INVAR: RANDOMIZATION-INVARIANT VISUAL REPRESENTATIONS FOR ZERO-SHOT SIM-TO-REAL

Anonymous authors
Paper under double-blind review

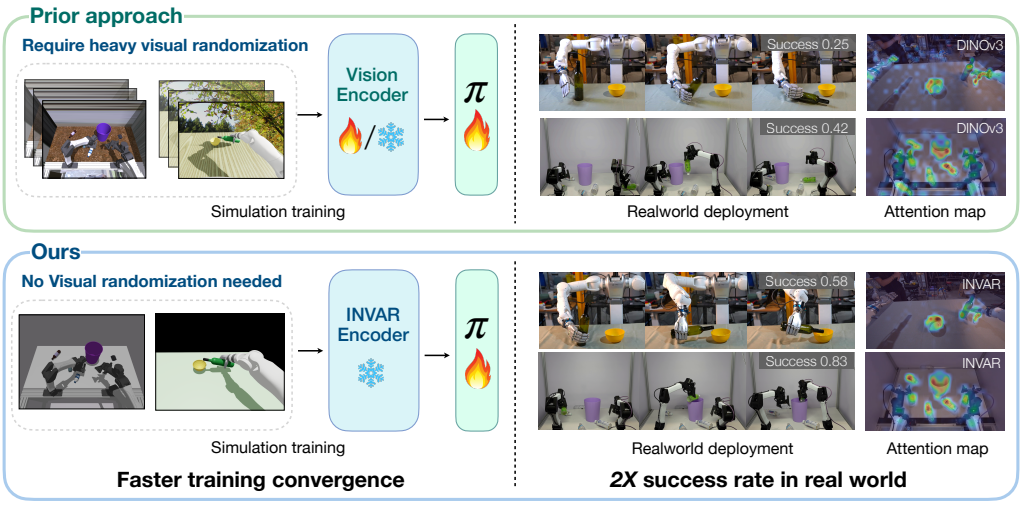


Figure 1: Prior sim-to-real approaches regenerate randomized simulation data and relearn visual invariance per new task. **INVAR instead learns a task-agnostic, randomization-invariant visual representation once, enabling policy training on clean (nominal) simulation data with a frozen encoder.** The resulting policies converge faster and achieve 2× higher real-world success across our dexterous-hand and parallel-gripper evaluations, while maintaining task-relevant attention under the sim-to-real appearance shift.

ABSTRACT

Humans know to act consistently across real and simulated environments because they can abstract away visual appearance to what matters for the task. Can robots achieve similar visual invariance? We present INVAR,[1] a visual representation for *zero-shot* sim-to-real transfer. While classic sim-to-real methods rely on task-specific domain randomization (DR) to train visual encoders and action policies, INVAR distills diverse simulation randomizations into a unified, *invariant* visual encoder that is self-supervised and scalable. It leverages self-distillation over an unlabeled corpus of simulated play data and robot rollouts without requiring real-world images during representation learning. A policy trained purely on nominal simulation data with a frozen INVAR encoder transfers zero-shot to the real world. INVAR nearly doubles the success rate of identical policies built on state-of-the-art pretrained encoders trained with standard DR, across both dexterous manipulation (0.61 vs. 0.30 mean success) and parallel-gripper manipulation (0.65 vs. 0.32). These results demonstrate that decoupling visual invariance into a task-agnostic encoder unlocks scalable, zero-shot sim-to-real transfer, similar to humans. Our website is available at `https://iclrinvar.github.io/invar/`.

1 INTRODUCTION

Simulation provides virtually unlimited interaction data at a marginal cost decoupled from physical robot-hours. However, visual sim-to-real transfer remains bottlenecked by appearance discrepancies. Policies that operate on raw pixels overfit to rendering artifacts such as lighting, material

[1] Invar is a nickel–iron alloy prized for holding its dimensions across a wide temperature range; our encoder holds its representation across a wide range of rendering conditions.

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.

shaders, and background textures, cues that immediately go out-of-distribution (OOD) with physical cameras.

The standard remedy is domain randomization (DR) (Tobin et al., 2017) which varies visual nuisance parameters during policy training so that the real world reads as another in-distribution draw. While effective, it is inefficient for policy optimization. Every new policy must independently re-discover and re-learn visual invariances from scratch. It compounds *optimization difficulty*, policy capacity is split between learning control and marginalizing over a high-variance visual input distribution. Finally, the learned invariance is coupled to task-specific weights and discarded with the policy.

In this work, we leverage a property unique to physics simulators that image datasets cannot replicate: the ability to generate paired renders from the exact same underlying state. Standard self-supervised learning (SSL) frameworks rely on heuristic augmentations such as random crops, color jitter, and Gaussian blur, to approximate visual equivalence (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025). A simulator, however, does not need to approximate. By fixing state s and rendering it under nominal parameters, visually aligned to match real-world conditions versus randomized variations, the resulting image pair differs exclusively along the nuisance dimensions the encoder ought to discard. This provides exact supervision without task labels, rewards, or expensive real-world robot data collection.

INVAR utilizes these paired renders as a training signal for representation learning via asymmetric self-distillation (§3). A teacher network processes the canonical render while a student network processes the randomized render. Pulling the student embedding toward the teacher's embedding maps randomized visual inputs into a canonical representation space. By optimizing this objective over $\approx$6.3M render pairs across three simulated embodiments (§3.2), we adapt a general vision backbone (e.g., DINOv3 (Siméoni et al., 2025)) to map its semantic priors into a representation space invariant to visual randomization, without any task supervision. Once pre-trained, the encoder is frozen, and visual DR can be completely removed for downstream control learning: policies are trained via simple behavior cloning in simulation. Crucially, this turns visual randomization from a per-policy requirement into a one-time representation-learning cost: the randomized simulation corpus is generated once, and the resulting encoder can be reused across downstream tasks trained only on nominal simulation.

In zero-shot real-world evaluation, policies using a frozen INVAR encoder trained only on clean simulation data achieve roughly $2\times$ the success rate of state-of-the-art general-purpose and manipulation-specific baselines trained with domain randomization. On 22-DoF Sharpa dexterous hand manipulation tasks, INVAR achieves a 0.61 mean success rate, a $2.0\times$ improvement over the strongest baseline evaluated on all six (0.30). On bimanual parallel-gripper manipulation tasks, INVAR reaches 0.65 versus 0.32 (§4.2). Notably, INVAR also outperforms a manipulation-specific encoder with over $10\times$ its parameter count. INVAR also accelerates downstream policy learning, reaching 90% of peak performance about $1.5\times$ faster than the fastest baseline. Qualitative analysis further shows that INVAR is robust to visual distractions and maintains consistent representations across the sim-to-real gap (§4.3). Together, these results suggest that our mechanism for learning visual invariance improves both policy optimization and sim-to-real transfer.

Our contributions are:

- **Domain randomization as self-supervision.** We reframe DR from a policy learning regularizer into a view-generation mechanism for representation learning, replacing heuristic image augmentations with state-matched re-renders.
- **Scalable, task-agnostic visual invariance.** We pre-train INVAR once across three simulated embodiments without task supervision or real-robot data. By confining visual randomization to a one-time pretraining stage, the same frozen encoder can be reused across downstream policies trained on nominal simulation, amortizing the cost of learning visual robustness across tasks.
- **Competitive zero-shot sim-to-real transfer performance.** Policies using the frozen INVAR backbone roughly double the real-robot success rates of prior general and manipulation-specific encoders across dexterous grasping and bimanual manipulation.

2 RELATED WORK

2.1 DOMAIN RANDOMIZATION

Domain randomization (DR) remains the de facto mechanism for visual sim-to-real transfer (Tobin et al., 2017; Sadeghi & Levine, 2016). While system-scale DR has driven landmark results in

dexterous (Akkaya et al., 2019; Chen et al., 2023) and bimanual manipulation (Chen et al., 2025), deploying it at the policy level has remained inefficient. The network must dedicate capacity to simultaneously solve control and re-discover ignoring task-irrelevant features. Further, this invariance is *ephemeral* and discarded with the policy. In this paper, we argue that DR is fundamentally a representation-learning tool but masquerades as a policy regularizer.

