ Real Link Poses
$\min_\theta \left\| \mathrm{F}_{\mathrm{oward}}\mathrm{K}_{\mathrm{inematics}}(\theta) - \mathrm{P} \right\|_2^2$
.1 Sec
θ
Encoder State as Optional Init.
a) Robot-Camera Pose
Robot-Render Overlay (tight)
b) Optimized Joints
c) Calib. = θ - θ
Sim ⇄ Real State Calibration

Fig. 2: **Overview of using Fiducial Exoskeletons for robot state estimation.** (1a, top-left) Each link is fitted with a *fiducial exoskeleton*, a 3D-printed mount with a flat marker plane and a known marker-to-link coordinate transformation, enabling single-image 6D pose estimation for each link without any iterative calibration (Section III-B). (1b, top-right) The exoskeleton for each link is printed and mounted on the robot. (2, bottom) From an RGB image, we estimate all link poses, also immediately yielding the camera pose directly from the base's marker [2a]. A fast optimization recovers the joint angles [2b] which best match the observed link poses (Section III-A). We also recover the robot's calibration offset [2c] by comparing the optimized joints to the raw encoder joints. The final joint estimate tightly matches the physical robot, observed by the rendered robot overlay (rightmost).

parameters) from a single RGB image, without depending on the robot's internal motor readings (which we consider here to be potentially noisy and unreliable). To achieve this goal of estimating the robot state from vision alone, we propose to first estimate the 6D pose of each link and then perform a global optimization over joint states to match the observed link poses. We also introduce the *fiducial exoskeleton* to facilitate link pose estimation, which is effectively a 3D-printed attachment for each link with a fiducial marker. Below, we first describe the global optimization to recover joint states from link pose estimates III-A, then how we estimate the poses of links using fiducial exoskeletons III-B, and conclude with how we leverage this state estimation to control the robot with precision III-D.

A. *Recovery of Robot Joint States from Link Poses*

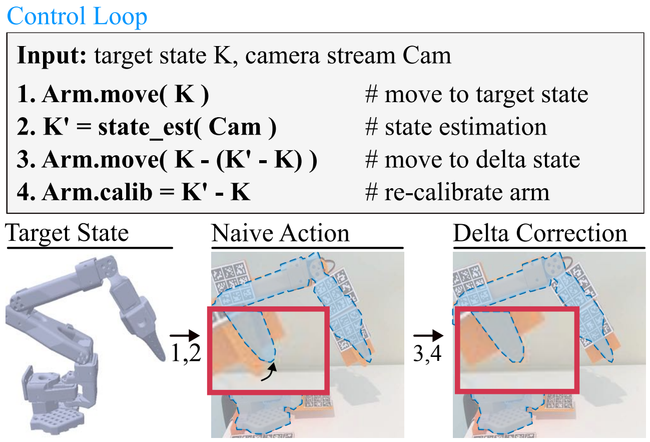
Visual estimation of the 6D pose for each link provides direct constraints on the joint parameters of the robot: by comparing the link poses induced by forward kinematics to those observed from vision, we can optimize the joint parameters to match the visual observations. Forward kinematics induces a set of per-link poses $\{T_j(\theta)\}_{j=1}^{L}$ by integrating the link transformations and joint angles down the kinematic chain. That is, for a d-DOF robot with joint parameters $\theta = (\theta_1, \ldots, \theta_d)$,

$$T_j(\theta) = \prod_{i=1}^{j} T_i^{i-1}(\theta_i), \tag{1}$$

Method	State Estimation Mask-IoU ↑	 Eff. Trans ↓	 Eff. Rot ↓
Ours Enc.	**.85**	**.06**	**.27**
Ours No-Enc.	.84	.07	.35
Just Enc.	.78	.18	1.1

Method	Control Mask-IoU ↑	 Eff. Trans ↓	 Eff. Rot ↓
Ours w/ Delta	**.79**	**.21**	**2.6**
Ours w/o Delta	.78	.22	2.8
Naive Move	.63	.37	3.8

