# CANONICAL OBSERVATION PRETRAINING FOR VISUOMOTOR CONTROL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Visuomotor policies learned from human demonstrations are brittle to camera viewpoint changes. Existing remedies each make a different assumption: novel-view synthesis requires a rendering pipeline, equivariant architectures require 3D inputs or symmetry hard-coded into the network, and view-invariant pretraining requires only multi-view RGB but does not model how a viewpoint change acts on the features. We present ***Canonical***, a self-supervised pretraining method that instead keeps viewpoint as a recoverable latent factor, modeling camera viewpoint change as a rotation in SO(3) and warping features into a shared canonical frame before a standard behavior-cloned policy that needs no viewpoint-specific changes. Pretraining consumes multi-view RGB and one rotation label per camera; at deployment an angle predictor recovers the rotation from the image itself, requiring no pose information. *Canonical* generalizes to camera viewpoints unseen by the policy, reaching $0.87$ success at pretraining-pool views and $0.78$ outside the pool on vision-only LIBERO-Goal against the strongest multi-view baseline's $0.65$ and $0.50$; on MetaWorld's main camera split it is the only tested visual representation that does not degrade a proprioceptive policy under viewpoint shift, and it remains the strongest method at viewpoints unseen by the policy on a real-world xArm robot.

## 1 INTRODUCTION

Visual imitation learning from human demonstrations has become a standard recipe for acquiring robotic manipulation skills (Chi et al., 2024; Lee et al., 2024). However, human demonstrations are expensive to collect, resulting in a greater emphasis on the visual representations used for downstream policy learning. This has motivated a large body of work on robot visual pretraining, ranging from Internet-scale pretraining (Nair et al., 2023; Xiao et al., 2022; Radosavovic et al., 2022; Majumdar et al., 2023) to in-domain self-supervised objectives that learn directly from demonstration trajectories (Cui et al., 2024; Chen et al., 2025). Despite these advances, policies trained on top of these pretrained visual features often remain brittle under distribution shifts, e.g., changes in camera viewpoint, object placement, or embodiment configuration can cause large performance drops even when the underlying task semantics are unchanged (Chen et al., 2024; Jiang et al., 2026; Xie et al., 2024). In particular, systematic evaluations show that camera viewpoint shifts can deteriorate policy success rates from above 80–90% to below 30% (Fei et al., 2026; Peri et al., 2024), a failure mode that persists across policy architectures and task suites. Visuomotor pretraining methods do not address this failure mode directly.

Prior work has tackled viewpoint robustness by expanding coverage through augmentation or novel-view synthesis (Chen et al., 2024; Tian et al., 2024), hard-coding geometric symmetries into the architecture (Wang et al., 2022b; 2024b), or factoring viewpoint out of the policy-facing representation (Pang et al., 2025; Jeong et al., 2026) (full taxonomy in Section 2). These approaches typically rely on external generative models, on known test-time pose or a symmetry group fixed in advance, or on an invariance objective, and none uses a learned rotation-conditioned operator in the policy-facing feature space.

Prior latent-dynamics, equivariant self-supervised learning (SSL), and multi-view representation-learning methods motivate preserving transformation structure rather than enforcing invariance (Cui et al., 2024; Aubret et al., 2024; Suau et al., 2023; Dangovski et al., 2022; Keller et al., 2023; Gao



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

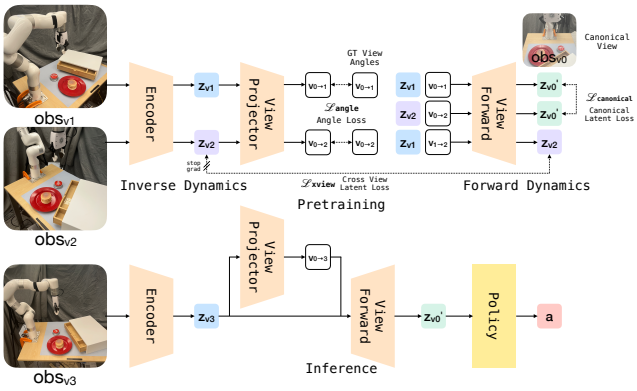



Figure 1: ***Canonical* pretraining and inference. (Top) Pretraining.** Paired views are encoded; the view predictor $g_\omega$ ("View Projector") estimates each camera's rotation $\mathbf{R}_k$ relative to the canonical view (drawn as $v_{0\to k}$), and the view forward module warps features across views and, conditioned on $\mathbf{R}_k^\top$, to the canonical frame. $\mathcal{L}_{\text{canonical}}$ aligns the two canonical predictions with each other ($\text{obs}_{v_0}$ is shown for reference, not as a target), alongside the angle loss and variance-invariance-covariance regularization (VICReg) (Section 3.2). **(Bottom) Inference.** The same modules infer the rotation of an arbitrary-viewpoint observation and warp its features to $\hat{\mathbf{z}}^{\text{can}}$ (drawn as $\mathbf{z}'_{v_0}$), the input the policy was trained on (Section 3.3).

et al., 2023). We apply this idea to camera viewpoint change by parameterizing it as an explicit SO(3) rotation that conditions a learned, single-hop latent canonicalization. Building on this, we present ***Canonical***, a self-supervised framework that maps observations in latent space from any camera viewpoint into a shared canonical feature space. At test-time, the encoder self-calibrates by inferring the camera rotation directly from observations and rotating features to the canonical view, without requiring external pose information. This enables single-viewpoint policies to generalize to camera angles unseen by the policy. Figure 1 illustrates the full pretraining and inference pipeline.

We evaluate *Canonical* on MetaWorld (Yu et al., 2020) and LIBERO-Goal (Liu et al., 2023a) benchmarks under systematic viewpoint perturbations, targeting generalization to camera angles unseen by the policy, including angles outside the encoder pretraining pool. On out-of-distribution views, *Canonical* reaches $0.78$ success on vision-only LIBERO-Goal against the strongest multi-view baseline's $0.50$; on MetaWorld it holds its success under viewpoint shift ($0.62$ to $0.64$) while ReViWo drops from $0.54$ to $0.35$ and the in-domain SSL baseline stays at $0.18$. We further validate *Canonical* on a real-world xArm robot, where it remains the strongest method at camera viewpoints unseen by the policy.

In summary, our contributions are:

1. We propose *Canonical*, a self-supervised framework that models camera viewpoint change as a latent rotation operator, making single-viewpoint policies robust to camera angles unseen by the policy without viewpoint-specific changes to the policy.
2. We show consistent viewpoint-generalization advantages over SSL and view-invariant baselines: absolute gains on LIBERO-Goal and xArm, and no degradation under viewpoint shift on MetaWorld.
3. Through ablations and representation-level probes, we show that removing the canonical warp costs $0.18$ mean success across three seeds and that removing both feature-prediction losses collapses performance to $0.30$/$0.32$; camera placement also matters more than camera count, with two well-separated pretraining views matching a four-view configuration ($0.517$), though below the twenty-view pool ($0.579$).

## 2 Related Work

**Equivariant and Geometric Robot Learning.** Incorporating geometric inductive biases through equivariant neural networks improves sample efficiency and generalization (Wang et al., 2022b;a),

extending from early $\mathrm{SE}(2)$ symmetries in spatial action spaces (Wang et al., 2021; Huang et al., 2022) to $\mathrm{SO}(3)$/$\mathrm{SE}(3)$ symmetries for 6-DoF manipulation (Wang et al., 2024b; Tie et al., 2025; Yang et al., 2024a;b), typically via specialized architectures such as steerable convolutions or spherical projections (Hu et al., 2025) operating on structured 3D inputs like point clouds or voxels (Wang et al., 2024a). Recent work such as RAVEN (Klee et al., 2026) and ISP (Hu et al., 2025) instead achieves equivariance from monocular RGB via ray-based tokenization and spherical projections, but still requires precise camera calibration or complex projection layers. In contrast, *Canonical* learns a rotation-conditioned latent transformation rather than enforcing equivariance through architectural constraints, acting on the input representation rather than the policy architecture, so it can be paired with any downstream policy, including non-equivariant ones.

**View-Invariant and Multi-View Representations.** A complementary line of work learns implicit view-invariance: ReViWo (Pang et al., 2025) decomposes observations into view-invariant and view-dependent components via a multi-view autoencoder, while VILA (Jeong et al., 2026) aligns latent action representations across viewpoints through action-guided contrastive objectives. These methods avoid hard-coded geometric constraints, but their objectives reward agreement across viewpoints rather than modeling how camera rotation acts on the latent state, and neither provides a mechanism for estimating camera rotation at inference. Augmentation-based approaches improve robustness through data diversity or explicit camera conditioning (Tian et al., 2024; Chen et al., 2024; Acar et al., 2024; Jiang et al., 2026; Yang et al., 2025; Xie et al., 2026), at the cost of rendering pipelines, multi-camera hardware, or known pose at test time (Appendix I.7 summarizes what each family of approaches requires and provides). A closely related family predicts or synthesizes one view from another as a self-supervised signal, either at the pixel level (Liu et al., 2023b) or through a rendered 3D-consistent object representation (Häni et al., 2020). *Canonical* also uses cross-view prediction, but performs it entirely in feature space and passes the resulting canonicalized latent directly to the policy, rather than synthesizing an image. Nothing needs to be rendered at test time, and no reconstruction quality bottleneck sits between the representation and control.

**Self-Supervised Dynamics Pretraining.** Our approach builds upon the DynaMo framework (Cui et al., 2024), which uses forward and inverse dynamics as a pretext task for visuomotor control. Broader pretraining methods use internet-scale or egocentric video to build general-purpose encoders (Nair et al., 2023; Xiao et al., 2022; Radosavovic et al., 2022; Majumdar et al., 2023; Tong et al., 2022; Assran et al., 2025), capturing visual features at the cost of domain specificity. Similar in-domain self-supervised methods have explored auto-encoding 3D transformations between views (Gao et al., 2023), temporal contrast across camera angles (Sermanet et al., 2018), and masked multi-view prediction (Seo et al., 2023). While DynaMo focuses on temporal transitions, it treats observations from all camera viewpoints uniformly, with no mechanism to exploit the geometric relationship between views. *Canonical* instead uses a DynaMo-style forward model to predict viewpoint changes in latent space, conditioned on relative camera rotation.

## 3 Canonical Observation Pretraining

From multi-view robot demonstration data, *Canonical* learns a feature space that is viewpoint-equivariant in a single-hop sense: a rotation-conditioned warp maps any one view to a shared canonical frame, though successive warps do not compose (Appendix I.4). *Canonical* consists of (a) a visual encoder, (b) a view predictor, which estimates the camera rotation relative to a canonical viewpoint from encoded latents, and (c) a view forward module that maps encoded latents from one viewpoint to another. The three modules are trained jointly on self-supervised objectives, namely viewpoint rotation prediction, canonical consistency, and cross-view dynamics prediction, computed between observation latents from different camera rotations, then frozen and used to canonicalize the policy's inputs at both training and deployment (Section 3.3).

### 3.1 Preliminaries

Let $\mathcal{D}_{\text{pre}} = \{(\mathbf{o}^{v_1}, \mathbf{o}^{v_2}, \mathbf{R}_1, \mathbf{R}_2)\}$ denote a pretraining dataset of synchronized two-view demonstration trajectories. Each sample consists of image observations from two cameras at rotations $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$, spanning azimuth and elevation variations. A downstream policy $\pi$ is trained on a separate dataset $\mathcal{D}_{\text{bc}} = \{(\mathbf{o}, \mathbf{a})\}$ collected from a canonical viewpoint with rotation $\mathbf{R}_0$. At evaluation, the camera is placed at an unknown rotation $\mathbf{R}_{\text{test}} \in SO(3)$, which may lie outside of

the pretraining distribution. Following prior work (Zhou et al., 2019), we represent each rotation as a 6D continuous vector $\mathbf{r}(\mathbf{R}) \in \mathbb{R}^6$ formed by concatenating the first two columns of $\mathbf{R}$; this representation is discontinuity-free and well-suited to regression with neural networks (alternative representations ablated in Appendix I.5). The visual encoder $f_\phi$ is shared across all views and timesteps, producing frame-level spatial feature maps $\mathbf{z} \in \mathbb{R}^{d \times H' \times W'}$. The view predictor and view forward module operate on these spatial maps, applying the rotation-conditioned warp across the feature grid. We describe the $SO(3)$ instantiation used on MetaWorld and xArm below; LIBERO-Goal, which varies azimuth only, uses a simpler $SO(2)$ instantiation on globally pooled features with a different canonical-loss weighting (Appendices E and I.10). $\mathbf{z}_{v_1}$ and $\mathbf{z}_{v_2}$ denote per-view spatial feature tensors.

### 3.2 Viewpoint-Equivariant Pretraining

**View prediction.** On MetaWorld, a lightweight head $g_\omega$ maps the stop-gradiented spatial feature map to a 6D output $\hat{\mathbf{r}} = g_\omega(\overline{\mathbf{z}}) \in \mathbb{R}^6$, where $\overline{(\cdot)}$ denotes stop-gradient (Chen & He, 2021; Cui et al., 2024). Gram-Schmidt orthonormalization of its two 3D column blocks then maps $\hat{\mathbf{r}}$ to a valid rotation $\hat{\mathbf{R}} \in SO(3)$ (Zhou et al., 2019). The head is supervised by the geodesic distance between the predicted and target rotations,

$$\mathcal{L}_{\text{angle}} = \frac{1}{2} \sum_{k \in \{1,2\}} \arccos\left( \frac{\operatorname{tr}(\hat{\mathbf{R}}_k^\top \mathbf{R}_k) - 1}{2} \right), \tag{1}$$

where $\mathbf{R}_k$ is the ground-truth rotation of camera $v_k$ relative to the canonical view, the same $\mathbf{R}_1, \mathbf{R}_2$ from $\mathcal{D}_{\text{pre}}$, and $\hat{\mathbf{R}}_k$ is the prediction for that camera. On LIBERO-Goal, $\mathcal{L}_{\text{angle}}$ is instead allowed to reach the encoder, which improves success there (Appendix E.3).

**View latent transformation.** The rotation projector $P_\eta : \mathbb{R}^6 \to \mathbb{R}^{d_r}$ maps a 6D rotation vector $\mathbf{r}(\Delta\mathbf{R})$ to a $d_r$-dimensional rotation latent ($d_r = 64$) through a lightweight MLP. The view forward module $h_\psi$ is a transformer that flattens the $H' \times W'$ spatial feature map into $H'W'$ position tokens, appends the rotation latent $v_{\text{rel}} = P_\eta(\mathbf{r}(\Delta\mathbf{R}))$ as an additional token, and predicts the target spatial feature map $\hat{\mathbf{z}}_{\text{tgt}} = h_\psi(\mathbf{z}_{\text{src}}, v_{\text{rel}})$, treating each frame independently.

For each batch of two-view pairs $(\mathbf{o}^{v_1}, \mathbf{o}^{v_2}, \mathbf{R}_1, \mathbf{R}_2)$, we compute encoder features $\mathbf{z}_{v_1}, \mathbf{z}_{v_2}$. We then form three predictions: the cross-view prediction conditioned on the relative rotation $\mathbf{R}_2\mathbf{R}_1^\top$, and two canonical predictions conditioned on the respective inverse absolute rotations $\mathbf{R}_k^\top$:

$$\hat{\mathbf{z}}_{v_2} = h_\psi\left(\mathbf{z}_{v_1}, P_\eta(\mathbf{r}(\mathbf{R}_2\mathbf{R}_1^\top))\right), \quad \hat{\mathbf{z}}_{v_1}^{\text{can}} = h_\psi\left(\mathbf{z}_{v_1}, P_\eta(\mathbf{r}(\mathbf{R}_1^\top))\right), \quad \hat{\mathbf{z}}_{v_2}^{\text{can}} = h_\psi\left(\mathbf{z}_{v_2}, P_\eta(\mathbf{r}(\mathbf{R}_2^\top))\right). \tag{2}$$

Here, $\hat{\mathbf{z}}_v^{\text{can}}$ denotes the predicted canonical-view features for view $v$; we drop the source-view subscript and write $\hat{\mathbf{z}}^{\text{can}}$ wherever the source view is clear from context.

The cross-view and canonical losses defined below are both built from one feature-prediction objective $\mathcal{L}_{\text{feat}}$, a mean cosine similarity between a predicted spatial feature map and a stop-gradient target:

$$\mathcal{L}_{\text{feat}}(\hat{\mathbf{z}},\, \mathbf{z}^\star) = 1 - \frac{1}{H'W'} \sum_{p=1}^{H'W'} \frac{\hat{\mathbf{z}}_p^\top\, \overline{\mathbf{z}^\star}_p}{\|\hat{\mathbf{z}}_p\|_2\, \|\overline{\mathbf{z}^\star}_p\|_2}, \tag{3}$$

where $\overline{(\cdot)}$ denotes stop-gradient and $p$ indexes spatial positions.

The *cross-view loss* applies this objective, batch-averaged over $N$ paired views, using the encoded target features as ground truth: $\mathcal{L}_{\text{xview}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{feat}}(\hat{\mathbf{z}}_{v_2,i},\, \mathbf{z}_{v_2,i})$.

The *canonical consistency loss* $\mathcal{L}_{\text{canonical}}$ applies the same objective symmetrically to both canonical predictions. The canonical camera belongs to the pretraining pool, so when a pair includes it, the cross-view loss predicts its recorded features. The canonical consistency loss, however, never uses a recorded canonical-view feature as a target: each view's canonical prediction is the pseudo-target for the other, so the canonical representation is defined by agreement between views:

$$\mathcal{L}_{\text{canonical}} = \frac{1}{2N} \sum_{i=1}^{N} \Big[ \mathcal{L}_{\text{feat}}\big(\hat{\mathbf{z}}_{v_1,i}^{\text{can}},\, \hat{\mathbf{z}}_{v_2,i}^{\text{can}}\big) + \mathcal{L}_{\text{feat}}\big(\hat{\mathbf{z}}_{v_2,i}^{\text{can}},\, \hat{\mathbf{z}}_{v_1,i}^{\text{can}}\big) \Big]. \tag{4}$$



Figure 2: **Evaluation benchmarks.** *(Left)* MetaWorld. *(Middle)* LIBERO-Goal. *(Right)* xArm real-world setup.

The two losses act as path-consistency constraints in latent space, in the spirit of cycle consistency (Zhou et al., 2016; Zhu et al., 2017; Adrian et al., 2024), except that each path is a single warp rather than a composition such as $v_1 \to v_2 \to v_0$ compared against the direct $v_1 \to v_0$. This agreement alone does not rule out convergence to a shared point decoupled from the canonical camera's actual features; Appendix H analyzes how it is anchored to the real canonical-view feature distribution. Mechanically, the canonical branch predicts features toward one fixed view, and the policy consumes its output directly, with no rendering.

**Regularization.** To prevent representation collapse, we apply the variance, invariance, and covariance terms from VICReg (Bardes et al., 2022) to three pooled feature sets on MetaWorld: the encoder features, the rotation latents, and the canonical predictions.

The full pretraining objective is

$$\mathcal{L} = \mathcal{L}_{\text{xview}} + \alpha_{\text{canonical}}\mathcal{L}_{\text{canonical}} + \alpha_{\text{angle}}\mathcal{L}_{\text{angle}} + \mathcal{L}_{\text{VICReg}}. \tag{5}$$

The canonical consistency and angle prediction losses are disabled for the first several epochs and enabled thereafter, which stabilizes training while the encoder features are relatively noisy. Appendix E gives the per-benchmark schedules and VICReg settings, including the LIBERO-Goal variant.

### 3.3 Policy Training and Inference

*Canonical* has three stages: pretraining trains the encoder $f_\phi$, view predictor $g_\omega$, rotation projector $P_\eta$, and view forward module $h_\psi$; behavior cloning keeps them frozen and trains only the policy $\pi$ on $\hat{\mathbf{z}}^{\text{can}}$; and at deployment all modules are fixed. Both in behavior cloning, on canonical-camera demonstrations, and at deployment, at an unknown rotation $\mathbf{R}_{\text{test}}$, every observation $\mathbf{o}$ passes through the same canonicalization pipeline: the encoder produces $\mathbf{z} = f_\phi(\mathbf{o})$, the view predictor recovers a rotation $\hat{\mathbf{R}} = \text{GramSchmidt}(g_\omega(\mathbf{z}))$, and the view forward module produces the canonical representation $\hat{\mathbf{z}}^{\text{can}} = h_\psi(\mathbf{z}, P_\eta(\mathbf{r}(\hat{\mathbf{R}}^\top)))$, which the policy maps to an action, $\mathbf{a} = \pi(\hat{\mathbf{z}}^{\text{can}})$. Using this predicted representation rather than the recorded canonical-view encoder feature, even during training where the ground-truth rotation is trivially known, is what lets the policy deploy unchanged at cameras it never saw: at test time it receives the same kind of input it was trained on, self-calibrating from visual input alone with no external pose sensor or camera calibration required. Per-benchmark policy architectures are given in Section 4.2; optimizers and schedules are in Appendix E.

## 4 Experiments

We evaluate whether *Canonical* generalizes to camera viewpoints unseen by the policy compared to prior SSL and multi-view baselines, whether viewpoint robustness holds across three benchmark environments using two policy architectures, and which components drive it.

