ROPES: Robotic Pose Estimation via Score-Based Causal Representation Learning

Pranamya Kulkarni* Google DeepMind

Puranjay Datta* Google DeepMind

Emre Acartürk Rensselaer Polytechnic Institute

Burak Varıcı Carnegie Mellon University

Karthikeyan Shanmugam Google DeepMind

Ali Tajer Rensselaer Polytechnic Institute

Abstract

We introduce **RO**botic **P**ose **E**stimation via **S**core-Based Causal Representation Learning (**ROPES**), a framework for recovering robot pose from raw images without sample-level labels. Existing vision-based estimators achieve high accuracy but rely on supervision or fiducials, limiting robustness due to domain shift, occlusion, and deployment at scale. ROPES adopts a generative view: images reflect latent factors such as geometry, lighting, background, and robot joints. The goal is to recover controllable latent variables, i.e., those linked to actuation. Interventional Causal Representation Learning (CRL) theory establishes that comparing distributions induced by interventions enables identifiability. In robotics, such interventions arise naturally by commanding actuators of various joints and recording images under varied controls. ROPES learns a disentangled 6-dimensional representation of a robot arm's state via a three-stage pipeline: (i) compressing images with an autoencoder, (ii) contrasting across interventional domains to estimate score differences, and (iii) refining these into six structured variables, where the final step is regularized using score differences to align estimated latent variables with the true joint angles. In semi-synthetic manipulator experiments, ROPES recovers latent representations that are highly disentangled, strongly correlated with true joint angles, and stable across settings. Crucially, this is achieved by leveraging only distributional changes, without using a single pose label at any step. This paper concludes by outlining challenges and positioning robot pose estimation as a near-practical testbed for measuring progress in CRL.

1 Introduction

Reliable knowledge of a robot's configuration, known as *pose*, is critical for a vast range of tasks, from robotic manipulation to safe human-robot interaction. Conventional pose estimation solutions include fiducial marker systems (e.g., ArUco [1]), geometric and sensor-fusion methods, and more recently, deep neural networks that regress poses or detect keypoints and infer joint angles. Pre-deep learning methods suffer from various issues. For instance, fiducial approaches require careful calibration and often fail at extreme orientations or under occlusion; even well-engineered marker pipelines can yield unstable estimates without tightly controlled imaging conditions.

While deep supervised methods have made substantial advances in the pose estimation problem, they typically rely on extensive labeled data and often depend on depth data or 3D CAD models. They are also sensitive to domain shifts, occlusions, and modeling assumptions (symmetry handling, reliance

*Equal contribution. Corresponding authors: `{pranamyapk,puranjaydatta}@google.com`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied World Models for Decision Making.

on depth, or bespoke post-processing). This reliance on specific conditions limits their generality: models trained for one workspace or lighting regime often degrade when deployed elsewhere, and bridging that sim-to-real gap remains an active challenge [2, 3]. Because such methods are tightly coupled to labeled regimes or engineered cues, they also offer limited guarantees about which internal representations the model learns (e.g., whether a latent dimension corresponds cleanly to a single physical joint). This raises the following question: can we recover interpretable and identifiable pose variables from images without per-sample supervision?

A generative viewpoint on pose. We frame pose estimation from a generative viewpoint, based on which each observed image x is generated by an unknown mapping $x = f(z)$ from a vector of latent variables z. This latent vector captures all factors of variation in the scene, e.g., lighting, camera intrinsics/extrinsics, background objects, and robot degrees of freedom (DoF). By modeling joint angles as latent factors embedded in a larger generative process, we can explicitly ask whether and how these variables can be recovered without explicit labels, and we can further allow *causal* interactions between those joints. This perspective reframes pose estimation as a problem of causal representation learning: recovering (a subset of) latent variables and their causal roles from high-dimensional observations, without direct supervision. Adopting this view offers a path toward not only label-free pose estimation, but also stronger interpretability and robustness properties grounded in formal identifiability results from the CRL literature.

Interventions as controllable variables. A central insight from recent CRL theory is that interventional data can enable identifiability of latent causal factors, even when the observation function f is unknown and highly nonlinear [4–6]. An *intervention* denotes a localized change in the latent data generation mechanism, and is a *distribution-level* phenomenon. Therefore, in contrast to per-sample labeling, this paradigm only requires dataset-level contrasts, e.g., sets of images recorded under different generative regimes or conditions. In robotics, interventions can be realized by grouping data collected under different actuation policies into different datasets. Each such control protocol yields a dataset whose distribution differs from others in ways that primarily reflect the changes in the altered, or *intervened*, latent factors.

Data availability and identifiability. While a robot's joint angles are part of the true data-generating factors, they are not the only ones; variables like camera pose and lighting also contribute if they are not fixed. Given this potentially large and unknown set of latent variables, we target the subset of latent variables for which interventions exist, i.e., controllable variables. This focus aligns with the practical reality of robotics, where an operator can command specific actuators to generate the necessary distributional contrasts. This approach is also grounded in fundamental CRL theory. Specifically, unsupervised disentanglement (in this case, the recovery of latent variables without mixing) is ill-posed without interventions or additional inductive biases [7]. Conversely, a wide variety of CRL results show that suitably designed interventional collections can yield identifiability for the intervened latents (up to well-understood ambiguities). Our approach is built on this principle: we do not claim to recover all latent factors (lighting, unknown background objects, sensor noise), but we do aim to recover the controllable subset, which are the variables directly implicated in robot pose and those that are manipulated by actuator-based interventions, and to align specific latent variables with those joints.

Methodology. We propose **RO**botic **P**ose **E**stimation via **S**core-Based Causal Representation Learning (**ROPES**) to learn a disentangled 6D representation of a robot arm's state from images. First, we collect interventional datasets by changing the distribution of one joint at a time. Then, we follow a three-stage pipeline summarized in Fig. 2. Inspired by score-based CRL [8], we leverage the sparsity of the (Stein) score function differences across these interventional datasets. A convolutional autoencoder compresses images into a latent space, and a classification network contrasts these latent mid-step representations to estimate score function differences, which is the key building block for score-based CRL. Then, a second autoencoder refines the initial latent encoding using these score signals into 6 variables, where each variable aligns with a joint angle (up to scale), enabling pose recovery.

Contributions. This paper makes the following contributions:

- **Formulation:** We formalize pose estimation as a CRL problem in which robot joint angles are treated as a controllable subset of latent causal variables embedded in a larger generative mapping $x = f(z)$.
- **Method:** We propose ROPES, an autoencoder-based architecture augmented with interventional regularizers based on pre- and post-intervention score functions that encourage individual latent dimensions to align with intervened degrees of freedom.

- **Empirical validation:** Through semi-synthetic experiments using a multi-joint manipulator rendered with realistic variability, we show that ROPES recovers latent variables that correlate strongly with ground-truth joint angles.
- **No reliance on pose labels:** Our work shows disentanglement by detecting distributional changes and therefore requires no conventional supervision from pose labels.