2.2 Visual Pretraining

To bypass sample-inefficient policy training, recent paradigms rely on visual pretraining (Nair et al., 2023; Xiao et al., 2022). General foundation models like DINOv3 (Siméoni et al., 2025) provide rich semantic priors but lack the invariances required for robot control. Action-centric models like CAIP Sharma et al. (2026) attempt to bridge this by aligning egocentric human video with 3D hand poses via contrastive learning. However, relying on human datasets to learn visual representations entangles learning visual robustness with learning human-to-robot kinematic correspondence. Rather than use out-of-domain human data to heuristically guess at visual invariance, our approach directly solves the sim-to-real appearance gap. Other works Radford et al. (2021); Yu et al. (2024) use natural language supervision to create appearance invariance. These approaches can learn sim-to-real invariant features but may lose task relevant information by compressing visual content into language. By finetuning DINOv3 via self-distillation on simulator-generated exact correspondence pairs, we map general visual features into an invariant state space for robot embodiments—achieving zero-shot sim-to-real transfer without needing human action proxies.

2.3 Self Distillation and Privileged Supervision

Self distillation is standard in self-supervised learning (Grill et al., 2020a; Chen & He, 2021), and privileged teacher-student frameworks are ubiquitous in robot sim-to-real transfer (Pinto et al., 2017; Chen et al., 2026; Lee et al., 2020; Kumar et al., 2021). However, standard robot distillation typically maps privileged low-dimensional simulator or sensor states (e.g., object poses, masses, or friction coefficients) into visual or latent policies (Kumar et al., 2021; Chen et al., 2026). Our architecture is intra-modal (vision-to-vision) and trained independent of control. Rather than passing down privileged physics parameters to guide control, the teacher network processes canonical visual renders to supervise the student's embedding of randomized renders. This decoupling ensures the learned representation remains task-agnostic. INVAR provides a canonical visual state to downstream BC policies, requiring neither rewards during pretraining nor privileged state estimators at inference.

2.4 Sim-to-Real Dexterous Manipulation

Visual sim-to-real transfer in dexterous manipulation is challenging; severe self-occlusions and complex geometry routinely collapse standard encoders, historically forcing reliance on massive DR (Akkaya et al., 2019; Chen et al., 2023). Recent real-world distillation architectures bypass this failure by aligning representations directly in the real world. For instance, Privileged Tactile Latent Distillation (PTLD) Chen et al. (2026) bridges the tactile domain gap via real-world distillation. However, this merely shifts the bottleneck, as it still requires per-task physical data collection. In contrast, by learning visual invariance from paired simulator renders, we resolve the appearance gap. Our results, ($1.8\times$ improvement) over action-centric priors (Sharma et al., 2026), demonstrate that constructing invariance in simulation offers a scalable path for purely vision-based dexterity than relying on per-task DR or physical data collection.

3 Method

INVAR formulates DR as a structured objective for representation learning (Fig. 2). We construct a pretraining dataset of paired observations, extracting both a canonical *clean* render and a domain-randomized render from identical underlying simulator states. Through a self-distillation objective (§3.1), we align these pairs to enforce an encoder representation that is invariant to visual perturbations parameterized by the DR distribution. The pretraining corpus aggregates these paired observations across three simulated embodiments (§3.2), and we select checkpoints with a linear action probe (§A). For downstream evaluation, the pretrained encoder is frozen, and a behavior-cloning (BC) policy is trained entirely with the invariant feature space (§4, Appendix B.2).

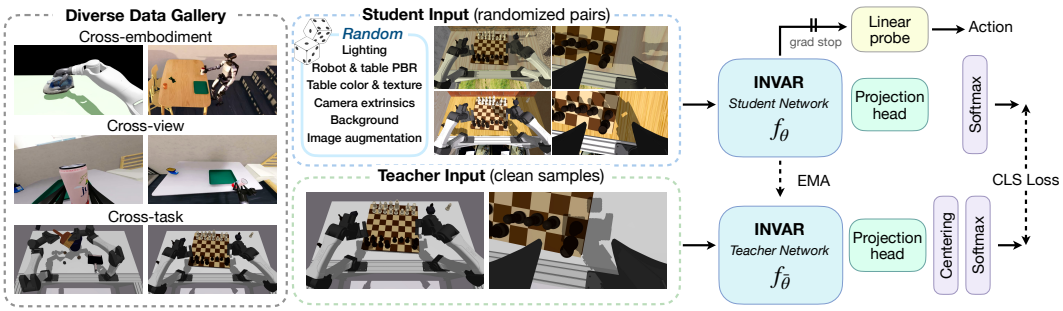


Figure 2: **INVAR self-distillation.** INVAR learns generalizable visual representations from state-matched renders across three robot embodiments (dexterous hand, parallel gripper, humanoid), diverse tasks, and camera views. A student network processes visually randomized views and minimizes a CLS self-distillation loss against an EMA teacher receiving clean renders. A detached linear action probe monitors action-relevant features without updating the encoder.

3.1 SELF-DISTILLATION FOR REPRESENTATION LEARNING

Self-distillation (Grill et al., 2020b; Caron et al., 2021; Baevski et al., 2022) trains a student encoder f_θ to match the output of an exponential moving average (EMA) teacher $f_{\bar{\theta}}$. Given two views v_s and v_t of a training sample, the objective minimizes the difference between their respective projections:

$$\min_{\theta} \; \mathbb{E}_{(v_t,\, v_s)} \, \ell\Big(h_\theta\big(f_\theta(v_s)\big),\; \mathrm{sg}\left[h_{\bar{\theta}}\big(f_{\bar{\theta}}(v_t)\big)\right]\Big). \tag{1}$$

To prevent representational collapse, this formulation relies on the stop-gradient operator $\mathrm{sg}[\cdot]$, EMA weight updates, and centering of teacher outputs (Caron et al., 2021). The two views inherently play asymmetric roles. The teacher typically receives the more informative view, such as global crops in DINO (Caron et al., 2021) or the unmasked signal in masked distillation (Zhou et al., 2022; Baevski et al., 2022; Sharma et al., 2025). Importantly, the invariances acquired by the encoder are dictated by the view-generation process. In standard visual SSL, views are generated via stochastic augmentations of an image (i.e. random cropping, color jitter) forcing the model to learn representations invariant to those specific transformations (Caron et al., 2021; Oquab et al., 2024).

INVAR keeps Eq. 1 unchanged but redefines the view-generating process. Instead of images, our training samples are simulator states s. By sampling visual DR parameters $\omega \sim \Omega$ (see Fig. 2), a renderer R produces a canonical appearance $v_t = R(s, \omega_0)$ and a randomized appearance $v_s = R(s, \omega)$ of the *same* geometry. Unlike standard DINO where some views are seen by both the teacher and student, in our formulation the teacher receives only clean renders, and the student receives only domain randomized renders. Optimizing Eq. 1 pulls the embedding of every randomized view toward its canonical counterpart, enforcing $f(R(s,\omega)) \approx f(R(s,\omega_0))$ for all ω. This yields a representation invariant to the simulator family of visual DR, learned without task labels, rewards, or real robot data collection.

We use DINO prototype cross-entropy loss (Caron et al., 2021), fine-tuning a pretrained DINOv3 ViT-S+/16 backbone (28.7M parameters) (Siméoni et al., 2025) rather than training from scratch. From each frame we take $T{=}2$ teacher crops of the clean render and $S{=}8$ student crops of the randomized render (§3.2) and apply the loss to every teacher–student pair:

$$\mathcal{L}_{\text{cls}} = \frac{1}{TS} \sum_{i=1}^{T} \sum_{j=1}^{S} H\big(P_{\bar{\theta}}(v_t^{(i)}),\; P_\theta(v_s^{(j)})\big), \tag{2}$$

where H is cross-entropy and $P_{\bar{\theta}}$, P_θ are the teacher and student softmax distributions over $K{=}65{,}536$ prototypes. The teacher output distribution is centered and sharpened as in DINO. A single projection head is shared across domains. Further details are listed in Tab. 6 (Appendix A).

