VISION-LANGUAGE MODELS PROVIDE TASK-CENTRIC PROMPTABLE REPRESENTATIONS FOR LATENT ACTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent Action Models (LAMs) have rapidly emerged as a promising approach for addressing the scarcity of high-quality action-labeled data. However, they fail when observations contain action-correlated distractors, often encoding noise instead of meaningful latent actions. Humans, on the other hand, can effortlessly distinguish task-relevant motions from irrelevant details in any video given only a brief task description. In this work, we propose to utilize the common-sense reasoning abilities of Vision-Language Models (VLMs) to provide promptable representations, effectively separating controllable changes from the noise. We use these representations as targets during LAM training and benchmark a wide variety of popular VLMs, revealing substantial variation in the quality of promptable representations as well as their robustness to different prompts and hyperparameters. Interestingly, we find that newer VLMs do not necessarily outperform older ones; however, all evaluated VLMs consistently outperform CLIP and DINOv2, highlighting the vital importance of language conditioning. Finally, we show that simply asking VLMs to ignore distractors can substantially improve latent action quality and downstream performance on Distracting MetaWorld.

1 INTRODUCTION

Latent Action Models (LAMs) (Schmidt & Jiang, 2023; Ye et al., 2024) have rapidly emerged as a promising approach for addressing the scarcity of high-quality action-labeled data (McCarthy et al., 2025). Unfortunately, as shown by Nikulin et al. (2025); Zhang et al. (2025); Klepach et al. (2025), without explicit supervision, LAMs struggle to disentangle controllable changes from noise, completely failing to produce meaningful latent actions in the presence of action-correlated distractors. Humans, however, interpret the world through semantics rather than raw pixels, and with only a brief task description can easily separate task-relevant features from irrelevant details. Wouldn't it also be convenient to simply ask LAM to focus on the task-centric features and ignore any other details?

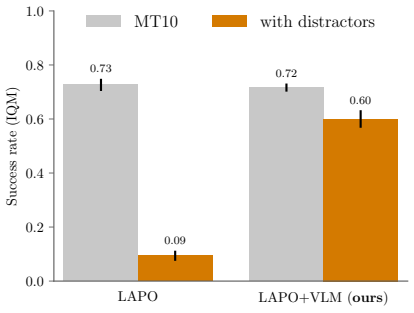


Figure 1: Success rate on MetaWorld-10 benchmark for LAPO and LAPO+Molmo, which uses promptable representations. We use three random seeds and report IQM and 95%-CI, following the Agarwal et al. (2021).

Inspired by the work of Chen et al. (2024a) on promptable representations, we propose to utilize the Vision-Language Models (VLMs) common-sense reasoning abilities as an unsupervised approach for effectively separating controllable changes from noise, thereby restoring the LAM's ability to recover ground-truth actions even in the presence of distractors. As a motivation for our approach, we begin from a simple demonstration experiment, showing that limitations of LAM can be mitigated with the clean target (Section 4). Using a controlled Distracting MetaWorld (Yu et al., 2020) setup, we then conduct large-scale benchmarking of different VLMs, comprising over 30k+ experiments, to assess their effectiveness at providing promptable representations (Section 5), revealing substantial variation in quality and robustness to hyperparameters. In addition, we show that VLMs significantly outperform self-supervised methods,

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

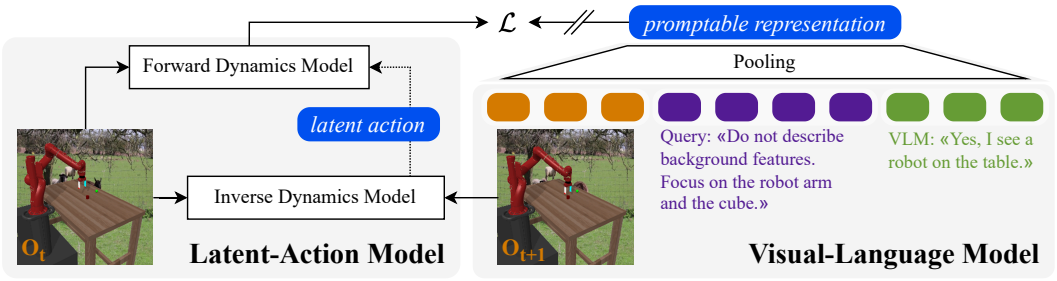


Figure 2: Task-centric promptable representations extraction from the VLMs and their subsequent use as targets during latent action learning. For pooling, we use simple fixed strategies, such as taking the mean over all output embeddings or taking only the embedding of the last token from either prompt or the generated answer.

such as DINOv2 (Oquab et al., 2023) or CLIP (Radford et al., 2021), confirming our hypothesis that language based filtering is essential for performance. Finally, using the best setup found, we demonstrate that, without any supervision, promptable representations can significantly improve latent action quality and downstream performance (Section 6), outperforming OTTER (Huang et al., 2025) and UniVLA (Bu et al., 2025) baselines, while increasing the success rate six-fold (see Figure 1).

