Imitation from Heterogeneous Demonstrations using Grounded Latent-Action World Models

Tianyou Wang **Anson Lei** **Joe Watson** **Ingmar Posner**
University of Oxford
{tianyou, anson, joewatson, ingmar}@robots.ox.ac.uk

Abstract: Imitation learning has emerged as a powerful paradigm for learning visuomotor policies, but its generalisation and stability are limited by the scale and quality of demonstration data needed. A promising direction is to leverage more abundant but heterogeneous data sources, which differ in action space and often lack action labels altogether. Existing co-training approaches that combine heterogeneous data sources rely on heuristic and hand-engineered alignment techniques. In contrast, we argue that action representations should be grounded in prediction: actions that produce the same effect on the environment should share the same representation, regardless of their sources. To this end, we instantiate this principle by using a grounded latent-action world model (GLAM), a pair of generative models with a shared latent action space across data sources that is grounded by predicting future observations consistently across sources. This latent action space is used to train downstream behavioural cloning (BC) policies which map observations to latent actions and decode them back to robot actions, providing a paradigm for learning from heterogeneous data. Empirically, we demonstrate that GLAM successfully learns an aligned latent action space that facilitates action transfer across data sources with and without action labels. Across five manipulation tasks in simulation and in the real world, GLAM-aligned policies significantly outperform BC baselines and prior latent-action methods, achieving an average of +48% improvement in task success rate with the same data-scarce setting. Videos and code are available at https://vicccccciv.github.io/glam/.

Keywords: world models, behavioural cloning, imitation learning

1 Introduction

Modern imitation learning has produced impressive visuomotor manipulation policies [1, 2, 3, 4], yet their generalisation remains tightly coupled to the scale and quality of expensive demonstration data. Even on simple tabletop tasks, behaviour cloning typically requires hundreds to thousands of teleoperated trajectories on the target robot to cross the threshold of reliable deployment [5, 6, 7]. Collecting such data is slow and costly, and the dominant bottleneck for scaling capable policies to the long tail of real-world tasks. One promising approach to sidestep this problem is to replace some of this expensive supervision with cheaper data, such as trajectories collected with portable handheld devices [8, 9, 10], in simulation [11, 7], or scraped from human video [12, 13]. These sources are abundant and far cheaper than demonstrations on the target robot. The central aim of this paper is to investigate how one can leverage heterogeneous data to supplement target-robot demonstrations.

Using heterogeneous data is challenging: different data sources carry different action spaces, frequently lack action labels altogether, and exhibit substantial visual variation in embodiment and scene that does not reflect the underlying task. These mismatches mean that naively pooling the data and training on the mixture rarely works. Prior work bridges these gaps with hand-crafted alignment recipes [14, 15, 16, 17]. For example, the sim-real co-training method [14] aligns simulation and real feature spaces with an optimal-transport loss, a DTW-based temporally-aligned sampling strategy, and an empirically tuned mixing ratio. In contrast, we advocate for a more principled approach: we posit that *what matters for manipulation is how an action affects the environment, regardless*

of where it comes from. Two actions that drive the manipulated object along the same trajectory should be represented similarly, regardless of whether they originate from different embodiments, real or simulated. This turns cross-source integration into a representation-learning problem, with environment transitions as the shared, physically grounded supervisory signal.

We argue that a world model, trained to predict how actions shape environment transitions, provides a natural way to ground latent actions in physical dynamics. Given labelled demonstrations in the target domain and auxiliary demonstrations without action labels, we treat actions as latent variables and learn two coupled generative models (grounded latent-action world model, GLAM): one over the full mixture of target and auxiliary data, and one over the labelled target data. The mixed-data model uses an inverse dynamics model (IDM) to infer latent actions directly from environment transitions, enabling source-invariant inference without action labels. The target-domain model uses an action encoder to infer latent actions from executable robot actions, ensuring that the latent space remains grounded in the target robot's action space. An asymmetric KL objective, together with shared forward dynamics, binds these two inference pathways into a single aligned latent action space. We then propose a GLAM-aligned behavioural cloning (BC) pipeline based on this unified action space. Specifically, using GLAM, we relabel all available data with source-agnostic, control-aware latent actions that serve as supervision signals for a downstream BC policy.

Empirically, we validate that GLAM unifies labelled and unlabelled actions into a single latent space across data sources. Latent actions inferred from auxiliary demonstrations can be transferred directly into executable actions for a target robot arm, making unlabelled auxiliary data usable as BC supervision. Moreover, we experimentally demonstrate that GLAM-aligned latent action policies are able to efficiently learn from auxiliary data sources, outperforming BC and other latent action baselines (up to $+69\%$) across three real-world and two simulated manipulation tasks.

2 Related Work

Imitation Learning from Heterogeneous Demonstrations. Modern robotic imitation learning spans action chunking [2], diffusion policies [1], regression-based variants [3], behaviour transformers [18], and generalist VLA models [19, 4, 20, 21], all of which rely on large amounts of action-labelled target-robot demonstrations. To reduce this dependence, recent work explores two directions for leveraging cheaper sources. A first line learns from auxiliary data alone: visual representation pretraining from web video relaxes the perception side [12, 22, 23], while portable hand-held devices [8, 9, 10] and in-the-wild human videos [24, 25, 26] supply trajectories collected entirely off the target robot. Hand-held devices, however, need extra hardware-matching to the target gripper and still rely on human collection, which remains time-costly. Human videos drop the device altogether but, lacking target-robot grounding, yield embodiment-agnostic affordances that struggle with precision, dexterity, and task-level learning. A second line co-trains the policy on a mixture of target and auxiliary sources to reduce teleoperation cost while preserving target-specific supervision [27, 11, 28, 29, 16]. For example, prior work [11] co-trains a single policy on a mixture of simulation and real data, relying on an empirically tuned mixing ratio and hand-aligned camera viewpoints between simulation and the real-world setup. These approaches typically navigate data mismatches through empirical mixing ratios, source-specific tokenizers, or heuristic and engineered embodiment alignment, treating cross-source integration as a tuning rather than a learning problem. GLAM instead leverages the world model's nature of modeling action and environment interactions to unify heterogeneous data into a shared physics space.

From Imagined Rollouts to Latent-Action Anchoring. World models in robot learning are predominantly used as imagined simulators for policy optimization [30, 31, 32] or as planners for trajectory optimization [33, 34, 35, 36, 37]. Both routes depend on substantial online interaction or massive pretraining to make rollouts reliable enough for control. GLAM instead uses the world model as a frozen latent-action anchor that supplies only per-step labels rather than rolled-out trajectories, sidestepping the online interaction needed by RL and the autoregressive drift driving the data demands of trajectory optimization, making the world model trainable from a few hundred

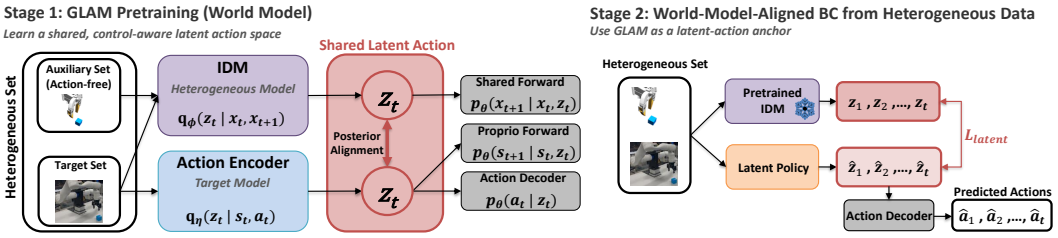

Figure 1: **GLAM-aligned imitation learning pipeline. Stage 1 (left):** GLAM is pretrained on a heterogeneous demonstration set; an IDM (posterior) and an action encoder (posterior) over a shared latent action z_t are aligned by an asymmetric KL and grounded by shared forward dynamics. **Stage 2 (right):** The frozen GLAM relabels every transition with z_t, which supervises a downstream BC policy that predicts latent action chunks and decodes them into executable actions.