3.2 PRETRAINING DATA

Our training corpus comprises clean–randomized image pairs across three simulated embodiments, requiring no task labels; exact episode and pair counts are detailed in Tab. 5 (Appendix A).

- **Dexterous grasping.** A 22-DoF SHARPA hand mounted on a 7-DoF Tianji arm grasps 20 household objects within IsaacLab (the environment used in §4.1), observed via a head camera. Trajectories are play data from a trained RL grasping policy: for each object, we roll out the policy checkpoint and replay them across randomized object poses keeping both successful and failed episodes; the encoder learns appearance invariance rather than the task, so failures are as useful as successes.
- **Parallel-gripper manipulation.** We utilize the bimanual parallel-jaw station, simulator, and episode data from ABC (Allshire et al., 2026). Recorded episodes of five tabletop tasks in MuJoCo/MJWarp (§4.1) are kinematically replayed and re-rendered under both appearance conditions from three camera viewpoints (one overhead, two wrist) using Isaac Renderer.
- **Humanoid loco-manipulation.** This domain features egocentric clips of a Unitree G1 transporting objects in reconstructed scenes, sourced from the VLK dataset (Wang et al., 2026). The G1 is equipped with a gripper and observed by head and wrist cameras. Clips are rendered in a nominal appearance alongside three re-lit variants. The nominal variant serves as the teacher view.

In total, the corpus contains ≈6.3M pairs (12.6M rendered frames). The pretraining data is strictly disjoint from downstream evaluation: none of the 20 grasp objects or five ABC pretraining tasks appear in §4, which evaluates novel objects and tasks on the same embodiments (the humanoid domain is not evaluated downstream). Because all pretraining data are generated entirely in simulation, the corpus can be produced without real-robot data collection and scaled efficiently across additional embodiments, tasks, and visual conditions.

Randomization and cropping. Render-time randomization (Fig. 2) leaves scene geometry and the manipulated object's appearance untouched. At *train time*, student crops undergo additional augmentations: a segmentation-guided background swap (using a random photograph (Everingham et al., 2010), solid color, or noise, without replacing robot or object pixels) and photometric augmentations. All views are generated as random resized crops, recentered to ensure the manipulated object remains in view.



Figure 3: **Real-world setups.** Dexterous-hand platform and bimanual parallel-gripper platform.

4 EXPERIMENTS

We evaluate Behavioral Cloning (BC) policies trained with frozen INVAR visual encoder across two real-world setups. Our experimental evaluation is structured into three parts:

1. **Main Experiments** (§4.2): We address two research questions: (1) Does INVAR improve *zero-shot sim-to-real* policy transfer, across robot embodiments, on objects and tasks absent from encoder pretraining? (2) Does INVAR accelerate policy training convergence?
2. **Qualitative Analysis** (§4.3): We investigate the representations learned by INVAR to understand the mechanisms driving its transfer capabilities.
3. **Ablations** (§4.4): We evaluate how performance scales with data diversity and dataset size, we verify compatibility across different network backbones, and we also disentangle scene in-distributionness with explicit DR invariance.

4.1 SETUP

Robot platform and tasks. We evaluate on two real-robot platforms (Fig. 3): a dexterous setup with a Sharpa hand mounted on a Tianji arm, and a bimanual parallel-gripper setup with YAM arms.

- **Dexterous manipulation suite:** We evaluate grasping on five objects not seen during INVAR encoder pretraining, together with an additional unseen task *pour wine into bowl*. Although grasping appears during encoder pretraining on 20 different objects, neither the evaluation objects nor the pouring task appear in the pretraining data.
- **Bimanual parallel-gripper manipulation suite:** We evaluate two tasks, *put the bottle in the bin* and *flip the mug*. Both the target objects and task concepts are absent from INVAR encoder pretraining, which uses five different ABC tasks.

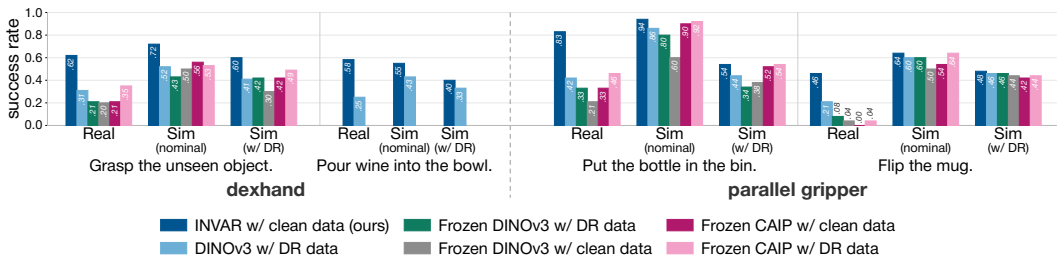


Figure 4: **INVAR improves sim-to-real transfer.** *Left*: dexterous-hand manipulation on five unseen grasping objects and the unseen *pour wine into the bowl* task. *Right*: bimanual parallel-gripper manipulation on two unseen tasks. Across both embodiments, INVAR achieves roughly $2\times$ the real-world success rate of the strongest baselines, with a smaller advantage in simulation, indicating improved robustness to the sim-to-real visual shift. Real-world results average 125 rollouts for dexterous grasping and 24 rollouts per method for the remaining tasks; simulation results use 50 episodes per evaluated checkpoint (full per-object and per-task results in Appendix Tab. 7 and 8).

Thus, all evaluations test zero-shot transfer to objects not seen during INVAR encoder pretraining, while the pouring and parallel-gripper evaluations additionally test transfer to unseen task concepts and motions. See Appendix B.1 for platform, simulation, and task details (Fig. 7).

Baselines. We benchmark two state-of-the-art pretrained visual encoders: **general-purpose** DINOv3 (Siméoni et al., 2025) and **manipulation-specific** CAIP (Sharma et al., 2026), to test whether strong visual pretraining already provides robustness to sim-to-real appearance shifts.

Across these encoders, we vary two factors: whether the visual encoder is frozen or finetuned, and whether the BC policy is trained on nominal or domain-randomized (DR) simulation data. Policies built on INVAR use a frozen encoder and clean simulation data. All methods use the same downstream BC architecture; further baseline and policy training details are provided in Appendix B.2.

Evaluation Protocol. To evaluate robustness, realworld evaluations systematically vary environmental factors including ambient lighting, table textures, object initializations, and background clutter (Fig. 8). Simulation success rate is reported under two conditions: *Nominal* (standard non-randomized renders) and *Visual DR* (randomized materials, lighting, background, and camera extrinsics).

4.2 MAIN RESULTS

INVAR improves zero-shot sim-to-real transfer to unseen objects and tasks across embodiments. On the **dexterous hand**, INVAR trained only on nominal simulation data improves over the strongest baseline by $1.8\times$ on grasping, averaged across five objects not seen during encoder training (0.62 vs. 0.35, *CAIP w/ DR data*), and by $2.3\times$ on *pour wine into bowl* (0.58 vs. 0.25, *DINOv3 w/ DR data*), which requires an unseen manipulation motion of pouring (Fig. 4). On the **bimanual parallel-gripper** tasks, INVAR similarly achieves approximately $2\times$ the mean real-world SR of the strongest baseline, *DINOv3 w/ DR data* (0.65 vs. 0.32). Thus, INVAR transfers strongly both to unseen objects and to unseen manipulation behaviors for different embodiments.

We observe that baseline failures are often caused by visually induced misalignment: inaccurate grasps and placements, as well as failed handovers between the two arms. These failures become more frequent under larger appearance shifts, such as moving backgrounds (e.g., a person walking behind the workspace) and changes in lighting. In contrast, INVAR remains robust under these visual distractions and maintains more consistent task execution. We hypothesize that INVAR is less sensitive to these disturbances because its encoder learns a more robust visual representation; we analyze this further in §4.3.

The advantage of INVAR is substantially larger on physical hardware than in simulation. For example, on *put the bottle in the bin*, CAIP nearly matches INVAR in nominal simulation (0.92 vs. 0.94), but falls far behind in the real world (0.46 vs. 0.83). Notably, despite using a manipulation-specific encoder over $10\times$ larger than INVAR, CAIP has a much larger sim-to-real performance drop, indicating that INVAR is more robust to visual differences between simulation and the real world.