TABLE I: State estimation (Left) and control (Right) results with fiducial skeletons are far stronger than using encoder-readings alone. For both state estimation and control, we compare the re-rendered robot using the estimated sensor state to the ground-truth robot mask (Mask IoU), as well as the rotation and translation distance for the estimated and true end-effector position. For state estimation, we compare the raw-encoder state (Just Enc.) to our method without using endoder states as initialization (Ours No-Enc.) and then with using the encoder states as well (Ours Enc.). For control, we compare the target robot state to the naive encoder motion (Naive Move), ours without the delta refinement (Ours w/o Delta), and with the delta refinement (Ours w/ Delta).



```
Input: target state K, camera stream Cam
1. Arm.move( K )                      # move to target state
2. K' = state_est( Cam )              # state estimation
3. Arm.move( K - (K' - K) )           # move to delta state
4. Arm.calib = K' - K                 # re-calibrate arm
```

Fig. 3: **The pseudo-code for our visual state-estimating control**. Top: the pseudo-code of our control loop, where we first move the arm naively to the target position, estimate the current robot state, move the delta between the observed robot and target robot state, and finally re-calibrate the robot offsets for the next target motion. Bottom: Illustrated steps of the control loop with the target state (Left), then the naive motion execution (Middle), and lastly the state-based refinement to better match the target state. We also highlight insets on the end-effector to better emphasize the differences between the target and physical states.

where $T_i^{i-1}(\theta_i) \in SE(3)$ is the transformation of joint i with respect to its parent frame $i-1$. Note that the forward kinematics integration is differentiable with respect to the joint states, an important property which we will leverage below.

In parallel, we assume access to a visual estimate of the same set of link poses,

$$\mathcal{P}_{\text{obs}} = \{P_j^{\text{obs}}\}_{j=1}^{L}, \tag{2}$$

where each $P_j^{\text{obs}} \in SE(3)$ is the 6D pose of link j.

We can then *solve* for the optimal joint states which best aligns the observed link poses to the ones induced by forward kinematics:

$$\theta^\star = \arg\min_\theta \sum_{j=1}^{L} d\big(T_j(\theta),\, P_j^{\text{obs}}\big)^2, \tag{3}$$

Method	1 Occ.	3 Occ.	4 Occ.	Upside Down
Ours (Enc)	**.88**	**.87**	**.87**	**.82**
Ours (No-Enc)	**.88**	**.87**	.75	**.82**
Just Enc	.81	.81	.81	.76

TABLE II: State estimation results (re-rendered robot mask IoU) under varying marker occlusions and robot orientation conditions. Note how even without any sensor information (Ours No-Enc), our optimization only falls into a degenerate minima when only the end-effector is visible.

where $d(\cdot,\cdot)$ is a distance metric on $SE(3)$. While this optimization is nonlinear, it is low-dimensional and can be solved with off-the-shelf nonlinear solvers (e.g. L-BFGS [33]) in real-time. Also, note that we can leverage encoder readings when available as the initial θ value used in the optimization, and just initialize the joint values with zeros otherwise. We find that the joint-state optimization typically converges to the same values with or without the encoder readings as initialization, and only observe an improvement in using the encoder readings as initialization when most links are occluded.

Also note that while the robot base is not technically a link, we assume the robot base pose is also estimated in this set of observed poses, and transform all link poses to the robot coordinate frame. In the next section, we also describe estimating the robot base pose with respect to external cameras.