mixed trajectories. Closer to our setting, Latent Action Models (LAMs) supply pseudo-action labels for behaviour cloning, extracting action latents from inter-frame transitions. Discrete latents tokenize inter-frame transitions for downstream policy learning [38, 39, 40, 41, 42, 43, 44], while continuous latents directly supervise downstream policies via regression [45, 46, 47, 48, 49]. Two representatives sharpen the design choice: LAPA [38] learns a discrete VQ codebook and utilizes it via large-scale VLM finetuning; CLAM [46] uses a continuous latent jointly trained with an action decoder for direct grounding, but in the single-embodiment setting. GLAM differs in three ways: (i) a single IDM trained across both target and auxiliary data delivers a source-invariant latent action space, rather than one tied to a single embodiment; (ii) a target-side action encoder bound to the IDM by asymmetric KL alignment and shared forward dynamics injects control semantics into the latent, avoiding post-hoc grounding of pixel-reconstruction latents loosely tied to actions; (iii) the resulting model serves as a frozen latent-action anchor that turns unlabelled auxiliary data into BC supervision, without large-scale pretraining.

3 Learning from Heterogeneous Demonstrations with World Models

We aim to learn a visuomotor policy on a target robot that benefits from auxiliary data $\mathcal{D}^{\text{aux}}$ in addition to a small set of target-robot data $\mathcal{D}^{\text{tar}}$. We assume access to the heterogeneous dataset $\mathcal{D} = \mathcal{D}^{\text{tar}} \cup \mathcal{D}^{\text{aux}}$. $\mathcal{D}^{\text{tar}}$ contains trajectories of $\tau = \{(\boldsymbol{o}_t, \boldsymbol{s}_t, \boldsymbol{a}_t)\}_{t=1}^{T}$, where $\boldsymbol{o}_t$ is the observation including the image and end-effector pose, $\boldsymbol{s}_t$ the proprioceptive robot state, and $\boldsymbol{a}_t$ the robot action. Note that $\mathcal{D}^{\text{aux}}$ lacks action labels $\boldsymbol{a}_t$ and robot state $\boldsymbol{s}_t$, so it contains trajectories of $\tau = \{\boldsymbol{o}_t\}_{t=1}^{T}$. We let auxiliary data co-supervise the downstream BC policy through a two-stage pipeline (Figure 1). In Stage 1 (Section 3.1), we treat actions as a latent variable and formulate a pair of generative models (GLAM): one for the heterogeneous dataset $\mathcal{D}$, and the other for the target dataset $\mathcal{D}^{\text{tar}}$, grounding the latent actions inferred from each model into a shared space. In Stage 2 (Section 3.2), we relabel every transition in $\mathcal{D}$ with its latent action and train a BC policy to predict latent actions, decoded back to executable robot actions via an action decoder.

3.1 Grounded Latent-Action World Models

A generative-model view of latent actions. Stage 1 (Figure 1, left) instantiates the shared latent action as a continuous variable $\boldsymbol{z}_t \in \mathbb{R}^d$ in two generative models over trajectory distributions:

$$\text{(heterogeneous)}\quad p(\boldsymbol{X}, \boldsymbol{O}, \boldsymbol{Z}) = p(\boldsymbol{x}_1) \prod_{t=1}^{T} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{z}_t)\, p(\boldsymbol{o}_t|\boldsymbol{x}_t)\, p(\boldsymbol{z}_t), \tag{1}$$

$$\text{(target)}\quad p(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{Z}) = p(\boldsymbol{s}_1) \prod_{t=1}^{T} p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{z}_t)\, p(\boldsymbol{a}_t|\boldsymbol{z}_t)\, p(\boldsymbol{z}_t), \tag{2}$$

where upper-case random variables refer to trajectories, e.g., $\boldsymbol{O} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T]$, and $\boldsymbol{x}_t$ is the latent state encoded from observations. The heterogeneous generative model has access only to observations $\boldsymbol{O}$ and is structured as a standard latent space dynamics model [50] with a latent action variable. Importantly, we use both $\mathcal{D}^{\text{tar}}$ and $\mathcal{D}^{\text{aux}}$ to jointly train this single generative model, ensuring that the transitions in both datasets are generated by latent action $\boldsymbol{z}_t$ from a unified space. To train this

heterogeneous model, we instantiate two learnable posteriors over the latent state and latent actions, $q_{\psi}(\boldsymbol{x}_t \mid \boldsymbol{o}_t)$ and $q_{\phi}(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{x}_{t+1})$, where the posterior over latent state $\boldsymbol{x}_t$ is a standard image encoder and the posterior over latent action $\boldsymbol{z}_t$ is an *inverse dynamics model* (IDM) that infers the action variable given the state transition. Crucially, the use of a shared inverse dynamics model and a shared forward model $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{z}_t)$ across the target and the auxiliary datasets ensures source-invariance of the latent action space, implementing our core principle that any action, regardless of the dataset of origin, should be represented based on how it affects the environment transition. This first generative model allows us to optimise the evidence lower bound objective (ELBO) [51],

$$\mathcal{J}_{\mathrm{H}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}, \mathcal{D}) = \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{E}\left[\log p_{\boldsymbol{\psi}}(\boldsymbol{o}_t|\boldsymbol{x}_t) - \mathbb{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\psi}}(\boldsymbol{x}_{t+1}|\boldsymbol{o}_{t+1}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{z}_t)\right)\right] - \mathbb{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \,\|\, p(\boldsymbol{z}_t)\right] - \mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\psi}}(\boldsymbol{x}_1|\boldsymbol{o}_1) \,||\, p(\boldsymbol{x}_1)]\Big], \quad (3)$$

where observation trajectory $\boldsymbol{O}$ is sampled from $\mathcal{D}$ and $\boldsymbol{z}_t, \boldsymbol{x}_t, \boldsymbol{x}_{t+1} \sim q_{\boldsymbol{\phi},\boldsymbol{\psi}}(\cdot|\boldsymbol{o}_t, \boldsymbol{o}_{t+1})$ where $q_{\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{z}_t, \boldsymbol{x}_t, \boldsymbol{x}_{t+1}|\boldsymbol{o}_t, \boldsymbol{o}_{t+1}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t+1})\, q_{\boldsymbol{\psi}}(\boldsymbol{x}_t|\boldsymbol{o}_t)\, q_{\boldsymbol{\psi}}(\boldsymbol{x}_{t+1}|\boldsymbol{o}_{t+1})$. This formulation is similar to the Dreamer world model architecture [52], with an additional IDM posterior that infers the latent action from observed transitions.