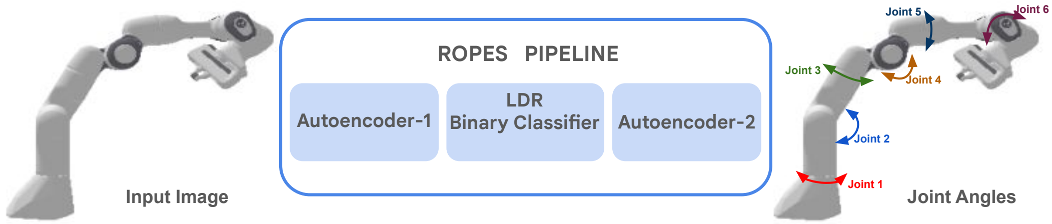


Figure 1: Conceptual overview of ROPES, highlighting its three-stage pipeline. The output visualization marks the specific joints targeted for intervention, along with their respective axes of rotation.

2 Related Work

Pose estimation in robotics. Various classes of methods, such as classical and learning-based, exist for robot pose estimation. Classical approaches, such as those using markers (ArUco [1]), rely on attaching fiducial markers to the robot's joints and locating them in the pixel space to estimate the joint positions. These methods are precise under controlled conditions but degrade with occlusion, calibration errors, or marker loss [9]. Learning-based approaches eliminate the need for physical markers by estimating the joint positions directly from images: DREAM [10] frames the problem as 2D keypoint detection, training a neural network to produce belief maps for each joint location using an L_2 loss against ground truth pixel coordinates. RoboPose [11] uses an iterative refinement strategy, directly minimizing an L_2 loss between the predicted and ground truth angles, while RoboPEPP [12] leverages a powerful pretrained encoder (I-JEPA [13]) before supervised regression. Despite their promising accuracy, all such methods depend on labeled data – either joint angles or keypoints – making them expensive to deploy broadly and sensitive to shifts in appearance or environment. In contrast, our approach learns directly from unlabeled images by exploiting distributional differences induced by robot actuation.

Interventional CRL. A substantial literature studies causal representation learning (CRL) with interventions, showing that latent factors become identifiable once data from multiple environments are available. Most results clarify conditions under which CRL is possible, and often remain theoretical or require assumptions impractical for complex domains. Among the algorithmic frameworks that can handle general transformations [4, 6, 14, 15], score-based CRL [8] provides a principled and practical approach: it avoids restrictive assumptions on the latent causal model and offers provable recovery guarantees, and therefore is the most amenable framework for this application.

Importantly, this paper helps close the gap between theory and practice in CRL by showing scaled-up results on the robot pose estimation problem. Specifically, most experimental settings in CRL literature typically operate on toy datasets (e.g., low-dimensional synthetic variables or image datasets) that fail to reflect real-world complexity. Although our experimental setting uses a simulator, the data consists of realistic, high-dimensional images of a robotic arm under diverse actuation regimes. By showing that CRL can successfully recover joint angles, we provide evidence that these methods can scale to visually rich and structured domains.

3 Problem Setting: CRL for Robotic Pose Estimation

In this work, we have $z \in \mathbb{R}^d$ as the d-dimensional movable joint angles of a robotic arm, and x as the RGB image captured by a camera mounted at a fixed position. Hence, f is the unknown rendering function that maps the joint angles to the corresponding image i.e. $x = f(z)$. For robot planning, reinforcement learning and applying control policies on the robotic arm, it is desirable to know the physical state space of the arm, i.e. z. As described in the related work, largely current works assume some form of supervision to learn f (or its inverse) in a calibration phase.

Hence, our objective is to recover the joint angles z from the image x using tools from interventional causal representation learning by estimating a good approximation $h \approx f^{-1}$ for inverse mapping. To do so, we assume that we can perform single-joint RCT-style interventions on a set of random robot poses (denoted by random vector Z) by manipulating one joint angle independently randomly at a time (akin to an RCT) while allowing the variations of other joint angles to be distributionally similar to original set of robot poses. This new set of poses forms the interventional dataset. Note that this is a realistic assumption in robotic applications, as we can typically manipulate a specific joint angle randomly and independently. Formally, if we have interventional dataset (thus created) for a joint i, the angle of which corresponds to Z_i, then we aim to learn a mapping $h_i : \mathcal{X} \to \mathbb{R}$, such that $h_i(X)$ is a function of Z_i only by simply using the original random set of poses and the intervened set of random poses considered as observational and interventional distribution. Except for the knowledge that some specific joint has been distributionally intervened on, *we require no pose label* (not even a single one). *In short, the goal is to recover joint angles using images from different interventional distributions in the latent space of joint angles, without requiring any explicit pose annotation.*

4 Methodology

Latent causal generative model. In causal representation learning, we consider a set of latent causal *random variables* $Z = [Z_1, \dots, Z_d]$ with an unknown transformation $f : \mathbb{R}^d \to \mathbb{R}^n$ that generates a high dimensional observation $X = [X_1, \dots, X_n]$ via $X = f(Z)$. Denote the probability density functions (pdfs) of Z, X by p, p_X, respectively. As standard in the CRL literature, we assume that $n \geq d$, and f is invertible and differentiable.

Distribution of latent random variable Z factorizes with respect to a directed acyclic graph (DAG) that consists of d nodes and is denoted by $\mathcal{G}$ where node $i \in [d]$ of $\mathcal{G}$ represents Z_i. Directed edges of $\mathcal{G}$ capture the cause-effect relationships in the sense that if one were to *physically* intervene (fix a specific joint angle to a specific angle), the descendants of the intervened joint will see a change while the other joints (non-descendants) will be undisturbed. Then, according to causal Bayesian network formalism [16], p factorizes as $p(z) = \prod_{i=1}^{d} p_i(z_i \mid z_{\mathrm{pa}(i)})$ where $\mathrm{pa}(i)$ denotes the set of parents of node i. Note that the generation of Z_i is governed by the conditional distribution $p_i(z_i \mid z_{\mathrm{pa}(i)})$, which is often called the causal mechanism of node i. The generic goal of CRL is to recover the latent variables Z and the causal graph $\mathcal{G}$ from samples of X. As this goal is known to be unachievable without additional supervision or diversity in the data [7], various branches of CRL consider different types of additional information (see [17, 18] for a detailed review). One important branch is *interventional* CRL, which is the setting of this paper.