### 4.1 Experimental Setup

**Benchmarks and protocol.** We evaluate on **MetaWorld** (Yu et al., 2020), a manipulation suite with fixed cameras spanning azimuth and elevation (Figure 2). Following ReViWo (Pang et al., 2025), encoders are pretrained on 17 tasks recorded from 20 fixed cameras, and downstream policies are trained on 4 representative tasks (*door-open*, *drawer-close*, *window-open*, *assembly*) from a canonical view ($\theta = 136°$) and evaluated on 6 ID and 4 OOD rollout cameras split into interpolation

and extrapolation, where ID means in the encoder's pretraining pool, OOD means outside it, and five of the six ID cameras are unseen by the policy (Appendix A). On MetaWorld, the policy additionally receives an 11-dimensional proprioceptive vector, consistent with the vision-plus-proprioception MetaWorld protocols of R3M and VC-1 (Nair et al., 2023; Majumdar et al., 2023); Section 4.4 isolates the visual contribution. We further evaluate on **LIBERO-Goal** (Liu et al., 2023a), a multi-task long-horizon suite with a distinct robot embodiment, pretraining on 20 azimuths within $\pm 20^\circ$. The vision-only policy is trained at $0^\circ$, goal-conditioned on a $0^\circ$ image, and tested at 10 cameras spanning $[-20^\circ, 25^\circ]$ (5 ID, 5 OOD), so each shifted-camera observation must be related to a $0^\circ$ goal feature (Appendix B). Finally, we validate *Canonical* on a UFACTORY **xArm** 7 robot across 3 tasks (*button press*, *pick-and-place*, *drawer close*), using VR-teleoperated demonstrations (Iyer et al., 2024), two pretraining cameras, a canonical view ($\theta = 300^\circ$), and 2 ID + 2 OOD test angles (Table 3). All three benchmarks use fixed, third-person, non-fisheye cameras. Additional camera geometry, wider-range and beyond-range azimuth sweeps, and intrinsics analysis are in Appendices C, G, M.2, and J.

### 4.2 Implementation Details

**Encoder pretraining.** The ResNet-18 backbone is trained from scratch as part of the self-supervised objective, with no ImageNet initialization; per-benchmark epoch counts are in Appendix E and checkpoint selection in Appendix F.

**Policy training.** We train a **DiTFlow** policy (a diffusion transformer (Peebles & Xie, 2023) trained with a flow-matching objective (Lipman et al., 2023)) on MetaWorld and xArm, and a **VQ-BeT** (Lee et al., 2024) policy (a vector-quantized behavior transformer) on LIBERO-Goal, testing *Canonical* with two distinct architectures.

### 4.3 Baselines

We compare *Canonical* against the following methods, all using the same policy architecture and training recipe per benchmark (feature interfaces in Appendix E.7). **Externally pretrained** (frozen features): ResNet-18 (supervised ImageNet), VC-1 (masked autoencoding on egocentric video and ImageNet) (Majumdar et al., 2023), DINOv2 (ViT-B/14, image self-distillation) (Oquab et al., 2024), and V-JEPA2 (ViT-B/16, latent prediction on internet video) (Assran et al., 2025), used at their released weights. **In-domain self-supervised**: DynaMo (Cui et al., 2024), a latent dynamics-based SSL encoder with no viewpoint-awareness, sharing *Canonical*'s architecture and training data. **Multi-view / view-invariant**: ReViWo (Pang et al., 2025) and VILA (Jeong et al., 2026) (described in Section 2), trained on our multi-view pool using the authors' released code and default hyper-parameters, with only dataset-interface adaptations and no per-benchmark search; ReViWo's frozen encoder has $\approx$ 6.6M parameters ($\approx$ 13.3M on LIBERO-Goal with its trained adapter), against $\approx$ 12.6M for *Canonical* at inference.

Table 3 uses one training seed per method, since real-robot data collection is manually intensive; its pooled counts therefore carry no seed-level uncertainty. Tables 1 and 2 use three policy-training seeds per method, with 10 trials per camera per seed (30 pooled trials per camera).

### 4.4 MetaWorld

In Table 1, *Canonical* reaches $0.62$ mean ID and $0.64$ mean OOD success across the four tasks; a pose- and count-matched single-seed OOD evaluation reaches $0.618$, numerically close to the $0.617$ three-seed ID mean (Appendix L.1). Under out-of-distribution cameras, the best multi-view baseline ReViWo falls to $0.35$ and the strongest in-domain SSL baseline DynaMo to $0.18$. *Canonical* remains flat from ID to OOD ($0.62/0.64$; $0.72$ at the policy's own training camera), whereas ReViWo drops $0.19$; this also holds across interpolation and extrapolation cameras. Across assembly, door-open, and window-open, every non-multiview baseline attains its per-camera maximum at the $136^\circ$ policy camera, whereas *Canonical* remains nonzero at all ten cameras; drawer-close is the exception. This flatness has a cost: at $136^\circ$ some baseline exceeds *Canonical* on three of the four tasks (by $0.03$–$0.33$; assembly is the exception), so *Canonical* trades peak training-camera success for robustness across views (Appendix L.2).

Table 1: Per-task success rates on MetaWorld. **ID/OOD**: as defined in Section 4.1. 30 trials per camera (3 seeds $\times$ 10 trials); mean$\pm$std across cameras. Per-camera results in Appendix L.*

| | Assembly | | Door Open | | Drawer Close | | Window Open | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **ID** | **OOD** | **ID** | **OOD** | **ID** | **OOD** | **ID** | **OOD** |
| *Pretrained Representations* | | | | | | | | |
| ResNet[7] | 0.06±0.12 | 0.01±0.01 | 0.16±0.36 | 0.00±0.00 | 0.61±0.44 | 0.53±0.40 | 0.17±0.37 | 0.02±0.03 |
| VC-1[1] | 0.02±0.04 | 0.02±0.03 | 0.14±0.31 | 0.21±0.36 | 0.46±0.46 | 0.54±0.38 | 0.15±0.25 | 0.10±0.17 |
| DINOv2[2] | 0.07±0.15 | 0.03±0.04 | 0.13±0.29 | 0.00±0.00 | 0.47±0.38 | 0.72±0.40 | 0.14±0.31 | 0.21±0.36 |
| V-JEPA2[3] | 0.03±0.06 | 0.00±0.00 | 0.10±0.22 | 0.00±0.00 | 0.84±0.16 | 0.82±0.19 | 0.07±0.15 | 0.00±0.00 |
| *In-Domain SSL* | | | | | | | | |
| DynaMo[4] | 0.03±0.06 | 0.00±0.00 | 0.10±0.22 | 0.00±0.00 | 0.48±0.48 | 0.66±0.34 | 0.09±0.17 | 0.07±0.13 |
| *Multi-View* | | | | | | | | |
| ReViWo[5] | 0.31±0.07 | 0.10±0.15 | 0.42±0.13 | 0.15±0.15 | 0.89±0.05 | 0.88±0.08 | **0.53±0.14** | 0.26±0.16 |
| VILA[6] | 0.03±0.05 | 0.00±0.00 | 0.46±0.37 | 0.22±0.17 | 0.27±0.36 | 0.02±0.03 | 0.17±0.15 | 0.13±0.19 |
| *Ours* | | | | | | | | |
| *Canonical* | **0.42±0.07** | **0.44±0.06** | **0.63±0.12** | **0.65±0.12** | **0.90±0.04** | **0.89±0.05** | 0.52±0.09 | **0.57±0.14** |

* Baseline citations (shared indices, Tables 1–3): [1] (Majumdar et al., 2023), [2] (Oquab et al., 2024), [3] (Assran et al., 2025), [4] (Cui et al., 2024), [5] (Pang et al., 2025), [6] (Jeong et al., 2026), [7] (Deng et al., 2009).

**Under viewpoint shift, baseline features pull the policy below the blind reference, whereas *Canonical*'s do not.** A proprioception-only policy (vision blanked) reaches 0.56 ID and 0.57 OOD over 3 seeds, because our MetaWorld setup carries substantial non-visual structure (fixed goal configurations for three of the four tasks, a 3D goal coordinate for the fourth, assembly); this blind policy is therefore the relevant behavioral reference, since every method in Table 1 adds visual features to it. ReViWo is at this reference ID (0.54) but well below it OOD (0.35), DynaMo is below it in both (0.18/0.18), and *Canonical* exceeds it in both; removing *Canonical*'s canonical warp instead drops success to 0.44 (Appendix I.3). On randomized-goal assembly, *Canonical* is likewise the strongest visual method and does not decline from ID to OOD ($0.42 \rightarrow 0.44$), whereas ReViWo falls from 0.31 to 0.10. Appendix I.11 gives the per-method comparison.

**Representation-level evidence for the canonical warp.** On 220,286 matched MetaWorld pairs, canonicalization raises feature similarity to the real canonical feature from 0.740 to 0.941, closing 77% of the gap. *Canonical* and VILA both retain azimuth information (circular RMSE $0.09^\circ$ and $1.76^\circ$, respectively), whereas ReViWo's view-invariant features do not ($90.8^\circ$; Figure 6). After the warp, source rotation stays MLP-decodable ($R^2 \geq 0.89$), so the warp aligns views while preserving viewpoint information. Appendix H reports the full alignment, nearest-neighbor, embedding, and reconstruction probes.

### 4.5 LIBERO-GOAL

*Canonical* reaches 0.87/0.78 ID/OOD success in Table 2, compared with 0.65/0.50 for the strongest baseline, ReViWo. Across the four OOD cameras through $20^\circ$, ReViWo holds its ID level (0.63 vs. 0.65) while *Canonical* leads by 0.07–0.40 per camera; the split-level gap widens at $25^\circ$, where ReViWo reaches 0.00 and *Canonical* 0.67 (Appendix M.1). The two LIBERO-specific objective adjustments (Appendix E.3) were selected using single-seed rollouts at the canonical camera only (the policy's training view and one of Table 2's five ID cameras); no OOD camera was consulted. The appendix reports the ablation cost of each adjustment under the full protocol. These averages are not driven by an unusually easy canonical camera: at $\theta = 0^\circ$ *Canonical* reaches only 0.77, below both its ID and OOD means, and excluding it raises the ID mean to about 0.90. Appendix M.3 retrains the policy under substituted warp rotations and separately intervenes on the warp at rollout.

### 4.6 XARM: REAL-WORLD ROBOT ENVIRONMENT

We test *Canonical* on a physical xArm setup to evaluate transfer outside simulation. At test time, a single uncalibrated RGB camera feeds the policy from one of four placements. *Canonical* leads every out-of-distribution split and in-distribution button press; ReViWo is the strongest in-distribution competitor on pick-and-place and drawer close (Table 3; task details and rollouts in Appendix D).

Table 2: Task success rates on LIBERO-Goal. **ID**: 5 pretraining-pool angles; **OOD**: 2 unsampled interior and 3 angles beyond the pool span. 30 trials per camera (3 seeds $\times$ 10 trials); mean$\pm$std across the 5 per-camera success rates within each split.

| Method | ID | OOD |
|---|---|---|
| *Pretrained* | | |
| ResNet[7] | $0.19\pm0.19$ | $0.05\pm0.03$ |
| VC-1[1] | $0.06\pm0.10$ | $0.00\pm0.00$ |
| DINOv2[2] | $0.31\pm0.28$ | $0.07\pm0.07$ |
| V-JEPA2[3] | $0.11\pm0.20$ | $0.00\pm0.00$ |
| *In-Domain SSL* | | |
| DynaMo[4] | $0.13\pm0.16$ | $0.01\pm0.03$ |
| *Multi-View / View-Invariant* | | |
| ReViWo[5] | $0.65\pm0.04$ | $0.50\pm0.27$ |
| VILA[6] | $0.40\pm0.10$ | $0.32\pm0.15$ |
| *Ours* | | |
| *Canonical* | $\mathbf{0.87\pm0.08}$ | $\mathbf{0.78\pm0.07}$ |

Table 3: Task success rates on xArm, pooled over 10 trials per camera $\times$ 2 cameras (20 per split). **ID**: the two encoder-pretraining cameras, $160^\circ$ and $300^\circ$ (the latter is also the policy's training camera); **OOD**: $280^\circ$ and $250^\circ$, outside the pretraining pool (per-camera results in Appendix Figure 11).

| | Button | | Pick-Place | | Drawer | |
|---|---|---|---|---|---|---|
| Method | ID | OOD | ID | OOD | ID | OOD |
| DynaMo[4] | $8/20$ | $1/20$ | $5/20$ | $2/20$ | $8/20$ | $7/20$ |
| ReViWo[5] | $7/20$ | $1/20$ | $\mathbf{9/20}$ | $3/20$ | $\mathbf{17/20}$ | $14/20$ |
| VILA[6] | $7/20$ | $2/20$ | $8/20$ | $2/20$ | $6/20$ | $3/20$ |
| ***Canonical*** | $\mathbf{10/20}$ | $\mathbf{8/20}$ | $8/20$ | $\mathbf{7/20}$ | $16/20$ | $\mathbf{17/20}$ |

### 4.7 Ablation Studies

We ablate *Canonical*'s pretraining objective by removing each self-supervised loss term in turn, holding the architecture, camera protocol, and policy fixed. MetaWorld success retains a substantial proprioceptive component (Appendix I.11), so Table 4 should be read as a relative comparison between variants; representation quality is assessed directly by the probes of Section 4.4.

Table 4: Objective and rotation-input ablations on MetaWorld (DiTFlow, 3 seeds). †/‡: both losses removed, warp kept/removed.

| Variant | ID | OOD |
|---|---|---|
| ***Canonical* (full)** | $\mathbf{0.62}$ | $\mathbf{0.64}$ |
| $-\mathcal{L}_{\text{xview}}$ | $0.59$ | $0.60$ |
| $-$canonical branch | $0.62$ | $0.59$ |
| $-$both$^\dagger$ | $0.30$ | $0.32$ |
| $-$both, no warp$^\ddagger$ | $0.29$ | $0.24$ |
| $\mathcal{L}_{\text{angle}}$ not detached | $0.60$ | $0.60$ |
| No warp | $0.43$ | $0.46$ |
| Identity conditioning | $0.44$ | $0.49$ |
| Fixed random rotation | $0.52$ | $0.53$ |

**Removing either predictive loss costs little; removing both collapses success.** Removing both drops success to $0.30/0.32$ even with the warp kept (Table 4), so at least one of the two losses is needed, whether to shape the features or to train the warp itself. Whether the angle loss updates the encoder is benchmark-dependent, and the cross-view stop-gradient and VICReg terms are null on MetaWorld success (Appendices E.3 and I.2). On MetaWorld, where proprioception carries much of the success, view-independent conditioning (encoder fixed, policy retrained) recovers $0.02$ of the warp's $0.18$ gain with identity, or $0.08$ with a fixed random rotation. On vision-only LIBERO-Goal the same test leaves the fixed random rotation near no warp ($0.62/0.58$ vs. $0.61/0.57$, against $0.87/0.78$ predicted, an OOD gap of $0.20$), so there the gain depends on the view-dependent rotation (Appendices I.3 and M.3).

**Camera placement matters more than camera count.** Pretraining on a single well-separated camera pair, [136,324], reaches $0.517$ mean success under a 3-seed mixed-view evaluation, matching a 4-camera configuration ($0.517$) but below the full 20-camera pool ($0.579$), while an adjacent pair reaches only $0.467$ and a clustered 3-camera configuration $0.404$ (Appendices I.8 and I.9).

**On a harder camera split, *Canonical* degrades less than ReViWo, the strongest baseline.** The main protocol's out-of-distribution cameras vary azimuth at fixed elevation and distance, so we also test six occlusion-verified cameras spanning elevation $-20^\circ$ to $-90^\circ$ and distance 0.8–3.5 m, evaluated both centered on the workspace and with each aim point translated off it by 0.22–0.48 m. On this stress test, every visual method falls below the $0.56$–$0.57$ blind reference. *Canonical* stays closest to it, reaching $0.543$ centered and $0.507$ translated (3 seeds), while ReViWo, the strongest baseline, reaches $0.268$ centered and collapses under translation ($0.063$); the remaining baselines stay below $0.16$ in both conditions. Within the centered condition, *Canonical*'s success falls from

$0.606$ on interpolation to $0.481$ on extrapolation cameras, a $21\%$ relative drop, versus $81\%$ for ReViWo across the same split. Appendices K and L.3 give the construction and full results (Table 42).

## 5 CONCLUSION

We presented *Canonical*, a self-supervised pretraining method that learns viewpoint-robust visual representations by treating camera viewpoint change as a rotation in $SO(3)$. A latent view forward predictor conditioned on relative rotation, a canonical-consistency objective, and an angle prediction head let *Canonical* deploy at camera rotations unseen by the policy without test-time calibration or viewpoint-specific policy changes. Results on MetaWorld, LIBERO-Goal, and a real xArm robot support modeling viewpoint change as a latent rotation, rather than enforcing invariance to it, as an inductive bias for generalizing visuomotor policies to camera viewpoints unseen by the policy.

**Limitations.** Like prior multi-view methods (Pang et al., 2025; Jeong et al., 2026), *Canonical* needs time-synchronized multi-view pretraining data; unlike them, it also needs one rotation label per view relative to the canonical camera, which on hardware means a multi-camera rig (or a transferred encoder) and a one-time rotation calibration rather than full extrinsics. Beyond these data requirements, *Canonical* canonicalizes only rotation and assumes fixed cameras with a known aim point, leaving translation and intrinsics to the encoder, which absorbs them at small cost (a $0.22$–$0.48$ m aim-point shift costs *Canonical* about $7\%$ success versus $77\%$ for the strongest baseline with headroom; Appendices K and J). The warp can add feature noise on high-precision tasks, and a steep canonical view lowers MetaWorld success to $0.53$ (Appendix I.6). On the evaluation side, the angle predictor's robustness to appearance change is untested, and the real-robot study uses one training seed and 20 trials per split. Finally, *Canonical* is evaluated within one internally consistent dataset per benchmark; scaling across independently defined canonical views is left to future work.

### AI USE STATEMENT

Generative AI tools were used to assist with code development and debugging, including implementing methods and scripts for collecting and reformatting simulation-replay data; with feedback on research methodology and experimental design originated by the authors; and with qualitative data analysis and results interpretation. They were also used for literature search, citation checking, drafting and editing paper text, suggesting paper structure, and identifying potential unsupported claims and anonymization issues. They were not used to develop the theoretical framework, formulate mathematical claims, propose or refine hypotheses, or originate the research methodology or experimental design; these were carried out by the authors. Use of generative AI for synthetic-data generation, for critical ingredients of mathematical proofs, for proof assistance, or for translation is not applicable to this work. All AI-assisted code, analyses, claims, references, and manuscript content were reviewed and verified by the authors, including testing AI-assisted code and data-processing scripts. The authors take responsibility for the final content of the paper, including text, claims, or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work evaluates viewpoint robustness on the simulated MetaWorld and LIBERO-Goal benchmarks and on a real xArm 7 robot in a tabletop manipulation setting. It involves no human participants or personal data, including in the multi-view data we plan to release, and we see no ethics concerns beyond the usual safety considerations for robotic manipulation research.

### REPRODUCIBILITY STATEMENT

Section 3 specifies the method. Appendix E gives per-benchmark architectures, optimizers, training schedules, and baseline implementation details, and Appendix F covers checkpoint selection. Camera geometries and evaluation splits are given in Appendices A, B, and C, and seed counts are disclosed for each benchmark and ablation family in the text or table captions. Code, model weights, and constructed datasets will be released publicly upon acceptance.

## REFERENCES

Cihan Acar, Kuluhan Binici, Alp Tekirdağ, and Yan Wu. Visual-policy learning through multi-camera view to single-camera view knowledge distillation for robot manipulation tasks. *IEEE Robotics and Automation Letters*, 9(1):691–698, 2024. doi: 10.1109/LRA.2023.3336245.

David B. Adrian, Andras Gabor Kupcsik, Markus Spies, and Heiko Neumann. Cycle-correspondence loss: Learning dense view-invariant visual features from unlabeled and unordered RGB images. *arXiv preprint arXiv:2406.12441*, 2024.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Arthur Aubret, Céline Teulière, and Jochen Triesch. Self-supervised visual learning from interactions with objects. In *European Conference on Computer Vision (ECCV)*, pp. 59–76, 2024.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. RoVi-Aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In *Conference on Robot Learning (CoRL)*, Munich, Germany, 2024.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, 2021.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19752–19763, October 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=vUrOuc6NR3.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gKLAAfiytI.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Haoran Ding, Anqing Duan, Zezhou Sun, Dezhen Song, and Yoshihiko Nakamura. Imagination at inference: Synthesizing in-hand views for robust visuomotor policy inference. *arXiv preprint arXiv:2509.15717*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026.

Xiang Gao, Wei Hu, and Guo-Jun Qi. Self-supervised multi-view learning via auto-encoding 3D transformations. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2023.

Nicolai Häni, Selim Engin, Jun-Jee Chao, and Volkan Isler. Continuous object representation networks: Novel view synthesis without target view supervision. In *NeurIPS*, 2020.

Boce Hu, Dian Wang, David Klee, Heng Tian, Xupeng Zhu, Haojie Huang, Robert Platt, and Robin Walters. 3D equivariant visuomotor policy learning via spherical projection. In *Advances in Neural Information Processing Systems*, 2025.

Haojie Huang, Dian Wang, Robin Walters, and Robert Platt. Equivariant transporter network. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022. doi: 10.15607/RSS.2022.XVIII.007.

Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. OPEN TEACH: A versatile teleoperation system for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

Youngjoon Jeong, Junha Chun, and Taesup Kim. Learning to act robustly with view-invariant latent actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R. Walter. Do you know where your camera is? View-invariant policy learning with camera conditioning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

T. Anderson Keller, Xavier Suau, and Luca Zappella. Homomorphic self-supervised learning. *Transactions on Machine Learning Research*, 2023.

David Klee, Boce Hu, Andrew Cole, Heng Tian, Dian Wang, Robert Platt, and Robin Walters. RAVEN: End-to-end equivariant robot learning with RGB cameras. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z8BN7KyaPl.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. In *International Conference on Machine Learning (ICML)*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, qiang liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023a. URL https://openreview.net/forum?id=xzEtNSuDJk.

Shaoteng Liu, Xiangyu Zhang, Tao Hu, and Jiaya Jia. Self-supervised learning by view synthesis. *arXiv preprint arXiv:2304.11330*, 2023b.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Skand Peri, Iain Lee, Chanho Kim, Li Fuxin, Tucker Hermans, and Stefan Lee. Point cloud models improve visual robustness in robotic learners. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *CoRL*, 2022.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning (ICML)*, 2023.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, 2018.

Xavier Suau, Federico Danieli, T. Anderson Keller, Arno Blaas, Chen Huang, Jason Ramapuram, Dan Busbridge, and Luca Zappella. DUET: 2D structured and approximately equivariant representations. In *International Conference on Machine Learning (ICML)*, 2023.

Stephen Tian, Blake Wulfe, Kyle Sargent, Katherine Liu, Sergey Zakharov, Vitor Guizilini, and Jiajun Wu. View-invariant policy learning via zero-shot novel view synthesis. In *Conference on Robot Learning (CoRL)*, 2024.

Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. ET-SEED: Efficient trajectory-level SE(3) equivariant diffusion policy. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://et-seed.github.io/.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. Spotlight.

Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. RISE: 3D perception makes real-world robot imitation simple and effective. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2870–2877, 2024a. doi: 10.1109/IROS58592.2024.10801678.

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant $Q$ learning in spatial action spaces. In *Conference on Robot Learning (CoRL)*, 2021. URL https://pointw.github.io/equi_q_page/.

Dian Wang, Mingxi Jia, Xupeng Zhu, Robin Walters, and Robert Platt. On-robot learning with equivariant models. In *6th Annual Conference on Robot Learning*, 2022a. URL https://openreview.net/forum?id=K8W6ObPZQyh.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022b. URL https://pointw.github.io/equi_rl_page/. Spotlight.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. In *8th Annual Conference on Robot Learning*, 2024b. URL https://openreview.net/forum?id=wD2kUVLT1g.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv:2203.06173*, 2022.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024. doi: 10.1109/ICRA57147.2024.10611331.

Yichen Xie, Yixiao Wang, Shuqi Zhao, Cheng-En Wu, Masayoshi Tomizuka, Jianwen Xie, and Hao-Shu Fang. Multi-camera view scaling for data-efficient robot imitation learning, 2026. URL https://arxiv.org/abs/2604.00557.

Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. EquiBot: SIM(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning (CoRL)*, 2024a.

Jingyun Yang, Congyue Deng, Jimmy Wu, Rika Antonova, Leonidas Guibas, and Jeannette Bohg. EquivAct: SIM(3)-equivariant visuomotor policies beyond rigid object manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9249–9255, 2024b.

Jonathan Yang, Chelsea Finn, and Dorsa Sadigh. Invariance co-training for robot visual generalization. *arXiv preprint arXiv:2512.05230*, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3D-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

# A METAWORLD CAMERA CONFIGURATION

MetaWorld encoder pretraining uses 20 fixed cameras. Azimuth increases counterclockwise (top-down view), with $0^\circ$ at the left side of the tabletop and $180^\circ$ at the right; cluster 1 cameras ($\approx 0^\circ$) therefore view the scene from the left, and cluster 2 cameras ($\approx 180^\circ$) from the right. Table 5 lists the azimuth, elevation, and distance for all 20 cameras: azimuths span the full circle and elevations vary in $[-56.3^\circ, -21.7^\circ]$, while camera distances are slightly varied in $[1.10, 1.49]$ m as an augmentation. This three-dimensional variation in camera pose is what motivates our $SO(3)$ treatment of viewpoint on MetaWorld.

The canonical view is at azimuth $136^\circ$ (marked $^*$ in Table 5). A harder stress test in Section 4.7 (Table 42) uses systematically chosen extrapolation cameras spanning a far wider elevation and distance range than the main protocol's fixed-elevation, fixed-distance out-of-distribution cameras. Table 6 lists the ten policy-evaluation cameras behind the main protocol (Table 1): six in-distribution cameras drawn from the pretraining pool, and four out-of-distribution cameras that share a fixed elevation and distance and vary only in azimuth. Pretraining covers 17 tasks: *assembly*, *basketball*, *button-press*, *coffee-button*, *coffee-push*, *dial-turn*, *door-open*, *door-unlock*, *drawer-open*, *faucet-open*, *handle-pull-side*, *pick-place*, *push*, *reach*, *soccer*, *window-close*, and *window-open*.

Table 5: MetaWorld camera azimuth, elevation, and distance (20 cameras), grouped by cluster. ($^*$) marks the canonical view (azimuth $136^\circ$). Figure 10 (top-left, "20-cam") shows the placements of these cameras, and Figure 6 shows that camera azimuth is linearly decodable from *Canonical*'s frozen features across this pool (*Canonical*, leftmost panel).

*Cluster 1* ($\approx 0^\circ$)

| **Az.** ($^\circ$) | **El.** ($^\circ$) | **Dist. (m)** |
|---|---|---|
| 8.0 | $-34.4$ | 1.11 |
| 11.0 | $-49.0$ | 1.19 |
| 17.0 | $-30.5$ | 1.37 |
| 20.0 | $-24.3$ | 1.13 |
| 28.0 | $-43.1$ | 1.11 |
| 35.0 | $-51.3$ | 1.30 |
| 324.0 | $-38.6$ | 1.49 |
| 333.0 | $-44.9$ | 1.32 |
| 339.0 | $-26.8$ | 1.35 |
| 351.0 | $-25.5$ | 1.33 |

*Cluster 2* ($\approx 180^\circ$)

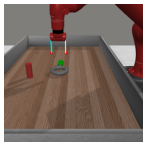
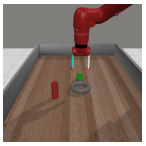
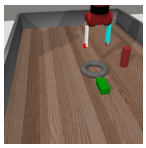
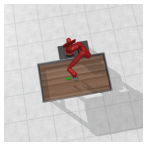
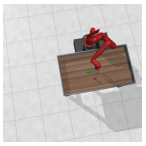
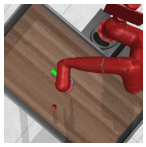
| **Az.** ($^\circ$) | **El.** ($^\circ$) | **Dist. (m)** |
|---|---|---|
| $136.0^*$ | $-25.0$ | 1.30 |
| 144.0 | $-34.0$ | 1.32 |
| 148.0 | $-51.2$ | 1.34 |
| 152.0 | $-27.6$ | 1.10 |
| 159.0 | $-27.8$ | 1.38 |
| 180.0 | $-46.4$ | 1.16 |
| 194.0 | $-21.7$ | 1.23 |
| 195.0 | $-56.3$ | 1.14 |
| 198.0 | $-26.1$ | 1.34 |
| 219.0 | $-27.7$ | 1.39 |

Table 6: MetaWorld policy evaluation cameras for the main protocol (Table 1; 30 trials per camera). The six in-distribution cameras are drawn from the pretraining pool (Table 5). The four out-of-distribution cameras share a fixed elevation ($-35^\circ$) and distance (1.25m), varying only in azimuth; they split into interpolation ($\{0^\circ, 170^\circ\}$, within the training clusters) and extrapolation ($\{125^\circ, 310^\circ\}$, in the inter-cluster gaps).

| **Split** | **Az.** ($^\circ$) | **El.** ($^\circ$) | **Dist. (m)** |
|---|---|---|---|
| ID | 8 | $-34.4$ | 1.11 |
| ID | 35 | $-51.3$ | 1.30 |
| ID | 136 | $-25.0$ | 1.30 |
| ID | 180 | $-46.4$ | 1.16 |
| ID | 219 | $-27.7$ | 1.39 |
| ID | 324 | $-38.6$ | 1.49 |
| OOD (interp) | 0 | $-35.0$ | 1.25 |
| OOD (interp) | 170 | $-35.0$ | 1.25 |
| OOD (extrap) | 125 | $-35.0$ | 1.25 |
| OOD (extrap) | 310 | $-35.0$ | 1.25 |

The stricter interpolation/extrapolation experiment in Section 4.7 (Table 42) uses six harder cameras, each evaluated with its aim point centered on the workspace and again shifted off it. Table 7 specifies all six.

Table 7: The six harder-split cameras (Section 4.7), each rendered on *assembly* at its centered aim point and again with the aim point translated (the translation test). Centered cameras aim at the default workspace point $(0,\ 0.5,\ 0)$; the shifted view applies the in-plane offset $(\Delta x,\ \Delta y)$ in the last column to that centered aim point, with a small vertical adjustment ($|\Delta z| \leq 0.15$m) so the workspace reaches the frame corners. Interpolation cameras $\{0°, 170°, 340°\}$ sit inside the pretraining azimuth clusters; extrapolation cameras $\{125°, 260°, 310°\}$ in the gaps between them. The two most extreme, $\mathrm{az}\,260°$ at 3.5m and the top-down $\mathrm{az}\,310°$, leave the workspace small and steeply foreshortened.

| Centered | Shifted | Az. (°) | El. (°) | Dist. (m) | Shifted offset $(\Delta x,\ \Delta y)$ (m) |
|---|---|---|---|---|---|
|  |  | 0 | −25 | 1.2 | $(0.15,\ 0.15)$ |
|  |  | 170 | −35 | 0.8 | $(-0.30,\ 0.30)$ |
|  |  | 340 | −45 | 1.4 | $(0.12,\ 0.20)$ |
|  |  | 125 | −20 | 2.5 | $(-0.25,\ -0.15)$ |
|  |  | 260 | −80 | 3.5 | $(0.45,\ 0.05)$ |
|  |  | 310 | −90 | 1.5 | $(-0.35,\ 0.00)$ |

## B LIBERO-GOAL CAMERA VIEWPOINT POOL

All LIBERO-Goal views share elevation $= 30°$ and distance $= 1.3$m from the scene center. The canonical viewpoint is $\theta_{\text{can}} = 0°$, defined as the front-facing view opposite the tabletop and robot arm, looking directly toward the manipulation workspace. We restrict viewpoint variation to azimuth within $\pm 20°$ because the scene contains many tabletop objects that begin to occlude the manipulation workspace as the camera moves beyond this range, limiting the meaningful span of camera motion. Given this comparatively narrow azimuth-only range, we keep the remaining camera dimensions (elevation, distance) fixed and model viewpoint as an $SO(2)$ rotation, which is sufficient in this regime and avoids the extra degrees of freedom of the full $SO(3)$ model used on MetaWorld. Table 8 lists the 20 sampled azimuths for the $\pm 20°$ pretraining pool; Table 9 lists those for the $\pm 40°$ pool.

Policies are evaluated at the ten azimuths in Table 10. The split follows pretraining coverage rather than a fixed angular range: the five ID angles are drawn from the $\pm 20°$ pool of Table 8, while the five OOD angles lie outside the pretraining pool. Two OOD angles ($\pm 10°$) interpolate between

sampled pool viewpoints, and three ($\pm 20^\circ$ and $25^\circ$) extrapolate past the pool's realized span of $[-18.72^\circ, 18.01^\circ]$.

Table 8: Azimuth angles in the $\pm 20^\circ$ pretraining pool (20 views).

| Azimuth (degrees) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $-18.72$ | $-17.95$ | $-15.45$ | $-13.55$ | $-10.53$ | $-8.65$ | $-6.22$ | $-5.83$ | $-3.16$ | $-1.94$ |
| 0.00 | 3.01 | 4.05 | 6.40 | 9.30 | 11.09 | 12.44 | 15.18 | 17.62 | 18.01 |

Table 9: Azimuth angles in the $\pm 40^\circ$ pretraining pool (20 views).

| Azimuth (degrees) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $-37.44$ | $-35.90$ | $-30.90$ | $-27.11$ | $-21.05$ | $-17.29$ | $-12.43$ | $-11.65$ | $-6.31$ | 0.00 |
| 0.50 | 4.12 | 8.87 | 14.02 | 16.11 | 20.80 | 26.60 | 30.18 | 32.88 | 38.36 |

Table 10: LIBERO-Goal policy evaluation azimuths (30 trials per camera). ID angles appear in the $\pm 20^\circ$ pretraining pool (Table 8); OOD angles do not.

| Split | Azimuth (degrees) |
|---|---|
| ID (seen in pretraining) | $-15.45$, $-5.83$, 0.00, 4.05, 15.18 |
| OOD (outside pool) | $-20.00$, $-10.00$, 10.00, 20.00, 25.00 |

## C xARM CAMERA CONFIGURATION

For the real-world xArm experiments, the encoder is pretrained from two fixed in-distribution viewpoints, the policy is trained from demonstrations at the $300^\circ$ canonical viewpoint only, and the policy is evaluated at both in-distribution and out-of-distribution camera placements. Table 11 lists the azimuth and elevation of each camera; all cameras are placed at a distance of approximately 1 m from the workspace. The OOD cameras shift both azimuth and elevation relative to the two ID cameras, probing generalization beyond the encoder pretraining pool.

Table 11: xArm camera azimuth and elevation angles. Both ID cameras are used for encoder pretraining; only the $300^\circ$ canonical camera is used for policy training. OOD cameras are used only at evaluation.

| Split | Azimuth ($^\circ$) | Elevation ($^\circ$) |
|---|---|---|
| ID | 160 | 50 |
| ID | 300 | 35 |
| OOD | 280 | 37 |
| OOD | 250 | 32 |

## D xARM TASK DESCRIPTIONS AND RESULT ANALYSES

We evaluate three real-world manipulation tasks of differing precision requirements and contact tolerance. For each task, we describe the object and success criterion, then summarize how success rate varies with camera viewpoint. Figure 3 shows representative rollouts for each task and method.

**Demonstrations and object placement.** We collect 80 demonstrations for *drawer close*, 80 for *pick-and-place*, and 40 for *button press*, recorded by VR-based teleoperation (Iyer et al., 2024) at 30 Hz from three calibrated cameras and resampled to 10 Hz via cubic spline interpolation to match the robot control frequency (only recordings from the two ID cameras of Table 11 are used; the third camera is excluded from training and evaluation). Episodes are typically $\approx 13$ s, $\approx 16$ s, and $\approx 10$ s long for the three tasks, respectively. The arm always starts from the same fixed home pose ($< 1$ mm variation across demos); diversity in the training data therefore comes from per-demo

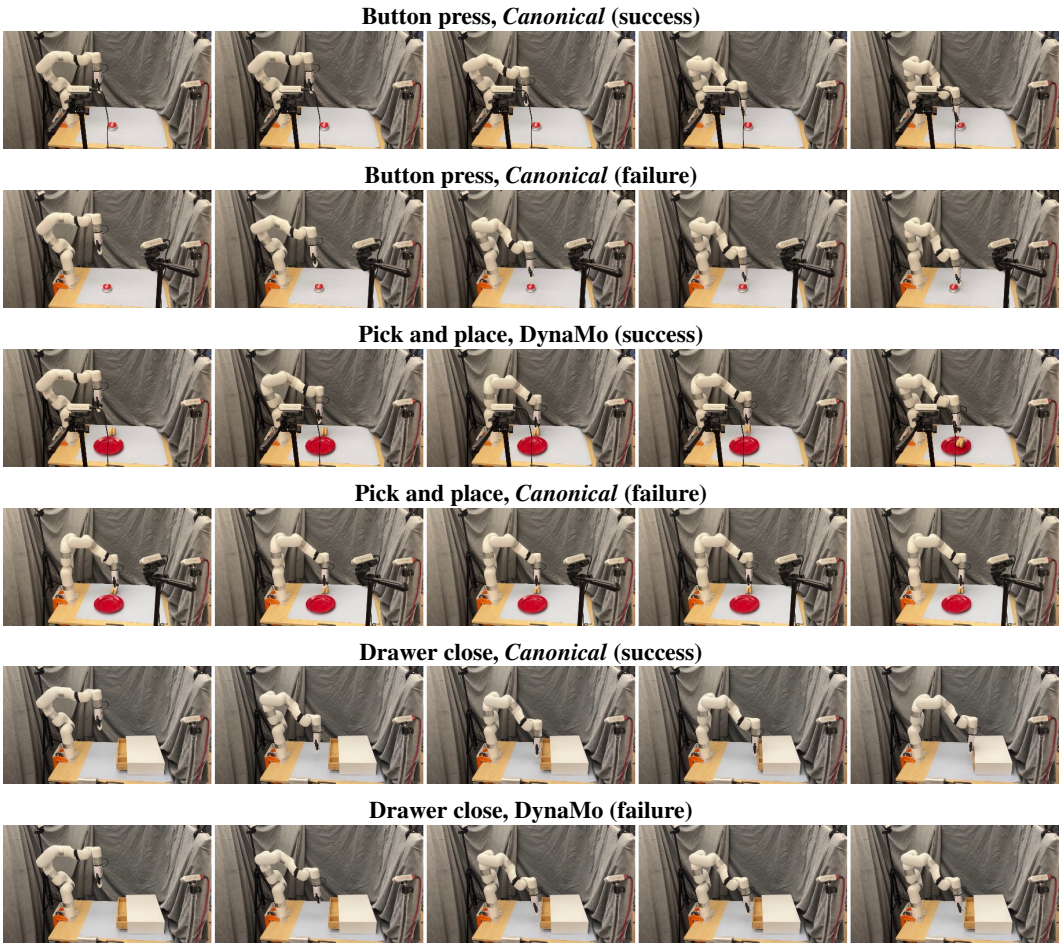



Figure 3: Representative xArm rollouts, with five frames sampled evenly from the first to the last frame of each episode. **Button press** (*Canonical* success vs. failure) illustrates the precision the task demands: the curved button surface can deflect off-center contacts, causing the gripper to slip and produce occasional failures. **Pick and place** (DynaMo success vs. *Canonical* failure) illustrates a representative failure mode on the deformable pancake, where shape mismatch and slippage can prevent a secure grasp. **Drawer close** (*Canonical* success vs. DynaMo failure) illustrates *Canonical*'s advantage at a challenging out-of-distribution camera.

object placement rather than arm initialization. For each task, the target object is placed within an in-distribution zone of roughly $12 \times 14$ cm (drawer handle), $10 \times 9$ cm (toy to be grasped in pick-and-place), and $10 \times 12$ cm (button) in the table plane. Real-robot rollouts in Table 3 are run on a structured grid of $\approx 5$ positions per task drawn from the edges of these envelopes, with the same positions reused across methods to control for placement noise.

**Button press.** The target is a toy button of 7.6 cm diameter. Successful actuation requires contacting the button near its center; off-center contact along the side causes the button to slip rather than depress, so the task demands moderate spatial precision. The contrasting *Canonical* rollouts in Figure 3 make this sensitivity concrete: the same policy succeeds when the gripper lands on center but slips off the curved surface otherwise. Every baseline's success falls sharply from the in-distribution to the out-of-distribution cameras, to at most 2 of 10 trials; *Canonical* is the only method that does not degrade, remaining at 4–6 of 10 trials across all four cameras (Figure 11).

**Pick-and-place.** The object is a deformable toy soufflé pancake of 7.6 cm diameter that must be grasped and placed onto a 25.4 cm plate. Its deformability complicates grasping, as shape mismatch and slippage frequently prevent a secure grip. *Canonical* has the highest pooled success of any

method on this task and leads at the out-of-distribution cameras (Table 3), although pick-and-place is its lowest-success xArm task; Figure 3 shows a representative failure on this object.

**Drawer close.** The task requires pushing a wooden box into an open drawer of dimensions $40.6 \times 25.4 \times 7.6$ cm. The large flat contact surface of the drawer front affords a high tolerance to imprecise contact, making the task relatively forgiving for a well-canonicalized policy. *Canonical* achieves the highest drawer-close success at the out-of-distribution cameras of any method (Table 3), with ReViWo the strongest in-distribution competitor.

# E Implementation Details

## E.1 MetaWorld: Encoder Pretraining

Tables 12–14 list the backbone architecture, optimizer/dataset (Table 13), and loss coefficients.