Table 1: **Policy convergence speed.** Mean epoch at which each method first reaches 90% of its own peak success rate, averaged across five training runs. Lower is faster.

	INVAR	DINOv3			CAIP	
Mean epoch to 90% peak SR ↓	frozen (clean data)	frozen (clean data)	finetuned (DR data)	frozen (DR data)	frozen (clean data)	frozen (DR data)
sim (nominal)	**29.4**	61.4	54.4	44.6	56.0	54.8
sim (w/ visual DR)	**27.6**	52.8	38.4	80.8	42.0	65.6

INVAR accelerates policy convergence. Averaged across the training runs, INVAR reaches 90% of its own peak performance about $1.5\times$ faster than the fastest baseline (Tab. 1). Using each method's own peak separates convergence speed from final performance. Notably, INVAR also converges faster than frozen baselines trained on either clean or randomized data. This indicates that it is the invariant representation learned by INVAR, rather than encoder freezing or clean-data training, that makes downstream policy optimization easier.

4.3 QUALITATIVE ANALYSIS

To better understand INVAR's performance gains, we probe the frozen encoders along three axes: *where it attends*, *how it organizes scene appearance*, and *whether the patch features correspond across the sim-to-real gap*.

Attention on manipulation-relevant structure. Fig. 5 shows last-block attention on an egocentric human image from EgoDex (CAIP's pretraining domain) and a real-world robot deployment image, both outside INVAR's pretraining distribution. Despite never being trained on human or real-world images, INVAR consistently concentrates on the manipulated objects and contact regions (cup and bottle; hammer head, nail block, and fingertips). In contrast, DINOv3 also allocates substantial attention to background textures, reflections, and lighting, and CAIP remains diffuse even on the egocentric human images it was pretrained on. This contrast explains why INVAR is more robust to lighting changes and background distractors during real-world deployment.

Robustness to appearance nuisances The patch-level PCA analysis Oquab et al. (2024) further shows that INVAR suppresses task-irrelevant appearance variation while preserving manipulation-relevant structure (Fig. 9). Under INVAR, visual nuisances, such as cast shadows and bystanders, merge into the background feature region, leaving only the gripper and manipulated object distinct. In contrast, DINOv3 allocates distinct feature directions to these nuisances, while CAIP yields spatially fragmented representations.

Zero-shot sim-to-real correspondence. We match manipulation-relevant objects and robot components (e.g., the bowl, bottle, fingertip, and forearm) from a nominal simulation frame to their nearest-neighbor patches in a real deployment frame using encoder feature space cosine similarity. (Fig. 6). INVAR correctly recovers the corresponding object and robot part for all four queries, preserving their identities across the sim-to-real visual shift. In contrast, DINOv3 confuses robot parts, matching the arm to the opposite arm, while CAIP maps several queries to background regions. These results suggest that the visual invariances learned by INVAR from clean–randomized simulation pairs generalize beyond randomizations in simulation to real-world observations, providing downstream policies with more consistent manipulation-relevant features across the sim-to-real gap.

Positional collapse of masked-patch objectives. While appending DINOv3's masked-patch (iBOT) loss (Zhou et al., 2022) to our objective is a natural extension, we find surprisingly, that it degrades dense representations. Training with iBOT leads to significant drops in patch-level metrics, while image-level retention remains stable (Tab. 9). This occurs because fixed-camera visual DR makes local patch appearance unpredictable across episodes, forcing iBOT to shortcut to spatial coordinate prediction (more details at Appendix C.2.1).

4.4 ABLATIONS

We ablate three primary factors of INVAR: (1) *pretraining dataset composition*, (2) *distillation objectives*, and (3) *encoder architecture*. All ablations fix the policy architecture and evaluate frozen encoders across two dexterous grasping tasks (hammer, spray bottle) and one parallel-gripper task (*bottle-in-bin*). Results are reported as averages across three tasks.

Dataset properties governing transfer. We isolate three axes of the pretraining data (Tab. 2–3):

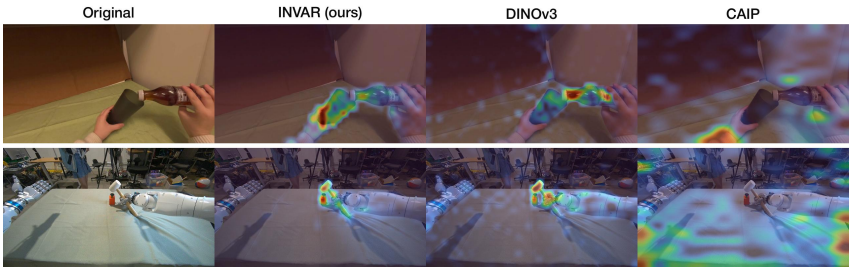


Figure 5: **Attention maps on out-of-distribution images.** Last-block CLS attention on an egocentric EgoDex image (top) and a real-world robot deployment image (bottom), neither seen during INVAR pretraining. Columns show the original image and attention maps from INVAR, DINOv3, and CAIP overlaid on the input. INVAR concentrates attention on manipulated objects and contact regions, while DINOv3 also attends to background appearance and CAIP remains diffuse.

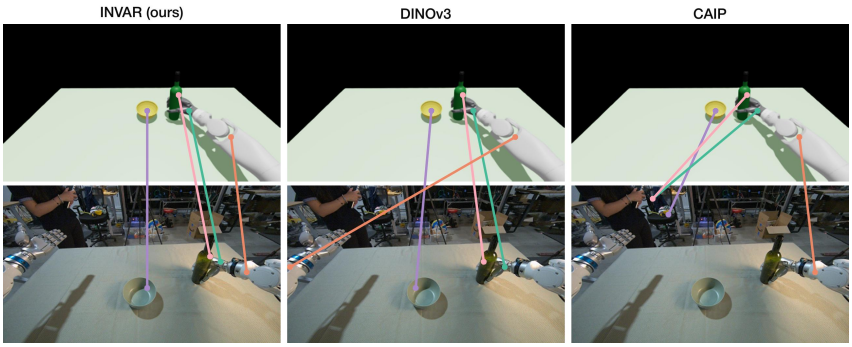


Figure 6: **Sim-to-real patch correspondence.** Top: four manipulation-relevant query patches from simulation (bowl, fingertip, bottle, and arm). Bottom: their nearest-neighbor matches in the real image under cosine similarity in each encoder's patch-feature space. INVAR recovers all correspondences, while DINOv3 confuses robot parts and CAIP maps several queries to background regions.

- *Embodiment diversity:* At a fixed pretraining budget, broader embodiment diversity substantially improves real-world transfer. On the dexterous platform, success drops from 0.84 with all three embodiments to 0.71 without humanoid data and 0.63 with single-embodiment pretraining; on the parallel-gripper platform, it similarly decreases from 0.83 to 0.67 and 0.58. Notably, although all downstream evaluations are manipulation tasks, adding even a small amount of humanoid loco-manipulation data improves performance on both platforms. This suggests that **representation pretraining benefits from broad embodiment and scene diversity**, even from embodiments far from the downstream robots.
- *Task and object diversity:* At a fixed pretraining budget, increasing within-embodiment diversity from 5 to 20 objects or 2 to 5 tasks yields only modest gains in real-world transfer. Together with the larger gains from multi-embodiment pretraining, this suggests that **broad cross-embodiment variation may be more valuable than adding semantic diversity within a single embodiment**.
- *Dataset scale:* Reducing the pretraining corpus from the full dataset to 50% and 25% consistently lowers real-world success on both platforms, indicating that **physical transfer benefits from scaling representation pretraining data**. Since all pretraining data are simulated, scaling is relatively inexpensive, requiring no additional real-world data collection.

Across these ablations, nominal simulation performance remains near saturation and therefore fails to reveal substantial differences in physical transfer.