Single-node stochastic hard interventions. This is the most commonly studied type of intervention in the CRL literature [5, 6, 8, 19]. Specifically, a stochastic hard intervention on node i removes the effects of its parents and replaces the causal mechanism $p_i(z_i \mid z_{\mathrm{pa}(i)})$ with a distinct mechanism $q_i(z_i)$. The resulting interventional distribution is given by:

$$q(z) = q_i(z_i) \prod_{i \neq j}^{d} p_j(z_j \mid z_{\mathrm{pa}(j)}) \, . \tag{1}$$

By modifying the latent distribution p in this way, such interventions introduce the desired statistical diversity in the observed data required for CRL. Question then becomes: *Given the local independencies created by hard interventions in the latent space, and observed in the X space through the same function f, can we invert this mapping f?* A recent line of work leverages the fact that a hard intervention renders the intervened variable independent of its ancestors. Therefore, when *multiple* hard interventions are applied to the same variable, the correct causal representation is characterized by a sparse change in the latent score space—the score functions of the latent variables. This induced sparsity can be exploited to learn causal representations. We briefly describe the key ideas formally below.

4.1 Score-Based CRL

Score function of the latent variables Z is given by $\nabla_z \log p(z)$, the gradient of the log-pdf. Following the methodology of Varıcı et al. [8], we adopt a *score difference* based technique for regularizing the latent space of the autoencoders for recovering the latent variables. In this framework, we use two hard stochastic interventions (specified in Equation (1)) for node z_i, denoted by q_i and $\bar{q}_i$. We require

q_i and $\bar{q}_i$ to be sufficiently different in distribution, formally defined via *interventional discrepancy* assumption [20, 8].

Definition 1 (Interventional Discrepancy). $\nabla_{z_i}(\log q_i(z_i) - \log \bar{q}_i(z_i))$ *can be zero only on a set of Lebesgue measure* 0.

Essentially, this condition states that the two interventions have different statistical imprints. Now, we aim to recover z_i (the joint angle in the pose estimation case). As such, for a particular joint of interest z_i, we create a pair of environments with the same camera modes but sampled from the two corresponding hard interventional distributions q and $\bar{q}$ for that particular joint i. Denote the score functions of these interventional distributions by $s_q, s_{\bar{q}} : \mathbb{R}^d \to \mathbb{R}^d$. As a direct consequence of Equation (1), [8, Lemma 7(iii)] implies the following sparse score changes property:

$$\mathbb{E}\left[\left|s_q(z) - s_{\bar{q}}(z)\right|\right]_j \neq 0 \quad \iff \quad j = i \; . \tag{2}$$

In other words, the score difference is a one-sparse vector in coordinate i. Therefore, a coordinate-wise scaled version of the true representation is a minimizer of the following loss (and it attains 0)

$$\mathcal{L} = \left\|\mathbb{E}\left[\left|s_q(z) - s_{\bar{q}}(z)\right|\right] - e_i\right\|_2^2 , \tag{3}$$

where e_i is the standard unit vector in dimension d with a one at position i.[2]

To algorithmically exploit this observation, we need to find the score difference in the true latent space. To this end, we propose to first find the score difference in the image space. Leveraging [8, Lemma 8], we can transform the score difference in X space to the unknown true Z space via

$$s_{Z_1}(z) - s_{Z_2}(z) = J_f(z)^\top \cdot (s_{X_1}(x) - s_{X_2}(x)) \, , \quad \text{where } x = f(z) \; , \tag{4}$$

where $J_f(z)$ denotes the Jacobian of f at point z. Let $h(x) = \hat{z}$ be the candidate encoder and $g(\hat{z}) = \hat{x}$ be the candidate decoder. Hence, the final loss function is derived from the score difference estimation transformation property Equation (4) and sparsity loss, which yields a score function regularized autoencoder learning problem given by optimization of the following loss:

$$\mathcal{L}(h, g) = \underbrace{\mathbb{E}\left[\|g \circ h(x) - x\|^2\right]}_{\text{Reconstruction Loss}} + \lambda \underbrace{\left\|\mathbb{E}\left[\left|J_g(\hat{z})^\top \cdot (s_q(x) - s_{\bar{q}}(x))\right|\right] - e_1\right\|^2}_{\text{Sparsity Loss}} . \tag{5}$$

This loss is a weighted sum of reconstruction loss of the autoencoder and score difference sparsity loss.

Theorem 1. *[8, Theorem 22 (reworded)] Assume that the latent distribution* p *has non-zero density over* $\mathbb{R}^d$*,* f *is a diffeomorphism onto its image and that pair* $(q, \bar{q})$ *satisfies interventional discrepancy. Then, the global optimizer* (h^*, g^*) *of* $\mathcal{L}(h, g)$ *recovers latent* z_i *up to an elementwise transform, that is,* $[h(x)]_i = \varphi(z_i)$ *for some* $\varphi : \mathbb{R} \to \mathbb{R}$*.*

We demonstrate the efficacy of this result in a scaled-up practical problem of robot pose estimation (using data generated by robot simulators) in the rest of the paper.

4.2 Data Generation

Our training process is unsupervised with respect to precise joint angles, meaning we do not require pose-labeled images. The core of our method relies on a dataset built from *hard interventions*. To create this data, it suffices to manipulate each robot joint individually and capture the resulting images. The images generated from hard interventions on joint i are labeled discretely based on the intervention identity. Following Section 4.1, for each joint i we have two interventional distributions, q_i and $\bar{q}_i$, and we assign labels '0' and '1' to the images drawn from q_i and $\bar{q}_i$, respectively. The observational distribution $p(\cdot)$ is formed by a random collection of robot arm poses. Then, for an interventional dataset on a specific joint, that joint's angle is resampled from a distribution different from its marginal in the observational setting. Importantly, our learning algorithm does not use metadata or even the statistics of these angle distributions, making it *unsupervised in the sense that it requires no pose labels*. Details of these distributions are described next, and exact parameterizations are given in the Appendix C.

[2]The nonzero score entry can be set to 1 by rescaling z appropriately: Multiplying z by c scales s_q by $1/c$.

We generated our dataset using a Franka Emika Panda arm in the panda-gym [21] simulator. The arm has six primary joint angles, which we set to create each image. The six joint angles have been marked in Fig. 1. Our data generation process was conducted in two stages. We initially focused on a simplified setup with a single camera, generating interventions for joints whose movements are largely confined to the camera's plane. To address the out-of-plane motions from other joints, which cannot be captured from a single viewpoint, we extended the dataset using a multi-view approach with two camera angles. The images in the dataset are converted to grayscale, resulting in a shape of $128 \times 128 \times 1$ for each image.

In-Plane Joints with a Single Camera. We first focused on a simplified task involving only joints 2, 4, and 6, as their movement primarily results in motion within a single plane for which a single camera view is sufficient. Each data point in this initial dataset is a set of six interventional images (2 for each joint) anchored by a single **observational state**. This observational state is generated by sampling a configuration for all six joints from a base truncated normal distribution. From this anchor, we create two distinct "hard interventions" for each of the target joints (2, 4, and 6). To perform an intervention on a specific joint, its angle is resampled from a distribution with a mean shifted far from the observational mean, while all other joint angles are held constant. This results in one observational image and six interventional images (3 joints $\times$ 2 interventions) per data point. Figure 6 in appendix shows a sample intervention on joint 4.