The second generative model grounds its latent action space using extra supervision signals from the robot state $\boldsymbol{S}$ and action labels $\boldsymbol{A}$, available only on $\mathcal{D}^{\mathrm{tar}}$. This ELBO jointly learns the *action encoder* $q_{\boldsymbol{\eta}}(\boldsymbol{z}_t \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ and dynamics model $p_{\boldsymbol{\theta}}(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{z}_t)$,

$$\mathcal{J}_{\mathrm{T}}(\boldsymbol{\theta}, \boldsymbol{\eta}, \mathcal{D}^{\mathrm{tar}}) = \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{E}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{z}_t) + \log p_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{z}_t)\right] - \mathbb{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\eta}}(\boldsymbol{z}_t|\boldsymbol{s}_t, \boldsymbol{a}_t) \,\|\, p(\boldsymbol{z}_t)\right]\Big], \quad (4)$$

where $(\boldsymbol{S}, \boldsymbol{A})$ is sampled from $\mathcal{D}^{\mathrm{tar}}$ and $\boldsymbol{z}_t \sim q_{\boldsymbol{\eta}}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The action encoder posterior together with the target generative model injects control-awareness into the action latent, ensuring it contains the information needed to recover the executable robot action in the target domain. The second action-reconstruction term $\log p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ enforces this latent-to-action decodability and is attached to the action encoder branch only, so that the IDM in Equation (3) remains free of target-specific control particularities. In our implementation, we set the prior, $p(\boldsymbol{z}_t)$, to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for both generative models. Implementation details of the exact model architecture are included in Appendix E.

Asymmetric alignment between posteriors. Finally, in order to bridge the pair of generative models and their respective approximate posteriors, we introduce an additional soft alignment constraint, $\mathcal{C}_{\mathrm{KL}}$, that encourages agreement between the two posteriors over the target dataset,

$$\mathcal{C}_{\mathrm{KL}}(\boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\psi}, \mathcal{D}^{\mathrm{tar}}) \;=\; \mathbb{E}_{\boldsymbol{X}\sim q_{\boldsymbol{\psi}}(\cdot|\boldsymbol{O}),\, \boldsymbol{O},\boldsymbol{S},\boldsymbol{A}\sim\mathcal{D}^{\mathrm{tar}}}\Big[\mathbb{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \,\|\, \mathrm{sg}[q_{\boldsymbol{\eta}}(\boldsymbol{z}_t|\boldsymbol{s}_t, \boldsymbol{a}_t)]\right]\Big], \quad (5)$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation. This deliberate asymmetry reflects that the action encoder carries privileged target-only signal, namely, the ground-truth action labels in the target space. Through this alignment, the IDM absorbs executable-action semantics on target transitions and transports them back to auxiliary ones. This preserves the IDM's source-invariance while keeping the action encoder undistorted by the more ambiguous IDM signal.

Training objective. In summary, GLAM is trained end-to-end by jointly minimising the two negative ELBO losses with respect to the latent action encoders, state encoders and dynamics models,

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\psi}, \boldsymbol{\theta}, \mathcal{D}, \mathcal{D}^{\mathrm{tar}}) \;= -\mathcal{J}_{\mathrm{H}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}, \mathcal{D}) - \mathcal{J}_{\mathrm{T}}(\boldsymbol{\theta}, \boldsymbol{\eta}, \mathcal{D}^{\mathrm{tar}}) + \lambda\, \mathcal{C}_{\mathrm{KL}}(\boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\psi}, \mathcal{D}^{tar}), \quad (6)$$

where soft constraint weighting $\lambda \geq 0$. When combined, these objectives unify $\mathcal{D}^{\mathrm{tar}}$ and $\mathcal{D}^{\mathrm{aux}}$ into a single, source-invariant, control-aware latent space from which any transition can be relabelled with a shared latent action.

Object masks as observations. Our hypothesis is that latent actions should be mapped to the same point if they affect the environment equally. In manipulation tasks, we can further sharpen this: the action representation should reflect how the manipulated object moves. To test this hypothesis, we also investigate a variant of our model that uses binary segmentation mask of the manipulated object extracted by an off-the-shelf segmentation model [53] in place of the RGB frame without changing any ELBO or the training objectives above. In Section 4.3, we empirically show that object masked transitions lead to further improved transfer between auxiliary and target dataset.

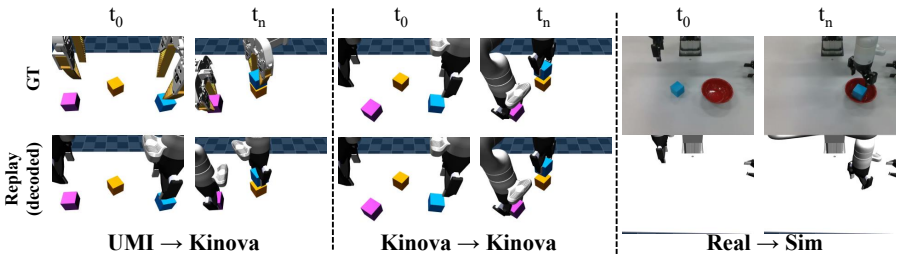

Figure 2: **Cross-source transfer through the shared latent action space.** For each group, an unseen episode (top: GT) is encoded by the IDM into latent actions, decoded by $p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$, which has only seen Kinova-sim data, and replayed open-loop on Kinova in simulation (bottom). Latents from **UMI**, **Kinova-sim**, and **Kinova-real** episodes all reproduce the original motion on Kinova-sim, validating action space alignment in cross-embodiment, in-distribution, and sim-real settings.

3.2 World-Model-Aligned Imitation Learning

Building on the pretrained GLAM in Section 3.1, we introduce a world-model-aligned imitation learning pipeline that turns heterogeneous data into BC supervision. The key idea is to learn latent policies that map observations to the GLAM aligned latent actions, and then decode the policy outputs to robot actions for execution. Concretely, we first relabel the available data, both auxiliary and target, using the IDM posterior mean as action labels,

$$\boldsymbol{z}_t \leftarrow \mu_{\boldsymbol{\phi}}^{\mathrm{IDM}}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{7}$$

which provides a large dataset of observation action pairs that serves as supervision signals for a downstream BC policy. Intuitively, the latent action labels distill information about how objects need to be manipulated for the task at hand while discarding source-specific information.

In principle, any downstream BC architectures can be used for learning policies from our augmented dataset. The specific policy structure used in our experiments is illustrated in Figure 1 (right). The policy operates directly on raw RGB images at both training and inference time. Specifically, two camera views are encoded by a jointly trained DINOv2 backbone [54], followed by a small learned projector, and concatenated with the proprioceptive state $\boldsymbol{s}_t$ to form the policy input $\boldsymbol{c}_t$. We adopt MIP [3], a lightweight two-step regression policy that combines stochasticity injection with supervised iterative computation, as our latent policy. MIP has been shown to match the performance of generative control policies, e.g. diffusion policies [1], while requiring no distribution fitting. Our latent-action policy predicts a chunk of latent actions $\hat{\boldsymbol{z}}_{t:t+H} \sim \pi_{\boldsymbol{\omega}}(\cdot \mid \boldsymbol{c}_t)$, which are further decoded into executable actions as $\hat{\boldsymbol{a}}_{t:t+H} = \boldsymbol{h}^a_{\boldsymbol{\omega}}(\hat{\boldsymbol{z}}_{t:t+H})$. The policy is trained end-to-end on $\mathcal{D}^{\mathrm{tar}} \cup \mathcal{D}^{\mathrm{aux}}$ with the objective:

$$\mathcal{L}_\pi(\boldsymbol{\omega}, \mathcal{D}^{\mathrm{tar}}, \mathcal{D}^{\mathrm{aux}}) = \mathbb{E}_{\tau\sim\mathcal{D}}\big[\|\hat{\boldsymbol{z}}_{t:t+H} - \boldsymbol{z}_{t:t+H}\|_2^2\big] + \mathbb{E}_{\tau\sim\mathcal{D}^{\mathrm{tar}}}\big[\|\hat{\boldsymbol{a}}_{t:t+H} - \boldsymbol{a}_{t:t+H}\|_2^2\big]. \tag{8}$$

The first term draws supervision from all of $\mathcal{D} = \mathcal{D}^{\mathrm{tar}} \cup \mathcal{D}^{\mathrm{aux}}$, so auxiliary trajectories shape $\pi_{\boldsymbol{\omega}}$ on equal footing with target ones; the second term is restricted to $\mathcal{D}^{\mathrm{tar}}$ where action labels are available.