B. Link Pose Estimation via Fiducial Exoskeletons

Above we illustrated how we can leverage 6D link poses to estimate the state of the robot. While there are indeed several potential parameterizations of the per-link pose regression, here we highlight another direction of *simply attaching a fiducial marker to each robot link*: fiducial markers provide SE(3) marker-to-camera pose estimates from a single RGB image, robust to partial occlusions and generalize trivially to a wide distribution of orientations. The key difficulty here, however, lies in registering the marker coordinate frame to the coordinate frame of the link they represent: while the standard approach of registering markers to the robot frame is the classical and multi-observation $AX = XB$

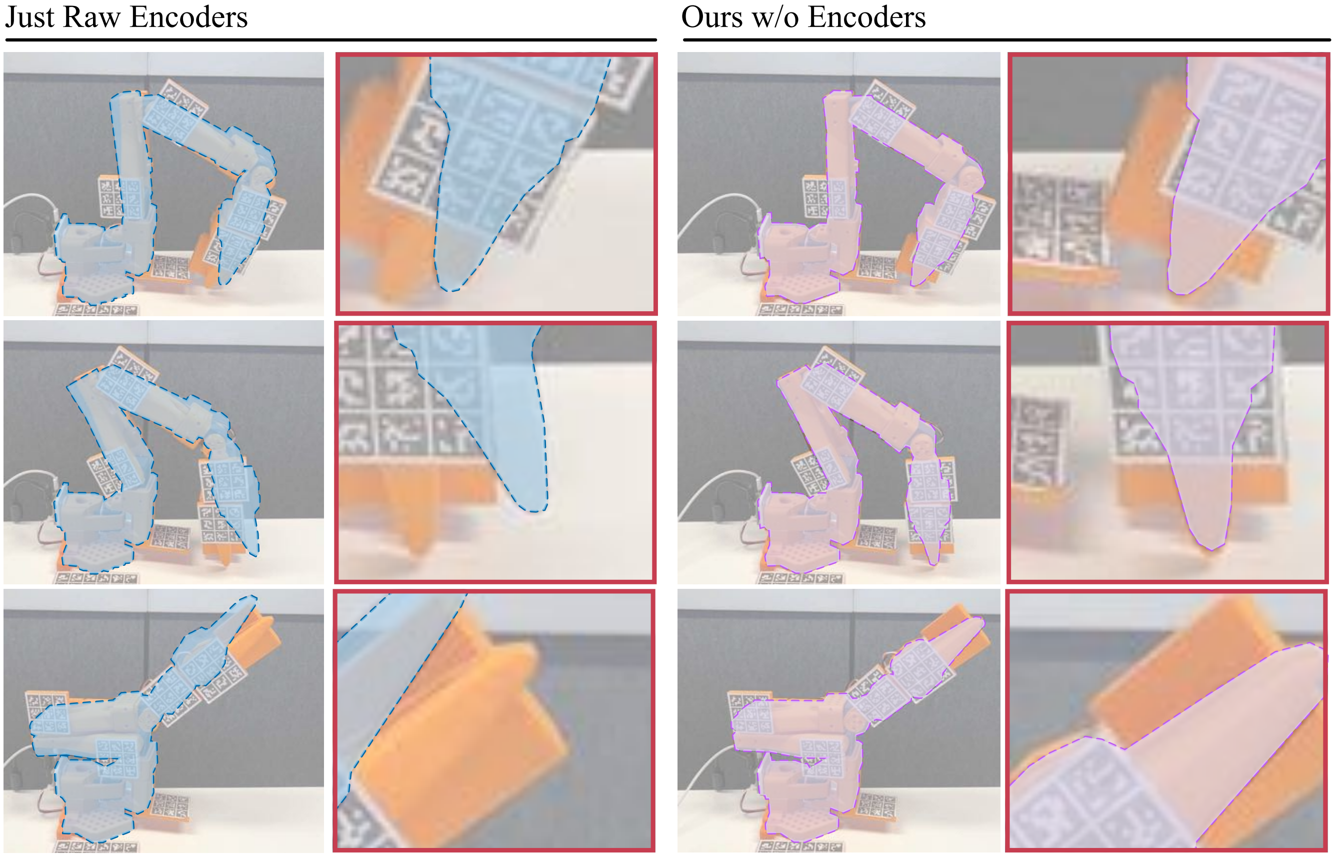


Fig. 4: **State estimation results.** (Left) Using raw encoder readings for state estimation (left), the rendered-robot is not well aligned with the physical robot. Using the fiducial exoskeletons, even without using any encoder readings as input, the robot re-render is aligned with the physical robot (Right). For each method, we plot the full-image overlay (inner left) and a highlighted inset on the end-effector (inner right).