Full 6-DOF with Two Cameras. To extend our analysis to all six degrees of freedom, we considered joints 1, 3, and 5, whose actuation causes significant out-of-plane motion. To ensure full observability, we captured every pose from two distinct camera angles ($45°$ and $135°$ yaw). The data generation process was extended accordingly. Each data point now begins with an observational pose captured from both cameras (2 images). Subsequently, we perform two hard interventions on each of the six joints ($j \in \{1, \ldots, 6\}$), with each of these 12 interventional pose distributions also captured from both camera angles. Thus, a single complete data point in this extended dataset consists of 26 images: 2 observational images plus 24 interventional images (6 joints $\times$ 2 interventions/joint $\times$ 2 cameras/pose). Figure 7 in the appendix shows a sample intervention on joint 3 captured from 2 camera angles.

4.3 ROPES End to End Framework

Autoencoder-1 (AE1): Dimensionality reduction. The first stage compresses the high-dimensional visual data into a more manageable latent space. We train a deep convolutional autoencoder (AE1) to map a grayscale image $x \in \mathbb{R}^{128\times128\times1}$ to latent features $z_1 \in \mathbb{R}^{8\times8\times1}$. The encoder ($E_1$) and decoder ($D_1$) are symmetric, featuring a multi-stage architecture with residual blocks and group normalization for stable training. AE1 is trained on the entire dataset $\mathcal{D}$ by minimizing the mean squared error (MSE) reconstruction loss:

$$\mathcal{L}_{\text{AE1}} = \mathbb{E}_{x\sim\mathcal{D}}\big\|x - D_1(E_1(x))\big\|_2^2 \,. \tag{6}$$

The trained encoder E_1 serves as a fixed feature extractor for the next stage. Note that this stage is identical for both the single-camera and two-camera cases in terms of its input, output, and latent shapes.

Score difference estimator. It is well-established that a binary classifier trained to distinguish between two distributions learns their log-density ratio [22]. We leverage this principle to estimate the score difference between our two intervention types for each joint. For each joint i, we train a separate binary classifier, a Log-Density Ratio (LDR) estimator, f_{LDR}, following the methodology of [8]. This classifier is trained to distinguish whether a given latent vector $z_1 = E_1(x)$ was generated from the first (q_i) or second $\bar{q}_i$ interventional distribution. The classifier is optimized using a standard binary cross-entropy loss. After training, the gradient of the classifier's logit output, $\nabla_{z_1} f_{\text{LDR}}(z_1)$, provides a direct estimate of the score difference between the two distributions. It is important to note that this score difference is computed in the latent space of AE1, not the original pixel space. The input to the LDR network is adapted based on the experimental setup. In the single-camera configuration, the LDR directly processes the $8 \times 8 \times 1$ latent feature map, z_1. In the two-camera configuration, any given "sample" yields two latent vectors corresponding to the two camera angles. These are concatenated along the channel axis to produce a single $8 \times 8 \times 2$ input tensor for the LDR.

Autoencoder-2 (AE2): Latent space disentanglement. In the last stage, AE2 is trained with the compressed image z_1 in stage 1 as the input with the main objective in Equation (5) to minimize the reconstruction loss and score difference sparsity loss. The input to AE2 also depends on the

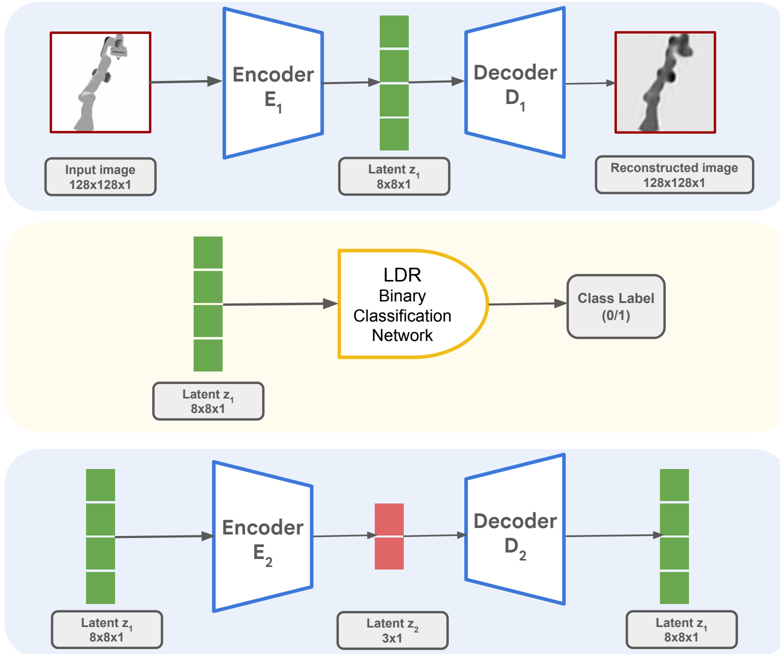


Figure 2: An overview of the ROPES pipeline, which begins with a shared Autoencoder 1 (AE1) compressing each $128 \times 128 \times 1$ image into an $8 \times 8 \times 1$ latent feature map. The subsequent data processing depends on the experimental setup. **For the single-camera case** (analyzing 3 joints), this $8 \times 8 \times 1$ map is directly fed into both the LDR network and Autoencoder 2 (AE2). AE2 then produces the final 3×1 disentangled pose vector. **For the two-camera case** (analyzing 6 joints), the latent maps from both camera views are concatenated along the channel axis, creating an $8 \times 8 \times 2$ input tensor for the LDR and AE2. In this pathway, AE2 outputs a 6×1 pose vector.

setup we are using: it is the $8 \times 8 \times 1$ latent map for single-camera experiments, or a channel-wise concatenation of the two views, forming an $8 \times 8 \times 2$ tensor, for the two-camera experiments. We perform a hyper-parameter search over the weight λ in Equation (5) for the best performance. Theoretically, the optimal latent space z_2's coordinate j gives us a *monotonic* transformation of the disentangled joint j angle. Empirically, we observe that this mapping is well-modeled by an affine transformation, enabling calibration via a small labeled dataset of ground truth samples.

Finally, the details of the AE1, LDR, and AE2 network architectures are given in Appendix A.