Asymmetric distillation vs. DR exposure. Do the transfer gains come from exposure to visually randomized data, or from our asymmetric training objective? We compare INVAR against a *symmetric-DR* ablation that uses the same pretraining data, visual randomization, architecture, and optimization budget. The ablation derives both teacher and student views from the same randomized render $R(s, \omega)$ using standard DINO-style cropping and photometric augmentations. In contrast, INVAR keeps the teacher view clean, $R(s, \omega_0)$, while the student receives the randomized render $R(s, \omega)$ of the same state, explicitly aligning appearance-perturbed observations to a canonical tar-

Table 2: **Pretraining data ablation: embodiment diversity and scale.** Real world success rate is reported over 12 rollouts for ablations.

		Dexterous hand			Parallel gripper		
axis	**corpus**	**real**	**sim nominal**	**sim w/ DR**	**real**	**sim nominal**	**sim w/ DR**
ours	dexterous hand + parallel gripper + humanoid	**0.84**	**0.96**	**0.67**	**0.83**	0.94	**0.54**
embodiment	dexterous hand + parallel gripper	0.71	0.84	0.62	0.67	0.92	0.48
	dexterous hand / parallel gripper	0.63	**0.96**	0.54	0.58	**0.96**	0.44
data scale	50% of pairs	0.71	0.89	0.63	0.67	0.90	0.46
	25% of pairs	0.58	0.90	0.61	0.50	0.94	0.40

Table 3: **Pretraining data ablation: object and task diversity.** See Figure 11 for the object and task selection of each corpus.

axis	**corpus**	**real**	**sim nominal**	**sim w/ DR**
objects, dexterous hand	20 objects	**0.63**	**0.96**	0.54
	10 objects	0.58	0.88	**0.55**
	5 objects	0.54	0.84	0.52
tasks, parallel gripper	5 tasks	**0.58**	**0.96**	0.44
	2 tasks	0.50	0.94	0.44

Table 4: **Method ablations: distillation objective, backbone, and policy training data.**

encoder / policy data	**real**	**sim nominal**	**sim w/ DR**
INVAR (clean)	**0.84**	0.95	**0.61**
Distillation algorithm			
symmetric-DR	0.42	0.94	0.43
Backbone (ConvNeXt)			
INVAR w/ clean data	0.65	**0.97**	0.58
ConvNeXt w/ DR data	0.38	0.73	0.39
Policy training data			
INVAR (DR)	0.79	0.85	**0.61**

get. While both methods perform similarly in nominal simulation (0.94 vs. 0.95 SR), symmetric DR reduces real-world success from 0.84 to 0.42. Since data scale, scene content and DR exposure are matched, the gap shows that exposure to randomized data and downstream-like scenes alone is insufficient; the clean-teacher/randomized-student alignment is what encourages appearance-invariant features that improve zero-shot sim-to-real transfer.

Generalization to convolutional backbones. Our paired distillation recipe is architecture-agnostic. Applying INVAR to a ConvNeXt-T backbone achieves a real-world SR of 0.65, substantially outperforming standard co-training on DR data (0.38 SR; Tab. 4). Although Vision Transformers yield higher absolute performance (0.84 SR), our objective consistently improves physical transfer across both architectural paradigms.

Representation invariance. If INVAR learns invariance to visual DR, applying DR during policy training should provide no further benefit. Evaluating a frozen INVAR encoder confirms this (Tab. 4): training the policy on DR renders yields comparable real-world transfer (0.79 DR vs. 0.84 clean) and identical performance under visual-DR simulation (0.61 SR). This suggests that the relevant visual invariance is already encoded in the frozen representation, making additional DR during downstream policy learning unnecessary. **Takeaway:** once visual robustness is encoded in INVAR, domain randomization becomes a one-time pretraining cost rather than a per-policy requirement, allowing the frozen encoder to be reused across downstream tasks trained on nominal simulation.

5 CONCLUSION

We presented INVAR, a visual encoder for zero-shot sim-to-real transfer trained by self-distilling 6.3M paired nominal and domain-randomized simulation renders. Without requiring real images, or task labels, frozen INVAR enables BC policies trained exclusively on clean simulation data to roughly double the real-world success rates of SOTA baselines across dexterous and bimanual platforms on novel tasks. Ablations show that this robustness stems from explicit cross-domain alignment rather than passive exposure to visual randomization, while nominal simulation performance remains a poor predictor of physical transfer. INVAR converts visual DR from a task-specific policy-training requirement into a reusable representation-pretraining cost: simulation is randomized once to train the encoder, after which new policies can be learned from nominal simulation alone.

Limitations and future work. INVAR isolates the visual reality gap, leaving geometric, dynamic, and actuation disparities unaddressed. Future work will extend our paired objective to physical and geometric randomizations, evaluate downstream RL policies, and scale pretraining embodiment diversity—which our analysis highlights as the critical driver of physical generalization.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with coding and running experiments (i.e. hyper parameter optimization), as well as literature review along with outlining and drafting text. We have not used generative AI tools for the final text in this paper or assistance on any of the core ideas, hypothesis or claims of this paper. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

To the best of our knowledge this work does not raise any ethical concerns. There were no human subjects, and discrimination, bias, or security concerns due to the nature of the work.

REPRODUCIBILITY STATEMENT

Code and encoder checkpoints will be made available at the time of publication.

REFERENCES

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, David McAllister, Justin Yu, Yiyuan Chen, Huang Huang, Pieter Abbeel, Xi Chen, Rocky Duan, Phillip Isola, Jitendra Malik, Fred Shentu, Guanya Shi, Philipp Wu, and Angjoo Kanazawa. Scalable behavior cloning with open data, training, and evaluation. *arXiv preprint arXiv:2606.27375*, 2026.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning (ICML)*, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Rosy Chen, Mustafa Mukadam, Michael Kaess, Tingfan Wu, Francois R Hogan, Jitendra Malik, and Akash Sharma. Ptld: Sim-to-real privileged tactile latent distillation for dexterous manipulation. *arXiv preprint arXiv:2603.04531*, 2026.

Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84): eadc9244, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020a.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

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? *arXiv preprint arXiv:2303.18240*, 2023.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

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 Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. Blog post.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Akash Sharma, Carolina Higuera, Chaithanya Krishna Bodduluri, Zixi Liu, Taosha Fan, Tess Hellebrekers, Mike Lambeta, Byron Boots, Michael Kaess, Tingfan Wu, François Robert Hogan, and Mustafa Mukadam. Self-supervised perception for tactile skin covered dexterous hands. *arXiv preprint arXiv:2505.11420*, 2025.

Yuvan Sharma, Dantong Niu, Anirudh Pai, Zekai Wang, Zhuoyang Liu, Baifeng Shi, Stefano Saravalle, Boning Shao, Ruijie Zheng, Jing Wang, Konstantinos Kallidromitis, Yusuke Kato, Fabio Galasso, Yuke Zhu, Danfei Xu, Linxi "Jim" Fan, Jitendra Malik, Trevor Darrell, and Roei Herzig. Contrastive action-image pre-training for visuomotor control, 2026. URL https://arxiv.org/abs/2606.17256.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Yen-Jen Wang, Jiaman Li, Sirui Chen, Takara E. Truong, Pei Xu, Pieter Abbeel, Rocky Duan, Koushil Sreenath, Angjoo Kanazawa, Carmelo Sferrazza, Guanya Shi, and C. Karen Liu. VLK: Learning humanoid loco-manipulation from synthetic interactions in reconstructed scenes. *arXiv preprint arXiv:2606.30645*, 2026.

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

Albert Yu, Adeline Foote, Raymond Mooney, and Roberto Martín-Martín. Natural language can help bridge the sim2real gap. *arXiv preprint arXiv:2405.10020*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023.

Gaoyue Zhou, Zichen Jeff Cui, Ada Langford, Bowen Tan, Yann LeCun, and Lerrel Pinto. Patch policy: Efficient embodied control via dense visual representations, 2026. URL `https://arxiv.org/abs/2607.18236`.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations (ICLR)*, 2022.

A Implementation Details

Linear action probes Closed-loop evaluation at every checkpoint is prohibitively expensive, therefore we score checkpoints with a linear action probe. For each checkpoint and each domain, we freeze the backbone, extract [CLS]-tokens on center-cropped frames of held-out episodes and fit a ridge regression probe to predict next-step actions. To prevent confounding capacity effects across training, the linear action probe is retrained from scratch at each checkpoint with no gradients propagated into the encoder.