4 Experiments

In this section, we present experimental validation of the proposed pipeline. We aim to answer two questions: first, whether GLAM learns a shared latent action space that is consistent across different data sources; and second, whether auxiliary, non action-labelled data can be used to augment training datasets and improve BC performance. We address the first question qualitatively in Section 4.2 through investigating cross-source latent action transfer, and the second in Section 4.3 through comparisons against BC and latent-action baselines.

4.1 Experimental Setup

Tasks. We evaluate our pipeline on five manipulation tasks spanning two domains. In the real world, we implement three tasks: *lifting*, picking up a cube; *pick-and-place*, transporting a cube into

a bowl; *knock-down*, knocking down a mustard bottle, which requires the gripper to approach at the correct angle since the bottle is stable when struck on its side. In simulation we use two tasks: *2-cube stacking* with a single arm and *3-cube stacking* with two arms.

Robot Platforms. The real-robot setup uses a 7-DoF single Kinova arm with a parallel-jaw gripper (Kinova), observed from two calibrated RealSense cameras. Simulation uses MuJoCo, where the single-arm tasks employ the same Kinova model as in the real setup, and the two-arm task employs two such arms in a shared workspace. The auxiliary data source uses a floating UMI gripper [8] simulated in MuJoCo without joint actions, which is to test whether GLAM can cross this large domain gap. Figure 8 shows the hardware and per-task visualizations.

Datasets. For each task, we collect a small target set $\mathcal{D}^{\text{tar}}$ of 100 demonstrations and a larger auxiliary set $\mathcal{D}^{\text{aux}}$ of 400 unlabelled trajectories. For the real world tasks, $\mathcal{D}^{\text{tar}}$ consists of 100 teleoperated Kinova trajectories; $\mathcal{D}^{\text{aux}}$ consists of 400 simulated UMI trajectories. The world model and policy are trained on $\mathcal{D}^{\text{tar}} \cup \mathcal{D}^{\text{aux}}$ and evaluated on the real robot. We additionally use 100 Kinova-sim trajectories as a third optional auxiliary set for multi-source experiment (Table 1). For simulated tasks, $\mathcal{D}^{\text{tar}}$ comprises 100 simulated Kinova trajectories and $\mathcal{D}^{\text{aux}}$ 400 UMI trajectories, with the world model and policy trained on both datasets and evaluated in simulation.

Baselines. To isolate the contribution of our world model design, we compare against prior latent action works that are implemented with the same network architecture, the same downstream policy, and the same training-data scale as GLAM. Concretely, for each prior work we extract its world-model components (the IDM, the FDM, and the action heads) and the only difference is the latent-action structure learned. The world models are then plugged into the same MIP-based BC policy as ours. Concretely, we evaluate the following models:

- **MIP [3].** A regression-based behaviour cloning policy. Since MIP requires action labels, we train it on the 100 target-robot trajectories of $\mathcal{D}^{\text{tar}}$ alone.
- **CLAM [46].** A latent action model that learns a continuous latent IDM and FDM together with a jointly trained action decoder. As CLAM is designed for the single-embodiment setting, we train it on $\mathcal{D}^{\text{tar}}$, label transitions with its IDM and train the MIP policy to predict these latents.
- **LAPA [38].** A VQ-VAE-based latent action model that learns a discrete codebook of inter-frame latent actions through next-frame reconstruction. Because the codebook is shared across all data sources by construction, we train its IDM and FDM on both $\mathcal{D}^{\text{tar}}$ and $\mathcal{D}^{\text{aux}}$, label every transition with its quantized latent, which the downstream MIP is trained to predict.
- **CLAM-O and LAPA-O.** We evaluate two object-masked variants of CLAM and LAPA, in which the world model input is replaced by the same binary object mask used in our method, isolating the contribution of our paired generative models from the object-mask advantage.
- **GLAM and GLAM-O.** GLAM uses our world model (Section 3.1) with raw RGB input; GLAM-O is the same model with the input replaced by an object-centric binary mask.

4.2 Cross-Source Transfer through a Shared Latent Action Space

To assess whether GLAM has learned a shared latent action space across sources, we investigate whether latent actions inferred from an unlabelled source can be executed on the target robot through its action decoder $p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$. For an unseen episode from a given source, we feed each transition into the IDM to obtain $\boldsymbol{z}_t$ by Equation (7), decode them through $p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$, and replay the result open-loop on Kinova in simulation. Since $p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ is trained only on $\mathcal{D}^{\text{tar}}$, latents from any other source must land within its training distribution to decode coherently. Figure 2 qualitatively demonstrates that the IDM maps all three sources (UMI, Kinova-sim, Kinova-real) into a unified latent action space, enabling the transfer of behaviours across embodiments *without* any action labels. Appendix A shows more cross-source transfer results.

4.3 World-Model Anchoring Enables BC to Generalize from Heterogeneous Data

World-model-aligned BC performance. We first compare our GLAM-aligned policy against the baselines discussed in Section 4.1 on the five manipulation tasks (Figure 3). GLAM(-O) outperforms all baselines on every task, with the largest absolute gains on the more complex tasks: GLAM-O

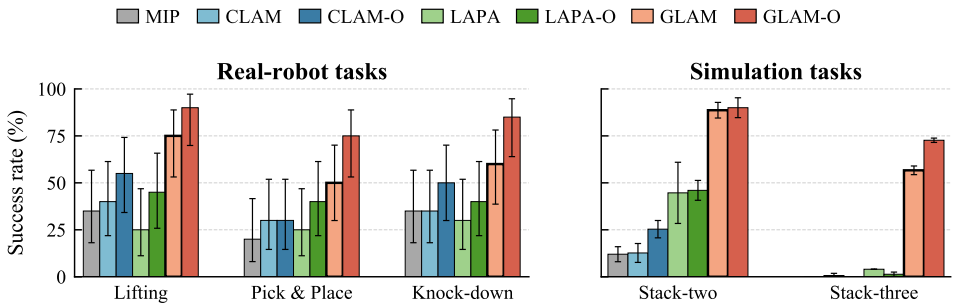

Figure 3: **Main results across three real-robot and two simulated manipulation tasks.** Success rate (%) of baselines and our method. **Real tasks** are evaluated over 20 trials per task; error bars show 95% Wilson score intervals [55]. **Simulation tasks** are trained with 3 training seeds and evaluated with 50 trials each; bars show mean and error bars show cross-training-seed standard deviation. Our method consistently outperforms all baselines and is the only approach to achieve non-trivial success on bimanual stack-three.

achieves an average +35% improvement over the best baseline across three real-world tasks; +44% on simulated stack-two; and +69% on bimanual stack-three. Remarkably, on the stack-three task, which typically requires hundreds to a thousand demonstrations to solve [6, 7], GLAM(-O) are the only methods that can successfully solve the task (72.7% vs. $\leq 4\%$ for all baselines). We perform statistical tests on the improvements using the Newcombe-Wilson hybrid-score 95% confidence intervals [56] on the difference (GLAM-O $-$ best baseline), which show significant directional improvements across all tasks.