4.4 Empirical Takeways

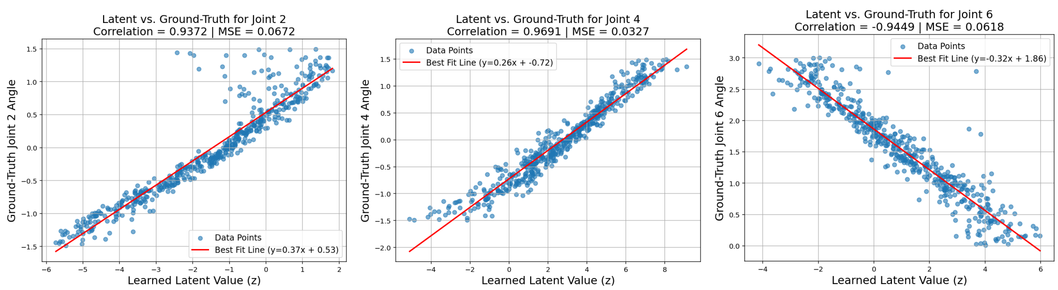
Before presenting our final results, we present our main empirical takeaways.

- In our experimental setup, we established distinct requirements for the two autoencoders. It is critical for the first autoencoder, AE1, to achieve high-quality data reconstruction. The second autoencoder, AE2, however, need not that highly optimized for reconstruction loss. Instead, its primary objective is to learn good enough representation that can effectively distinguish the dynamics of different joint movements.
- A significant challenge arose from the nature of the interventions used for training. When the interventions were starkly different, the model learned to distinguish them easily, reflected by a strong LDR loss signal. However, this apparent success was misleading, as the lack of any overlap between the intervention data led to poor performance in the actual joint disentanglement task. This issue was particularly evident in the six-joint experiments, where we found that two camera angles were necessary to resolve ambiguity between interventions. Certain joints performed out-of-plane rotations, which a single camera could not capture. The additional viewpoint provided crucial information that gave a lower cross-entropy for the optimized LDR loss, enabling the model to correctly differentiate the movements.

- On a practical note, the training process for AE2 exhibited high sensitivity to hyperparameter selection, especially the learning rate and the relative weights of the loss terms. We did a learning rate sweep from 10^{-3} to 10^{-6}. Other hyperparameters like the choice of the optimizer, architecture are described in the supplement.
- Furthermore, we found that integrating residual networks into the encoder and decoder architectures was an effective strategy for reducing reconstruction error and stabilizing training.
- Our framework disentangled only those joint angles whose interventions were used in the loss function Equation (5). Our framework achieves such partial disentanglement empirically at this scale. We are not aware of any larger scale demonstration of causal representation learning that shows such partial and incremental disentanglement when relevant interventions/actions/changes are available.

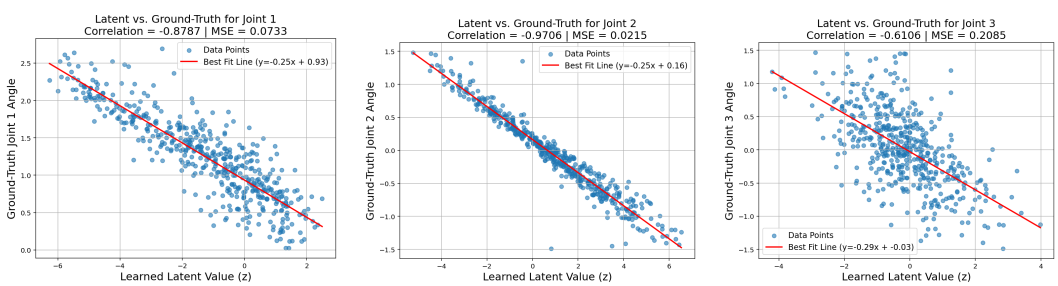
5 Results

We provide a qualitative analysis of the model's reconstruction performance in Appendix D. The results show good quality reconstructions for both experimental setups, with a minor performance decrease on the more visually diverse two-camera dataset, as detailed in the appendix.

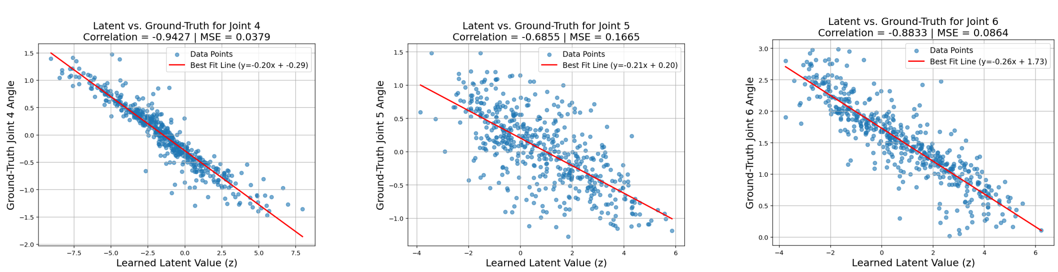


(a) Joint 2 (b) Joint 4 (c) Joint 6

Figure 3: Single camera: Scatter plots of ground truth and estimated angles for joints 2, 4, and 6.



(a) Joint 1 (b) Joint 2 (c) Joint 3



(d) Joint 4 (e) Joint 5 (f) Joint 6

Figure 4: Two camera angles: Scatter plots of ground truth and estimated angles for all six joints

To quantitatively assess disentanglement, we plot the learned latent variables against their corresponding ground-truth joint angles. Each scatter plot is annotated with the Mean Correlation Coefficient (MCC) and the Mean Squared Error (MSE) with respect to a linear best-fit line. Figure 3 presents these results for the three in-plane joints from the single-camera experiments. We observe that MCC is consistently above 0.93 for the single camera setup which is much better than results reported in prior experiments in CRL on much smaller scale image datasets before (see Table 14 in [8]). Figure 4 shows the results for all six joints from the multi-view experiments. Table 1 also presents these MCC and MSE values. In the two-camera experiments, we observe that the MCC scores for joints 3 and 5 are notably lower than those for the other joints. We attribute this performance discrepancy to less precise score estimates from the LDR network, an interpretation supported by the fact that these two joints exhibited a comparatively higher classification loss during LDR training. This suggests that the images of the hard interventions for joints 3 and 5 are less distinct, making them inherently more challenging to classify. Notably, the MCC scores for joints 2, 4, and 6 remain robust when transitioning from the single-camera to the two-camera setup. This result is significant because we are now disentangling 6 joint angles instead of 3 joint angles. The stability of these scores suggests that our method scales effectively, leveraging the multi-view information.