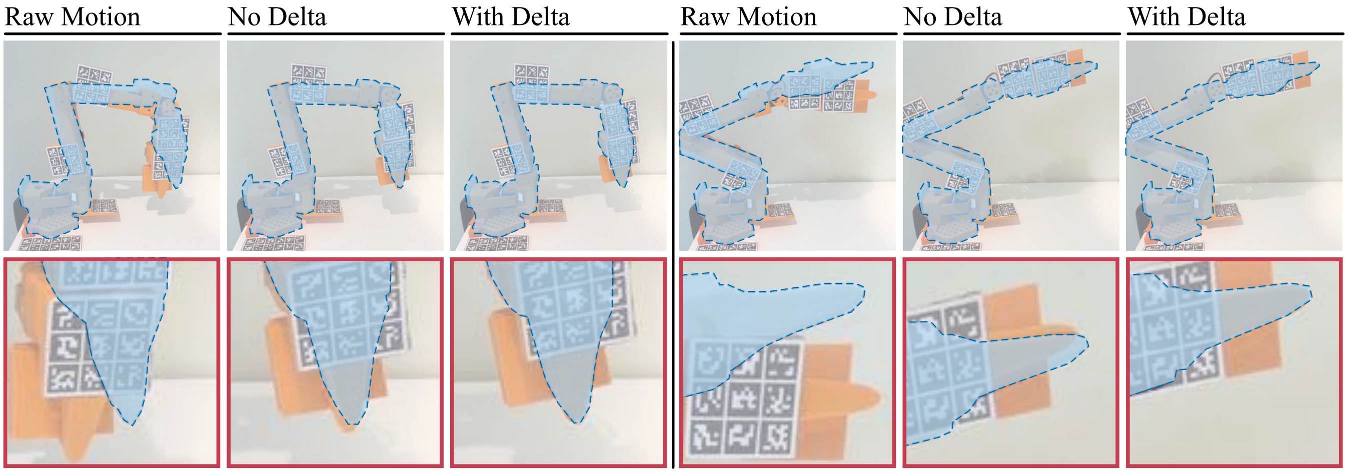


Fig. 5: **Robot control illustrations**. For each target state (blue outline) we compare the difference to the physical robot in using just the raw naive motion encoders (Raw Motion), the execution with our fiducial exoskeleton but without our delta refinement (No Delta), and finally with our delta refinement (With Delta). We also plot highlighted insets on the end-effector (Bottom). Without delta correction, the end-effector location is still close to the target, and the delta refinement further closes the gap between the target and executed motion.