Both MIP and CLAM(-O) are designed to be single-source only and therefore cannot make use of the auxiliary dataset. As such, these methods tend to overfit to the limited $\mathcal{D}^{\text{tar}}$ and therefore cannot generalise to unseen task configurations. On the other hand, LAPA(-O) is trained on both sources, but its latent actions are supervised purely by visual reconstruction. While prior work [38] shows that this reconstruction-based approach is sufficient when there is large-scale data, our results demonstrate that IDM reconstruction alone cannot induce reliable transfer between data sources. In contrast, GLAM's paired ELBOs together with the posterior alignment inject control semantics, yielding a latent space that is source-invariant and control-aware.

Object-mask observation improves cross-source transfer. Overall, we observe that GLAM alone, without object masking, consistently outperforms the baselines. However, as discussed in Section 3.1, we conjecture that grounding latent actions based on how *objects* should be manipulated can further improve the quality of the learned action space. Here, Figure 3 shows that object-mask input consistently improves task performance across all 5 tasks. In particular, a one-sided Newcombe-Wilson 95% interval test indicates significant improvements in the knock-down (+25%) and bimanual stack-three (+16%) tasks. These results corroborate our claim that grounding latent action at the object level can improve transfer across datasets.

Heterogeneous data substitutes for target-robot teleoperation. Our central hypothesis is that using auxiliary data with aligned latent action improves data-efficiency in the target domain. To further investigate this, we perform experiments on the stack-two task with varying amounts of available data to characterise the data-efficiency of the proposed method. First, we demonstrate that the baseline MIP is able to reliably solve the task when enough target data is available (Figure 4(a)). Here, by leveraging auxiliary data, GLAM-O is able to match the final performance of the BC baseline with only a small fraction of the target data. In Figure 4(b), we present a more fine-grained result on the performance gains offered by unlabelled data, comparing the performance of GLAM-O given a varying mix of target and auxiliary data. Starting from 100 target demonstrations, we compare the performance gains from adding more target data with those from adding auxiliary unlabelled data, up to a total of 500 trajectories. Note that both cases use the same pretrained GLAM-O model, which is trained on a mixture of target and auxiliary datasets. We observe similar scaling behaviour across the two data sources: increasing the amount of auxiliary data yields performance gains comparable

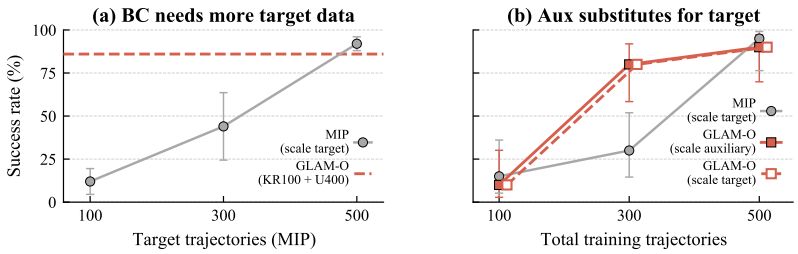

Figure 4: **Heterogeneous data closes BC's data gap on stack-two.** **(a)** Target-data scaling: MIP needs $5\times$ more target trajectories than GLAM-O to reach the same success rate. **(b)** Auxiliary substitutes for target: for GLAM-O, scaling auxiliary UMI data matches scaling target Kinova data trajectory-for-trajectory (the two curves coincide); MIP scales only with target data. Evaluated on 20 fixed unseen initial configurations; error bars are 95% Wilson intervals.

to increasing the amount of target data. This supports our claim that the aligned GLAM action space enables auxiliary data to serve as a viable substitute for target data. Interestingly, Figure 4(b) also shows that latent policies predicting GLAM latent actions is more data-efficient than vanilla BC. We hypothesize that this is because GLAM has access to more data (500 total episodes) during the pretraining phase which induces a more smooth and informative action space that is more conducive for downstream learning. In Appendix B, we provide further analysis at the motion-execution level.

5 Conclusion

In this paper, we introduce GLAM, a pair of latent-action generative models that leverage heterogeneous demonstrations for supervising downstream imitation learning. GLAM aligns an IDM posterior with an action encoder posterior via paired ELBOs and grounds the action latent in shared forward dynamics, then serves as a frozen latent-action anchor that relabels every transition into source-invariant, control-aware supervision for a BC policy. Trained on a few hundred trajectories without large-scale pretraining, GLAM lets cheap auxiliary data substitute for expensive target teleoperation, alleviating BC's data demand and delivering consistent gains across real-robot and simulated manipulation tasks.

Limitations. There are several limitations to the current work. First, our auxiliary trajectories share the deployment camera placement and tabletop scene; the framework has not been tested on truly in-the-wild data such as web video [13, 27], where viewpoint and scene drift would require additional invariance pressure. Second, the framework has not been tested across morphologically distinct end-effectors such as multi-fingered or dexterous hands. A natural extension is to co-train the world model (WM) with human-hand demonstrations [9, 57] as a bridge to multi-fingered embodiments. Third, our auxiliary set shares both the manipulated object and task semantics with the target set; we have not tested auxiliary data sharing only skill primitives with the target task. Since the WM is a general latent-action anchor rather than task-specific, scaling it to many tasks with a shared skill vocabulary, then anchoring single-task policies, is a promising direction. Finally, GLAM grounds the latent in visual object motion, end-effector pose, and proprioception, that suffice for free-space manipulation but cannot capture contact forces or tactile feedback. An interesting extension is to use the WM as a multi-modal bridge: tactile or force signals from instrumented target rollouts [58, 59] flow through the shared latent, lifting vision-only auxiliaries via the unified action space without modality matching on every source.

Acknowledgments

This research was supported by an EPSRC Programme Grant (EP/V000748/1). The authors would like to acknowledge the use of the SCAN computing cluster in carrying out this work. Ingmar Posner holds concurrent appointments as a Professor of Applied AI at the University of Oxford and as an Amazon Scholar. This paper describes work performed at the University of Oxford and is not associated with Amazon.

References

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[3] C. Pan, G. Anantharaman, N.-C. Huang, C. Jin, D. Pfrommer, C. Yuan, F. Permenter, G. Qu, N. Boffi, G. Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. *arXiv preprint arXiv:2512.01809*, 2025.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pages 54877–54910, 2025.

[6] D. Wang, S. Hart, D. Surovik, T. Kelestemur, H. Huang, H. Zhao, M. Yeatman, J. Wang, R. Walters, and R. Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

[7] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[8] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[9] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[10] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025.

[11] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.

[12] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[13] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16939–16947, 2025. doi:10.1109/ICRA55743.2025.11128283.

[14] S. Cheng, L. Ma, Z. Chen, A. Mandlekar, C. Garrett, and D. Xu. Generalizable domain adaptation for sim-and-real policy co-training. *Advances in Neural Information Processing Systems*, 38:11905–11933, 2026.