Table 1: MCC and MSE (in degrees) for single and two-camera setups for different joints

	Training Setup	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
MCC	Single Camera	-	0.94	-	0.97	-	0.94
	Two Cameras	0.88	0.97	0.61	0.94	0.69	0.88
MSE	Single Camera	-	3.85	-	1.87	-	3.54
	Two Cameras	4.20	1.23	11.94	2.17	9.53	4.95

6 Conclusion

In this work, we extend the Causal Representation Learning (CRL) approach from toy datasets to a semi-synthetic, close-to-real-world robotics simulator. We demonstrate that our method can recover the majority of robot joints, achieving a significantly high MCC and a very low MSE as shown in equation 1. Notably, this recovery is accomplished using only interventions, eliminating the need for explicit labels. This advancement holds significant potential for recent video world models in robotics, such as DreamGen [23]. These models typically operate by imagining future trajectories in a high-dimensional image space, which then serves as input for a Vision-Language-Action model to predict subsequent states. Our CRL-based approach can enhance this pipeline by enabling direct prediction of the robot's underlying state, i.e., its joint configuration. This dimensionality reduction of the observation space—from high-dimensional images to as few as six joint values—can substantially accelerate the learning process for the RL-diffusion policies. However, a key challenge is the sensitivity of the CRL framework to the quality of score estimates learned by the LDR module and hence the gradient calculation of the LDR output in the forward pass w.r.t the latent of Autoencoder (AE1). This sensitivity renders hyperparameter tuning for stable training to be particularly difficult. Furthermore, we observe instances where some joints exhibit slightly inferior recovery performance compared to others, even when their visual reconstructions appear to be of high fidelity. We believe this work opens several promising avenues for future research. One direction involves improving the model architecture, for instance, by replacing the CNN-based autoencoder with more advanced Vision Transformer (ViT) based networks. Concurrently, developing more robust and accurate methods for score estimation would directly address the aforementioned stability issues. Another compelling direction is to explore interventions in the action space rather than the state space of the robotic simulator. This shift could provide finer control over the generated trajectories and unlock the potential to leverage the causal relationships between actions to accomplish specific tasks more effectively.

References

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

[2] Kostas Ordoumpozanis and George A Papakostas. Reviewing 6d pose estimation: Model strengths, limitations, and application fields. *Applied Sciences*, 15(6), 2025. ISSN 2076-3417. doi: 10.3390/app15063284. URL `https://www.mdpi.com/2076-3417/15/6/3284`.

[3] Kai Chen, Rui Cao, Stephen James, Yichuan Li, Yun-Hui Liu, Pieter Abbeel, and Qi Dou. Sim-to-real 6d object pose estimation via iterative self-training for robotic bin picking. In *Proceedings of the European Conference on Computer Vision*, pages 533–550, Tel Aviv, Israel, 2022. Springer.

[4] Burak Varici, Emre Acartürk, Karthikeyan Shanmugam, and Ali Tajer. General identifiability and achievability for causal representation learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2314–2322. PMLR, 2024.

[5] Julius von Kügelgen, Michel Besserve, Wendong Liang, Luigi Gresele, Armin Kekić, Elias Bareinboim, David M Blei, and Bernhard Schölkopf. Nonparametric identifiability of causal representations from unknown interventions. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, December 2023.

[6] Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, December 2023.

[7] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. International Conference on Machine Learning*, Long Beach, CA, June 2019.

[8] Burak Varıcı, Emre Acartürk, Karthikeyan Shanmugam, Abhishek Kumar, and Ali Tajer. Score-based causal representation learning: Linear and general transformations. *Journal of Machine Learning Research*, 26(112):1–90, 2025.

[9] Pablo García-Ruiz, Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, Manuel J Marín-Jiménez, and Rafael Medina-Carnicer. Fiducial objects: Custom design and evaluation. *Sensors*, 23(24):9649, 2023.

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

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

[12] Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. Robopepp: Vision-based robot pose and joint angle estimation through embedding predictive pre-training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6930–6939, 2025.

[13] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv:2301.08243*, 2023.

[14] Jiaqi Zhang, Chandler Squires, Kristjan Greenewald, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, December 2023.

[15] Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from multiple distributions: A general setting. In *Proc. International Conference on Machine Learning*, Vienna, Austria, July 2024.

[16] Judea Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2009.

[17] Dingling Yao, Dario Rancati, Riccardo Cadei, Marco Fumero, and Francesco Locatello. Unifying causal representation learning with the invariance principle. In *Proc. International Conference on Learning Representations*, Singapore, April 2025.

[18] Aneesh Komanduri, Xintao Wu, Yongkai Wu, and Feng Chen. From identifiable causal representations to controllable counterfactual generation: A survey on causal generative modeling. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=PUpZXvNqmb`.

[19] Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *Proc. International Conference on Machine Learning*, Honolulu, Hawaii, July 2023.

[20] Wendong Liang, Armin Kekić, Julius von Kügelgen, Simon Buchholz, Michel Besserve, Luigi Gresele, and Bernhard Schölkopf. Causal component analysis. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, December 2023.

[21] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-source goal-conditioned environments for robotic learning. *arXiv:2106.13687*, 2021.

[22] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2), 2012.

[23] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv:2505.12705*, 2025.

A Architecture Details

Table 2 details the architecture of the first autoencoder (AE1), which is identical for both the single- and two-camera experiments. The architectures for the Log-Density Ratio (LDR) network and the second autoencoder (AE2), which are adapted for each setup, are presented in Table 3 and Table 4, respectively.

Table 2: Autoencoder1 architecture ResNet-style with GroupNorm

Component	Layer-wise Details
Block Def.	**ResBlockGN(f)**: GroupNorm → ReLU → Conv(features=f, ks=3, pad='SAME') → GroupNorm → ReLU → Conv(features=f, ks=3, pad='SAME') → Add residual input *(Note: 'ks'=kernel size, 's'=stride, 'pad'='SAME')*
Encoder	**Input: Image** $X \in \mathbb{R}^{128\times128\times1}$ Conv(features=64, ks=3, pad='SAME') ResBlockGN(64) x 2 Conv(features=64, ks=3, s=2, pad='SAME'), ReLU *// Downsample 128 → 64* Conv(features=128, ks=3, pad='SAME') ResBlockGN(128) x 2 Conv(features=128, ks=3, s=2, pad='SAME'), ReLU *// Downsample 64 → 32* Conv(features=256, ks=3, pad='SAME') ResBlockGN(256) x 2 Conv(features=256, ks=3, s=2, pad='SAME'), ReLU *// Downsample 32 → 16* Conv(features=512, ks=3, pad='SAME') ResBlockGN(512) x 2 Conv(features=1, ks=3, s=2, pad='SAME'), ReLU *// Downsample 16 → 8* **Output: Latent** $z_1 \in \mathbb{R}^{8\times8\times1}$
Decoder	**Input: Latent** $z_1 \in \mathbb{R}^{8\times8\times1}$ Conv(features=512, ks=3, pad='SAME') ResBlockGN(512) x 2 ConvTranspose(features=512, ks=4, s=2, pad='SAME'), ReLU *// Upsample 8 → 16* Conv(features=256, ks=3, pad='SAME') ResBlockGN(256) x 2 ConvTranspose(features=256, ks=4, s=2, pad='SAME'), ReLU *// Upsample 16 → 32* Conv(features=128, ks=3, pad='SAME') ResBlockGN(128) x 2 ConvTranspose(features=128, ks=4, s=2, pad='SAME'), ReLU *// Upsample 32 → 64* Conv(features=64, ks=3, pad='SAME') ResBlockGN(64) x 2 ConvTranspose(features=64, ks=4, s=2, pad='SAME'), ReLU *// Upsample 64 → 128* **Conv(features=1, ks=3, pad='SAME'), ReLU** *// Final convolution to 1 channel* Reshape to (batch, $128 \times 128 \times 1$) **Output: Reconstructed Image** $\hat{X} \in \mathbb{R}^{128\times128\times1}$
Training	Adam optimizer with learning rate = 1×10^{-4}

Table 3: LDR Network Architectures for Single- and Two-Camera Setups.