Pretraining corpus. Table 5 gives the exact size of each domain of the corpus in §3.2. A pair is one clean and one randomized render of the same frame; the pair count for the parallel-gripper domain includes all three of its cameras.

Table 5: Pretraining corpus, per embodiment.

domain	trajectory source	cameras	episodes	pairs
dexterous grasping (20 objects)	RL grasps, replayed	head	10,216	4,425,179
parallel-gripper manipulation (5 tasks)	ABC (Allshire et al., 2026)	overhead, 2 wrist	3,615	1,852,485
humanoid loco-manipulation	VLK (Wang et al., 2026)	head, wrist	131	18,324
total			13,962	6,295,988

Distillation hyperparameters. Table 6 lists the full configuration of the encoder distillation.

Table 6: Encoder distillation: implementation details. The auxiliary action heads ride on detached tokens as gradient-stopped monitors during training; checkpoint selection uses only the linear probe of §A.

component	value
backbone	pretrained DINOv3 ViT-S+/16 (28.7M, 384-d), fine-tuned end to end
input resolution	224×416 (patch grid 14×26)
views per frame	2 teacher (clean render) + 8 student (randomized render)
crop areas (teacher/student)	$0.80/0.70$ (grasping, ABC); $0.90/0.80$ (humanoid)
projection head	3-layer MLP, 256-d ℓ_2 bottleneck, weight-normed prototypes
prototypes K	$65{,}536$ (one head shared across domains)
teacher temperature τ_t	0.04
teacher centering	EMA of the mean teacher logit, momentum 0.9
student temperature τ_s	0.1
teacher EMA momentum	$0.996 \rightarrow 1$, cosine
optimizer	AdamW (β_1=0.9, β_2=0.999), bf16 autocast
learning rate	backbone peak 10^{-4}, layer-wise decay 0.75; heads 5×10^{-4}
schedule	1.5k-step linear warmup, cosine decay to a 0.01 floor; 25k steps
weight decay	0.04 (none on 1-d parameters)
gradient clipping	3.0
batch	12 frames/GPU $\times$ 8 GPUs $=$ 96 global, domain-homogeneous
checkpoints	every 1,000 steps

Train-time augmentation (student crops only). Where a segmentation mask exists (grasping and ABC), each student crop draws a background recipe: keep the rendered background, or replace the background segment with a random photograph (Everingham et al., 2010), a random solid colour, or Gaussian noise. Robot and object pixels are never replaced, and apertures in the gripper are closed before compositing so no swapped background bleeds through the hand. The humanoid renders carry no segmentation, so their appearance variation comes from the re-lit variants alone. All student crops further receive scale and position jitter of the crop window (an image-space proxy for camera perturbation) and photometric augmentation: colour jitter, Gaussian blur, grayscale, and pixel noise. Teacher crops are cropped and normalized only. Every crop, teacher and student, is re-centred so the manipulated object remains in view, using the object segmentation where available and a centre-biased anchor for the humanoid.

B Experiment Details

B.1 Evaluation Platforms and Tasks

Our two real-world evaluation platforms are shown in Fig. 3, and Figure 7 shows all eight evaluated tasks.

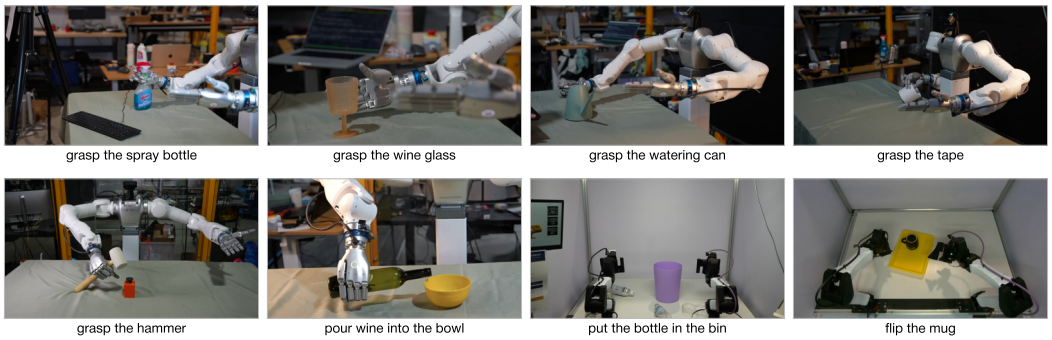
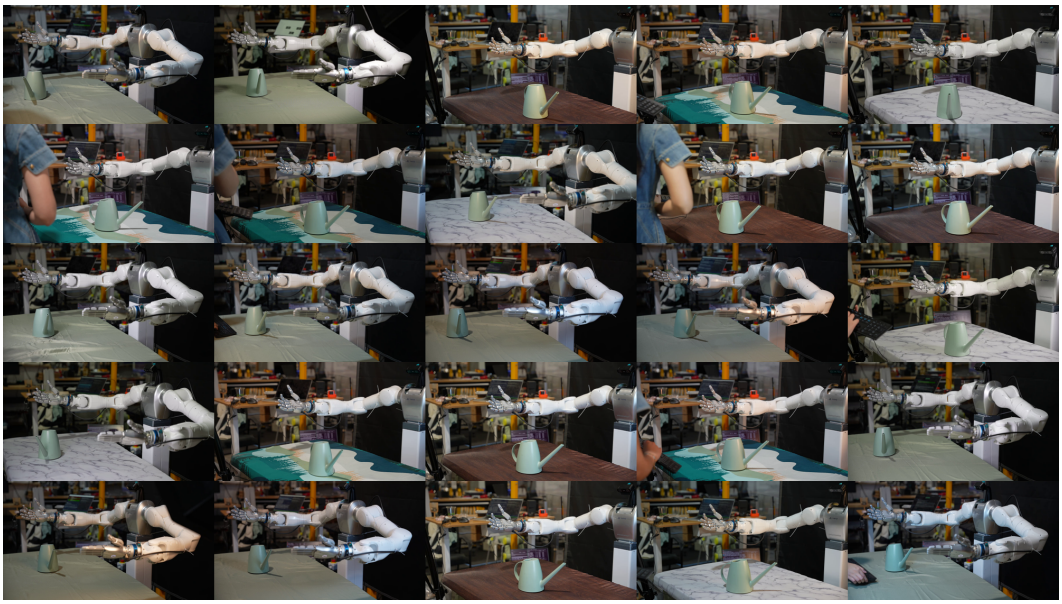
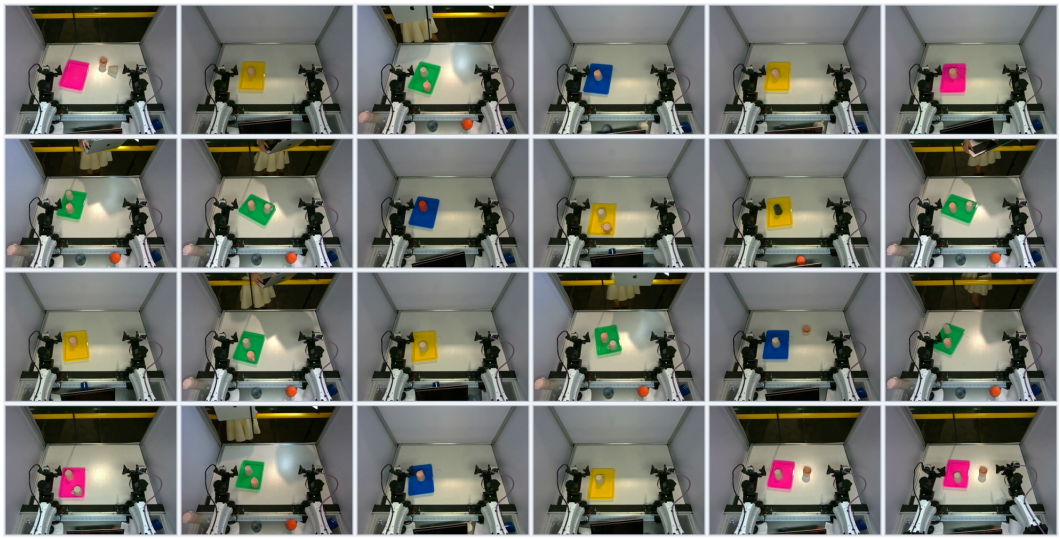


Figure 7: Eight real-world evaluation tasks. **Top row and bottom left:** the dexterous-hand suite — grasping the five unseen objects and the unseen *pour wine into the bowl* task. **Bottom right:** the two bimanual parallel-gripper tasks.