Table 12: MetaWorld encoder architecture.

| **Component** | **Value** |
|---|---|
| Backbone | ResNet-18 |
| Image size | $224 \times 224$ |
| Encoder output dim ($d$) | 512 |
| Spatial feature map ($H' \times W'$) | $7 \times 7$ |
| View projector (in $\to$ hid $\to$ out) | $6 \to 64 \to 64$ |
| View SSL transformer (L/H/dim) | 4 / 4 / 128 |

Table 13: MetaWorld encoder optimizer/dataset.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | AdamW |
| Learning rate | $2 \times 10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | 0.0 |
| Gradient clip | 0.1 |
| LR schedule | Constant |
| Batch size | 256 (4×RTX 6000 Ada) |
| Training epochs | 500 |
| Checkpoint used | Epoch 400 |
| Loss warmup | 10 epochs |
| Window size $T$ | 5 |
| Tasks $\times$ demos | $17 \times 15 = 255$ trajectories |
| Camera pool | 20 fixed cameras |
| Azimuth distribution | Two clusters ($\approx 0^\circ$ and $\approx 180^\circ$); $0^\circ$ is the left side of the table |
| Elevation range | $[-56.3^\circ, -21.7^\circ]$ (sampled per camera) |
| Canonical | $136^\circ$ |

## E.2 MetaWorld: Policy Training (DiTFlow)

Tables 15–17 list the policy architecture, optimizer/dataset (Table 16), and per-task training budget. MetaWorld tasks are divided into three tiers based on difficulty: Easy and Medium tasks have fixed goal configurations, while Hard tasks have randomized goals and additionally condition on a 3D proprioceptive goal (the target position).

Table 14: MetaWorld encoder loss coefficients. The canonical oversample weight upweights training pairs in which one of the two views is the canonical camera, within $\mathcal{L}_{\text{canonical}}$ only. The angle head is a detached readout (stop-gradient), so $\mathcal{L}_{\text{angle}}$ does not update the encoder.

| Coefficient | Value |
|---|---|
| $\alpha_{\text{angle}}$ | 0.5 |
| $\alpha_{\text{canonical}}$ | 0.5 |
| $\alpha_{\text{xview}}$ | 1.0 |
| Canonical oversample weight | 4.0 |
| Variance reg. coef. | 1.0 |
| Covariance reg. coef. | 0.04 |
| VICReg scope | enc. + canonical + rot. latents |

Table 15: MetaWorld DiTFlow policy architecture. Easy/Medium tasks have fixed goals; Hard tasks require a 3D proprioceptive goal (target position). obs_dim is *Canonical*'s per-token width; baselines enter at their own feature width (Table 25).

| Component | Easy/Med | Hard |
|---|---|---|
| Transformer (L/H/dim) | 8/8/256 | 8/8/256 |
| obs_dim | 512 | 512 |
| goal_dim | 0 | 3 |
| act_dim | 4 | 4 |
| Window size $T_{\text{obs}}$ | 2 | 2 |
| Action chunk size | 4 | 4 |
| Diffusion inference steps | 10 | 30 |

Table 16: MetaWorld DiTFlow policy optimizer and dataset. Easy/Medium tasks have fixed goals; Hard tasks add a 3D proprioceptive goal (target position).

| Hyperparameter | Value |
|---|---|
| Optimizer | AdamW |
| Learning rate | $1 \times 10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | $1 \times 10^{-3}$ |
| LR warmup steps | 1000 |
| Gradient clip | 1.0 |
| Batch size | 64 |
| Image augmentation | Random crop, pad $= 8$ |
| Env evals (train / final) | 10 / 10 |
| Demonstrations per task | 50 (canonical view) |
| Canonical | 136° |

Table 17: MetaWorld per-task training budget (DiTFlow). Easy/Medium tasks train for 600 epochs; Hard tasks (assembly) train for 1000. All tasks evaluate in-environment every 100 epochs. $T_{\text{mean}}$ is the mean demonstration length.

| Task | Tier | $T_{\text{mean}}$ | Epochs | Eval freq |
|---|---|---|---|---|
| door-open-v2 | Easy | 79.6 | 600 | 100 |
| window-open-v2 | Easy | 86.1 | 600 | 100 |
| drawer-close-v2 | Easy | 78.4 | 600 | 100 |
| assembly-v2 | Hard | 93.6 | 1000 | 100 |

### E.3 LIBERO-GOAL: ENCODER PRETRAINING

Tables 18, 19, and 20 list the backbone architecture, optimizer, dataset, and loss coefficients. $\mathcal{L}_{\text{canonical}}$ and $\mathcal{L}_{\text{angle}}$ are disabled for the first 5 epochs and enabled thereafter; $\mathcal{L}_{\text{xview}}$ and VICReg are active from epoch 1.

Table 18: LIBERO-Goal encoder architecture.

| **Component** | **Value** |
|---|---|
| Backbone | ResNet-18 |
| Image size | $224 \times 224$ |
| Encoder output dim ($d$) | 512 |
| Feature interface | Global average pool (no spatial map) |
| View projector (in $\rightarrow$ hid $\rightarrow$ out) | $4 \rightarrow 64 \rightarrow 64$ |
| View SSL transformer (L/H/dim) | 6 / 6 / 120 |

Table 19: LIBERO-Goal encoder optimizer and dataset.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | AdamW |
| Learning rate | $1 \times 10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | 0.0 |
| Gradient clip | 0.1 |
| LR schedule | Constant |
| Batch size | 128 (2×RTX 6000 Ada) |
| Epochs ($\pm 20°$) | 60 |
| Epochs ($\pm 40°$) | 80 |
| Loss warmup | 5 epochs |
| Window size $T$ | 5 |
| Train fraction | 0.95 |
| Tasks $\times$ demos | $10 \times 50 = 500$ |
| Camera pool | 20 cameras |
| Azimuth range | $\pm 20°$ (primary) / $\pm 40°$ (wider) |
| Elevation | $30°$ (fixed) |
| Canonical camera | $\theta = 0°$ (frontal) |

Table 20: LIBERO-Goal encoder loss coefficients. Unlike MetaWorld (Table 14), canonical pairs are not oversampled here and $\alpha_{\text{canonical}}$ is doubled (1.0 vs. 0.5); every pair contributes equally to $\mathcal{L}_{\text{canonical}}$. Also unlike MetaWorld, the angle head is not stop-gradiented (so $\mathcal{L}_{\text{angle}}$ reaches the encoder) and VICReg's variance terms are applied only to the rotation latents, while covariance regularization on the encoder features is retained; the latter two are benchmark-adaptive choices justified in Appendix E.3.

| **Coefficient** | **Value** |
|---|---|
| $\alpha_{\text{angle}}$ | 0.5 |
| $\alpha_{\text{canonical}}$ | 1.0 |
| $\alpha_{\text{xview}}$ | 1.0 |
| Canonical oversample weight | 1.0 (none) |
| Variance reg. coef. | 1.0 |
| Covariance reg. coef. | 0.04 |
| VICReg scope | rotation latents + encoder covariance |

**Benchmark-adaptive choices.** Two objective settings differ on LIBERO-Goal because all 500 of its demonstrations share a single kitchen scene. Applying VICReg's variance term to the encoder and canonical features then opposes canonical alignment, since it penalizes the batch-variance reduction

that $\mathcal{L}_{\text{canonical}}$ drives, so we restrict the variance terms to the rotation latents, keeping encoder covariance regularization; and because visual diversity is low, allowing $\mathcal{L}_{\text{angle}}$ to shape the encoder rather than detaching it (as on MetaWorld) makes the features more rotation-discriminative. Both benchmark-adaptive choices were selected using single-seed VQ-BeT policy rollouts at the canonical camera only ($\theta = 0°$), a narrower validation check than the full 10-camera protocol in Table 2. This canonical view is itself one of Table 2's five ID headline-evaluation cameras, rather than a separate validation camera. In the full 10-camera ablation, adding encoder/canonical VICReg collapses success from $0.87/0.78$ (ID/OOD) to $0.16/0.05$, while detaching the angle head lowers it to $0.73/0.63$ (a $0.14/0.15$ ID/OOD drop, respectively). The canonical-loss weight ($1.0$ vs. $0.5$) and the removal of canonical-pair oversampling were set in advance and not tuned, and no OOD camera was consulted for any of these choices. Keeping every other LIBERO-Goal setting and only widening the angle predictor to output an $SO(3)$ rotation instead of an $SO(2)$ angle reaches $0.82\pm0.10$ ID and $0.71\pm0.08$ OOD (3 policy seeds, 10 trials per camera per seed; $\pm$ is the standard deviation across the five per-camera success rates in each split), below the $SO(2)$ variant ($0.87/0.78$) but above every baseline in Table 2 (strongest, ReViWo: $0.65/0.50$).

### E.4 LIBERO-GOAL: POLICY TRAINING (VQ-BET)

Following DynaMo, no image augmentation is applied during LIBERO-Goal policy training for any method. Table 21 lists the policy architecture and optimizer.

Table 21: LIBERO-Goal VQ-BeT policy: architecture (left) and optimizer/schedule (right). Observation and goal widths are *Canonical*'s; baselines enter at their own feature width (Table 25).

| **Component** | **Value** |
|---|---|
| Transformer (L/H/dim) | 6 / 6 / 120 |
| Obs. dim / goal dim | 512 / 512 |
| Action dim | 7 |
| Obs. window size | 10 |
| Action chunk size | 1 |
| VQ-VAE codebook/groups | 16 / 2 |
| VQ-VAE fit steps | 884 |

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | AdamW |
| Learning rate | $5.5 \times 10^{-5}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | $2 \times 10^{-4}$ |
| Batch size | 64 |
| Training epochs | 50 |
| Train fraction | 0.9 |
| Eval frequency | Every epoch |
| Env evals (train / final) | 10 / 10 |

### E.5 XARM: ENCODER PRETRAINING

Tables 22 and 23 list the encoder architecture and optimizer. Training follows the MetaWorld encoder procedure above at a constant learning rate, for 400 epochs. A single encoder is pretrained jointly over the three tasks and shared by all three task policies, and the final checkpoint is used.

Table 22: xArm encoder architecture.

| **Component** | **Value** |
|---|---|
| Backbone | ResNet-18 |
| Image size | $224 \times 224$ |
| Encoder output dim ($d$) | 512 |
| Spatial feature map ($H' \times W'$) | $7 \times 7$ |
| View projector (in $\to$ hid $\to$ out) | $6 \to 64 \to 64$ |
| View SSL transformer (L/H/dim) | 4 / 4 / 128 |

### E.6 XARM: POLICY TRAINING (DITFLOW)

Table 24 lists the DiTFlow policy optimizer and training settings for xArm.

Table 23: xArm encoder optimizer and dataset.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | AdamW |
| Learning rate | $2 \times 10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | 0.0 |
| Gradient clip | 0.1 |
| LR schedule | Constant |
| Batch size | 256 (4×RTX 6000 Ada) |
| Workers per GPU | 12 |
| Training epochs | 400 |
| Checkpoint used | Epoch 400 (final) |
| Loss warmup | 10 epochs |
| Window size $T$ | 5 |
| Tasks | *drawer close*, *pick-and-place*, *button press* |
| Camera pool | 2 cameras (canonical = 300°) |

Table 24: xArm DiTFlow policy optimizer and dataset.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | AdamW |
| Learning rate | $2 \times 10^{-4}$ |
| $(\beta_1, \beta_2)$ | $(0.9,\ 0.999)$ |
| Weight decay | $1 \times 10^{-3}$ |
| LR warmup steps | 250 |
| Gradient clip | 1.0 |
| Batch size | 256 |
| Image augmentation | Random crop, pad = 16 |
| Demonstrations per task | 40–80 (see Appendix D) |
| Canonical | 300° |

### E.7 Feature Interface of Each Method

Table 25 lists the feature each method passes to the policy. For the externally pretrained encoders we use each model's standard image-level output: global average pooling (GAP) for ResNet-18, the CLS token for VC-1 and DINOv2, and the mean of the patch tokens for V-JEPA2, which receives $384\times384$ inputs. Every feature enters the policy at its own width through the policy's input linear layer, which is trained with the rest of the policy; the 512-dimensional observation width in Tables 15 and 21 is *Canonical*'s. On MetaWorld, a flat feature forms one observation token per timestep, concatenated with the 11-dimensional proprioceptive state. Spatial features (ReViWo and *Canonical*) use a spatial variant of the same DiTFlow, in which a shared linear layer projects each token, a learned 2D position embedding is added, and the proprioceptive state enters as a separate token. Across methods, only the input layer and, for spatial features, the position embedding depend on the feature shape. On LIBERO-Goal, the observation and goal images pass through the same encoder, and the two features are concatenated before VQ-BeT's input layer.

ReViWo on LIBERO-Goal is the one case with an adapter between encoder and policy. Its 196 post-quantization tokens (64-D each) are flattened to 12544 dimensions and mapped to 512 by a two-layer MLP ($12544 \to 512 \to 512$, with LayerNorm and ReLU; 6.7M parameters), applied to both observation and goal features. The MLP is trained jointly with the policy at the policy's learning rate and weight decay, and the ReViWo encoder stays frozen. It is the only trainable module outside the policy on MetaWorld and LIBERO-Goal. On MetaWorld, ReViWo's tokens enter the spatial policy directly.

Table 25: Baseline feature interface: the feature each frozen encoder passes to the policy, and its width. Each feature enters the policy's own input layer at this width. †Trained MLP adapter (see text).

| **Method** | **MetaWorld (DiTFlow)** | **LIBERO-Goal (VQ-BeT)** |
|---|---|---|
| ResNet-18 (ImageNet) | GAP, 512 | GAP, 512 |
| VC-1 | CLS token, 768 | CLS token, 768 |
| DINOv2 | CLS token, 768 | CLS token, 768 |
| V-JEPA2 | mean patch tokens, 768 | mean patch tokens, 768 |
| DynaMo | GAP, 512 | GAP, 512 |
| ReViWo | $196\times64$ tokens | 12544 flat $\rightarrow$ MLP $\rightarrow$ 512† |
| VILA | GAP, 512 | GAP, 512 |
| *Canonical* | $49\times512$ tokens, warped | GAP, 512, warped |

Table 26: Encoder quality on the $\pm20^\circ$ pretraining pool. *Canonical* and DynaMo are taken at epoch 40; ReViWo and VILA use their final checkpoints trained on our pool with the authors' released code. Probe $R^2$: coefficient of determination from a 2-layer MLP predicting robot proprioceptive state from frozen features. **ID**: probe trained and tested at the same in-pool viewpoint; **OOD**: the same probe evaluated on features from the OOD view at $25^\circ$ (for *Canonical*, the self-calibrated warped features the policy consumes). Only *Canonical* stays positive on all three targets out of distribution; DynaMo and ReViWo fall below chance on all three, while VILA falls below chance only on gripper state. Effective rank: $\exp(H)$ where $H$ is the entropy of the normalized eigenvalue spectrum of the feature covariance. Avg NN dist: mean distance to the nearest neighbor in each method's feature space; values are descriptive because feature scale and dimension differ across methods. Feature dimension: DynaMo, *Canonical*, VILA 512; ReViWo 64 (mean-pooled VQ tokens; its policy-facing adapter maps the 12544-D flattened tokens to 512-D, Appendix E.7). Effective rank is bounded by feature dimension.

| | **ID (same view), $R^2$** | | | **OOD ($25^\circ$ view), $R^2$** | | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | eef pos | eef quat | gripper | eef pos | eef quat | gripper | **Eff. rank** | **NN dist** |
| DynaMo | 0.986 | 0.964 | 0.809 | $-0.59$ | $-5.30$ | $-14.4$ | $\approx5$ | 0.003 |
| ReViWo | 0.981 | 0.969 | 0.779 | $-0.52$ | $-0.24$ | $-0.43$ | $\approx5$ | 0.073 |
| VILA | **0.995** | **0.987** | **0.912** | 0.75 | 0.27 | $-0.06$ | $\approx7$ | 0.040 |
| ***Canonical*** | 0.993 | 0.971 | 0.782 | **0.86** | **0.38** | **0.43** | $\approx64$ | 0.089 |

## F Checkpoint Selection

Encoder checkpoints are fixed in advance for each benchmark and are not selected on downstream success: epoch 400 of 500 on MetaWorld, the final epoch (400) on xArm, and epoch 40 on LIBERO, used for both *Canonical* and DynaMo. As a convergence check on LIBERO, a 2-layer MLP probe ($512 \rightarrow 256 \rightarrow$ output) predicting proprioceptive state from frozen features on a held-out split had plateaued by epoch 40 for both *Canonical* and DynaMo. For the wider $\pm40^\circ$ encoders of Appendix G, we use epoch 80 for *Canonical* and epoch 60 for DynaMo, by which point the probe had also plateaued.

## G Effect of Wider Encoder Training Range

We additionally trained encoders on a wider $\pm40^\circ$ azimuth pool using the same 20-camera configuration described in Appendix B, to test whether more pretraining coverage helps. Downstream success did not consistently improve over the $\pm20^\circ$ pool used in the main results, so we retain the $\pm20^\circ$ configuration used throughout the paper.

## H Encoder Quality Analysis

Table 26 reports encoder quality at epoch 40 for the $\pm20^\circ$ pretraining range.

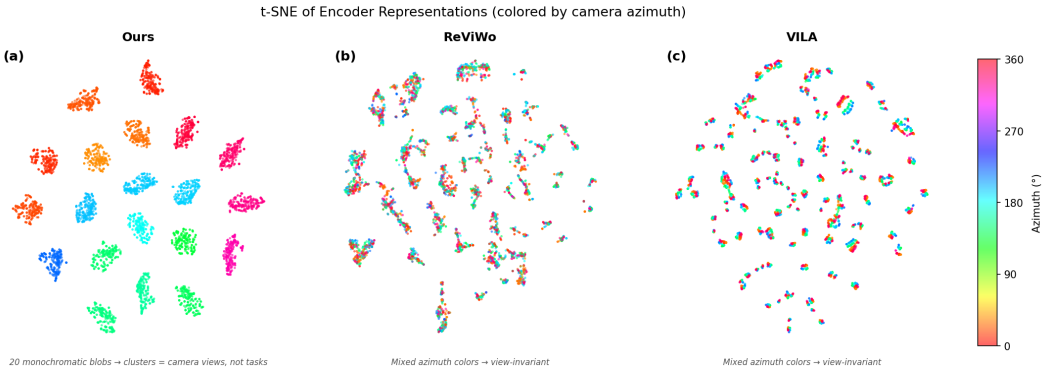



Figure 4: **t-SNE of encoder representations colored by camera azimuth** (5,100 points; 20 cameras; MetaWorld features). *Canonical*'s features form ≈20 distinct, monochromatic blobs organized by azimuth, confirming that the encoder is explicitly viewpoint-sensitive. ReViWo shows mixed-azimuth clusters, reflecting a view-invariant encoder that suppresses viewpoint information; VILA's clusters are less mixed, and its azimuth remains recoverable below, consistent with a softer invariance pressure than ReViWo's. Unlike invariance, *Canonical*'s equivariance keeps viewpoint available to the warp, enabling canonical alignment at test time without external pose sensors, though pretraining still requires a pose label per camera (Section 5). See Appendix H for the complementary task-colored t-SNE (Figure 5).

All four methods achieve high same-view end-effector-position probe $R^2$ ($\geq$ 0.98), with VILA leading across all three targets, so same-view probe $R^2$ does not separate the methods and does not measure view invariance. When the same probe is evaluated on features from the 25° OOD view (Table 26, right), DynaMo and ReViWo fall to negative $R^2$ on all three targets, far worse than predicting the mean: DynaMo drops to $-5.3$ and $-14.4$ on orientation and gripper. VILA stays positive on position and orientation (0.75, 0.27) but also falls negative on gripper ($-0.06$). *Canonical*'s raw encoder collapses there too, by design (it is equivariant, not invariant); only its self-calibrated (warped) features stay predictive, and *Canonical* with the warp is the only method positive on all three proprioceptive targets.

Figure 5 provides a complementary view of the representation structure, colored by task rather than azimuth.

DynaMo has the smallest average nearest-neighbor distance in Table 26 (0.003, versus 0.089 for *Canonical*, 0.073 for ReViWo, and 0.040 for VILA), so its features are locally concentrated in its own feature space. Its negative OOD probe values in the same table show that a probe trained on the ID view does not transfer to the OOD view. Because nearest-neighbor distance depends on feature scale and dimension, we report it descriptively and do not use it to rank representation quality.

**Linear decodability of azimuth.** To quantify how explicitly each encoder represents viewpoint, we fit linear regressors predicting $\cos(\theta)$ and $\sin(\theta)$ of camera azimuth $\theta$ from frozen encoder features, then plot each sample at its predicted $(\cos, \sin)$ location colored by ground-truth azimuth. A representation that linearly exposes camera azimuth would map observations to a rainbow-ordered unit circle. We report circular RMSE, which accounts for 360° wraparound. Figure 6 shows the result. *Canonical*'s features retain azimuth with circular RMSE 0.09°. VILA also retains azimuth (circular RMSE 1.76°), whereas ReViWo collapses to a central blob (circular RMSE 90.8°), consistent with a view-invariant encoder that explicitly suppresses azimuth. VILA's policy-facing features also decode to the source camera rather than to the canonical view ($\Delta\text{SSIM} < 0$ on all three benchmarks, Figure 7 and the decoder probe below), so it retains viewpoint without mapping views to a shared frame. The self-calibrating angle predictor relies on this decodability at inference.