Component	Layer-wise Details
Input Processing	The input shape depends on the camera setup: **Single-Camera:** Input $z_1 \in \mathbb{R}^{8\times8\times1}$ is used directly. **Two-Camera:** Two z_1 maps are concatenated to form an input $\in \mathbb{R}^{8\times8\times2}$.
Core Architecture	The following layers are applied to the processed input: Conv(features=32, ks=3), ReLU *// Spatial dim: 8x8 → 6x6* Conv(features=64, ks=3), ReLU *// Spatial dim: 6x6 → 4x4* Conv(features=128, ks=3), ReLU *// Spatial dim: 4x4 → 2x2* Flatten Dense(features=128), ReLU Dense(features=1) **Output: Logit** $\in \mathbb{R}^1$
Training	Adam optimizer with learning rate = 1×10^{-3}. Minimize binary cross-entropy with logits loss on the output.

Table 4: Autoencoder 2 (AE2) Architectures for Single- and Two-Camera Setups.

Component	Layer-wise Details
Block Def.	**ResBlockGN(f)**: GroupNorm → ReLU → Conv(features=f, ks=3, pad='SAME') → GroupNorm → ReLU → Conv(features=f, ks=3, pad='SAME') → Add residual input *(Note: 'ks'=kernel size, 's'=stride, 'pad'='SAME')*
Encoder	**Input:** $z_1 \in \mathbb{R}^{8\times8\times C_{in}}$, where C_{in} is 1 (single-cam) or 2 (two-cam). Conv(features=64, ks=3, pad='SAME') ResBlockGN(64) x 2 Conv(features=64, ks=3, s=2, pad='SAME'), ReLU *// Downsample 8 → 4* Conv(features=128, ks=3, pad='SAME') ResBlockGN(128) x 2 Conv(features=128, ks=3, s=2, pad='SAME'), ReLU *// Downsample 4 → 2* Conv(features=256, ks=3, pad='SAME') ResBlockGN(256) x 2 Conv(features=256, ks=3, s=2, pad='SAME'), ReLU *// Downsample 2 → 1* Flatten to (batch, 256) Dense(features=D_{latent}), where D_{latent} is 3 (single-cam) or 6 (two-cam). **Output: Latent** $z_2 \in \mathbb{R}^{D_{latent}}$

Decoder	**Input: Latent** $z_2 \in \mathbb{R}^{D_{latent}}$
	Dense(features=256), ReLU
	Reshape to (batch, $1 \times 1 \times 256$)
	Conv(features=512, ks=3, pad='SAME')
	ResBlockGN(512) x 2
	ConvTranspose(features=512, ks=4, s=2, pad='SAME'), ReLU *// Upsample 1 → 2*
	Conv(features=256, ks=3, pad='SAME')
	ResBlockGN(256) x 2
	ConvTranspose(features=256, ks=4, s=2, pad='SAME'), ReLU *// Upsample 2 → 4*
	Conv(features=128, ks=3, pad='SAME')
	ResBlockGN(128) x 2
	ConvTranspose(features=128, ks=4, s=2, pad='SAME'), ReLU // Upsample 4 → 8
	Conv(features=C_{in}, ks=3, pad='SAME'), ReLU
	Reshape to (batch, $8 \times 8 \times C_{in}$)
	Output: Reconstructed $\hat{z}_1 \in \mathbb{R}^{8\times8\times C_{in}}$

Training	Adam optimizer with learning rate = 7×10^{-5}.

B Training Details

All models were trained on TPUs. We performed a hyperparameter search for the optimal learning rate, testing values in the range of $1e-7to1e-3$. Our dataset consists of 10,000 observational images. As detailed in Section 4.2, we generated corresponding interventional images for two experimental setups. In the single-camera setup, each observational image yields 6 interventional images. In the two-camera setup, it yields 24 interventional images. The training process involved three stages. First, Autoencoder-1 was trained on 70k images (single-camera) and 250k images (two-camera) with a batch size of 256. Second, we trained a separate Low-Dimensional Representation (LDR) model for each joint, using 20k samples (single-camera) and 40k samples (two-camera) with a batch size of 64. Finally, for each joint, we trained a corresponding Autoencoder-2 alongside its specific LDR. This final stage used 30k samples (single-camera) and 50k samples (two-camera) consisting of both observational and interventional images, with a batch size of 128.

C Interventional Distributions

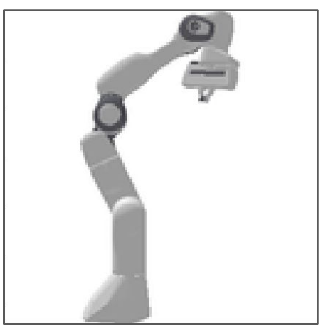
Tables 5 and 6 detail the observational and interventional distributions used to sample the joint angles for the single- and two-camera setups, respectively. We use a truncated normal distribution, denoted by $\mathcal{TN}_{[a,b]}(\mu, \sigma^2)$, which represents a normal distribution with mean μ and variance σ^2 truncated to the interval $[a, b]$.

Table 5: Sampling distributions for observational and interventional settings for single camera setup.

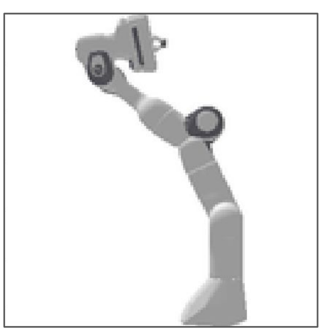
Joint	**Scenario**	**Distribution**
2	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\,1)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(-0.75,\,0.5)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ 0.75,\,0.5)$
4	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\,1)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(-0.75,\,0.5)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ 0.75,\,0.5)$
6	Observational	$\mathcal{TN}_{[0,\,3]}(1.5,\,1)$
	Intervention 1	$\mathcal{TN}_{[0,\,3]}(2.25,\,0.5)$
	Intervention 2	$\mathcal{TN}_{[0,\,3]}(\ 0.75,\,0.5)$

Table 6: Sampling distributions for observational and interventional settings where we include two camera angles.