(a) dexterous hand (watering can)



(b) parallel gripper (mug flipping)

Figure 8: Real-world evaluation configurations on the two platforms.

Dexterous manipulation suite. The dexterous platform consists of a 22-DoF SHARPA hand mounted on a 7-DoF Tianji arm, with observations captured from a head-mounted Zed X mini camera. Policies are trained in IsaacLab (Mittal et al., 2025) and deployed on the physical system zero-shot. The evaluation suite contains five grasping objects—spray bottle, wine glass, watering can, tape, and hammer—with a dedicated BC policy for each object, together with the additional non-grasping task *pour wine into bowl*. The five evaluation objects are distinct from the 20 objects used during INVAR encoder pretraining, and the pouring task and motion are also absent from pretraining.

Parallel-gripper manipulation suite. The parallel-gripper platform is a bimanual setup with YAM arms from the ABC benchmark (Allshire et al., 2026). It is simulated in MuJoCo (Todorov et al., 2012) with Isaac rendering. The evaluation includes two tasks: *put the bottle in the bin* and

flip the mug. Both the target objects and task concepts are distinct from the five ABC tasks used during INVAR encoder pretraining.

B.2 POLICY TRAINING DETAILS

All baselines use the same BC policy architecture and training hyperparameters, consisting of a visual encoder followed by a flow-matching action head (Zhou et al., 2026).

We evaluate DINOv3 under three configurations: *frozen + nominal*, *frozen + DR*, and *finetuned + DR*. In the first two settings, the visual encoder remains fixed while the BC policy is trained on nominal or domain-randomized simulation renders, respectively. In *finetuned + DR*, the encoder and BC policy are jointly trained on DR data, corresponding to the standard end-to-end sim-to-real pipeline.

For CAIP, we evaluate *frozen + nominal* and *frozen + DR*. We keep CAIP frozen because its backbone is over $10\times$ larger than that used by DINOv3 and INVAR, making downstream finetuning substantially more computationally expensive. CAIP has been shown to outperform prior general-purpose and robot-specific visual encoders (Tong et al., 2022; Oquab et al., 2024; Zhai et al., 2023; Tschannen et al., 2025; Qwen Team, 2026; Nair et al., 2023; Xiao et al., 2022; Majumdar et al., 2023).

C ADDITIONAL RESULTS

C.1 MAIN-EXPERIMENT FULL RESULTS

Table 7: Full per-object results for Figure 4 dexhand tasks.

(a) Dexterous grasping, **realworld**.

		DINOv3			CAIP	
object	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
spray_bottle	**0.76**	0.18	0.36	0.20	0.24	0.36
wine_glass	**0.40**	0.16	0.12	0.04	0.04	0.28
watering_can	**0.40**	0.16	0.24	0.20	0.18	0.30
tape	**0.60**	0.00	0.16	0.00	0.04	0.12
hammer	**0.92**	0.50	0.67	0.63	0.56	0.70
mean	**0.62**	0.20	0.31	0.21	0.21	0.35
mean (4 obj)	**0.54**	0.13	0.22	0.11	0.12	0.27

(b) Dexterous grasping, **sim, nominal**.

		DINOv3			CAIP	
object	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
spray_bottle	**0.94**	0.62	0.50	0.46	0.84	0.50
wine_glass	0.42	0.24	0.30	0.20	**0.64**	0.52
watering_can	**0.58**	0.28	0.44	0.30	0.14	0.30
tape	**0.74**	0.66	0.48	0.46	0.48	0.48
hammer	**0.90**	0.70	0.88	0.72	0.68	0.84
mean	**0.72**	0.50	0.52	0.43	0.56	0.53
mean (4 obj)	**0.67**	0.45	0.43	0.36	0.53	0.45

(c) Dexterous grasping, **sim, w/ visual DR** (randomised lighting, materials and camera).

		DINOv3			CAIP	
object	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
spray_bottle	**0.76**	0.38	0.52	0.62	0.52	0.50
wine_glass	**0.34**	0.12	0.16	0.16	0.28	**0.34**
watering_can	0.38	0.16	0.28	0.20	0.26	**0.44**
tape	**0.62**	0.34	0.36	0.50	0.36	0.42
hammer	**0.92**	0.50	0.72	0.60	0.70	0.74
mean	**0.60**	0.30	0.41	0.42	0.42	0.49
mean (4 obj)	**0.53**	0.25	0.33	0.37	0.36	0.43

Table 8: Full per-task for Figure 4, **bimanual parallel gripper**.

(a) Parallel-gripper manipulation, **realworld**.

		DINOv3			CAIP	
task	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
put bottle in bin	**0.83**	0.21	0.42	0.33	0.33	0.46
flip mug	**0.46**	0.04	0.21	0.08	0.00	0.04
mean	**0.65**	0.13	0.32	0.21	0.17	0.25

(b) Parallel-gripper manipulation, **sim, nominal**.

		DINOv3			CAIP	
task	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
put bottle in bin	**0.94**	0.60	0.86	0.80	0.90	0.92
flip mug	**0.64**	0.50	0.60	0.60	0.54	**0.64**
mean	**0.79**	0.55	0.73	0.70	0.72	0.78

(c) Parallel-gripper manipulation, **sim, w/ visual DR**.

		DINOv3			CAIP	
task	INVAR (ours)	clean, frozen	DR	DR, frozen	clean	DR
put bottle in bin	**0.54**	0.38	0.44	0.34	0.52	**0.54**
flip mug	**0.48**	0.44	0.46	0.46	0.42	0.44
mean	**0.51**	0.41	0.45	0.40	0.47	0.49

C.2 ADDITIONAL ANALYSIS RESULTS

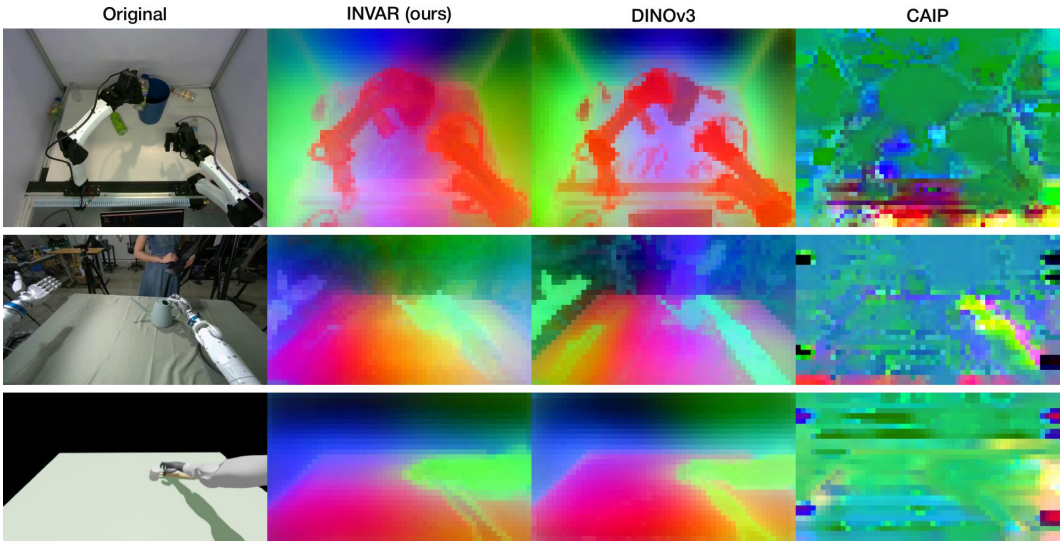


Figure 9: **Patch-level feature PCA under appearance shifts.** Top three PCA components of frozen patch features for a real parallel-gripper rollout (top), a real dexterous-hand rollout (middle), and a clean simulation render (bottom), with the basis fit per clip. With INVAR, cast shadows in each scene blend into the surrounding background feature region; in the middle row, the human bystander in the background is similarly suppressed, while the robot and manipulated object remain distinct. In contrast, DINOv3 assigns separate feature regions to these appearance nuisances, while CAIP produces spatially fragmented features.