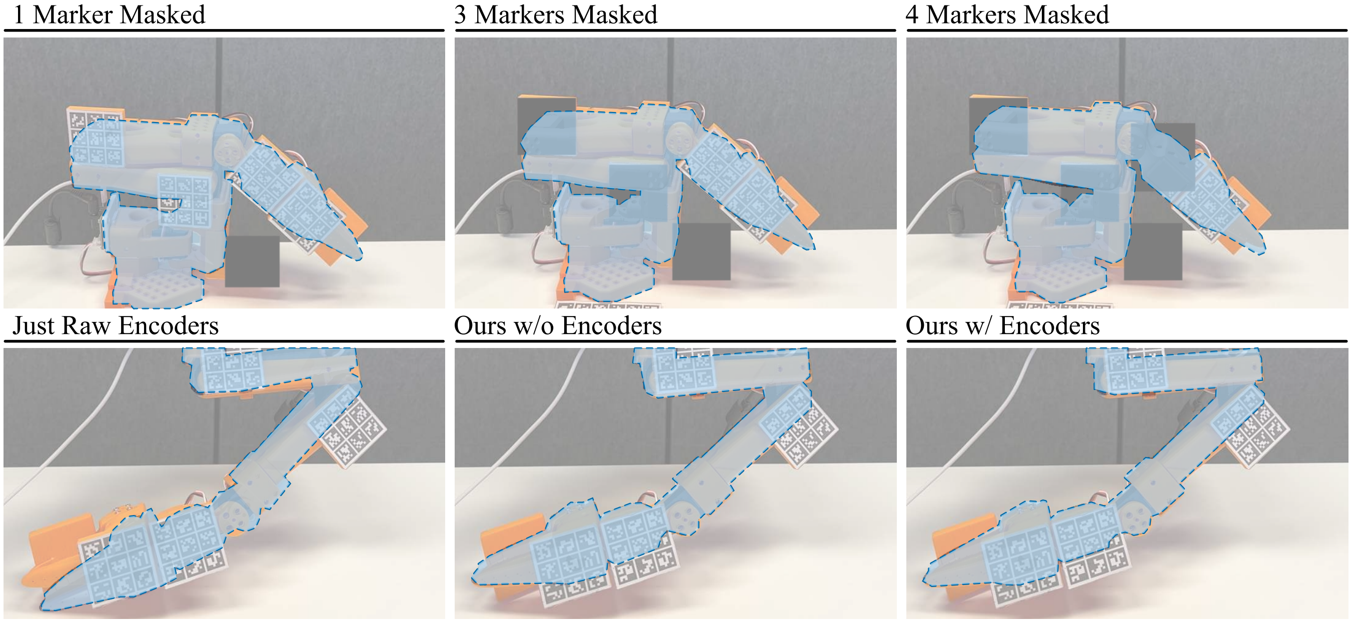


Fig. 6: **Robustness study**. (Top) State estimation results when masking out 1, 3, and 4 fiducial markers. Even with all but one marker occluded, our vision-based state estimation still produces strong results. Even with significant occlusion, our optimization remains stable, with encoder initialization helping in the most under-constrained cases. (Bottom) Unlike learning-based approaches which may struggles with out-of-distribution robot configurations, here we show state estimation even in an ‘upside-down’ state is similarly accurate for our method’s estimation.

calibration, this assumes access to accurate joint kinematics and a known camera pose. In the case of imprecise motors, accurate kinematic state or camera pose is difficult to obtain for such calibration.

To remedy these issues, we introduce the *fiducial exoskeleton*. For each link, we 3D-model a lightweight 3D-printed mount that attaches unambiguously onto the link. The fiducial exoskeleton consists of two parts: the component which mounts to the link, and a flat plane which holds the fiducial marker (see Fig. 2 for illustration). Importantly, this 3D model provides us the direct transformation between the plane to the link coordinate frame, which allows us to recover the 6D pose of the link from the 6D marker estimate, without any iterative calibration, kinematic state, or camera pose. Formally, the link pose in the camera frame is simply:

$$T_{\text{link}}^{\text{cam}} = T_{\text{aruco}}^{\text{cam}} \; T_{\text{exo}}^{\text{aruco}} \; T_{\text{link}}^{\text{exo}}, \tag{4}$$

where $T_{\text{aruco}}^{\text{cam}}$ is the pose of the fiducial marker in the camera frame, $T_{\text{exo}}^{\text{aruco}}$ is the transform from the marker to the exoskeleton,

C. Camera Robot Pose Estimation and Robot Calibration from a Single Image for Free

1) Camera Robot Pose: In the same way we attach fiducial exoskeletons to each link, we attach one to the robot base as well. This immediately delivers robot-to-camera extrinsics, rendering it as simple as estimating the pose of the base’s fiducial marker:

$$T_{\text{cam}}^{\text{robot}} = (T_{\text{aruco}}^{\text{cam}} \; T_{\text{exo}}^{\text{aruco}} \; T_{\text{robot}}^{\text{exo}})^{-1}. \tag{5}$$

This direct estimation eliminates the need for tedious and iterative hand-eye calibration procedures.