**The canonical warp anchors features to a shared frame while retaining scene content.** The analyses above show that *Canonical*'s encoder preserves viewpoint; the canonical warp is what aligns this equivariant representation onto a shared canonical frame at inference. On MetaWorld, we probe both stages of a single SO(3) *Canonical* model, comparing the raw encoder features against

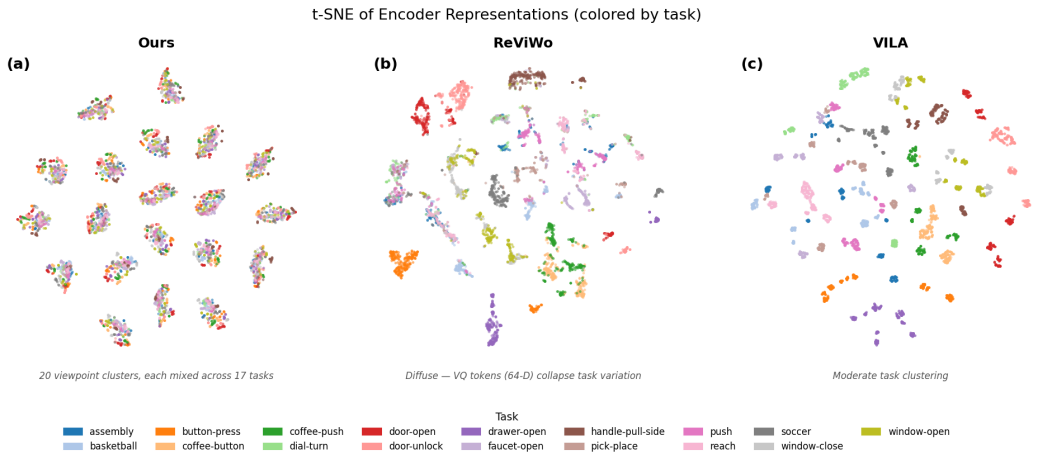



Figure 5: **t-SNE of encoder representations colored by task** (5,100 points; 17 tasks; 20 cameras; same features as Figure 4). *Canonical*'s features form 20 tight clusters organized by camera azimuth; within each cluster, multiple tasks are mixed, showing that task content and viewpoint are factored: the same task at different viewpoints falls in different azimuth clusters, while the same viewpoint groups tasks together. This structure is what lets the canonical warp align different views onto a shared frame while preserving task-relevant content. ReViWo's VQ bottleneck discards both task and viewpoint structure; VILA achieves moderate task clustering while keeping azimuth decodable (below), so the two are less separated than in *Canonical*'s encoder.

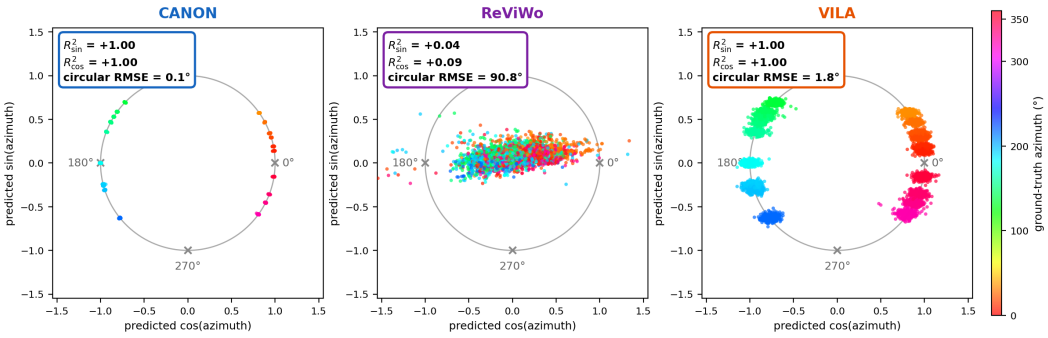



Figure 6: **Linear decodability of camera azimuth from frozen features.** Linear regressors predict $\cos(\theta)$ and $\sin(\theta)$ from each encoder's embeddings; each point is placed at its predicted $(\cos, \sin)$ location and colored by ground-truth azimuth. A representation that linearly exposes azimuth produces a rainbow-ordered unit circle. *Canonical* (left) traces the circle near-perfectly ($R^2_{\sin} = R^2_{\cos} = 1.00$, circular RMSE $0.09°$); ReViWo (center) collapses to a central blob ($R^2_{\sin} = 0.04$, $R^2_{\cos} = 0.09$, $90.8°$), reflecting its view-invariant design; VILA (right) recovers azimuth well but with more scatter ($R^2 = 1.00$, $1.76°$).

the same features after the canonical warp. Table 27 reports two representation-level measures on frozen features: cross-view feature alignment, the mean cosine similarity between the same physical state seen from different cameras, and the linear decodability of object position, a scene variable that is absent from the 11-dimensional proprioceptive state and so can only come from vision. The warp raises cross-view alignment from $0.80$ to $0.95$, confirming that it maps different views of a state to nearly the same feature, while object position stays linearly decodable from the canonicalized features ($R^2 = 0.81$; the small decrease from the raw encoder's $0.91$ reflects the warp discarding view-specific detail).

That cross-view alignment is an average over all camera pairs. A stricter test is whether the warped feature matches the encoder feature of a real canonical-view image of the same physical state. We measure this directly on $220{,}286$ held-out matched frame pairs ($149$ held-out trajectories $\times$ $19$ source cameras, $6$ tasks), pairing every non-canonical frame with the real $136^\circ$ frame of the identical simulator state. The warp raises cosine similarity to the real canonical feature from $0.740$ to $0.941$, closing $77\%$ of the gap; split by source azimuth, it closes $84\%$ for the thirteen cameras at $144^\circ$–$351^\circ$ and $64\%$ for the six at $8^\circ$–$35^\circ$. This grouping is by azimuth range only; it does not correspond to the in-distribution and out-of-distribution splits, and the gap closed does not track angular distance from the canonical camera. One camera of the nineteen runs the other way: at azimuth $8^\circ$, $128^\circ$ from the canonical view, the warp hurts ($\Delta = -0.068$, $n = 11{,}594$), while the remaining eighteen all reach between $0.90$ and $0.96$. The same probe on LIBERO-Goal, over $12{,}842$ held-out frames perturbed to the OOD $25^\circ$ view, gives $0.9664 \rightarrow 0.9785$ against the real $0^\circ$ frame, positive in all $10$ tasks and about $60$ times the standard error, which closes $36\%$ of the remaining gap. The absolute movement is smaller than MetaWorld's, consistent with LIBERO's milder $25^\circ$ perturbation leaving the no-warp control at $0.966$; the gap-closed framing is what makes the two benchmarks comparable.

Table 27: Representation-level probes of a single SO(3) *Canonical* model on MetaWorld, before and after the canonical warp (frozen features). Cross-view alignment: mean cosine similarity between matched states from different cameras. Object-position $R^2$: linear decodability of object $xyz$, a variable absent from proprioception.

| **Features** | **Cross-view alignment** | **Object-pos $R^2$** |
|---|---|---|
| Encoder (equivariant) | 0.80 | 0.91 |
| Canonical (after warp) | 0.95 | 0.81 |

**What anchors the canonical representation, and how does this relate to view synthesis?** The canonical-consistency objective (Eq. 4) aligns two predictions with each other, not against a recorded canonical-view feature, so by itself it does not rule out convergence to a shared point unrelated to the actual canonical camera. $\mathcal{L}_{\text{xview}}$ provides the anchor: whenever a sampled pair includes the canonical camera, cross-view supervision pins the corresponding prediction to that camera's recorded feature, biasing the agreement point toward the true canonical distribution rather than an arbitrary attractor; on MetaWorld such pairs are also upweighted (Table 14). This is still feature-space cross-view prediction, and so is mechanically related to view synthesis. It differs in four ways: the target is a frozen feature, not pixels or a rendered image; the policy consumes the predicted canonical feature directly, with no reconstruction step; the consistency objective is symmetric, unlike a one-way reconstruction loss; and the representation is validated for control and task-state retention (Table 28) instead of image fidelity. We therefore view *Canonical* as latent canonicalization for control, not as an image-synthesis pipeline.

**The canonical prediction specifically identifies the canonical camera.** The alignment number above shows that the warped feature moves toward the real canonical feature, but closeness alone does not show that the canonical camera is uniquely identified among the camera pool. We test this directly by comparing each warped feature against the real encoder features from all 20 pretraining-pool cameras of the same held-out state, and asking whether the nearest neighbor is the true canonical camera. Pooled over $220{,}286$ source-camera/timestep pairs, excluding the source camera's own feature from the candidate pool, the warped feature selects the real canonical-camera feature $91.9\%$ of the time, with mean rank $1.655$ among the $19$ candidates and mean margin $+0.085$ over the nearest non-canonical camera. The unwarped feature selects the canonical camera only $0.5\%$ of the time, so this specificity is induced by the warp rather than already present in the raw encoder.

The same pattern holds for the encoder used in Appendix I.3: $92.2\%$ hit rate, mean rank $1.647$, and margin $+0.125$, versus $0.4\%$ for the unwarped feature. This identification is not uniform. One source camera, at azimuth $8^\circ$ and $128^\circ$ from the canonical camera, fails almost completely ($7.7\%$ hit rate, negative margin), while every other camera reaches $85$–$100\%$. Door-open is also consistently weaker than the other tasks ($81$–$82\%$ hit rate versus $93$–$95\%$). A state-mismatch control, comparing the warped feature against the real canonical feature from a different randomly sampled held-out state, gave indistinguishable cosine similarity from the matched-state comparison, so we treat it as inconclusive rather than evidence that the warp matches state identity.

**Anchoring helps most between widely separated views.** The measurement above uses the canonical camera as one endpoint. We also evaluate Eq. 3 in its general form, predicting one arbitrary view's features from another via $h_\psi(\mathbf{z}_{v_1}, \mathbf{R}_2\mathbf{R}_1^\top)$, on $25{,}684$ directed LIBERO-Goal pairs. Pooled over all pairs the warp raises mean cosine similarity from $0.9888$ to $0.9922$, a small margin, because most sampled pairs are near-adjacent and the unwarped features are already close to ceiling there. Stratifying by angular separation is more informative: the gap grows monotonically, and in the widest bucket ($\geq 30^\circ$, $n = 962$) the warp improves $0.9593$ to $0.9849$, roughly seven times the pooled margin.

**Canonicalized features still encode the source viewpoint.** Anchoring is not invariance. An MLP probe recovers the source camera rotation from the canonicalized feature with $R^2 = 0.991$ on LIBERO-Goal and $R^2 = 0.896$ on MetaWorld (a linear probe recovers less, $R^2 = 0.917$ and $0.078$ respectively, so the residual information is present but not linearly accessible on MetaWorld). The warp's objective is agreement between views, not removal of information, and the residual does not translate into viewpoint-dependent control: *Canonical* spans $0.67$ to $1.00$ across the ten LIBERO-Goal cameras with no collapse (Tables 45 and 46). ReViWo makes the converse point: its encoder exposes little azimuth linearly ($R^2_{\sin} = 0.04$, $R^2_{\cos} = 0.09$; Figure 6), yet it is still the strongest baseline overall and still reaches only $0.35$ at MetaWorld's out-of-distribution cameras against *Canonical*'s $0.64$, so discarding viewpoint does not by itself buy robustness to it.

**Canonicalization preserves the state information a policy needs.** *Canonical* freezes the encoder and trains the policy separately, so the encoder never sees the task. A reasonable worry is that canonicalization discards exactly the content control depends on. We test this by probing six state targets from the frozen LIBERO-Goal features, before and after the warp (Table 28): end-effector position and orientation, gripper aperture, object state, each object's position relative to the end-effector, and distance to the nearest object. The last two are contact geometry, which decides whether a grasp is possible at all, and we probe relative position as its own target because accurate absolute positions can hide relative error.

Table 28: Task-relevant information retained through the canonical warp on LIBERO-Goal. **Retained**: variance-weighted ratio of post-warp to pre-warp probe $R^2$, which weights each dimension by its variance rather than averaging dimensions of very different scales. Chance is $\approx 0$ for every target.

| **Target** | **Raw $R^2$** | **Warped $R^2$** | **Retained** |
|---|---|---|---|
| End-effector position | $0.985$ | $0.980$ | $99.5\%$ |
| End-effector orientation | $0.936$ | $0.932$ | $99.5\%$ |
| Gripper aperture | $0.573$ | $0.557$ | $97.1\%$ |
| Object state | $0.435$ | $0.334$ | $93.1\%$ |
| Object position rel. to end-effector | $0.952$ | $0.943$ | $99.2\%$ |
| Distance to nearest object | $0.939$ | $0.924$ | $98.3\%$ |

Retention runs from $93.1\%$ to $99.5\%$, and every target stays far above chance after the warp. Contact geometry is among the best preserved, at $99.2\%$ and $98.3\%$. The metric matters here: recomputing object state as a plain ratio of the two $R^2$ columns gives $77\%$ rather than $93.1\%$, because a plain ratio weights near-constant dimensions as heavily as informative ones. We report the variance-weighted figure throughout. Freezing the encoder is also standard for this class of method, including for the DynaMo baseline, so the two-stage design is not a concession specific to *Canonical*.

**What the canonicalized feature looks like.** To see what the canonical warp does, we train a small decoder on each frozen encoder and ask it to reconstruct an image from the feature the policy actually consumes. Each method decodes its own policy-facing representation, so the decoders differ in input shape by design and reconstruction quality is not comparable across methods; we compare only which viewpoint each reconstruction resembles. We quantify that as $\Delta$SSIM, the similarity of a reconstruction to the canonical-view ground truth minus its similarity to the source-camera ground truth, so a positive value means the representation decodes to the canonical view rather than to the camera it came from (Figure 7).

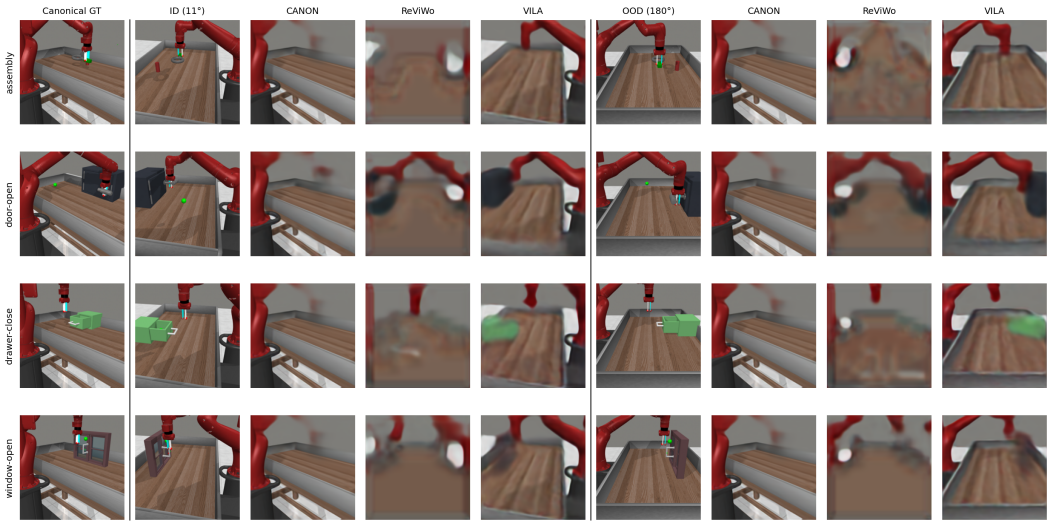



Figure 7: Decoder reconstructions from each method's policy-facing features, for two input cameras (11° and 180°, both in the encoder's pretraining pool), across four tasks (one per row). Each method decodes its own policy-facing representation, so reconstruction quality is not comparable across methods; we compare only which viewpoint each reconstruction resembles. *Canonical*'s reconstructions resemble the canonical view (leftmost column) from both input cameras, while the baselines' track the camera the observation came from.

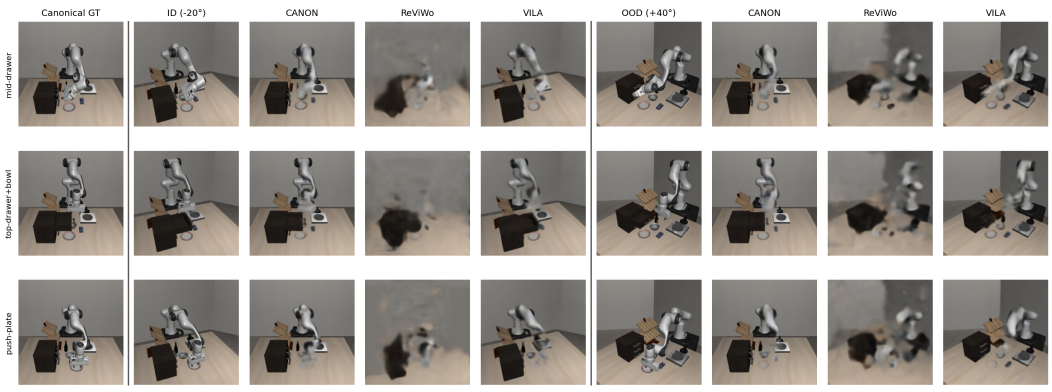



Figure 8: Decoder probe on LIBERO-Goal, for a near-pool ($-20°$) and a far out-of-distribution ($+40°$) input camera, across three tasks (one per row): open-middle-drawer, open-top-drawer-and-place-bowl, and push-plate. As on MetaWorld and xArm, *Canonical*'s reconstructions resemble the canonical view (leftmost column) from both input cameras, while ReViWo's collapse and VILA's track the source camera the observation came from.

Averaged over four tasks, *Canonical* is the only method with a positive $\Delta$SSIM: $+0.350$ from in-distribution cameras and $+0.295$ from out-of-distribution ones, against $-0.057$ and $-0.077$ for ReViWo and $-0.322$ and $-0.271$ for VILA. *Canonical*'s reconstructions from in-distribution and

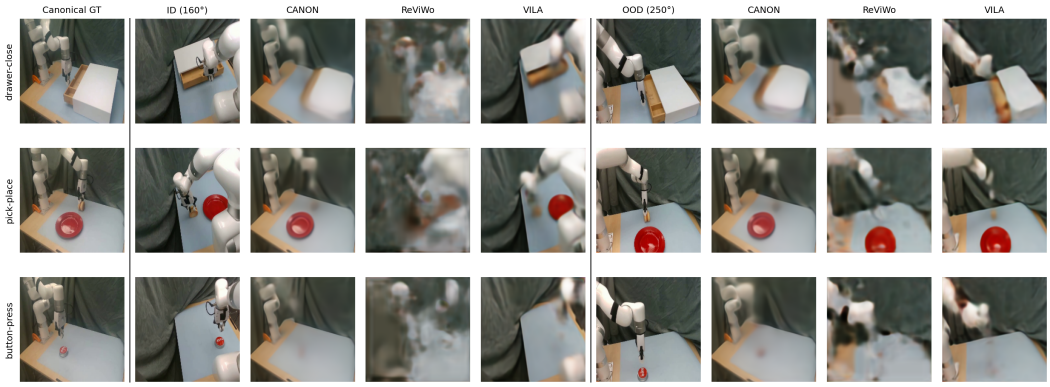



Figure 9: Real-robot replication of the decoder probe on xArm, for an in-distribution ($160°$) and an out-of-distribution ($250°$) input camera, across three tasks (one per row): drawer-close, pick-and-place, and button-press. As in simulation, *Canonical*'s reconstructions resemble the canonical view (leftmost column) from both input cameras, while ReViWo's collapse and VILA's track the source camera the observation came from.

out-of-distribution inputs also resemble each other far more than they resemble their own source frames, as intended. The effect does not depend on how the decoder was trained: repeating it with $L_1$, $L_2$ and perceptual reconstruction losses moves *Canonical*'s $\Delta$ negligibly. The same probe on LIBERO-Goal shows the same pattern (Figure 8): from a near-pool ($-20°$) and a far out-of-distribution ($+40°$) input camera, *Canonical* is again the only method positive on both camera sets ($+0.260$ near-pool and $+0.313$ far out-of-distribution); ReViWo is positive near-pool but collapses far out-of-distribution ($+0.010$ / $-0.136$), and VILA is negative on both ($-0.229$ / $-0.267$). The same probe on the real xArm reproduces the pattern across all three tasks (Figure 9): averaged over the three tasks, *Canonical* is again the only method with a positive $\Delta$SSIM ($+0.384$ from in-distribution and $+0.296$ from out-of-distribution cameras), against ReViWo ($-0.005$ / $-0.145$) and VILA ($-0.372$ / $-0.271$).

We claim from this figure only which viewpoint a representation decodes to, and nothing about reconstruction fidelity or object-level content. *Canonical* is trained entirely in the latent feature domain: its self-supervised objective predicts camera rotation from frozen features and never optimizes pixel reconstruction, whereas ReViWo's pretraining carries an explicit decoder reconstruction loss. *Canonical*'s representation was therefore never trained to preserve task-specific appearance detail, and the post-hoc decoder can only surface what the frozen feature already encodes. This evidence is qualitative and covers ten tasks across three benchmarks.