C.2.1 FROZEN-REPRESENTATION PROBES AND POSITION-CODE COLLAPSE

Table 9 quantifies the qualitative analysis of §4.3 with four standard frozen-feature probes, comparing INVAR against stock DINOv3, CAIP, and a variant of our distillation that adds DINOv3's

masked-patch (iBOT) loss. *ImageNet linear* trains a linear classifier on frozen features (DINOv2 protocol; general semantics retained); *NYU-Depth* reads metric depth out of frozen patch tokens with a linear head (geometry retained); *DAVIS* propagates a first-frame object mask through a video purely by patch nearest neighbors (temporal correspondence, the property policy heads consume); and the *clean↔DR matching* probe measures exactly what the distillation trains for — each DR patch must find its pixel-aligned counterpart in the clean render — with a derangement control (*content gap*: aligned minus shuffled-episode accuracy) that separates content-based invariance from degenerate position codes.

Three observations. First, INVAR buys its invariance cheaply: relative to the stock encoder it gives up 6.5 points of ImageNet linear accuracy, 0.06 m of depth RMSE, and 8.6 DAVIS points, while raising clean/DR matching from 0.76 to 0.90 with the largest genuine content gap (0.27). Second, CAIP trails every probe by a wide margin despite its $10\times$ larger backbone — its features are neither general nor DR-invariant, consistent with its policy results. Third, the iBOT-recipe variant appears to win the matching probe with a perfect 1.00, but its content gap of 0.005 exposes the score as a position code rather than content correspondence (analyzed below); it also pays the largest temporal-correspondence penalty (DAVIS 42.1). This trade-off is why INVAR distills at the CLS level on paired renders rather than with masked-patch objectives.

Table 9: Frozen-representation probes. IN1k = ImageNet-1k linear top-1 (%, ↑); NYU = monocular-depth linear probe (RMSE in m, ↓; δ_1, ↑); DAVIS = semi-supervised label propagation (J&F, ↑); clean↔DR = patch-matching accuracy at radius 1 on pixel-aligned pairs and its content gap (both ↑; high accuracy with gap ≈ 0 indicates position-code collapse, marked †).

	IN1k lin.	NYU RMSE	NYU δ_1	DAVIS J&F	DR acc.	content gap
DINOv3 (frozen)	**78.7**	**0.449**	**0.850**	**69.4**	0.763	0.181
CAIP	30.7	0.851	0.550	43.1	0.482	0.146
INVAR + masked-patch (iBOT)	74.3	0.569	0.744	42.1	1.000†	0.005
INVAR	72.2	0.513	0.796	60.8	**0.898**	**0.272**

Position-code collapse under masked-patch objectives. Evidence for the analysis in §4.3: adding DINOv3's masked-patch (iBOT) loss to the paired distillation makes the patch tokens degenerate into *position codes* on our domain-randomized corpus. The iBOT-recipe encoders score a perfect 1.00 on the clean↔DR matching probe of Table 9 but retain essentially the same accuracy when the clean frame is swapped for one from a *different episode*: their content gap is 0.000–0.005, versus 0.27 for INVAR and 0.18 for stock DINOv3. The collapsed patch embeddings exhibit high cross-image similarity at identical grid locations (cosine 0.97 over 64 frames; stock 0.82, INVAR-family 0.68) but near-orthogonality across locations within a single image (0.06, vs. 0.60 for DINOv3): the features encode location over semantics, which is why cross-domain matching succeeds even between misaligned episodes. Figure 10 visualizes the collapse as a static coordinate lattice devoid of semantic content.

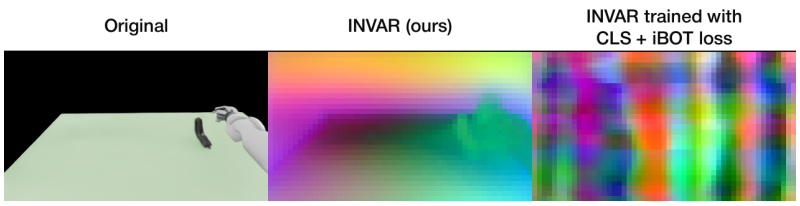


Figure 10: Patch-feature PCA (top-3 components, basis fit per encoder over the clip) on a domain-randomized drill frame. INVAR produces a smooth scene-geometry map in which the arm and object carry their own feature directions. INVAR trained with the CLS + iBOT loss instead renders a repeating position lattice that ignores the arm and object, i.e. the patch embedding encodes grid coordinates rather than content.

C.3 ENCODER ABLATIONS

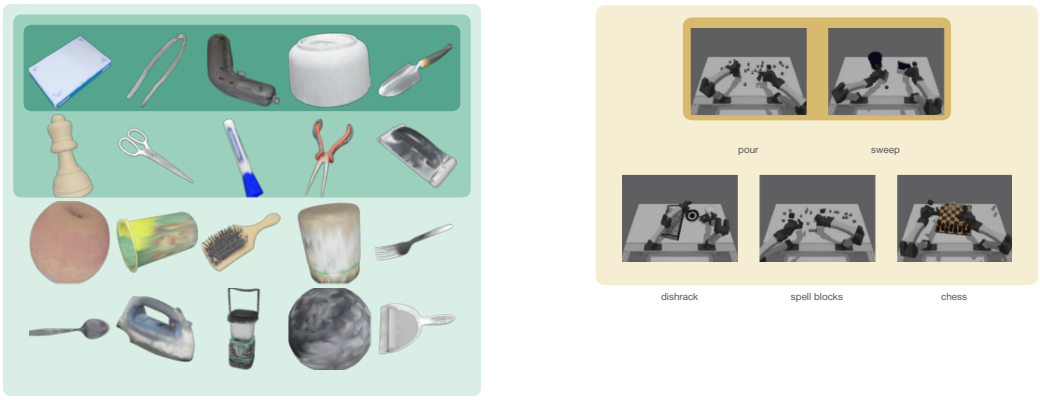


Figure 11: The corpus contents behind the two per-embodiment diversity axes of the data ablation: the 20 dexhand objects (**left**) and the 5 gripper tasks (**right**), with each smaller corpus drawn as a nested chip ($5 \subset 10 \subset 20$ objects; pour+sweep $\subset$ 5 tasks). The success rates of the resulting frozen encoders are in Table 3.

Table 10: Encoder ablations.

		real			sim nominal			sim w/ DR		
axis	encoder	dex	grip	**avg**	dex	grip	**avg**	dex	grip	**avg**
	INVAR	0.84	0.83	**0.84**	0.96	0.94	0.95	0.67	0.54	**0.61**
	INVAR enc., DR BC data	0.83	0.75	0.79	0.80	0.90	0.85	0.69	0.52	**0.61**
embodiment	no humanoid	0.71	0.67	0.69	0.84	0.92	0.88	0.62	0.48	0.55
objects, dexhand	20 objects	0.63	–	0.63	0.96	–	**0.96**	0.54	–	0.54
	10 objects	0.58	–	0.58	0.88	–	0.88	0.55	–	0.55
	5 objects	0.54	–	0.54	0.84	–	0.84	0.52	–	0.52
tasks, gripper	5 tasks	–	0.58	0.58	–	0.96	**0.96**	–	0.44	0.44
	2 tasks	–	0.50	0.50	–	0.94	0.94	–	0.44	0.44
data scale	50% of pairs	0.71	0.67	0.69	0.89	0.90	0.90	0.63	0.46	0.55
	25% of pairs	0.58	0.50	0.54	0.90	0.94	0.92	0.61	0.40	0.51
algorithm	crop-only views	0.42	0.42	0.42	1.00	0.88	0.94	0.46	0.40	0.43
backbone	ConvNeXt-T, ours	0.63	0.67	0.65	0.93	1.00	**0.97**	0.63	0.52	0.58
	ConvNeXt stock, unfroz.+DR	0.42	0.33	0.38	0.51	0.94	0.73	0.47	0.30	0.39