2 Preliminaries

Problem setting. We consider a setting of imitation learning from observation (Liu et al., 2018; Torabi et al., 2019). Our goal is to pre-train a policy $\pi(a|o)$, given a large dataset of expert trajectories $\mathcal{D} \coloneqq \{(o_i^n)\}_{i=1}^{\tau}$, containing observations but not actions (e.g. videos), and a limited number of real action labels. Ideally, the pre-trained agent should achieve maximum performance (e.g. success rate) in the environment after fine-tuning with a minimum amount of action-labeled data. The commonly considered ratio of labeled to unlabeled data is around $2-10\%$ in the existing work (Zheng et al., 2023; Nikulin et al., 2025), while in our experiments, we consider a ratio as low as $< 1\%$.

Promptable representations. We follow the Chen et al. (2024a) and define promptable representations *simply as a process of obtaining observation embeddings from the VLMs given a task-specific prompt and some extraction and aggregation strategy* (see Figure 2). We obtain such representations from the last and next-to-last layers (Chen et al., 2024a). In contrast to the Chen et al. (2024a); Huang et al. (2025) we cannot learn pooling from the data to better predict true actions or obtain better reward. Thus, we experiment only with simple fixed strategies, such as taking the mean over all output embeddings or taking only the embedding of the last token from either prompt or the generated answer. Thus, the final promptable representations of an observation o_t is always just a single vector $s_t \in R^D$, where D is a VLMs hidden size.

Latent action models. Given the dataset of observations $\mathcal{D} \coloneqq \{(o_i^n)\}_{i=1}^{\tau}$, LAMs (Rybkin et al., 2018; Edwards et al., 2019; Schmidt & Jiang, 2023) try to infer latent actions z_t such that they are maximally predictive of observed transitions (o_t, o_{t+1}) while being minimal (Schmidt & Jiang, 2023), i.e. describe changes only relevant to control. After pre-training, LAM is used to supply latent actions for behavioral cloning (BC) on unlabeled dataset to obtain useful behavioral priors. As a final stage, decoder is trained to map from latent to ground-truth actions on a small labeled dataset. Modern LAMs (Bruce et al., 2024; Ye et al., 2024; Cui et al., 2024; Chen et al., 2024c;b; Gao et al., 2025) mostly follow the same high-level architecture introduced by LAPO (Schmidt & Jiang, 2023), which uses a combination of inverse (IDM) and forward (FDM) dynamics models to infer latent actions. Given a transition (o_t, o_{t+1}), IDM first infers latent action $z_t \sim f_{\text{IDM}}(\cdot|o_t, o_{t+1})$, which FDM further uses to predict the next observation $\hat{o}_{t+1} \sim f_{\text{FDM}}(\cdot|o_t, z_t)$. Both models are trained jointly to minimize the loss $\mathcal{L}_{\text{MSE}} = \mathbb{E}_{(o_t, o_{t+1})\sim\mathcal{D}}\left[\|f_{\text{FDM}}(f_{\text{IDM}}(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}), \boldsymbol{o}_t) - \boldsymbol{o}_{t+1}\|^2\right]$.

3 Benchmarking Setup

Latent action model. We use the architecture proposed by Schmidt & Jiang (2023), omitting action quantization (Nikulin et al., 2025; Liang et al., 2025; Yang et al., 2025b). We use frame stacking, but only in IDM, while FDM uses only the current frame and latent action to predict the next frame, as

in Chen et al. (2024b). Other than that, in our main experiments, we do not use any improvements upon LAPO (unless explicitly stated otherwise), such as augmentations or multi-step predictions in FDM (Nikulin et al., 2025; Chen et al., 2024b; Ye et al., 2024; Cui et al., 2024), to eliminate possible confounders on latent action quality. We use ResNet (He et al., 2016) or ViT (Dosovitskiy, 2020) as IDM and FDM backbones. When predicting in the latent space instead of images, we follow Nikulin et al. (2025) and use multiple MLP blocks similar to those used in Transformers (Vaswani et al., 2017). For action decoder head, we use a small three-layer MLP. For actor itself, we always used ResNet (He et al., 2016). We follow the standard three-stage pipeline (Schmidt & Jiang, 2023; Ye et al., 2024; Nikulin et al., 2025): (1) pre-train the LAM, (2) train behavioral cloning (BC) agent on latent actions, and (3) train a decoder head on a small number of true-action labels. For hyperparameters see Section G in the Appendix.

Environments. We use MetaWorld (Yu et al., 2020) Multi-Task 10 (MT10) suite as our benchmark, as it allows us to conduct experiments in controlled setting while being lightweight enough for exhaustive experimentation with VLMs under limited resources. As we will demonstrate in Section 5, while MetaWorld is simple, adding distractors almost completely breaks LAMs and makes it possible to distinguish between different VLMs, judging by the quality of promptable representations. We modify MetaWorld to include distracting videos in the background, using the DAVIS (Pont-Tuset et al., 2017) videos, which depict different real world scenes, including humans. We also move the default camera position farther back and remove borders around the table to include more of the background video in the observation, making latent action learning more challenging. See Figure 3 for a visualization.

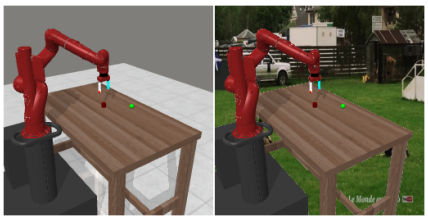


Figure 3: Visualization of observations with and without distractors in our modification of MetaWorld environment.