**The rotation predictor degrades smoothly outside its training range, with one reproducible exception.** Because the warp is driven by $g_\omega$'s estimate, its failure mode matters. Measured at inference on LIBERO-Goal, the mean angular error is $1.2°$ at in-distribution cameras and $11.7°$ at out-of-distribution ones, growing smoothly with distance from the pretraining pool and with no abrupt failure at any tested angle. On MetaWorld, the same geodesic-error probe gives $0.072°$ mean error on the six ID cameras, including the canonical view ($n = 720$). Three of the four OOD cameras show moderate degradation: $7.9°$ at azimuth $0°$, $14.5°$ at $125°$, and $20.3°$ at $310°$, for a mean of $14.3°$ over those cameras ($n = 360$). The remaining OOD camera, at azimuth $170°$, is a reproducible exception: its error is $174.0°$, near the maximum possible geodesic error, and an independently trained checkpoint shows the same pattern ($0.086°$ ID mean, $14.4°$ OOD mean excluding $170°$, and $169.1°$ at $170°$). We report the $170°$ camera as a reproducible outlier rather than folding it into a single four-camera OOD mean. Control success on MetaWorld does not register this error. At the $170°$ camera, *Canonical* averages $0.59$ across the four MetaWorld tasks ($0.47$–$0.80$; Appendix L.2), close to the $0.57$ blind OOD reference, despite the $174°$ rotation error supplied to the warp at that camera. The $8°$ camera makes the same point from the alignment probes: it is the one source camera where the warp lowers similarity to the real canonical feature and identifies the canonical camera only $7.7\%$ of the time (Appendix H), yet *Canonical* averages $0.62$ there ($0.40$, $0.77$, $0.93$, $0.40$ on the four tasks; Table 40), at its ID mean. This is consistent with Appendix I.3,

where view-independent conditioning recovers $0.02$–$0.08$ of the warp's $0.18$ MetaWorld benefit, and it shows that MetaWorld control success does not directly measure rotation-estimation accuracy. A separate intervention on LIBERO-Goal (Appendix M.3), which substitutes only the rollout-time rotation without retraining the policy, shows that replacing the predicted rotation with the identity costs $0.140$ overall, concentrated at the largest azimuth offsets rather than uniformly. The matched retrained LIBERO-Goal ladder (Table 48) shows a different pattern: there the view-independent variants recover little of the warp's benefit.

# I Ablation Studies

This appendix extends the self-supervised objective ablation of Section 4.7 (Table 4) along six further axes: a budget-matched view-invariant objective, the causal role of the warp's rotation input with the encoder held fixed, the rotation representation used during encoder pretraining, the choice of canonical camera, the number and placement of pretraining cameras, and the encoder's feature interface. Unless noted otherwise, all variants use the same 6 ID + 4 OOD MetaWorld camera protocol as Table 1 and the DiTFlow policy, and single-seed cells are noted explicitly.

## I.1 A Budget-Matched View-Invariant Objective

The baselines in Section 4.3 differ from *Canonical* in architecture, pretraining data and capacity as well as in objective, so they cannot isolate the contribution of equivariant canonicalization on their own. We therefore train a direct-invariance control that differs from *Canonical* in the objective alone: the same ResNet-18 encoder, the same multi-view pretraining pool, the same feature dimensionality, and the same downstream policy, training recipe and compute budget, but with the rotation-conditioned warp replaced by a view-agnostic one, so that the encoder is pushed to make different views of a state predictable from each other without any viewpoint information (Table 29). Concretely, the angle and canonical losses are removed and the view forward module receives a constant zero token $c$ in place of the rotation latent, predicting each view's features from the other:

$$\mathcal{L}_{\text{direct}} = \frac{1}{2N} \sum_{i=1}^{N} \Big[ \mathcal{L}_{\text{feat}}(h_\psi(\mathbf{z}_{v_1,i}, c),\, \mathbf{z}_{v_2,i}) + \mathcal{L}_{\text{feat}}(h_\psi(\mathbf{z}_{v_2,i}, c),\, \mathbf{z}_{v_1,i}) \Big], \tag{6}$$

with the stop-gradient on the targets as in Eq. 3. The total objective is $\mathcal{L} = \mathcal{L}_{\text{direct}} + 1.0\,\mathcal{L}_{\text{var}} + 0.04\,\mathcal{L}_{\text{cov}}$, where $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{cov}}$ are the VICReg variance and covariance terms on the encoder features (the first of the three pooled sets in Section 3.2), with the same coefficients as *Canonical* (Table 14) but applied to the encoder features only.

Table 29: Budget-matched view-invariant control on MetaWorld, evaluated on the same 10-camera protocol as Table 1 (300 trials per task). The control is a single seed (one encoder and one policy per task); the *Canonical* row is the 3-seed reference. Only the pretraining objective differs between the two rows.

| Pretraining objective | ID | OOD | Mean |
|---|---|---|---|
| Direct invariance | 0.30 | 0.31 | 0.30 |
| ***Canonical*** | **0.62** | **0.64** | **0.63** |

The control reaches $0.30$ in distribution and $0.31$ out of distribution, roughly half *Canonical*'s success in both conditions.

The result is only meaningful if the control is a working model rather than a failed training run, so we check its representation directly. At the raw encoder stage the two are indistinguishable on the rotation probe: camera rotation is recovered from both at $R^2 = 0.999$, showing that the control's raw features encode camera rotation as completely as *Canonical*'s. The difference appears at each method's own trained output, where a probe still recovers rotation from the control at $R^2 = 0.974$ (Ridge) and $0.998$ (MLP). The matched objective thus never achieved the invariance it optimizes for, even though it trained successfully by every other measure. We therefore read this as evidence that the matched objective is not a viable route to viewpoint robustness at this budget, not as evidence that invariant representations must fail in general.

### I.2 Null Objective Terms on MetaWorld

Two ablations of the pretraining objective leave MetaWorld success essentially unchanged. We report them here rather than in Table 4 because MetaWorld success retains a substantial proprioceptive component (Appendix I.11), so a null success rate does not imply a null effect on the representation.

Table 30: Objective ablations with no effect on MetaWorld success, single seed, same protocol as Table 4. *Canonical*'s reference is 0.62 ID and 0.64 OOD.

| Variant | ID | OOD |
|---|---|---|
| cross-view target stop-gradient off | 0.63 | 0.63 |
| VICReg off | 0.60 | 0.62 |

**Neither term moves MetaWorld success.** Disabling the cross-view stop-gradient leaves success unchanged (0.63 ID and OOD, against *Canonical*'s 0.62 and 0.64), and disabling VICReg alone costs 0.02 (0.60 ID, 0.62 OOD). On LIBERO-Goal, by contrast, applying VICReg to the encoder and canonical features is actively harmful, so it is restricted to the rotation latents there (Appendix E.3).

### I.3 Rotation-Input Ablation

Section 4.7 ablates *Canonical*'s pretraining objective. Here we instead isolate, over three matched seeds, the causal role of the rotation fed to the canonical warp, holding a single SO(3)-pretrained *Canonical* encoder fixed. This encoder's own predicted-rotation performance reproduces the headline MetaWorld success rates (Table 1) within seed noise, so it is a faithful proxy for the full method. The encoder, view predictor, and view forward module stay frozen at their pretrained *Canonical* weights for every row; we change only the rotation passed into the warp, at behavior-cloning training and rollout, and retrain the policy on it, so differences in Table 31 are attributable to the warp's conditioning signal alone.

Table 31: Rotation-input ablation on MetaWorld (DiTFlow policy; same 6 ID + 4 OOD camera protocol as Table 1). Each row replaces the rotation fed to the canonical warp with a different signal, holding the frozen *Canonical* encoder and all other factors fixed. Values: mean±std success rate over 3 seeds. $\Delta$: paired per-seed difference from *Canonical*.

| Rotation input | ID | OOD | Mean | $\Delta$ vs. *Canonical* |
|---|---|---|---|---|
| *Canonical* (predicted) | 0.62±0.03 | 0.64±0.04 | 0.63±0.04 | ref. |
| No warp | 0.43±0.05 | 0.46±0.02 | 0.44±0.03 | −0.18±0.02 |
| Identity | 0.44±0.05 | 0.49±0.02 | 0.46±0.04 | −0.16±0.03 |
| Random | 0.52±0.06 | 0.53±0.06 | 0.53±0.06 | −0.10±0.04 |
| Ground truth | **0.67±0.05** | **0.66±0.05** | **0.67±0.05** | +0.04±0.04 |

**Canonicalization is necessary.** Removing the warp entirely and passing raw, un-warped features to the policy costs $0.18 \pm 0.02$ overall success, showing that the warp drives most of the viewpoint robustness beyond the contribution of visual features alone.

**View-independent conditioning recovers 0.02–0.08 of the warp's 0.18 benefit.** Identity conditioning costs $0.16 \pm 0.03$, while one random rotation held fixed across all views costs $0.10 \pm 0.04$. Relative to no warp, these controls recover 0.02 and 0.08, respectively. Their separation is about 0.06–0.07, comparable to the roughly 0.05 run-to-run scale estimated below, so these data support only a range of about one tenth to one half for the view-independent share.

**The ground-truth row calibrates a roughly 0.05 run-to-run floor.** Supplying the true camera-to-canonical rotation scores $0.67 \pm 0.05$ ID and $0.66 \pm 0.05$ OOD, against $0.62 \pm 0.03$ and $0.64 \pm 0.04$ for the predicted rotation. At the ID cameras, the predicted rotation is within $0.072^\circ$ of the truth on average, so the 0.05 ID difference provides a practical estimate of run-to-run variation in policy training. At the OOD cameras, where prediction errors range from $7.9^\circ$ to $174.0^\circ$, the exact rotation adds 0.02. We treat differences of about 0.05 or less elsewhere in this appendix as unresolved at this noise scale.

### I.4 The Warp Canonicalizes a Single View Accurately but Does Not Compose

The ablation above asks which rotation the warp reads best; here we ask what the warp does with a rotation it is given. We freeze the $SO(3)$ *Canonical* encoder and its view forward module $h_\psi$ and, using the ground-truth relative rotations among the 20 pretraining-pool cameras, measure over roughly 400 (state, camera-triple) samples how faithfully a single warp maps one view onto another and whether two warps compose into one, as a true group action would.

**A single warp is an accurate geometric canonicalization.** Conditioned on the relative rotation $\mathbf{R}_{v_1 \to v_2}$, $h_\psi$ maps view $v_1$'s features onto view $v_2$'s encoder features at cosine 0.96, against a no-warp floor of 0.47; warping directly to a third view reaches that target equally well (0.96).

**Two warps do not compose.** Chaining $h_\psi(h_\psi(\mathbf{z}_{v_1}, \mathbf{R}_{v_1 \to v_2}), \mathbf{R}_{v_2 \to v_3})$ reaches the third view at only 0.59, far below the 0.96 of the equivalent direct warp $h_\psi(\mathbf{z}_{v_1}, \mathbf{R}_{v_1 \to v_3})$, and its agreement with the direct warp (0.64) is indistinguishable from an unrelated-rotation control (0.64). Thus $h_\psi$ is not a strict $SO(3)$ group action. It is a learned operator trained only on single-hop encoder-feature inputs, so feeding it its own output, an off-distribution input never seen in training, degrades the result, with latent-space errors compounding and no reconstruction anchor to correct them.

**Scope of the $SO(3)$ claim.** *Canonical* models viewpoint change as an $SO(3)$ rotation and its warp accurately canonicalizes any single view under that rotation, which is the only warp the method applies at inference. We therefore claim the $SO(3)$ parameterization and single-view canonicalization, not the stronger property of a rotation group acting compositionally on the latent. A reconstruction-anchored objective is a plausible route to closing the compositional gap and is left to future work.

### I.5 Rotation Representation

Appendix I.3 varies the rotation supplied to the warp while holding a fixed, SO(3)-pretrained encoder. Here we instead vary the rotation representation the encoder is pretrained to predict: SO(2) (azimuth only) and an SO(3) 7D variant that adds an explicit scale term to the 6D representation used throughout the paper. Table 32 reports single-seed results under the same evaluation protocol as Table 1, with *Canonical*'s own 3-seed result shown in bold for reference.

Table 32: Rotation representation used during encoder pretraining, on MetaWorld (DiTFlow policy; variants single seed). *Canonical*'s default 6D representation (bold) is the 3-seed reference from Table 31.

| Representation | ID | OOD | Mean |
|---|---|---|---|
| ***Canonical* (SO(3), 6D)** | **0.62** | **0.64** | **0.63** |
| SO(2) (azimuth only) | 0.62 | 0.55 | 0.59 |
| SO(3), 7D | 0.62 | 0.61 | 0.62 |

**SO(2) shows a larger in-distribution to out-of-distribution gap in this comparison.** The single-seed SO(2) variant matches *Canonical*'s 3-seed result on ID cameras (0.62 vs. 0.62) and reaches 0.55 OOD against *Canonical*'s 0.64. This comparison does not identify the cause of the OOD deficit. An elevation shift does not explain it: the ID cameras span elevations from $-25.0°$ to $-51.3°$ and SO(2) matches *Canonical* there, while the OOD cameras share a fixed $-35°$ elevation inside that range and vary only in azimuth (Table 6). The 7D representation, which adds an explicit scale term to the 6D encoding, reaches 0.62 overall against *Canonical*'s 0.63. Both representation comparisons are directional because the variants use one seed and the reference uses three.

**The 7D scale term adds no robustness even when camera distance varies.** The standard evaluation holds camera distance nearly fixed, so the 7D representation's extra scale degree of freedom is rarely exercised. To probe it directly, we re-evaluate the fixed 6D and 7D policies on cameras whose distance instead varies across the pretraining range (1.10 to 1.49 m) and on cameras that extrapolate beyond it (0.85 to 1.85 m), changing only the evaluation distance. Moving from fixed to varying distance changes overall success by at most 0.02 for either representation, and the 7D warp reduces this change no more than 6D does; on the extrapolated distances the two differ by 0.03, within single-seed noise. These runs are evaluation-only and single-seed, so we read the within-representation change across distances rather than absolute success levels.

### I.6 CANONICAL-FRAME CHOICE

The canonical frame is a per-dataset convention: each dataset picks its own target view, and the view predictor is trained against that choice. We use azimuth $136^\circ$ on MetaWorld, $0^\circ$ on LIBERO-Goal and $300^\circ$ on xArm. Within a dataset, any pretraining-pool camera can serve as the warp's target. Table 33 varies this choice among three azimuths (single seed each), holding everything else fixed; *Canonical* uses $136^\circ$.

Table 33: Canonical-frame choice on MetaWorld (DiTFlow policy; variants single seed). *Canonical*'s default $136^\circ$ frame (bold) is the 3-seed reference from Table 31.

| Canonical view | ID | OOD | Mean |
|---|---|---|---|
| ***Canonical*** ($136^\circ$) | **0.62** | **0.64** | **0.63** |
| $324^\circ$ | 0.62 | 0.62 | 0.62 |
| $219^\circ$ | 0.61 | 0.55 | 0.59 |
| $35^\circ$ (steep, $-51^\circ$ elevation) | 0.54 | 0.52 | 0.53 |

**The canonical-frame choice matters little except at a steep elevation.** Two alternative canonical views ($324^\circ$, $219^\circ$) perform close to *Canonical*'s default of $136^\circ$ (overall $0.62$ and $0.59$ vs. $0.63$). The one clear outlier is $35^\circ$, whose steep $-51^\circ$ elevation drops overall success to $0.53$: a canonical view near the edge of the pretraining pool's elevation range gives the warp a harder target to reconstruct. A steep canonical view should therefore be avoided.

**Which components drive out-of-distribution robustness.** *Canonical* shows no in-distribution to out-of-distribution drop ($0.62$ ID, $0.64$ OOD), whereas three ablated variants match *Canonical* on in-distribution cameras but degrade specifically on out-of-distribution ones: the azimuth-only SO(2) representation ($0.62$ to $0.55$), the alternative canonical frame at $219^\circ$ ($0.61$ to $0.55$), and removing the canonical branch (Table 4; $0.62$ to $0.59$). These ablations associate out-of-distribution robustness with the continuous SO(3) representation, canonical-frame choice, and canonical branch. The SO(2) and $219^\circ$ comparisons are single-seed and therefore directional, and the canonical-branch ID-to-OOD difference ($0.03$) is below the roughly $0.05$ run-to-run scale in Appendix I.3, so we do not interpret it.

### I.7 APPROACHES TO VIEWPOINT ROBUSTNESS

We do not run experimental comparisons against the novel-view-synthesis or equivariant-architecture families in Table 34. They assume a rendering pipeline, or 3D/point-cloud input and an architecture-level symmetry constraint, so they are not drop-in alternatives to a feature-space, RGB-only pretraining method, and a head-to-head run would compare assumptions instead of isolating the contribution evaluated here.

### I.8 CAMERA PLACEMENT AND COUNT

We ablate the number and placement of cameras used for encoder pretraining on the MetaWorld benchmark. All variants use 3 policy seeds per task and are evaluated under a mixed-view protocol with 30 trials per camera (3 seeds $\times$ 10 trials at each of the 10 rollout cameras from the evaluation pool, 6 ID + 4 OOD), and all use the canonical-warp inference pipeline with evaluation warped to the canonical view. Camera indices refer to the 20-camera pretraining pool defined in Table 5; for readability we label each variant by the azimuth angles of its cameras.

Figure 10 visualizes the camera placements for each variant, and Table 35 reports per-task success rates.

**Placement matters more than count.** A single opposing camera pair, 2-cam opp [136,324] ($0.517$), matches the 4-cam configuration ($0.517$) and reaches $89\%$ of the full 20-cam pool ($0.579$), with the remaining gap concentrated on assembly ($0.16$ vs. $0.38$) and window-open ($0.34$ vs. $0.53$). Placement dominates: the opposing pair clearly outperforms the clustered 3-cam [136,180,198] ($0.404$) and the adjacent 2-cam [35,136] ($0.467$), since diverse, widely-separated viewpoints span more of the rotation space.

Table 34: Approaches to camera-viewpoint robustness, by what each one requires and what that buys. *Canonical* occupies the remaining cell: it takes the same cheap multi-view RGB as view-invariant pretraining, adds only the camera angle, and models how viewpoint change acts on the features without a rendering pipeline, a 3D input, or a constrained policy architecture.

| Family | Requires | Strength | Limitation |
|---|---|---|---|
| Novel-view synthesis (Tian et al., 2024; Chen et al., 2024; Ding et al., 2025) | Rendering or generative pipeline | Broad viewpoint coverage | Cost and fidelity of the pipeline |
| Equivariant architectures (Wang et al., 2024b; Tie et al., 2025; Hu et al., 2025) | 3D point clouds, or symmetry hard-coded into the network (steerable convolutions, spherical projections) | Exact geometric symmetry | Constrains the policy architecture |
| View-invariant pretraining (Pang et al., 2025; Jeong et al., 2026; Seo et al., 2023) | Multi-view RGB only, no camera parameters | Cheapest data; task-centric features | No shared frame: viewpoint is discarded or left unmodeled in the features |
| *Canonical* (ours) | Multi-view RGB and the camera angle | Latent equivariance from RGB; policy unchanged; calibration-free at test time | Rotation only; translation and intrinsics are absorbed, not canonicalized |

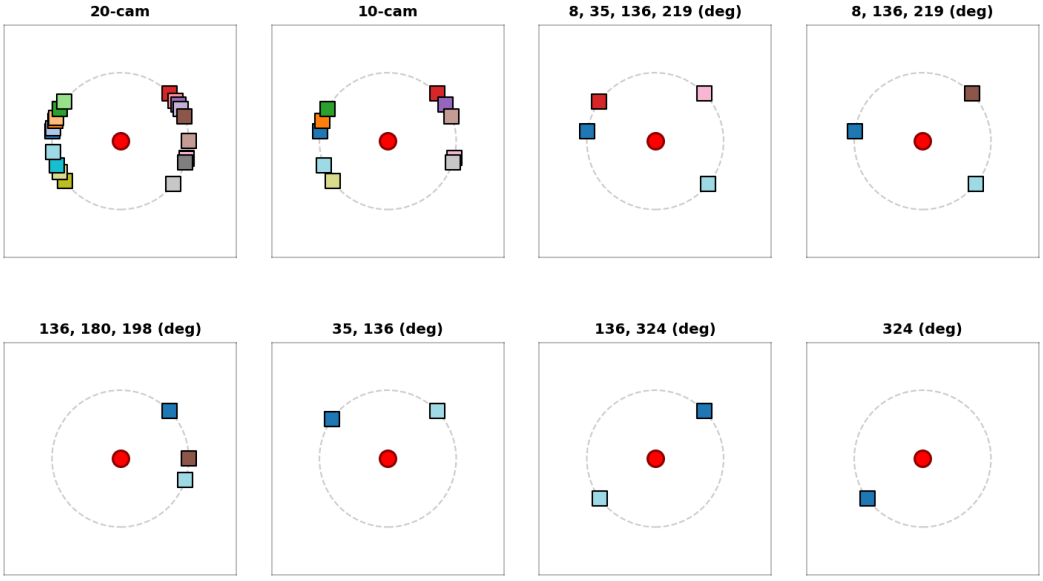



Figure 10: Camera placements for the count and placement ablations. Each panel shows the selected cameras (squares) on the viewing circle around the robot (center), titled by their azimuth angles; dense configurations are labelled by camera count.

**A single camera is insufficient.** The 1-cam [324] variant collapses to 0.412, matched only by the clustered 3-cam [136,180,198] configuration (0.404) as the two weakest results in the sweep: at least one non-trivial camera pair is required to train the view transformation that the canonical warp relies on, and a cluster of cameras without angular diversity offers no advantage over a single one.

**Door-open and assembly vary most across configurations.** Door-open ranges from 0.22 to 0.62, and assembly ranges from 0.06 to 0.38, while drawer-close remains high (0.81–0.95) across all camera configurations.