2) Robot Calibration: Also note that since our joint estimation does not require known encoder states, we can perform robot calibration from a single image as well. The joint offset is recovered as the difference of the raw encoder joints θ^{Enc} and the optimized joints $\theta^{\star}$:

$$\Delta\theta = \theta^{\star} - \theta^{Enc}. \tag{6}$$

Note this calibration procedure is performed from a single image and is dramatically simpler than current calibration procedures, which often involve manually moving the robot to a ‘target pose’ for alignment or require iterative and slow homing procedures.

D. Accurate Control via State Estimation Refinement

For the same reason that state estimation of a robot with potentially inaccurate sensors can be difficult – local regions of backlash, drift, etc. – commanding the robot to a target kinematic state θ^{Targ} with precision can be difficult as well.

We introduce a simple control loop to leverage our state estimation for pose refinement. For each kinematic target, we perform on-the-fly robot calibration, command the robot to the target state θ^{Targ}, estimate the current robot state $\theta^{*'}$, and move the robot with the delta between the observed and target state $\theta^{Targ} - (\theta^{*'} - \theta^{Targ})$.

The procedure pseudocode and visual results are described in Fig. 3. With this control loop, we are able to greatly increase the precision at which we can move the robot to a target kinematic state compared to naive execution without requiring high-precision encoders.

IV. Experiments

We benchmark our method in both robot state estimation and control on a low-cost robot in Figures 4,5 and Tables I,II. For state estimation, we evaluate how well we can estimate the joint parameters and 6D link positions in IV-A, and for control IV-C, how accurately we can move the robot links to target kinematic states or 6D link positions.

A. Estimating Robot State

For estimating the robot state, we compare the traditional approach of using FK ('Just Enc.') for the 6D position of each link, to our method – both with using the raw encoder states as input (Ours Enc.) and then without (Ours No-Enc). To compare the state estimation, we plot the robot mesh render overlayed on the robot in Fig. 4, and report Mesh IoU as well as the 6D pose error (position and rotation losses) for the end-effector, on a dataset of diverse robot configurations, in Table I. Our method produces state estimation which are much more aligned with the physical robot than that of naively using FK-based integration. See Fig. 4 to qualitative observe how much more aligned the robot re-rendering is with the physical robot. And quantitatively, see Tab. I where we report a ∼75% decrease in error in estimating the end-effector's SE(3) position.

B. Robustness Studies

One reasonable question is how our method performs when some subset of the links are occluded, or the robot is in 'out-of-distribution' positions. We cover various number of link markers (1, 3 and 4 markers), as well as move the robot to an upside-down position, and report accuracy in Table II and plot illustrations in Table II. When all links are occluded, our method reduces to the case of just using the raw encoder states, and when even one link (such as the end-effector) is visible, even this one constraint increases the state estimation accuracy significantly. We only find that in the case of not using any encoder state (not relevant in practice and more so a demonstration of surprising robustness) and only one link is visible, our estimation can fall into a degenerate minima. And since our state estimation does not involve any dataset or learning, there is no notion of out-of-distribution pose estimation: our pose estimation reduces to that of fiducial marker estimation, a classical method with robust implementation, and so this 'upside-down' state is trivially recovered as well. See Fig. 6 (bottom) to observe that our estimation in this 'upside-down' configuration is similarly aligned with the physical robot.