[15] Y. Liu, W. C. Shin, Y. Han, Z. Chen, H. Ravichandar, and D. Xu. Immimic: Cross-domain imitation from human videos via mapping and interpolation. *arXiv preprint arXiv:2509.10952*, 2025.

[16] R. Punamiya, D. Patel, P. Aphiwetsa, P. Kuppili, L. Y. Zhu, S. Kareer, J. Hoffman, and D. Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *Human to Robot: Workshop on Sensorizing, Modeling, and Learning from Humans*, 2025.

[17] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[18] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[19] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[20] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[21] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[22] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

[23] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[24] H. Chen, B. Sun, A. Zhang, M. Pollefeys, and S. Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27661–27672, 2025.

[25] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024.

[26] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[27] X. Cai, R.-Z. Qiu, G. Chen, L. Wei, I. Liu, T. Huang, X. Cheng, and X. Wang. In-n-on: Scaling egocentric manipulation with in-the-wild and on-task data. *arXiv preprint arXiv:2511.15704*, 2025.

[28] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233, 2025. doi:10.1109/ICRA55743.2025.11127989.

[29] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[30] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

[31] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[32] W. Zhang, G. Wang, J. Sun, Y. Yuan, and G. Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, 2023.

[33] L. Maes, Q. L. Lidec, D. Scieur, Y. LeCun, and R. Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

[34] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[35] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pages 47376–47405, 2024.

[36] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[37] U. Sobal, W. Zhang, K. Cho, R. Balestriero, T. G. Rudner, and Y. LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *Advances in Neural Information Processing Systems*, 38:43905–43941, 2026.

[38] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[39] X. Chen, H. Wei, P. Zhang, C. Zhang, K. Wang, Y. Guo, R. Yang, Y. Wang, X. Xiao, L. Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[40] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

[41] X. Chen, J. Guo, T. He, C. Zhang, P. Zhang, D. C. Yang, L. Zhao, and J. Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

[42] Y. Chen, Y. Ge, W. Tang, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. 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*, pages 19752–19763, 2025.

[43] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg. Amplify: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025.

[44] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[45] D. Schmidt and M. Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pages 9379–9395, 2024.

[46] A. Liang, P. Czempin, M. Hong, Y. Zhou, E. Biyik, and S. Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

[47] B. Tharwat, Y. Nasser, A. Abouzeid, and I. Reid. Latent action pretraining through world modeling. *arXiv preprint arXiv:2509.18428*, 2025.

[48] H. Kim, J. Kang, H. Kang, M. Cho, S. J. Kim, and Y. Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. *arXiv preprint arXiv:2505.08787*, 2025.

[49] Z. J. Cui, H. Pan, A. Iyer, S. Haldar, and L. Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024.

[50] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[51] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[52] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

[53] N. Carion, L. Gustafson, Y.-T. Hu, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[54] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[55] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[56] R. G. Newcombe. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Statistics in medicine*, 17(8):873–890, 1998.

[57] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[58] C. Higuera, S. Arnaud, B. Boots, M. Mukadam, F. R. Hogan, and F. Meier. Visuo-tactile world models. *arXiv preprint arXiv:2602.06001*, 2026.

[59] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.

[60] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet. On the analysis of movement smoothness. *Journal of neuroengineering and rehabilitation*, 12(1):112, 2015.

[61] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1810–1816. IEEE, 2021.

[62] J. Watson and J. Peters. Inferring smooth control: Monte carlo posterior policy iteration with gaussian processes. In *Conference on Robot Learning*, pages 67–79. PMLR, 2023.

A Additional Cross-Source Transfer Results

Figure 5 extends the qualitative analysis of Section 4.2 by showing the full episode trajectories. Across UMI, Kinova-sim, and Kinova-real sources, the decoded actions drive the Kinova robot through the entire task open-loop and reach the goal state, indicating that the IDM consistently produces target-executable latent actions for episodes drawn from any source.

This qualitative experiment of latent quality also guided two design choices in GLAM. First, without the asymmetric alignment in Equation (5), the two posteriors collapse toward each other, dragging the action encoder posterior toward the looser, source-mixed IDM; we observed that latents from both the action encoder and the IDM replay unseen Kinova trajectories markedly worse, and UMI latents from the IDM fail to transfer to Kinova. Second, attaching the action-reconstruction term $\log p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ to the IDM posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{x}_{t+1})$, as in CLAM [46], rather than to the action encoder injects a supervision signal that is not unified across sources into the heterogeneous-model IDM posterior (action labels exist only on $\mathcal{D}^{\text{tar}}$), so UMI latents from the IDM fail to transfer to Kinova and the auxiliary data becomes unreliable for downstream BC. Both led to our final design: control semantics enter through the privileged, target-only action encoder, keeping the IDM source-invariant yet decodable into executable actions.

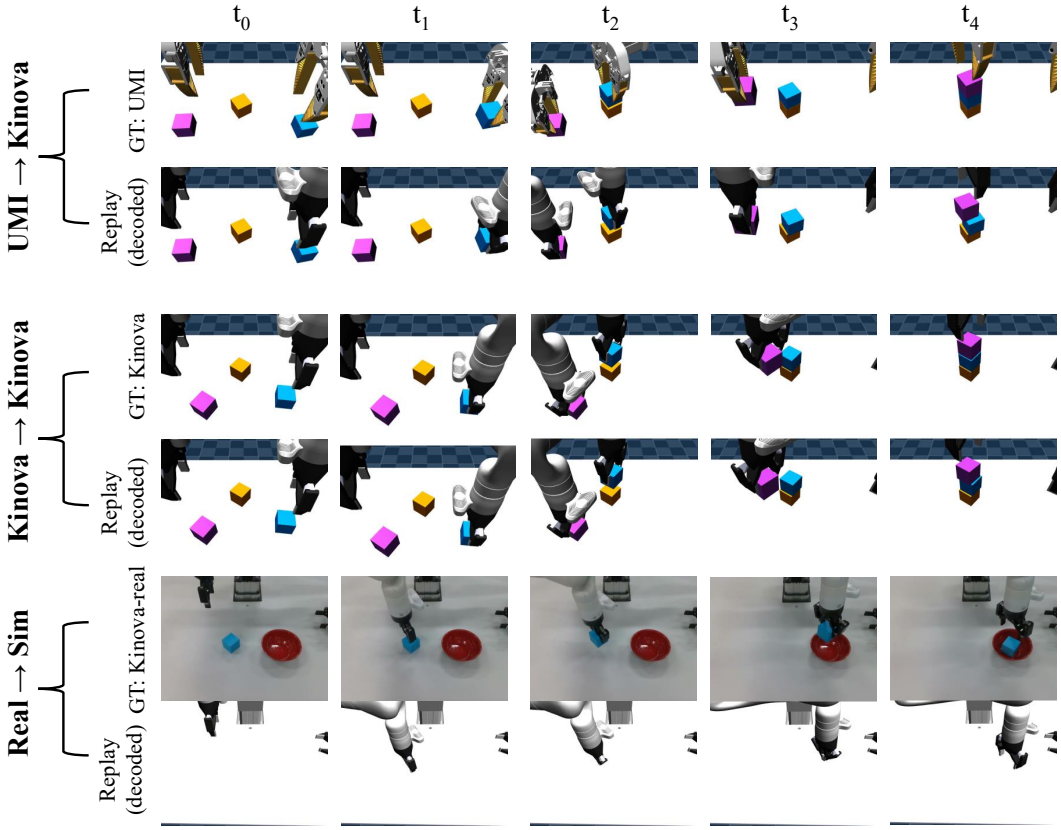

Figure 5: **Extended cross-source transfer results.** For each row group, an unseen episode (top: GT) is encoded by the IDM into latent actions, decoded by the Kinova-sim action decoder $p_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$, and replayed open-loop on Kinova in simulation (bottom). **UMI → Kinova**: latents inferred from an unseen UMI episode reproduce the manipulation on Kinova, demonstrating cross-embodiment transfer. **Kinova → Kinova**: latents from an unseen Kinova-sim episode replay the original motion, an in-distribution sanity check. **Real → Sim**: latents from an unseen Kinova-real episode transfer to Kinova in simulation, demonstrating sim-real invariance.