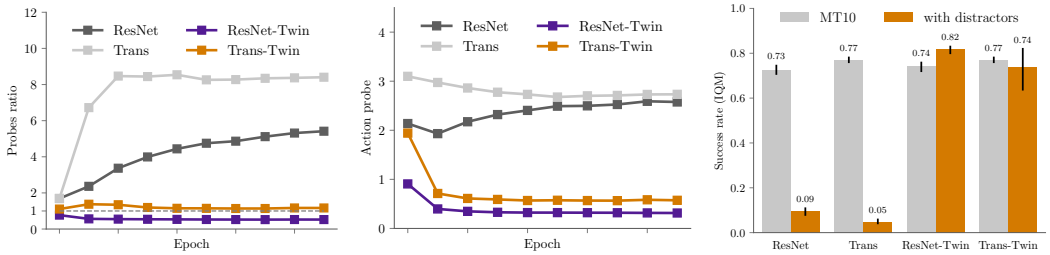
Datasets. For each task, we collect 5k trajectories from the scripted experts provided by MetaWorld and up to 16 additional action-labeled trajectories for the final stage, which is less than 1% of the full datasets. During reset, we sample a random video uniformly, as well as an initial direction (forward or reverse). We then advance the video frame by frame at each step. Most videos are shorter than environment episodes, so we simply reverse the playback direction and replay them. In addition, we render and save same observation with and without distractors. For validation, we use videos from the DAVIS validation set. We will release the code with environment modifications and datasets collection.

Evaluation. For evaluation we use **action probe** and **success rate** (Nikulin et al., 2025). Specifically, we train linear probes to predict real actions from the latent ones during LAM training, while stopping the gradient through the latent actions. The final MSE serves as our quality metric, as it indicates whether the latent actions encode the real ones. However, as Nikulin et al. (2025) notes, linear probing has a key limitation: it can reveal whether true actions are present in the latent space, but it does not guarantee minimality, meaning that noise may still be encoded. Thus, to measure the true usefulness of pretraining with latent actions, we evaluate the success rate in the environments after fine-tuning on true action labels.

4 The Importance of the Right Target

We begin with a simple demonstration that the limitation of LAMs in the presence of distractors arises entirely from the poor FDM target (e.g. next frame in pixel space), rather than from flaws in the overall idea or architecture. By construction, latent actions are optimized to maximally explain the dynamics. Therefore, the root of the failure to recover true actions lies in the dynamics we predict, which is directly determined by the target in FDM: $\hat{o}_{t+1} \sim f_{\text{FDM}}(\cdot|o_t, z_t)$. What would be the ideal target for FDM? And if it exists, what would be the final performance? Could LAM possibly recover the ground-truth actions despite distractors in the input observations?

To answer these questions we feed observations with distractors as inputs to IDM and FDM, but as the target for FDM we use next observation without distractors. As the actual controllable changes are preserved (the underlying state is the true next state), it serves as a target with ideal disentanglement of controllable features from the noise (see Figure 3). To show that existing limitations are agnostic



(a) w/ to w/o distractors probe ratio (b) Probes w/ distractors (c) Success rate on MT10

Figure 4: Demonstration that quality of latent actions learned by LAPO completely degrades in the presence of distractors, which results in almost zero success rate. We show that with the ideal target for FDM, which perfectly disentangles controllable features from the noise, performance may be restored, serving as a main motivation for us to explore promptable representations. Action probe represents MSE of a linear probe trained to predict real actions from latent actions. See Section 3 for detailed explanation. We use three random seeds and report IQM and 95%-CI, following the Agarwal et al. (2021). See Section 4 for details.

to the architecture, we explore both ResNet (Schmidt & Jiang, 2023) and transformer (Bruce et al., 2024; Ye et al., 2024) backbones for IDM and FDM.

Results. First of all, as can be seen in Figure 5, we confirm that simply adding distractors results in complete degradation of latent actions quality. This subsequently leads to almost zero success rate after fine-tuning on true actions (see Figure 4c). Next, in Figures 4a and 4b we show the effect of using perfect targets during LAPO training (with -Twin postfix). To better illustrate the trend, in Figure 4a we report the ratio of probes with and without distractors for each method: $\frac{\texttt{mse_clean}}{\texttt{mse_distractors}}$. With the ideal target action probes immediately drop to the level of LAPO without distractors and the ratio converges to one. Finally, improvements in latent action quality directly result in rising of success rates (see Figure 4c). Overall, this result supports the hypothesis that the right target is the key to unlock latent action learning in the presence of distractors.