C. Control Precision Studies

To measure how precisely we can move the robot to a target 6D and kinematic state, we record a diverse dataset of target positions, and move the robot to each target state using our control loop. We similarly plot the robot mesh re-render of the target state onto the final reached state in Fig. 5, as well as quantitative Mesh IoU and end-effector 6D position accuracy, in Table I. We find that each component of the control loop is vital for accurate positioning, comparing our full method to the naive method of just moving the robot to the target state, as well as not using our state-estimating delta position refinement. Specifically, we find that even without using the delta-correction (effectively just re-calibrating the robot before moving), our method reduces the end-effector control error by ∼40%, and adding our delta-correction further decreases error by an additional ∼5%.

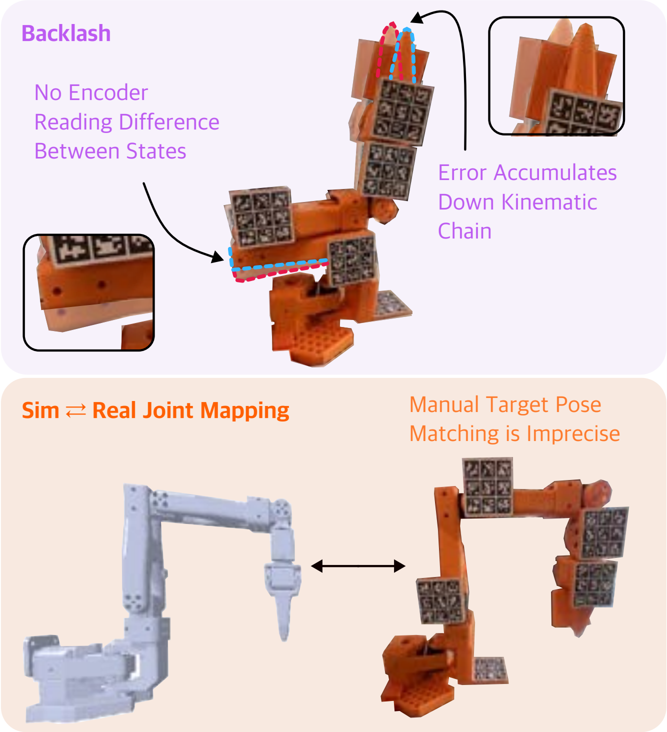

Backlash
No Encoder Reading Difference Between States
Error Accumulates Down Kinematic Chain
Sim ⇄ Real Joint Mapping
Manual Target Pose Matching is Imprecise

Fig. 7: **Challenges of 3D state estimation on low-cost robots.** (**Top**) *Backlash:* due to imperfect mechanical alignment in the internal motors, each joint has a margin of rotation in which encoders register no change. This unmeasured change accumulates down the kinematic chain, leading to large state error at the end-effector. (**Bottom**) *Calibration:* joint offsets are often calibrated by manually matching one or a small set of 'target' poses, yielding imperfect mappings.

V. Discussion

We have introduced Fiducial Exoskeletons, a simple and robust design for robot state estimation and control which does not depend on highly accurate and expensive motors, instead leveraging holistic image-based estimation. Our method is significantly more accurate the standard method (forward kinematics with raw sensor encoders) for robot state estimation and control on a low-cost 6DoF robot arm. We believe our design paves the way towards simpler and more accurate 3D robot state estimation and control, as well as increases capabilities for lower-cost robot arms.

A. Limitations

Fiducial Exoskeletons have several limitations that suggest exciting future explorations. First, the robot has to be observed by an external camera in which at least the base marker and one other marker are clearly visible: exploring the camera mount design to ensure proper visibility is an interesting constraint and future design consideration. One such solution could be to use one mounted camera just focused on observing the robot state and another for the robot and the rest of the scene together. The markers also have to be designed per-embodiment and per-link, which is laborious but is a constant design time per embodiment and can be distributed to all robot operators via 3D CAD files. And while aesthetics are not the primary concern of future robot policy learning, the markers are large and occlude much of the robot; future work on integrating more seamless and subtle marker design is interesting as well.