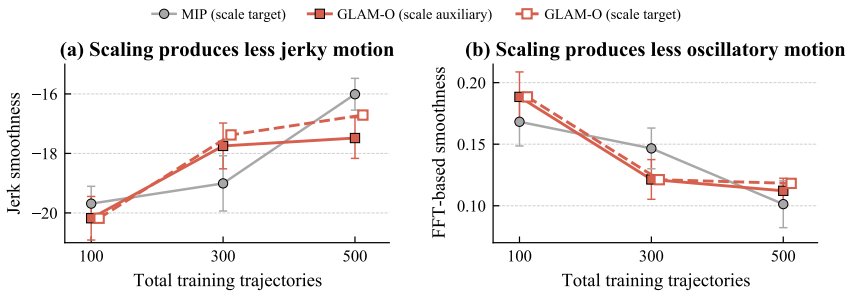

Figure 6: **End-effector motion smoothness across scaling regimes on stack-two**, evaluated on the same 20 unseen initial configurations as Figure 4 (b). **(a)** Jerk-based score [60]; less negative is smoother. **(b)** FFT-based score [61, 62]; lower is smoother.

B Motion Smoothness Corroborates the Scaling Trends

We examine whether the scaling differences in Figure 4 are also visible at the motion level, using two complementary smoothness metrics on the end-effector speed profile $v(t) = \|\dot{\boldsymbol{x}}_t^e\|$: a jerk-based score [60], $-\ln|T^5/v_{\text{peak}}^2 \cdot \int_0^T (d^2v/dt^2)^2\,dt|$ (less negative is smoother); and an FFT-based score [61, 62], $S_m = (2/(nf_s))\sum_i M_i f_i$ (lower is smoother). The two metrics come from disjoint mathematical families, time-domain integration vs. frequency-domain weighting, so concurrent improvement on both helps avoid measurement artifacts.

Figure 6 mirrors Figure 4(b) in three ways. (i) *Scaling-curve shape matches success rate.* MIP's smoothness gain is modest from 100 to 300 trajectories and large from 300 to 500; GLAM-O's is the opposite. Smooth end-effector motion is essential for stable grasping and placement on stack-two, so motion smoothness directly explains the success-rate trends. (ii) *GLAM-O is data-efficient at the motion level too.* The training-scale band over which smoothness sharply improves sits at $300 \to 500$ for MIP but at $100 \to 300$ for GLAM-O, matching the data-efficiency gap reported in Figure 4(b). (iii) *Target and auxiliary scaling are kinematically interchangeable.* The two GLAM-O curves nearly coincide on both metrics, reinforcing the success-rate coincidence in Figure 4(b) and confirming that auxiliary data can substitute for target data as BC supervision under GLAM.

We also inspect the joint-action trajectories to explain the success-rate gap at 300 trajectories in Figure 4(b) and Figure 6. Figure 7 shows that GLAM-O's joint trajectories follow the overall trend of the expert better and settle into final configurations consistent with a successful stack; MIP shows abrupt excursions on joint actions (such as J6, which is essential for aligning the end-effector with the cube) and a chattering gripper that never sustains a stable grasp, leading to failure.

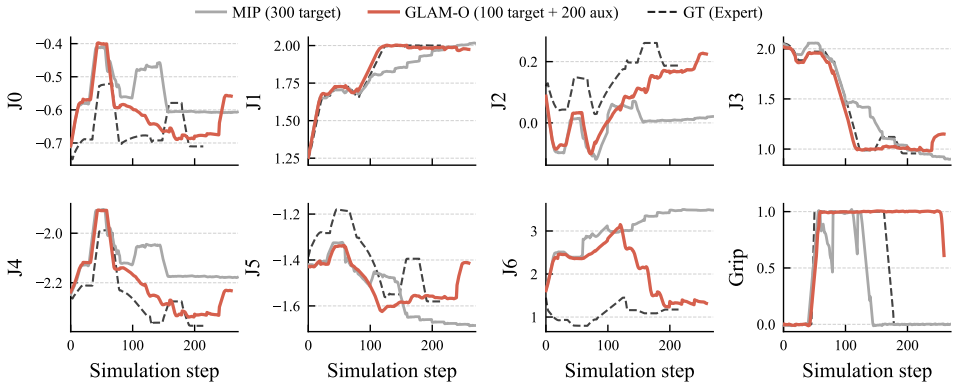

Figure 7: Assessing policy generalisation by comparing online joint space rollouts against a validation expert demonstration. In this episode, MIP fails to stack the two cubes while GLAM-O succeeds, despite neither perfectly reproducing the demonstration. However, the GLAM-based policy fits the unseen demonstration better and is notably smoother. Both policies use 300 total training trajectories, but GLAM-O only uses 100 robot trajectories.

C Co-training with an Additional Data Source

GLAM's source-invariance claim predicts that adding another data source should help, not hurt. We test this by augmenting the 500-trajectory mix with 100 additional Kinova-sim trajectories (Table 1). Co-training improves both pick-and-place (+3) and knock-down (+3), while lifting holds at $18/20$, already near the per-task ceiling; no task degrades. Because GLAM encodes every source into the same shared latent space, the third source enters the same training pipeline without architectural or weighting changes and contributes additional latent-action labels that enrich the downstream policy's supervision. In principle, the same pipeline accommodates further auxiliary sources across embodiments and domains.

Table 1: Adding a third source (100 Kinova-sim trajectories) to GLAM training mix maintains or improves all three real-robot tasks without any architectural changes. (KR = Kinova-real, K = Kinova-sim, U = UMI; *e.g.* KR100+K100+U400 mixes 100 real-robot + 100 sim-robot + 400 UMI trajectories.)

Task	**Two sources** *KR100 + U400*	**Three sources** *KR100 + K100 + U400*	Δ
Lifting	**18/20**	**18/20**	0
Pick & Place	15/20	**18/20**	+3
Knock-down	17/20	**20/20**	+3

D Experimental Setup Details

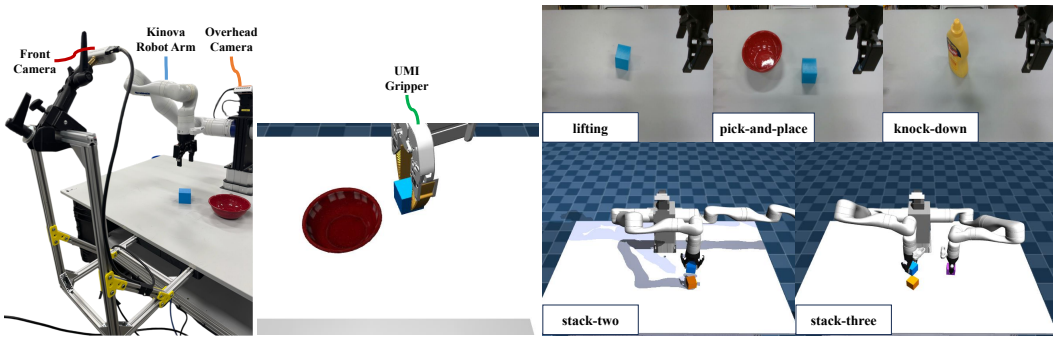

Figure 8: **Hardware setup and per-task visualizations.** Left: the Kinova Gen3 with parallel-jaw gripper and two RealSense cameras (overhead, front). Middle: the UMI gripper in MuJoCo for auxiliary demonstrations collection. Right: example scenes for each of the five tasks.