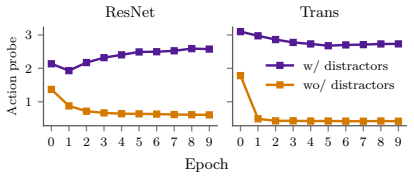


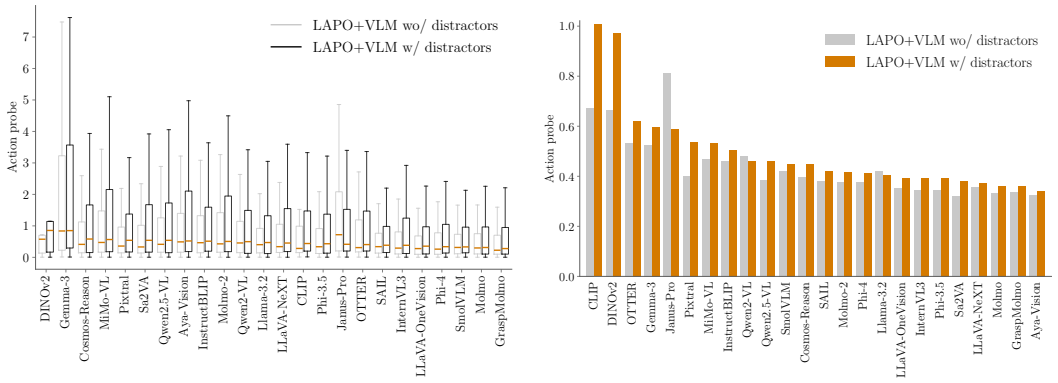
Figure 5: Baseline LAPO action probes on MT10. Averaged over 3 random seeds. Action probe represents MSE of a linear probe trained to predict real actions from latent actions. See Section 3 for detailed explanation.

Although these experiments may seem obvious in hindsight, they allow us to convey a key empirical observation, which supports the intuition that originally led us to explore promptable representations. Our main hypothesis is that representations provided by VLMs can serve as an effective and fully unsupervised way to disentangle controllable features from the noise.

5 The Promise of Promptable Representations

Although modern VLMs can identify and describe the robotic arm in images (like Figure 3) even noise on the background, valid text output does not guarantee their embeddings are suitable for LAMs. An effective LAM target representation should (1) contain task-centric visual information, (2) be minimal by filtering out visual details irrelevant to the prompt, and (3) remain consistent across observations to represent changes caused by real actions. Unfortunately, current VLMs are known to struggle with visual focus (Rahmanzadehgervi et al., 2024; Sim et al., 2025) and pixel-level understanding (Gou et al., 2024; Dahou et al., 2025; Li et al., 2025). Given these limitations, we begin by benchmarking a wide variety of modern VLMs to assess their viability. As a baseline, we included representations from self-supervised methods such as CLIP (Radford et al., 2021) and DINOv2 (Oquab et al., 2023), which are not promptable but were pre-trained on large amounts of visual data and used by recent LAMs (Cui et al., 2024; Bu et al., 2025; Lee et al., 2026).

Setup. Conducting large scale VLMs evaluation on the full datasets would be a prohibitively expensive task. Instead, we train LAPO+VLM on small subsets of 64 trajectories that are taken from the full 5k trajectories dataset for each task. Then, we measure the resulting latent action quality with the help of MSE probes. As we will show in Section 6, hyperparameter rankings obtained this way



(a) Aggregation over all hyperparameters explored

(b) Aggregation over the best hyperparameters

Figure 6: Benchmarking the effectiveness of promptable representations provided by different VLMs for latent action learning on all tasks from MT10. Action probe represents MSE of a linear probe trained to predict real actions from latent actions. See Section 3 for detailed explanation. Overall, all VLMs provide some improvement over LAPO, with Molmo performing the best and Gemma-3 the worst. For details and exact experimental protocol see Section 5. We additionally provide the ranking for each combination of hyperparameters in the Figure 18 in the Section E.

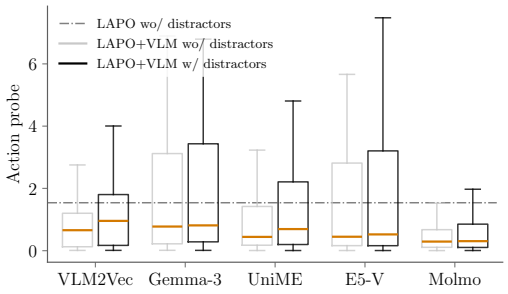


Figure 7: Benchmarking the effectiveness of promptable representations provided by recent *embedding* VLMs for latent action learning on all tasks from MT10. Overall, embedding VLMs, despite their promise, do not deliver any substantial gains compared to traditional VLMs, such as Molmo. We include Gemma-3 and Molmo results here for convineince.

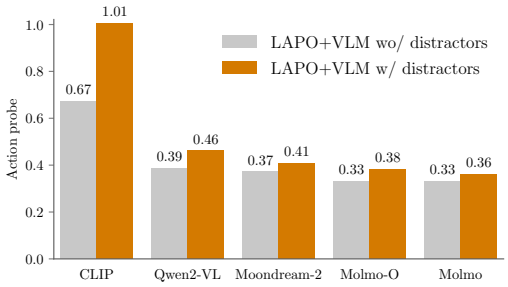


Figure 8: Molmo performance investigation, using aggregation over the best hyperparameters. We benchmark both Molmo versions, which both use CLIP but differ in LLM backbones (OLMo vs Qwen2), as well as Qwen2-VL. Since both Molmo share the pre-training data but differ in architecture, we conclude that the likely source of their superior performance lies in the data.

transfer reasonably well to the full dataset performance. For each VLM, we evaluated eight prompt variants designed in different styles to exploit diverse VLM capabilities (e.g., CLIP-style captions, pointing, segmentation; see Table 6 in Section F). We further varied the source of representations (last vs. next-to-last layer, prompt vs. generated embeddings) and the aggregation strategy (averaging vs. last non-padding token). This yields 64 runs per VLM, per task, per dataset. The full list of VLMs, including exact model names, sizes, and prompt templates, is provided in Section F.