While there are indeed physical design limitations here, we hope this direction of 3D robot co-design increases the breadth of robots by which we can leverage analytical robot control and 3D-based policy learning, especially in the lower-cost regime of robots with less precise motors.

References

[1] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[2] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se(3)-equivariant object representations for manipulation," 2022.

[3] A. Wilcox, M. Ghanem, M. Moghani, P. Barroso, B. Joffe, and A. Garg, "Adapt3r: Adaptive 3d scene representation for domain transfer in imitation learning," 2025.

[4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "Robot dynamics and control," 2004.

[5] T. Edition and J. J. Craig, "Introduction to robotics," 2005.

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

[7] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, 2011.

[8] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, S. Palma, P. Kooijmans, M. Aractingi, M. Shukor, D. Aubakirova, M. Russi, F. Capuano, C. Pascal, J. Choghari, J. Moss, and T. Wolf, "Lerobot: State-of-the-art machine learning for real-world robotics in pytorch." https://github.com/huggingface/lerobot, 2024.

[9] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, p. 195–210, June 1995.

[10] R. K. Lenz and R. Y. Tsai, "Calibrating a cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 11, no. 09, pp. 916–928, 1989.

[11] 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.

[12] R. G. Goswami, P. Krishnamurthy, Y. LeCun, and F. Khorrami, "Robopepp: Vision-based robot pose and joint angle estimation through embedding predictive pre-training," 2025.

[13] Y. Tian, J. Zhang, G. Huang, B. Wang, P. Wang, J. Pang, and H. Dong, "Robokeygen: Robot pose and joint angles estimation via diffusion-based 3d keypoint generation," 2024.

[14] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," 2023.

[15] J. Lu, Z. Liang, T. Xie, F. Ritcher, S. Lin, S. Liu, and M. C. Yip, "Ctrnet-x: Camera-to-robot pose estimation in real-world conditions using a single camera," 2024.

[16] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render and compare," 2021.

[17] S. Ban, J. Fan, X. Ma, W. Zhu, Y. Qiao, and Y. Wang, "Real-time holistic robot pose estimation with unknown states," 2024.

[18] I. Bilić, F. Marić, I. Marković, and I. Petrović, "A distance-geometric method for recovering robot joint angles from an rgb image," 2023.

[19] F. Widmaier, D. Kappler, S. Schaal, and J. Bohg, "Robot arm pose estimation by pixel-wise regression of joint angles," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 616–623, IEEE, 2016.

[20] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, "Craves: Controlling robotic arm with a vision-based economic system," 2025.

[21] R. Liu, A. Canberk, S. Song, and C. Vondrick, "Differentiable robot rendering," 2024.

[22] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, pp. 1–14, 2023.

[23] X. Zhang *et al.*, "Nerf in robotics: A survey," *arXiv preprint arXiv:2405.01333*, 2024.

[24] F. Remondino *et al.*, "A critical analysis of nerf-based 3d reconstruction," *Remote Sensing*, 2023.

[25] S. Li, Y. Gao, D. Sadigh, and S. Song, "Unified video action model," 2025.

[26] N. Research, "Dreamgen." Project page, 2025. https://research.nvidia.com/labs/gear/dreamgen/.

[27] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg, "Amplify: Actionless motion priors for robot learning from videos," 2025.

[28] J. Jang *et al.*, "Dreamgen: Unlocking generalization in robot learning through neural trajectories," *arXiv preprint arXiv:2505.12705*, 2025.

[29] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo, "Latent action pretraining from videos," 2025.

[30] A. Villar-Corrales *et al.*, "Playslot: Learning inverse latent dynamics for controllable object-centric video prediction and planning," in *ICML*, 2025.

[31] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang, "Predictive inverse dynamics models are scalable learners for robotic manipulation," 2024.

[32] Y. Wen, J. Lin, Y. Zhu, J. Han, H. Xu, S. Zhao, and X. Liang, "Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation," 2024.

[33] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.