Table 35: Camera count and placement ablation on MetaWorld (DiTFlow policy, 3 seeds, mixed-view evaluation with 30 trials per camera). The 20-camera row comes from training runs separate from Table 1, so its per-task values differ from that table's values pooled over the six ID and four OOD cameras, within seed noise. Variants are labelled by the azimuth angles of the pretraining cameras. Values: mean±std across the 3 seeds. **Bold**: best mean. The mean column averages five tasks, including button-press, whose per-task column is omitted for space; it is therefore not the average of the four columns shown.

| Variant | Assembly | Door | Drawer | Window | Mean |
|---|---|---|---|---|---|
| 20-cam (full pool, default) | 0.38±0.05 | 0.62±0.07 | 0.89±0.05 | 0.53±0.04 | **0.579** |
| 10-cam (even) | 0.36±0.07 | 0.52±0.08 | 0.85±0.06 | 0.43±0.08 | 0.537 |
| 4-cam [8,35,136,219] | 0.28±0.09 | 0.53±0.08 | 0.95±0.05 | 0.31±0.09 | 0.517 |
| 3-cam [8,136,219] | 0.21±0.09 | 0.44±0.08 | 0.90±0.05 | 0.37±0.05 | 0.465 |
| 3-cam [136,180,198] | 0.07±0.04 | 0.40±0.07 | 0.81±0.06 | 0.29±0.06 | 0.404 |
| 2-cam adj [35,136] | 0.06±0.04 | 0.54±0.05 | 0.92±0.03 | 0.26±0.06 | 0.467 |
| 2-cam opp [136,324] | 0.16±0.09 | 0.61±0.07 | 0.94±0.03 | 0.34±0.07 | 0.517 |
| 1-cam [324] | 0.07±0.04 | 0.22±0.06 | 0.89±0.06 | 0.34±0.05 | 0.412 |

### I.9 Why Camera Count Has Limited Effect

The ablations above show that BC performance depends more on camera placement than on camera count once a good pair is chosen. We offer two complementary intuitions.

**The canonical warp equalizes encoder diversity at inference.** The policy never consumes raw encoder features; at inference it always applies the canonical warp, mapping features from any camera to their canonical-view equivalent before the policy acts. What matters is therefore whether the view forward module can perform this mapping accurately, not how many cameras the encoder observed during pretraining. The canonical consistency loss trains exactly this mapping, and a single non-trivial camera pair already supplies the signal needed to learn it; additional cameras yield diminishing returns. The 1-camera collapse is the converse of this: with only the angle loss active, the view forward module receives no useful gradient, the canonical warp breaks at inference, and success rate drops sharply.

**The SO(3) inductive bias may support interpolation to OOD views.** The 6D continuous rotation representation (Zhou et al., 2019), combined with the geodesic angle loss, encodes the continuous structure of SO(3) into the network. The rotation projector maps rotations continuously into the latent space, and the view forward module, conditioned on this latent, inherits the same smoothness. We therefore expect OOD camera viewpoints to be handled by interpolation on the rotation manifold rather than by lookup, which would let two well-placed training cameras cover the evaluation azimuths; we have not tested a discrete rotation code against this.

### I.10 Encoder Feature Interface

*Canonical*'s view predictor and view forward module operate on the spatial 7×7 feature map from the encoder's last convolutional block (Section 3.1). With the encoder's pretraining objective held fixed, Table 36 tests three alternative interfaces, each trained with a single seed: an earlier, higher-resolution spatial map (14×14, from an earlier block), a fusion of both blocks, and a global-average-pooled variant that removes spatial structure at the policy input, compared against *Canonical*'s own 3-seed result from Table 31 (0.63 overall).

Table 36: Encoder feature interface on MetaWorld (DiTFlow policy; variants single seed). *Canonical*'s default 7×7 layer-4 interface (bold) is the 3-seed reference from Table 31.

| Interface | ID | OOD | Mean |
|---|---|---|---|
| ***Canonical*** **(layer-4, 7×7 spatial)** | **0.62** | **0.64** | **0.63** |
| Layer-3, 14×14 | 0.53 | 0.55 | 0.54 |
| Layer-3+4 fusion, 14×14 | 0.59 | 0.56 | 0.58 |
| Layer-4, pooled (7×7 collapsed) | 0.60 | 0.61 | 0.60 |

**The spatial $7\times7$ interface is a reasonable default.** An earlier, higher-resolution layer-3 map costs $0.09$ overall, concentrated on the hardest task; fusing layer-3 and layer-4 recovers $0.04$ of this gap ($0.58$), and collapsing the layer-4 map with global average pooling before the policy costs $0.03$ ($0.60$). We keep the spatial $7\times7$ layer-4 map by default on this SO(3) instantiation, since the view predictor and view forward module operate on the spatial feature grid (Section 3.1), for which a pooled vector is insufficient during pretraining; LIBERO-Goal's simpler SO(2) instantiation does not require this spatial correspondence and uses a pooled vector instead (Appendix E.3).

### I.11 Vision Contribution

Section 4.4 notes that *Canonical* exceeds a proprioception-only policy on MetaWorld. Table 37 compares each method's frozen features, added to the same proprioceptive DiTFlow policy, against the blind (vision-blanked) reference, and also passes *Canonical*'s features without the canonical warp; the complementary vision-only (proprioception-blanked) control was run for *Canonical* alone. The blind and *Canonical* rows are 3-seed means; baseline rows are the headline three-seed Table 1 values.

Table 37: Vision contribution on MetaWorld: success of the same proprioceptive DiTFlow policy augmented with each method's frozen visual features (rows), pooled over the in-distribution and out-of-distribution camera sets. **Proprio only** blanks vision (the blind reference); **+*Canonical* (no warp)** passes *Canonical*'s features without the canonical warp. Baseline rows are the headline per-method evaluation (Table 1, mean over 3 seeds); the blind and *Canonical* rows are input-decomposition controls (mean over 3 seeds). Every baseline drops the policy below the blind reference at out-of-distribution cameras, whereas *Canonical* does not.

| Policy input | ID | OOD |
|---|---|---|
| Proprioception only (blind reference) | 0.56 | 0.57 |
| + DynaMo features | 0.18 | 0.18 |
| + ReViWo features | 0.54 | 0.35 |
| + VILA features | 0.23 | 0.09 |
| + *Canonical* features (no warp) | 0.43 | 0.46 |
| **+ *Canonical* features** | **0.62** | **0.64** |

**MetaWorld success is largely solvable from proprioception.** A blind policy already reaches $0.56$ ID and $0.57$ OOD, and *Canonical* adds a modest absolute improvement over proprioception ($0.62$ ID and $0.64$ OOD vs. $0.56$ and $0.57$); a separate single-method diagnostic that blanks proprioception for *Canonical* itself shows vision alone is weak ($0.18$ aggregate over four tasks). Vision still matters, however: Table 37 shows every baseline's features, added to the same proprioceptive policy, drop it below the blind reference at out-of-distribution cameras, whereas *Canonical*'s do not, and removing *Canonical*'s own canonical warp reproduces that failure ($0.43$ ID, $0.46$ OOD). Absolute success rate is therefore a weak instrument for visual-representation quality on this benchmark, and we assess visual content and viewpoint structure more directly with the representation-level probes of Appendix H.

## J Camera Intrinsics

*Canonical* canonicalizes rotation and assumes intrinsics are shared between pretraining and deployment. A camera with a different field of view breaks that assumption, so we measure how much it costs. This analysis varies field of view under a shared perspective projection model, matching our third-person, conventional-optics cameras (Section 4.1). Fisheye lenses are more common on mobile robots and wrist-mounted (eye-in-hand) cameras, where a wide field of view offsets navigation range or close-range mounting; static third-person setups like ours do not share that constraint, and camera models with this distortion are outside what we test. We render the evaluation cameras at fields of view from $30^\circ$ to $100^\circ$ around the $45^\circ$ used for training, and evaluate each policy twice: on the raw wide or narrow frame, and on a rectified frame geometrically resampled to the training field of view. Rectification is the standard remedy, so the question is how much each method depends on it. This sweep uses five tasks, adding *button-press* to the four in Table 1, with 120 episodes per task per cell.

Table 38: Field-of-view sweep on MetaWorld: pooled five-task success (120 episodes per task per cell), field of view as columns; each cell is the mean over the ID and OOD camera sets. **rect**: frame resampled to the $45°$ training field of view; **raw**: unmodified frame. $45°$ is the training field of view, where rectification is the identity and serves as a null control. The OOD set copies the ID elevations, distance and aim point and moves only azimuth (a mild intrinsics control, not a viewpoint-robustness benchmark). The top row previews the workspace at each field of view.



| | 30° | | 45° | | 60° | | 75° | | 100° | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | rect | raw | rect | raw | rect | raw | rect | raw | rect | raw |
| ReViWo | 0.218 | 0.306 | 0.426 | 0.416 | 0.410 | 0.270 | 0.404 | 0.138 | 0.411 | 0.114 |
| VILA | 0.200 | 0.065 | 0.259 | 0.265 | 0.249 | 0.142 | 0.253 | 0.081 | 0.237 | 0.052 |
| ***Canonical*** | **0.533** | **0.487** | **0.546** | **0.545** | **0.526** | **0.553** | **0.545** | **0.501** | **0.542** | **0.457** |

The informative quantity is how much each method needs rectification. Averaged over the three fields of view wider than training and over both camera sets, rectification is worth $+0.034$ to *Canonical*, against $+0.234$ to ReViWo and $+0.155$ to VILA, roughly seven and five times more. *Canonical*'s benefit even changes sign across fields of view, so it is not distinguishable from zero. Read the other way, without rectification *Canonical* retains most of its native-field-of-view score across these wider fields, while ReViWo and VILA retain only about $0.4$ and $0.3$ of theirs. At $45°$ rectification is an identity operation and reproduces the raw score to within $0.01$ for every method, which validates the rectification path itself. The two camera sets behave the same throughout, so an intrinsics change and a viewpoint change do not compound. Taking each method's unrectified score relative to its own native $45°$ score, the OOD and ID penalties differ only at the scale of the standard error and with no consistent sign. This is a property of the setup rather than an advantage of *Canonical*, since the baselines show it too.

**What these numbers do not show.** This experiment has three limits. First, on ID cameras at $100°$ *Canonical*'s unrectified score ($0.445$) exceeds the best rectified baseline ($0.430$) by less than one standard error, which is parity rather than a win; on OOD cameras the same comparison is $0.468$ against $0.393$, which is outside noise. Second, the OOD camera set in Table 38 shifts azimuth by only $8°$ to $34°$ and copies elevation and distance from the ID set, so its small penalties are not a measure of viewpoint robustness and should not be compared with Section 4.7's harder split. Third, on the narrow side rectification must invent nothing, so unobserved periphery is zero-padded; its effect is method-dependent in direction, and *Canonical*'s own sign is not stable across the two camera sets, so we claim nothing about the narrow side beyond that. These evaluations are single-seed.

## K Camera Translation

Section 4.7 reports that shifting a camera's aim point by $0.22$ to $0.48$ m costs *Canonical* about $7\%$ of its success (Table 42, $0.543 \rightarrow 0.507$) where ReViWo, the strongest baseline with headroom for the comparison to mean anything, loses $77\%$ ($0.268 \rightarrow 0.063$); every other baseline sits near floor in both conditions, so their small changes carry no information in either direction. This appendix gives the geometric and mechanistic detail behind that result.

*Canonical* models viewpoint as a rotation, so a camera that is physically moved rather than rotated falls outside what the warp represents. For lookat-centered tabletop cameras at roughly meter scale, aim-point shifts of $0.1$ to $0.4$ m move image content by $20$ to $77$ pixels but leave only about $1$ pixel of residual once a global homography is fitted. A low homography residual shows only that the scene is close to planar, since a camera translation induces a homography exactly when it is. It does not show that the induced image change is rotation-like or that a rotation-conditioned warp can absorb it. What the result shows remains empirical: over the range tested, moderate off-axis translation is

absorbed without collapsing the canonicalization. Close-range or wrist-mounted cameras and scenes with strong relief would produce far more parallax and are not covered by this result.

The same scale caveat applies as for the field of view (Appendix J): both numbers sit below the $0.56$–$0.57$ blind proprioceptive reference (Appendix I.11), which no camera perturbation can move, so the contrast is between degrees of interference rather than between absolute robustness levels. The view forward module has no explicit translation or depth mechanism, and we do not attribute the translation tolerance to any particular component.

We tested whether modeling translation explicitly would help. A 9D encoder conditioned on translation as well as rotation, trained on a translation-diverse dataset against a matched 6D control, scores $0.594$ against the control's $0.590$, and the warp is insensitive to the translation channel even when given ground-truth translation. This is a single behavior-cloning seed per arm, and we read the 9D channel as a self-consistent pose parameterization rather than as metric camera translation. We therefore treat explicit $SE(3)$ modeling as outside the regime tested here, and as future work for strong-parallax settings.

These results and the intrinsics sweep (Appendix J) are consistent with *Canonical*'s design: the encoder keeps camera pose available and the warp uses it to place features in a shared frame, instead of trying to make the representation invariant to viewpoint. They do not diagnose why the baselines fail. These evaluations use Table 1's three frozen policy seeds per task, with 10 trials per camera per seed.

# L METAWORLD: ADDITIONAL EVALUATIONS

## L.1 IS THE OUT-OF-DISTRIBUTION SPLIT EASIER?

The main protocol's ID cameras vary elevation and distance as well as azimuth, whereas its OOD cameras hold elevation and distance fixed (Section 4.1). The OOD set might therefore be easier by construction, which would flatter the ID-to-OOD comparison in Table 1. To test this we build a matched-diversity set of six OOD azimuths $\{0°, 45°, 125°, 170°, 235°, 310°\}$, each inheriting the exact elevation and distance of one ID camera, so the two splits have identical elevation and distance ranges by construction and differ only in azimuth (Table 39).

Table 39: Matched-diversity cameras on MetaWorld: six OOD azimuths that inherit the ID cameras' exact elevation and distance values. Pooled over four tasks, 30 trials per camera (720 episodes per method), single seed (one encoder and one policy per task).

| **Method** | **OOD** |
|---|---|
| *Pretrained* | |
| ResNet (ImageNet) | 0.133 |
| VC-1 (Majumdar et al., 2023) | 0.089 |
| DINOv2 (Oquab et al., 2024) | 0.190 |
| V-JEPA2 (Assran et al., 2025) | 0.154 |
| *In-Domain SSL* | |
| DynaMo (Cui et al., 2024) | 0.140 |
| *Multi-View / View-Invariant* | |
| ReViWo (Pang et al., 2025) | 0.375 |
| VILA (Jeong et al., 2026) | 0.083 |
| ***Canonical*** | **0.618** |

*Canonical* scores $0.618$ on this single-seed OOD set, numerically close to Table 1's $0.617$ three-seed ID mean. Camera count and pose diversity may therefore explain the apparent OOD advantage in Table 1, although the comparison is not seed-matched. On this same fairness-controlled set *Canonical* reaches $0.618$ against $0.375$ for the strongest baseline, ReViWo, and at most $0.19$ for every other method; it is the only method above $0.4$. A deliberately harder split that varies elevation and distance far more widely stresses this further (Tables 43 and 44).

| Task / Method \ Azimuth |  8° |  35° |  136° |  180° |  219° |  324° |
|---|---|---|---|---|---|---|
| *Assembly* | | | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | 0.33±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.10±0.05 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.40±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.17±0.07 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.17±0.07 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| ReViWo | 0.27±0.08 | **0.37±0.09** | 0.40±0.09 | 0.27±0.08 | 0.37±0.09 | 0.20±0.07 |
| VILA | 0.07±0.05 | 0.00±0.00 | 0.13±0.06 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| *Canonical* | **0.40±0.09** | **0.37±0.09** | **0.57±0.09** | **0.40±0.09** | **0.43±0.09** | **0.37±0.09** |
| *Door Open* | | | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | **0.97±0.03** | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| VC-1 | 0.03±0.03 | 0.00±0.00 | 0.83±0.07 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.77±0.08 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.60±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.60±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| ReViWo | 0.50±0.09 | 0.17±0.07 | 0.47±0.09 | 0.37±0.09 | 0.47±0.09 | **0.57±0.09** |
| VILA | **0.83±0.07** | 0.00±0.00 | **0.97±0.03** | 0.53±0.09 | 0.00±0.00 | 0.43±0.09 |
| *Canonical* | 0.77±0.08 | **0.67±0.09** | 0.67±0.09 | **0.63±0.09** | **0.67±0.09** | 0.37±0.09 |
| *Drawer Close* | | | | | | |
| ResNet | 0.60±0.09 | **1.00±0.00** | **1.00±0.00** | **1.00±0.00** | 0.03±0.03 | 0.00±0.00 |
| VC-1 | 0.83±0.07 | 0.00±0.00 | 0.97±0.03 | 0.97±0.03 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.43±0.09 | **1.00±0.00** | 0.43±0.09 | **0.90±0.05** | 0.07±0.05 |
| V-JEPA2 | 0.90±0.05 | 0.97±0.03 | **1.00±0.00** | 0.90±0.05 | 0.53±0.09 | 0.73±0.08 |
| DynaMo | 0.90±0.05 | 0.00±0.00 | **1.00±0.00** | 0.00±0.00 | 0.00±0.00 | **0.97±0.03** |
| ReViWo | **0.93±0.05** | 0.83±0.07 | 0.97±0.03 | 0.93±0.05 | 0.87±0.06 | 0.83±0.07 |
| VILA | 0.43±0.09 | 0.00±0.00 | **1.00±0.00** | 0.17±0.07 | 0.00±0.00 | 0.03±0.03 |
| *Canonical* | **0.93±0.05** | 0.90±0.05 | 0.97±0.03 | 0.87±0.06 | 0.83±0.07 | 0.90±0.05 |
| *Window Open* | | | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | **1.00±0.00** | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.67±0.09 | 0.00±0.00 | 0.23±0.08 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.83±0.07 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.40±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.07±0.05 | 0.00±0.00 | 0.47±0.09 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| ReViWo | **0.50±0.09** | **0.63±0.09** | 0.63±0.09 | 0.27±0.08 | 0.47±0.09 | **0.67±0.09** |
| VILA | 0.00±0.00 | 0.17±0.07 | 0.37±0.09 | 0.37±0.09 | 0.00±0.00 | 0.13±0.06 |
| *Canonical* | 0.40±0.09 | 0.53±0.09 | 0.67±0.09 | **0.43±0.09** | **0.60±0.09** | 0.47±0.09 |

Table 40: **Per-camera success rates on MetaWorld tasks, In-Distribution cameras (DiTFlow, 30 trials per camera).** Images show the robot workspace from each rollout camera; the angle row gives the camera azimuth. Cells: mean±SE (Bernoulli SE). **Bold**: best per column within each task block. In-distribution: encoder-pretraining cameras (policy trained at 136° only) spanning cluster 1 (≈ 0°) and cluster 2 (≈ 180°).

### L.2 Per-Camera Results by Task

Tables 40 and 41 show per-camera success rates for all reported MetaWorld tasks, split by in- and out-of-distribution cameras. Drawer-close is the task on which baselines most often succeed at cameras other than their training camera: DINOv2 reaches 1.00 at 0° and 170°, and ResNet, VC-1, V-JEPA2, DynaMo and ReViWo each reach 1.00 at one or more out-of-distribution cameras. The task is forgiving (it stays at 0.81–0.95 under every pretraining-camera configuration, Appendix I.8), so high drawer-close success says little about viewpoint robustness.

### L.3 Harder-Split Per-Camera Results

Table 42 pools the harder interpolation/extrapolation split (Section 4.7) into one number per method per condition; Tables 43 and 44 break those out by camera, for the centered and translated conditions respectively. The six cameras (previewed with full specifications in Table 7) are each evaluated centered on the workspace and again with the aim point translated, at 30 trials per camera (10 trials per seed). Column means over all six cameras and four tasks reproduce Table 42.

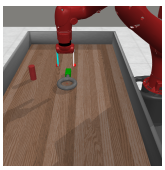
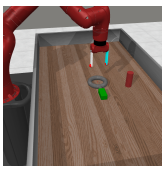
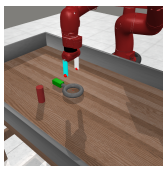
| Task / Method \ Azimuth |  0° |  170° |  125° |  310° |
|---|---|---|---|---|
| *Assembly* | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | 0.03±0.03 | 0.00±0.00 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.10±0.05 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| ReViWo | 0.03±0.03 | 0.37±0.09 | 0.00±0.00 | 0.00±0.00 |
| VILA | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| *Canonical* | **0.50±0.09** | **0.47±0.09** | **0.33±0.09** | **0.47±0.09** |
| *Door Open* | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | **0.83±0.07** | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| ReViWo | 0.33±0.09 | 0.27±0.08 | 0.00±0.00 | 0.00±0.00 |
| VILA | 0.13±0.06 | 0.27±0.08 | 0.47±0.09 | 0.00±0.00 |
| *Canonical* | **0.67±0.09** | **0.50±0.09** | 0.60±0.09 | **0.83±0.07** |
| *Drawer Close* | | | | |
| ResNet | 0.80±0.07 | 0.30±0.08 | 0.00±0.00 | **1.00±0.00** |
| VC-1 | 0.40±0.09 | 0.00±0.00 | **1.00±0.00** | 0.77±0.08 |
| DINOv2 | **1.00±0.00** | **1.00±0.00** | 0.87±0.06 | 0.03±0.03 |
| V-JEPA2 | 0.87±0.06 | 0.93±0.05 | **1.00±0.00** | 0.50±0.09 |
| DynaMo | 0.30±0.08 | 0.33±0.09 | **1.00±0.00** | **1.00±0.00** |
| ReViWo | 0.90±0.05 | 0.83±0.07 | 0.80±0.07 | **1.00±0.00** |
| VILA | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 |
| *Canonical* | 0.90±0.05 | 0.80±0.07 | 0.93±0.05 | 0.93±0.05 |
| *Window Open* | | | | |
| ResNet | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 | 0.00±0.00 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.40±0.09 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | **0.83±0.07** | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.30±0.08 | 0.00±0.00 |
| ReViWo | 0.27±0.08 | 0.20±0.07 | 0.07±0.05 | 0.50±0.09 |
| VILA | 0.07±0.05 | 0.47±0.09 | 0.00±0.00 | 0.00±0.00 |
| *Canonical* | **0.50±0.09** | **0.60±0.09** | 0.40±0.09 | **0.77±0.08** |

Table 41: **Per-camera success rates on MetaWorld tasks, Out-of-Distribution cameras (DiT-Flow, 30 trials per camera).** Images show the robot workspace from each rollout camera; the angle row gives the camera azimuth. Cells: mean±SE (Bernoulli SE). **Bold**: best per column within each task block. Out-of-distribution: $\{0°, 170°\}$ interpolation; $\{125°, 310°\}$ extrapolation.