Benchmarking VLMs. We summarize our main benchmarking results in Figure 6 and provide full per-hyperparameter rankings in Figure 18 (Section E). As it can be seen in Figure 6a, in general every VLM provides some degree of improvement over LAPO in terms of the median action probe. However, some of them, especially Molmo (Deitke et al., 2025), are better in terms of lower variance, indicating higher robustness to different sets of hyperparameters. In Figure 6b we visualize ranking by averaging best scores for each task. While this changes ranking a bit, we still observe the following trend among VLMs: Gemma-3 (Team et al., 2025) is the worst and Molmo (Deitke et al., 2025) is consistently the best. Based on Figure 18, we conclude that in general, promptable representations aggregated by averaging next-to-last layer in prompt embeddings perform the best. From a practical standpoint, this is beneficial, as it eliminates the additional overhead for answer generation and requires only one forward pass.

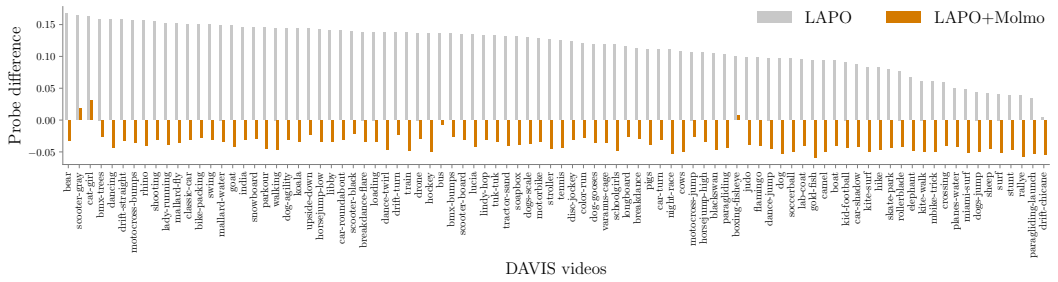


Figure 9: Differences between action probes of LAPO and LAPO+Molmo with distractors and LAPO probes without distractors. We use the LAPO action probe without distractors as a baseline and compute the difference between the probes. Values below zero mean that LAPO+Molmo with distractors outperforms LAPO without distractors in terms of the latent action quality.

Ironically, our experiments showed that the best prompt is *Do not describe background features. Focus on the robot arm and the [task-obj]*, which explicitly asks VLM to filter out distractors. Moreover, Table 1 shows that for LAPO+Molmo, the quality of latent actions strongly depends on how task-centric the prompt is.

This brings us to a striking conclusion that state-of-the-art VLMs do not necessarily provide better promptable representations. For example, InstructBLIP (Dai et al., 2023) outperforms both Gemma-3 (Team et al., 2025) and Pixtral (Agrawal et al., 2024), despite being considerably older. Furthermore, Cosmos-Reason (Azzolini et al., 2025) results indicate that explicit fine-tuning on robotics data is not sufficient to guarantee improved representations. We believe that our results, besides relevance to LAMs, reveal a large blind spot in how VLMs are currently evaluated, which has critical implications for current robotics and VLA research.

In addition, DINOv2 and CLIP (see Figure 6) results highlight the vital importance of language conditioning. Due to large-scale pretraining both may possess helpful inductive biases, for example by attending to moving objects, but without language conditioning there is no guarantee that these objects are the ones that could be controllable by an agent. Both methods achieve the worst latent action quality among all approaches we considered. For instance, OTTER (Huang et al., 2025) uses the same CLIP model but applies simple training-free filtering using text CLIP embeddings. This small modification significantly improves latent action quality, although it is still not enough to beat the level of VLMs, such as Molmo.

Table 1: LAPO+Molmo median action probes on MT10 with and without distractors for prompts with different task-centricity. All hyperparameters except prompt are fixed. We used next-to-last layer prompt embeddings. Lower is better.

Prompt	Without	With
Point to the robotic arm gripper.	0.21	0.32
Do you see a robot?	0.49	0.87
What do you see in the foreground?	0.56	0.93
What do you see in the background?	0.45	1.20
What is the perfect morning routine?	1.53	1.62

Benchmarking prompt robustness. To disentangle sensitivity to exact wording from sensitivity to semantic focus, we vary only the prompt while fixing Molmo and the representation-extraction configuration. As shown in Table 2, a meaning-preserving paraphrase remains close to the canonical object-agnostic prompt, while performance progressively degrades as prompts become less aligned with controllable features. Exact task-object names are therefore not required, but semantic alignment remains important.