For the real-world tasks, the target platform is a 7-DoF Kinova Gen3 arm with a parallel-jaw gripper, observed by two Intel RealSense cameras (overhead and front). Target demonstrations are collected via teleoperation, while the auxiliary set is generated by scripted policies in simulation, using a UMI gripper [8] simulated as a free-floating end-effector (Figure 8: middle). The simulated tasks use MuJoCo (Figure 8: right bottom): the single-arm task reuses the Kinova Gen3 model and the bimanual *three-cube-stacking* task adds a second arm in a shared workspace, both serving as the target robot, while the auxiliary set again uses the UMI gripper. We evaluate each real-world task on 20 unseen object placements and each simulated task on 5 unseen test seeds of 10 rollouts each (50 trials per training seed).

E Implementation Details

This section reports module architectures and hyperparameter values for the symbols introduced in Sections 3.1 and 3.2.

Module architectures. Table 2 summarises all modules. The posteriors $q_{\psi}, q_{\phi}, q_{\eta}$ from Section 3.1 correspond to the visual encoder, IDM, and action encoder, respectively; $p_{\psi}(\boldsymbol{o}_t \mid \boldsymbol{x}_t)$ is realised by the visual decoder, and p_{θ} covers the shared forward model, proprio forward, and action decoder. The latent state $\boldsymbol{x}_t$ in Equation (1) is formed by mapping the image part of $\boldsymbol{o}_t$ through q_{ψ} to a visual feature $\boldsymbol{y}_t \in \mathbb{R}^{d_o}$ and concatenating with the end-effector pose $\boldsymbol{x}_e^t$, giving $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{x}_e^t)$; the image is reconstructed from $\boldsymbol{y}_t$ via $p_{\psi}(\boldsymbol{o}_t \mid \boldsymbol{x}_t)$, and we found that KL regularisation on the latent state was not required in practice. The shared forward model $p_{\theta}(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{z}_t)$ uses this same state composition and has a dual head predicting residuals $\Delta\boldsymbol{y}_{t+1}$ and $\Delta\boldsymbol{x}_e^{t+1}$, which are added to $\boldsymbol{x}_t$ to form the next latent state. We replaced the transition-model negative log-likelihood with a deterministic next-state prediction trained by an MSE to simplify the implementation. We additionally sample $\boldsymbol{z}_t \sim q_{\eta}$ during training and pass it through this same forward model, so that both the IDM and action-encoder posteriors drive consistent forward predictions. These two posteriors, q_{ϕ} and q_{η}, share the same Gaussian prior $p(\boldsymbol{z}_t) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, regularising them into the same bounded region of $\boldsymbol{z}$-space.

The two action decoders, $p_{\theta}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ and the policy decoder $\boldsymbol{h}_{\omega}^a$ (Equation (8)), share the same architecture but decode latents from different posteriors: $p_{\theta}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ is grounded on action-encoder latents during world-model pretraining, whereas $\boldsymbol{h}_{\omega}^a$ must decode the IDM latents that relabel the data, so we train it from scratch end-to-end with the policy. We also observed that initialising $\boldsymbol{h}_{\omega}^a$ from the pretrained $p_{\theta}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$ yields comparable results, with the action-reconstruction loss converging quickly in either case.

Table 2: **Module architectures of GLAM and the downstream policy.** "N ResMLP / H" denotes N stacked residual MLP blocks of hidden size H, each block LayerNorm → Linear → GELU → Linear with a residual connection. Conv layers use kernel 4, stride 2, padding 1. The MIP policy follows the design of MIP [3].

Module	Symbol	Backbone	Input → output
GLAM world model (Section 3.1)			
Visual encoder	$q_{\psi}(\boldsymbol{x}_t \mid \boldsymbol{o}_t)$	4 Conv / ReLU	image → $\boldsymbol{y}_t \in \mathbb{R}^{d_o}$
Visual decoder	$p_{\psi}(\boldsymbol{o}_t \mid \boldsymbol{x}_t)$	4 ConvT / ReLU	$\boldsymbol{y}_t$ → image
IDM	$q_{\phi}(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{x}_{t+1})$	5 ResMLP / 256	$(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \to (\mu, \log\sigma^2) \in \mathbb{R}^d$
Action encoder	$q_{\eta}(\boldsymbol{z}_t \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$	5 ResMLP / 256	$(\boldsymbol{s}_t, \boldsymbol{a}_t) \to (\mu, \log\sigma^2) \in \mathbb{R}^d$
Forward model	$p_{\theta}(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{z}_t)$	4 ResMLP / 256, dual head	$(\boldsymbol{x}_t, \boldsymbol{z}_t) \to \Delta\boldsymbol{y}_{t+1}$, $\Delta\boldsymbol{x}_e^{t+1}$
Proprio forward	$p_{\theta}(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{z}_t)$	5 ResMLP / 256	$(\boldsymbol{s}_t, \boldsymbol{z}_t) \to \Delta\boldsymbol{s}_{t+1}$
Action decoder	$p_{\theta}(\boldsymbol{a}_t \mid \boldsymbol{z}_t)$	4 MLP / 256, ReLU	$\boldsymbol{z}_t \to \boldsymbol{a}_t$
Downstream latent-action policy (Equation (8))			
Vision encoder	—	DINOv2-S/14 + MLP	Per view DINOv2-S/14 (CLS, 384-d), jointly trained; 2-view concat → Linear(768→256)-LN-GELU-Linear(256→64)-LN
MIP policy	$\pi_{\omega}(\hat{\boldsymbol{z}}_{t:t+H} \mid \boldsymbol{c}_t)$	10 MLP / 256, LN+Mish	vision (64-d) $\oplus$ $\boldsymbol{s}_t$ → $\hat{\boldsymbol{z}}_{t:t+H} \in \mathbb{R}^{H \cdot d}$
Action decoder	$\boldsymbol{h}_{\omega}^a$	4 MLP / 256, ReLU	$\hat{\boldsymbol{z}}_{t:t+H} \to \hat{\boldsymbol{a}}_{t:t+H}$

Hyperparameter values. Table 3 lists all hyperparameters. The latent-action dimension d is the only quantity that varies by task. The KL-to-prior weights on the IDM and action encoder posteriors in Equations (3) and (4) are both set to 10^{-3}, which prevents posterior collapse and preserves task-relevant information in the latent action.

Table 3: **Hyperparameter values for the symbols introduced in Sections 3.1 and 3.2.**

Symbol	Meaning	Value
d	Latent action dim	8; 16 for stack-three
d_o	Visual feature dim	128
H	Policy chunk size	20
λ	Alignment KL weight (Equation (6))	1
$-$	KL-to-prior weight in ELBOs (Equations (3) and (4))	10^{-3}