Joint	Scenario	Distribution
1	Observational	$\mathcal{TN}_{[0,\,3]}(1.2,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[0,\,3]}(2.0,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[0,\,3]}(\ 0.6,\ 0.4)$
2	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(0.7,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ -0.7,\ 0.4)$
3	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(0.7,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ -0.7,\ 0.4)$
4	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(0.9,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ -0.9,\ 0.4)$
5	Observational	$\mathcal{TN}_{[-1.5,\,1.5]}(0,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[-1.5,\,1.5]}(0.9,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[-1.5,\,1.5]}(\ -0.9,\ 0.4)$
6	Observational	$\mathcal{TN}_{[0,\,3]}(0.5,\ 0.4)$
	Intervention 1	$\mathcal{TN}_{[0,\,3]}(2.4,\ 0.4)$
	Intervention 2	$\mathcal{TN}_{[0,\,3]}(\ 0.7,\ 0.4)$

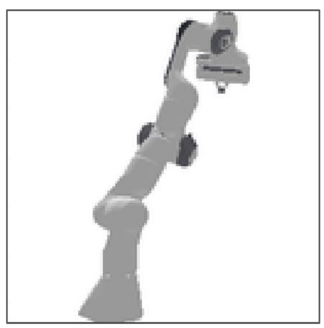


(a) Intervention 1 (label 0)

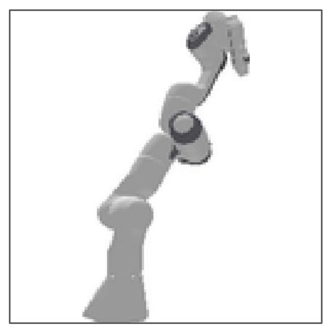


(b) Intervention 2 (label 1)

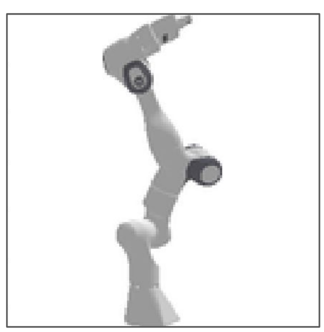
Figure 6: Two different hard interventions on Joint 4.



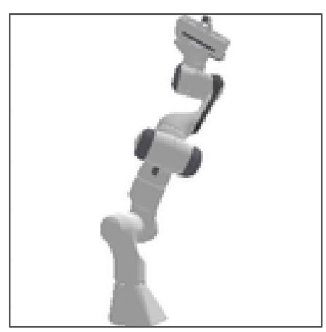
(a) Intervention 1 (45°)



(b) Intervention 2 (45°)



(c) Intervention 1 (135°)

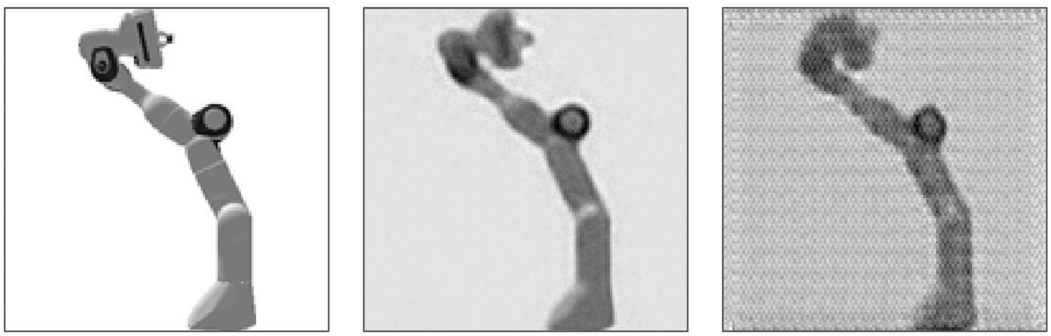


(d) Intervention 2 (135°)

Figure 7: Two different hard interventions on Joint 3, each shown from two camera angles.

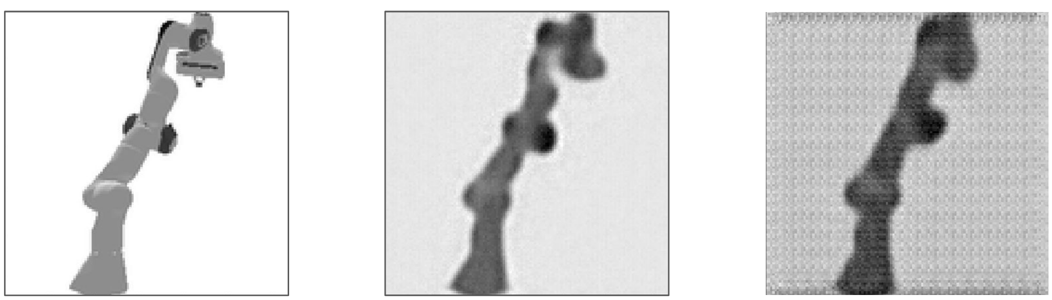
D Additional Dataset Figures

Figures 8 and 9 provide a qualitative analysis of our pipeline's reconstruction performance. Specifically, Figure 8 compares an original image to its reconstructions from AE1 and AE2 for the single-camera setup, while Figure 9 shows the equivalent comparison for the two-camera setup. We observe a slight degradation in the reconstruction quality of AE1 when trained on the two-camera dataset compared to the single-camera setup. As the final autoencoder, AE2, is trained on the latent representations from AE1, this reduction in quality of AE1 consequently affects the quality of the final AE2 reconstructions as well. We hypothesize that this performance difference is attributable to the increased data complexity of the multi-view dataset. The inclusion of multiple viewpoints introduces greater visual variance, presenting a more challenging reconstruction task for the autoencoder compared to the more constrained single-view data.



(a) Observational Image (b) AE1 Reconstruction (c) AE2 Reconstruction

Figure 8: Visual comparison of the reconstruction quality at each stage of our pipeline. (a) The original input image. (b) The reconstruction from the first autoencoder (AE1). (c) The final reconstruction from the second autoencoder (AE2)



(a) Observational Image (b) AE1 Reconstruction (c) AE2 Reconstruction

Figure 9: Visual comparison of the reconstruction quality at each stage of our pipeline using the models trained on the dataset generated using 2 camera angles

E Licenses

1. Panda-gym :
    - Citation : Galloućdec et al. [21]
    - License : MIT

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims reflect the paper's scope and contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in Section 5 and 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we have disclosed all the information needed to reproduce the main experimental results of the paper in Section 4, 5, and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We provide sufficient amount of details about hyper parameters and detailed architecture to be reproducible at this point.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the required experimental details have been provided in Section 4, 5, and Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to lack of resources at the moment we were unable to perform multiple experiments needed for error bar calculation. However we will include it in the final version.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details have been provided in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the guidelines in the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no direct societal impact of the work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no risk of misuse

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the packages (like panda-gym) used and have also included the licenses in the Appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets are being introduced in the paper. Existing open source framework panda-gym is being used for data generation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We have not used LLMs in our research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.