Benchmarking embedding VLMs. In our main benchmark (see Figure 6), we evaluated conventional VLMs, which were not explicitly trained to produce strong unified representations and therefore required heuristics such as embedding pooling. Recently, a new class of embedding VLMs has emerged (Jiang et al., 2024b; Meng et al., 2025). These models are designed specifically to learn high-quality, promptable, and multimodal embeddings for zero-shot retrieval. Given the similarity of their objective to ours, one might expect them to perform better. To test this, we evaluated three recent state-of-the-art models (Jiang et al., 2024a; Meng et al., 2025; Gu et al., 2025) using the same protocol as earlier, but separately as they require different prompt formats. As can be seen in Figure 7, such models do not deliver any substantial gains. In fact, VLM2Vec-V2 (Meng et al., 2025), which is considered one of the best model in its class, performed worse than Gemma-3, which was the weakest model in the main benchmark, and none of the models surpassed Molmo. Our results

indicate that embedding VLMs do not actually encode only prompt-specific visual information into the embeddings and fail to deliver the anticipated benefits for our use case.

Why does Molmo perform so well? Given Molmo's strong performance, it is natural to ask what drives its improved representations. Directly answering this question is difficult, but we can gather indirect evidence suggesting that the gains stem primarily from pre-training data rather than from the specific architecture. Fortunately, Molmo-team provide two variants of their model: one is Molmo-D, which uses Qwen2 as its backbone (Yang et al., 2025a), and another is Molmo-O, which uses OLMo (Groeneveld et al., 2024). They both employ CLIP (Radford et al., 2021) as the vision encoder. In contrast, Qwen2-VL (Wang et al., 2024) does not use CLIP, offering a useful comparison point to disentangle architectural effects. In Figure 8 we show that CLIP alone performs the worst, Molmo-O ranks second after Molmo-D, and Qwen2-VL still performs the lowest among VLMs. Since the Molmo variants share the same pre-training data but differ in backbone architecture, we conclude that the likely source of their superior performance lies in the data rather than the architecture.

Table 2: Prompt perturbation study on MT10. Molmo and the representation-extraction configuration are fixed; only the prompt is varied. Mean action-probe MSE across tasks; lower is better.

Category	Prompt	MSE ↓
Object-agnostic	Point to the robotic arm gripper.	**0.64**
Paraphrase	Indicate where the robotic arm gripper is.	0.78
High-level task	The robot's task is to [task]. Focus on the visual information needed to complete the task.	0.91
Correct task object	Do not describe background features. Focus on the robot arm and [task_object].	1.08
Relevant objects	Do not describe background features. Focus on the robot arm and the objects relevant to the task.	1.19
Incorrect task object	Do not describe background features. Focus on the robot arm and [wrong_object].	1.31
Background-directed	Describe the objects and motion in the background.	1.60

The challenge of camera motion. To further analyze the benefits of promptable representations, we focused on pick-place-v2 and collected 32 trajectories per DAVIS video using the procedure from Section 3, with each dataset using a single video as the distractor. First, we assessed the difficulty of each video, where difficulty is measured by the extent to which action probe quality degrades in the presence of distractors. After a brief analysis of the video content, we formed the hypothesis that difficulty for LAPO is mostly related to camera motion, as the easiest videos (e.g., sheep, drift-chicane, etc.) mostly depict a static camera, whereas the most difficult ones (e.g., scooter-gray, dancing, etc.) depict highly dynamic scenes. We measured camera motion using a simple proxy: we computed dense optical flow between frames[1], took the median flow magnitude, and averaged it across each video (see Figure 10). The correlation coefficient for LAPO is around 0.4, which is usually considered weak but still indicative of an existing linear association. For LAPO+Molmo, the coefficient is only ~ 0.2, which is below the threshold at which a linear relationship can typically be assumed. This simple analysis suggests that, unlike LAPO, LAPO+Molmo can handle the camera movement well.

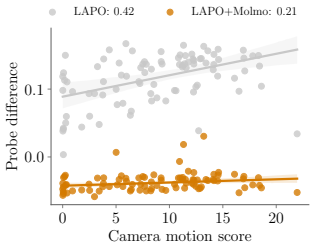


Figure 10: Differences between action probes of LAPO and LAPO+Molmo with distractors and LAPO without distractors, compared to the camera motion estimate for each DAVIS video.

6 Promptable Representations Unlock Task-Centric Latent Actions

Setup. Based on the benchmark results, we selected multiple VLMs for further experiments: Phi-4, Molmo and GraspMolmo. Although all of them achieved improvements in latent action quality upon LAPO on small datasets, it remains necessary to validate whether this performance transfers to the full 5k datasets and yields improved success rates, as this is not guaranteed (Nikulin et al., 2025). We chose the best hyperparameters for each VLM and trained LAPO+VLM on the full datasets, using three random seeds. As specified in Section 3 we used 16 labeled trajectories with ground-truth actions for final fine-tuning. See Section G for hyperparameters.

[1]Using cv2.calcOpticalFlowFarneback from OpenCV library (Bradski, 2000).