Splitting the centered condition into its three interpolation cameras ($\{0°, 170°, 340°\}$, inside the pretraining azimuth clusters) and three extrapolation cameras ($\{125°, 260°, 310°\}$, in the gaps between them), *Canonical*'s success falls from $0.606$ to $0.481$, a $21\%$ relative drop, unlike the flat interpolation-to-extrapolation pattern on the main protocol (Section 4.4).

# M LIBERO-Goal: Additional Evaluations

## M.1 Per-Camera Success Rates

Tables 45 and 46 give the per-camera success rates behind Table 2, for the policy trained at $\theta_{\text{train}} = 0°$.

## M.2 Beyond the Evaluation Range

LIBERO-Goal's evaluation cameras span $[-20°, 25°]$ because tabletop objects begin to occlude the workspace past roughly that point (Appendix B). That is a property of the scene rather than of any

Table 42: Harder camera set on MetaWorld (720 episodes per method per condition), pooled over the four tasks, 3 seeds (Table 1's three frozen policy seeds per task, 10 trials per camera per seed). **Centered**: the six harder cameras aimed at the workspace; **Shifted**: the same cameras with each aim point translated off the workspace (the translation test; camera details in Table 7). This harder camera set differs from Table 1's main protocol, so the two are not directly comparable.

| Method | Centered | Shifted |
|---|---|---|
| *Pretrained* | | |
| ResNet (ImageNet) | 0.114 | 0.097 |
| VC-1 (Majumdar et al., 2023) | 0.043 | 0.090 |
| DINOv2 (Oquab et al., 2024) | 0.126 | 0.099 |
| V-JEPA2 (Assran et al., 2025) | 0.151 | 0.043 |
| *In-Domain SSL* | | |
| DynaMo (Cui et al., 2024) | 0.100 | 0.115 |
| *Multi-View / View-Invariant* | | |
| ReViWo (Pang et al., 2025) | 0.268 | 0.063 |
| VILA (Jeong et al., 2026) | 0.074 | 0.056 |
| ***Canonical*** | **0.543** | **0.507** |

method, but it does leave open how performance falls off further out. We therefore evaluate at $\pm 30^\circ$ and $\pm 40^\circ$, well beyond the range any method was built for, at 30 trials per camera (Table 47).

*Canonical* retains small but measurable success at $\pm 30^\circ$ (0.17 and 0.20), where the baselines are at or near zero, and reaches zero itself by $\pm 40^\circ$. Performance is near zero past roughly $30^\circ$, where tabletop objects substantially occlude the workspace; these results do not distinguish an occlusion limit from representation or policy failure, and they are not comparable to the robustness margins on MetaWorld, where the camera can circle the scene freely. Every cell in this range lies between 0.00 and 0.20, and the only non-zero baseline cells (ResNet 0.07 at $+30^\circ$ and 0.10 at $+40^\circ$, ReViWo 0.07 at $+30^\circ$, VILA 0.07 at $+40^\circ$, DynaMo 0.03 at $+40^\circ$) are within noise of zero, so we do not draw a ranking beyond noting that *Canonical* is the only method reaching 0.15 or above anywhere in the range.

### M.3 LIBERO-Goal Rotation-Input Ablations

On vision-only LIBERO-Goal, removing the warp costs 0.26 ID and 0.21 OOD. Identity conditioning and a fixed random rotation shared by all views stay within 0.05 of no warp ($0.65/0.62$ and $0.62/0.58$). The view-independent variants recover little of the warp's benefit: fixed random rotation recovers $0.01/0.01$ and identity conditioning recovers $0.04/0.05$, in contrast to MetaWorld, where view-independent conditioning recovers 0.02–0.08 of 0.18 (Appendix I.3). On these means, the large gap to the predicted rotation is consistent with the LIBERO-Goal benefit depending on view-dependent conditioning.

On MetaWorld, Appendix I.3 isolates the warp by retraining the behavior-cloning policy under each substituted rotation. Table 48 repeats that design on LIBERO-Goal, and we additionally run a narrower deployment-time intervention on *Canonical*'s LIBERO-Goal policy from Table 2: behavior cloning, trained with the predicted warp, is left unchanged, and only the rotation supplied to the warp at rollout is replaced with the identity. This measures how much the deployed policy depends on the canonicalization it was trained to consume. Unlike in the retrained ladder of Table 31, the policy here cannot adapt to the substituted rotation.

Table 49 reports both arms over the ten evaluation cameras, 30 trials each. The predicted arm scores 0.810 overall, consistent with the 0.827 ten-camera mean in Tables 45 and 46; the 0.017 difference is within the trial-sampling noise of 30 rollouts per camera. Replacing the predicted rotation with the identity costs 0.140 overall. That cost is not spread evenly across cameras. At the canonical view the identity is the correct warp, and it scores higher than the predicted arm in this finite 30-trial sample (0.93 against 0.73). Away from the canonical view the identity intervention degrades sharply, reaching 0.37 against 0.73 at $20^\circ$ and 0.30 against 0.57 at $25^\circ$. The deployed policy's dependence

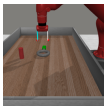
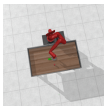
| Task / Method \ Azimuth |  0° |  170° |  340° |  125° |  260° |  310° |
|---|---|---|---|---|---|---|
| *Assembly* | | | | | | |
| ResNet | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | 0.33 | 0.00 | 0.23 | 0.00 | 0.00 | 0.00 |
| VILA | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **0.47** | **0.23** | **0.50** | **0.07** | 0.00 | **0.10** |
| *Door Open* | | | | | | |
| ResNet | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | **0.73** | 0.33 | 0.30 | 0.00 | 0.00 | 0.00 |
| VILA | 0.43 | 0.00 | 0.20 | 0.00 | 0.00 | 0.00 |
| *Canonical* | 0.60 | **0.43** | **0.67** | **0.33** | **0.33** | **0.73** |
| *Drawer Close* | | | | | | |
| ResNet | 0.07 | 0.43 | 0.77 | 0.00 | 0.37 | 0.97 |
| VC-1 | 0.73 | 0.00 | 0.00 | 0.27 | 0.03 | 0.00 |
| DINOv2 | **1.00** | 0.00 | 0.47 | 0.47 | 0.10 | **1.00** |
| V-JEPA2 | 0.33 | **1.00** | **1.00** | 0.07 | 0.03 | 0.97 |
| DynaMo | 0.13 | 0.17 | **1.00** | 0.80 | 0.07 | 0.00 |
| ReViWo | 0.80 | 0.77 | 0.90 | 0.27 | 0.03 | 0.60 |
| VILA | 0.20 | 0.27 | 0.10 | 0.00 | 0.00 | 0.00 |
| *Canonical* | 0.90 | 0.93 | 0.93 | **0.87** | **0.87** | 0.90 |
| *Window Open* | | | | | | |
| ResNet | 0.00 | 0.13 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.23 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.23 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | 0.40 | 0.13 | **0.47** | 0.00 | 0.00 | 0.13 |
| VILA | 0.43 | 0.13 | 0.00 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **0.67** | **0.57** | 0.37 | **0.60** | **0.40** | **0.57** |

Table 43: **Harder-split per-camera success rates on MetaWorld, centered (DiTFlow, 3 seeds, 10 trials per camera per seed).** Each of the six cameras is evaluated centered on the workspace; the top row previews the viewpoint. Interpolation cameras $\{0°, 170°, 340°\}$ sit inside the pretraining azimuth clusters, extrapolation cameras $\{125°, 260°, 310°\}$ in the gaps; all span elevation $-20°$ to $-90°$ and distance 0.8–3.5 m. **Bold**: best per column within each task block; all-zero columns are left unbolded.

on the predicted warp is therefore concentrated at off-canonical views, especially the largest azimuth offsets.

## N xARM: ADDITIONAL EVALUATIONS

Figure 11 breaks the pooled Table 3 down into per-camera success rates for all four methods. Per-task, per-camera cells ($n$=10) are too small to support significance claims individually; since tasks and cameras plausibly differ in base difficulty, we test across them with a Cochran–Mantel–Haenszel test (6 strata: 3 tasks $\times$ 2 out-of-distribution cameras) rather than pooling raw counts into a single contingency table. Out of distribution, *Canonical*'s advantage is significant against all three baselines (ReViWo: $\chi^2$=7.76, $p\approx$0.005; DynaMo: $\chi^2$=18.1, $p\ll$0.001; VILA: $\chi^2$=22.8, $p\ll$0.001); in distribution, the same test shows no difference from ReViWo (34/60 vs. 33/60; $\chi^2$=0.0, $p$=1.0). Object placements (the $\approx$5-position grid of Appendix D) are reused across methods (a paired rather than independent design), and in-distribution/out-of-distribution here refers to the encoder's pretraining cameras, not the policy's single training camera.

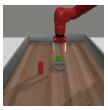
| Task / Method \ Azimuth |  0° |  170° |  340° |  125° |  260° |  310° |
|---|---|---|---|---|---|---|
| *Assembly* | | | | | | |
| ResNet | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VILA | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **0.07** | 0.00 | **0.57** | **0.07** | 0.00 | 0.00 |
| *Door Open* | | | | | | |
| ResNet | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | 0.00 | 0.00 | 0.00 | 0.07 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | 0.03 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 |
| VILA | 0.33 | 0.00 | 0.40 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **0.70** | **0.40** | **0.73** | **0.37** | **0.30** | **0.23** |
| *Drawer Close* | | | | | | |
| ResNet | 0.33 | 0.07 | **0.93** | 0.00 | **1.00** | 0.00 |
| VC-1 | 0.77 | 0.60 | 0.00 | 0.23 | 0.00 | 0.00 |
| DINOv2 | **1.00** | 0.00 | 0.10 | **0.97** | 0.00 | 0.20 |
| V-JEPA2 | 0.13 | 0.03 | 0.20 | 0.00 | 0.03 | 0.60 |
| DynaMo | 0.47 | 0.70 | 0.03 | 0.47 | 0.03 | **0.97** |
| ReViWo | 0.13 | 0.10 | 0.37 | 0.23 | 0.07 | 0.03 |
| VILA | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **1.00** | **0.93** | 0.87 | 0.93 | 0.87 | 0.87 |
| *Window Open* | | | | | | |
| ResNet | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| VC-1 | **0.50** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DINOv2 | 0.00 | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| V-JEPA2 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 |
| DynaMo | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ReViWo | 0.17 | 0.17 | 0.17 | 0.00 | 0.00 | 0.00 |
| VILA | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| *Canonical* | **0.50** | **0.63** | **0.40** | **0.77** | **0.43** | **0.53** |

Table 44: **Harder-split per-camera success rates on MetaWorld, shifted (translation test) (DiTFlow, 3 seeds, 10 trials per camera per seed).** Each of the six cameras has its aim point translated off the workspace (the translation test); the top row previews the viewpoint. Interpolation cameras $\{0°, 170°, 340°\}$ sit inside the pretraining azimuth clusters, extrapolation cameras $\{125°, 260°, 310°\}$ in the gaps; all span elevation $-20°$ to $-90°$ and distance 0.8–3.5 m. **Bold**: best per column within each task block; all-zero columns are left unbolded.

Table 45: LIBERO-Goal per-camera success rate for the $\theta_{\text{train}} = 0°$ policy, in-distribution cameras (mean±SE, binomial SE from 30 trials per camera). **Bold**: best per camera. The top row renders the workspace from each evaluation camera. Exact angles are in Table 10, and note that $\pm 10°$ are unsampled interior angles rather than beyond the range. The column means of this table reproduce Table 2.

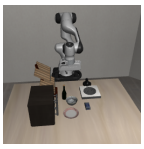
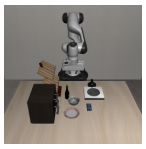
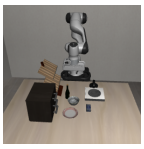
| |  |  |  |  |  |
|---|---|---|---|---|---|
| **Method** \ **Azimuth** | $-15$ | $-6$ | $0$ | $4$ | $15$ |
| *Pretrained* | | | | | |
| ResNet (ImageNet) | 0.07±0.05 | 0.07±0.05 | 0.57±0.09 | 0.17±0.07 | 0.07±0.05 |
| VC-1 | 0.00±0.00 | 0.03±0.03 | 0.27±0.08 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.10±0.05 | 0.30±0.08 | **0.83±0.07** | 0.23±0.08 | 0.07±0.05 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.50±0.09 | 0.03±0.03 | 0.00±0.00 |
| *In-Domain SSL* | | | | | |
| DynaMo | 0.00±0.00 | 0.03±0.03 | 0.40±0.09 | 0.20±0.07 | 0.00±0.00 |
| *Multi-View / View-Invariant* | | | | | |
| ReViWo | 0.67±0.09 | 0.60±0.09 | 0.70±0.08 | 0.67±0.09 | 0.60±0.09 |
| VILA | 0.23±0.08 | 0.40±0.09 | 0.53±0.09 | 0.43±0.09 | 0.40±0.09 |
| *Canonical* | **0.87±0.06** | **1.00±0.00** | 0.77±0.08 | **0.83±0.07** | **0.90±0.05** |

Table 46: LIBERO-Goal per-camera success rate for the $\theta_{\text{train}} = 0°$ policy, OOD cameras (marked $^{\dagger}$: outside the encoder pretraining pool) (mean±SE, binomial SE from 30 trials per camera). **Bold**: best per camera. The top row renders the workspace from each evaluation camera. Exact angles are in Table 10, and note that $\pm 10°$ are unsampled interior angles rather than beyond the range. The column means of this table reproduce Table 2.

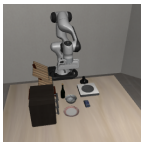
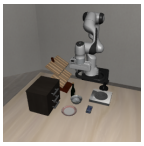
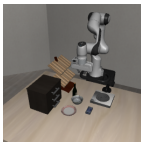
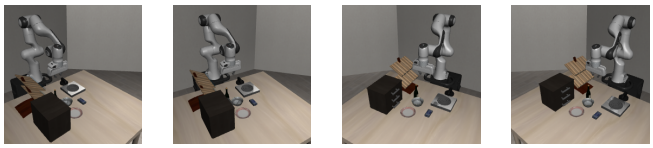
| |  |  |  |  |  |
|---|---|---|---|---|---|
| **Method** \ **Azimuth** | $-20^{\dagger}$ | $-10^{\dagger}$ | $10^{\dagger}$ | $20^{\dagger}$ | $25^{\dagger}$ |
| *Pretrained* | | | | | |
| ResNet (ImageNet) | 0.00±0.00 | 0.07±0.05 | 0.07±0.05 | 0.10±0.05 | 0.03±0.03 |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.07±0.05 | 0.17±0.07 | 0.13±0.06 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| *In-Domain SSL* | | | | | |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 | 0.00±0.00 | 0.00±0.00 |
| *Multi-View / View-Invariant* | | | | | |
| ReViWo | 0.47±0.09 | 0.73±0.08 | 0.70±0.08 | 0.60±0.09 | 0.00±0.00 |
| VILA | 0.40±0.09 | 0.30±0.08 | 0.40±0.09 | 0.47±0.09 | 0.03±0.03 |
| *Canonical* | **0.87±0.06** | **0.80±0.07** | **0.83±0.07** | **0.73±0.08** | **0.67±0.09** |

Table 47: Success beyond the evaluation range on LIBERO-Goal (30 trials per camera, 3 seeds × 10 trials; mean±SE). All methods use the policy trained at the canonical 0° view. **Bold**: best per camera; all-zero columns are left unbolded. Every method collapses to at or near zero across this range; among the pretrained baselines only ResNet is non-zero, at +30° (0.07) and +40° (0.10).



| Method \ Azimuth | $-40°$ | $-30°$ | $+30°$ | $+40°$ |
|---|---|---|---|---|
| *Pretrained* | | | | |
| ResNet (ImageNet) | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 | **0.10±0.05** |
| VC-1 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| DINOv2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| V-JEPA2 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 |
| *In-Domain SSL* | | | | |
| DynaMo | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.03±0.03 |
| *Multi-View / View-Invariant* | | | | |
| ReViWo | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 | 0.00±0.00 |
| VILA | 0.00±0.00 | 0.00±0.00 | 0.00±0.00 | 0.07±0.05 |
| ***Canonical*** | 0.00±0.00 | **0.17±0.07** | **0.20±0.07** | 0.00±0.00 |

Table 48: Retrained rotation-input ladder on LIBERO-Goal (VQ-BeT, 3 seeds, 10 trials per camera per seed; same ten cameras as Table 2). The predicted-rotation row is *Canonical*'s Table 2 policy; the other rows retrain the policy on the substituted rotation. The encoder is fixed for every row. Results are mean±std across the 5 per-camera success rates within each split, as in Table 2.

| Variant | ID | OOD |
|---|---|---|
| ***Canonical* (predicted rotation)** | **0.87±0.08** | **0.78±0.07** |
| No warp | 0.61±0.05 | 0.57±0.06 |
| Identity conditioning | 0.65±0.04 | 0.62±0.05 |
| Fixed random rotation | 0.62±0.06 | 0.58±0.05 |

Table 49: Deployment-time warp intervention on LIBERO-Goal, using *Canonical*'s $\theta_{\text{train}} = 0°$ policy from Table 2 with behavior cloning left unchanged. Only the rotation fed to the canonical warp at rollout is altered: the predicted rotation, or the identity. Success rate per evaluation camera, 30 trials each, azimuths in ascending order ($-20°$ to $25°$, the same ten cameras as Tables 45 and 46). The policy is not retrained under the intervention, so these rows measure deployment-time dependence on the warp and are not comparable to the retrained ladders in Tables 31 and 48.

| Warp rotation | $-20$ | $-15$ | $-10$ | $-6$ | 0 | 4 | 10 | 15 | 20 | 25 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Predicted | 0.80 | 0.80 | 0.90 | 0.93 | 0.73 | 0.80 | 0.93 | 0.90 | 0.73 | 0.57 | 0.81 |
| Identity | 0.47 | 0.67 | 0.80 | 0.87 | 0.93 | 0.83 | 0.77 | 0.70 | 0.37 | 0.30 | 0.67 |

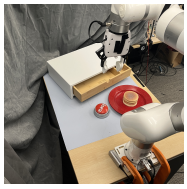
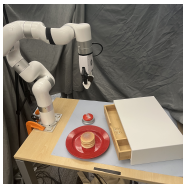
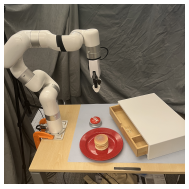
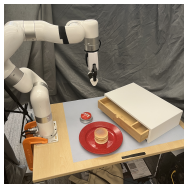
| | In-Distribution | | Out-of-Distribution | |
|---|---|---|---|---|
| |  |  |  |  |
| **Task / Method** \ **Azimuth** | 160° | 300° | 280° | 250° |
| *Button Press* | | | | |
| DynaMo | 3/10 | 5/10 | 1/10 | 0/10 |
| ReViWo | 3/10 | 4/10 | 1/10 | 0/10 |
| VILA | 2/10 | 5/10 | 2/10 | 0/10 |
| *Canonical* | **4/10** | **6/10** | **4/10** | **4/10** |
| *Pick-and-Place* | | | | |
| DynaMo | 2/10 | 3/10 | 2/10 | 0/10 |
| ReViWo | **4/10** | **5/10** | 2/10 | 1/10 |
| VILA | 3/10 | **5/10** | 1/10 | 1/10 |
| *Canonical* | 3/10 | **5/10** | **4/10** | **3/10** |
| *Drawer Close* | | | | |
| DynaMo | 3/10 | 5/10 | 3/10 | 4/10 |
| ReViWo | **7/10** | **10/10** | 8/10 | 6/10 |
| VILA | 2/10 | 4/10 | 2/10 | 1/10 |
| *Canonical* | **7/10** | 9/10 | **10/10** | **7/10** |

Figure 11: Per-camera success rates on xArm (DiTFlow, 10 trials per camera). Columns are the four evaluation cameras: 160° and 300° are ID encoder-pretraining cameras, while 280° and 250° are OOD; the snapshots (top) show each camera's view. **Bold**: best per column within each task. This is the per-camera breakdown of the pooled Table 3.