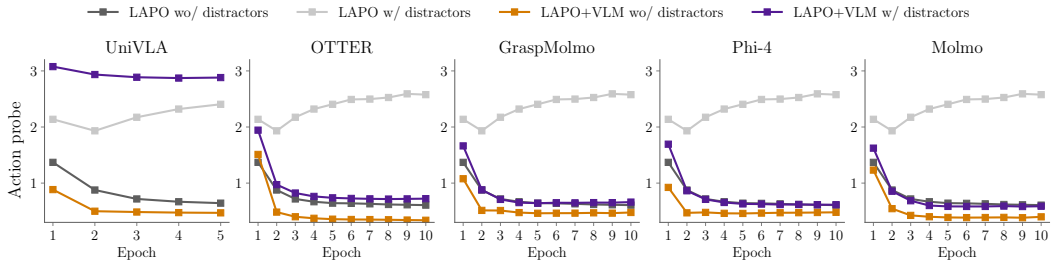


Figure 11: Action probes comparison for LAPO, baselines and LAPO with promptable representations on full datasets for all tasks in MT10. Results are averaged over three random seeds. Promptable representations significantly improves upon LAPO in terms of the latent actions quality, and without any supervision with true actions closes the gap with LAPO without distractors. While all VLMs bring improvements, Molmo achieve best results overall, especially given it high robustness to hyperparameter choices (see Figure 6a). UniVLA uses two-stage pipeline, each half of total epochs and we visualize the second stage, where task-centric latent actions should be learned. For resulting success rates see Figure 12. We provide per-environment probes in Figure 17, Section E.

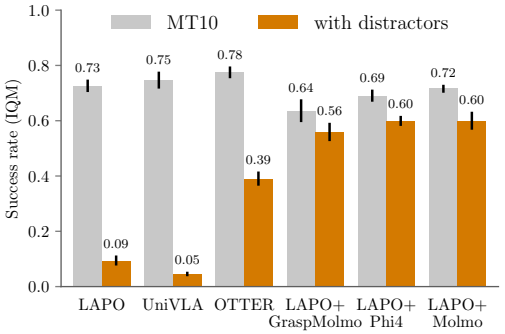


Figure 12: Success rate on MT10 for LAPO and LAPO+VLMs, which uses promptable representations. We use three random seeds and report IQM and 95%-CI, following the Agarwal et al. (2021).

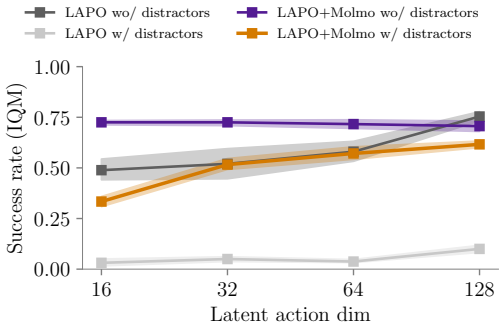


Figure 13: Success rate on MT10 for LAPO and LAPO+Molmo, with varying latent action dimension to control the information bottleneck. We use three random seeds and report IQM and 95%-CI, following the Agarwal et al. (2021).

Baselines. As our primary baselines, we additionally evaluated OTTER (Huang et al., 2025) and UniVLA (Bu et al., 2025). We note, however, that we are not competing with the VLAs proposed in these works. For UniVLA, we re-implemented their proposed two-stage task-centric LAM training pipeline using our transformer backbone and task instructions provided by MetaWorld (Yu et al., 2020), which we encoded by a pretrained T5 text encoder (Raffel et al., 2020). For OTTER, we adopted their CLIP training-free text-aware visual feature extraction approach to construct a target for FDM in LAPO. All stages other than latent action learning are unified (e.g. share BC actor architecture) across all considered methods to ensure a fair comparison.

Results. We present the resulting action probes in Figure 11 and the final success rates after fine-tuning on real actions in Figure 12. As shown in Figure 11, LAPO with promptable representations achieves a substantial improvement in latent action quality, both with and without distractors. In the presence of distractors, it nearly closes the gap to LAPO trained without distractors, and without distractors, it slightly outperforms it (e.g. Molmo).

Among the baselines, only UniVLA does not show any improvement in the presence of distractors. As discussed in **??**, UniVLA adds a language task instruction embedding to the IDM and FDM inputs, which may help disentangle task-relevant videos from noise at a high level but does not provide the per-step learning signal required to accurately recover low-level ground-truth actions. As a result, during single-task learning the task instruction remains a constant vector, and UniVLA effectively reduces to LAPO.

Crucially, the improvement in action probes on full datasets carries over to downstream performance (see Figure 12): success rates increase by a factor of six at max in the presence of distractors, while remaining almost unchanged without them. Interestingly, we found Phi-4 to outperform GraspMolmo,

despite having worse probes on small datasets. On full datasets (Figure 11), however, Phi-4 is better. This indicates, that while results from a small dataset may carry over to a larger one with some error, probes on the full dataset predict the final success rate with high accuracy. We provide additional experiments with change in lighting as an additional distractor in Section C. Overall, our results confirm the viability of promptable representations as a clean target for latent action modeling in the presence of distractors.

Single set of hyperparameters transfers robustly across tasks. Note that our main results use the best hyperparameters, which may be difficult to select without ground-truth actions in real-world settings. To further assess the robustness of promptable representations, we rerun the LAPO+Molmo experiments across 3 random seeds using a single fixed, task-agnostic hyperparameter setting. We use *Point to the robotic arm gripper* as the prompt and take the mean of the penultimate-layer prompt embeddings as the final representation (see Table 3). As shown, LAPO+Molmo remains stable, only slightly underperforming the best-hyperparameter setting (0.55 vs 0.60 IQM success rate; see Figure 12), while still substantially outperforming LAPO with distractors, even at larger action-label budgets.

Table 3: Success rates on MT10 for LAPO+Molmo with one shared config and varying budgets of labeled trajectories. We report success rate IQM and 95%-CI following the Agarwal et al. (2021).

# Ep.	No distractors		With distractors	
	LAPO	+Molmo	LAPO	+Molmo
16	0.75 (0.74, 0.77)	0.67 (0.65, 0.69)	0.10 (0.08, 0.11)	**0.55** (0.53, 0.56)
32	0.79 (0.77, 0.81)	0.78 (0.76, 0.79)	0.11 (0.10, 0.13)	**0.66** (0.65, 0.68)
64	0.83 (0.78, 0.88)	0.86 (0.84, 0.87)	0.14 (0.12, 0.17)	**0.75** (0.70, 0.78)
128	0.87 (0.78, 0.94)	0.93 (0.92, 0.94)	0.17 (0.13, 0.20)	**0.84** (0.75, 0.89)
256	0.97 (0.95, 0.98)	0.96 (0.95, 0.97)	0.28 (0.26, 0.30)	**0.93** (0.91, 0.94)

Varying the latent action information bottleneck. In all previous experiments, we fixed the latent action dimension to 128 to ensure a consistent level of latent action minimality across all methods. However, this choice was inherited from LAPO and may not be optimal. Moreover, as shown by LAPA (Ye et al., 2024), smaller latent action spaces make the perception-and-language-to-action generation problem significantly easier to learn. We therefore conducted additional experiments that varied the latent action dimension, using the full datasets and three random seeds. The results are summarized in Figure 13. We observe that promptable representations not only increase success rates in the presence of distractors, but also significantly outperform LAPO without distractors under stronger minimality constraints, such as 16 action dimensions. These results provide further evidence that VLMs help filter out information irrelevant to controllable changes, enabling a more compact latent action space.

Preliminary real-world validation. We additionally evaluate the best fixed LAPO+Molmo configuration on 100 real-world DROID episodes (`lerobot/droid_100` from HuggingFace). Table 4 shows that promptable representations improve the 10-epoch action probe at every tested latent-action dimension, with the largest gain under the strongest bottleneck. While this does not replace downstream real-robot evaluation, it provides preliminary evidence that the effect persists with real textures, lighting, camera motion, robot appearance, and background activity.

Table 4: Action-probe MSE on 100 real-world DROID episodes using the same fixed best LAPO+Molmo configuration across latent-action dimensions. Results are reported after 10 epochs; lower is better.

Latent action dim	LAPO	LAPO+Molmo
8	0.131	**0.067**
16	0.080	**0.061**
32	0.080	**0.062**
64	0.074	**0.069**

7 Conclusion

In this work, we demonstrated that promptable representations provided by Vision-Language Models can effectively filter out action-correlated distractors, enabling task-centric latent actions. Our experiments on the Distracting MetaWorld benchmark confirmed that using task-centric promptable representations as targets for LAPO substantially improves both latent action quality and downstream success rates. We hope that our results will inspire the community to explore promptable representations at scale. We discuss the related work in Section A and limitations in the Section B.

AI USE STATEMENT

In this work, we used generative AI tools for final text editing, polishing and grammar checking. We have not used generative AI tools for experiments, idea generation, inital text writing and image generation. 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.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions, 2025. URL https://arxiv.org/abs/2505.06111.

William Chen, Oier Mees, Aviral Kumar, and Sergey Levine. Vision-language models provide promptable representations for reinforcement learning. *arXiv preprint arXiv:2402.02651*, 2024a.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024b.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 2024c.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *arXiv preprint arXiv:2409.12192*, 2024.

Yasser Dahou, Ngoc Dung Huynh, Phuc H Le-Khac, Wamiq Reyaz Para, Ankit Singh, and Sanath Narayan. Vision-language models can't see the obvious. *arXiv preprint arXiv:2507.04741*, 2025.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 91–104, 2025.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International conference on machine learning*, pp. 1755–1763. PMLR, 2019.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Chenhui Gou, Abdulwahab Felemban, Faizan Farooq Khan, Deyao Zhu, Jianfei Cai, Hamid Rezatofighi, and Mohamed Elhoseiny. How well can vision language models see image details? *arXiv preprint arXiv:2408.03940*, 2024.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Tiancheng Gu, Kaicheng Yang, Ziyong Feng, Xingjun Wang, Yanzhao Zhang, Dingkun Long, Yingda Chen, Weidong Cai, and Jiankang Deng. Breaking the modality barrier: Universal embedding learning with multimodal llms. *arXiv preprint arXiv:2504.17432*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024a.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024b.

Albina Klepach, Alexander Nikulin, Ilya Zisman, Denis Tarasov, Alexander Derevyagin, Andrei Polubarov, Nikita Lyubaykin, and Vladislav Kurenkov. Object-centric latent action learning. *arXiv preprint arXiv:2502.09680*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. Mvp-lam: Learning action-centric latent action via cross-viewpoint reconstruction. *arXiv preprint arXiv:2602.03668*, 2026.

Wenyan Li, Raphael Tang, Chengzu Li, Caiqi Zhang, Ivan Vulić, and Anders Søgaard. Lost in embeddings: Information loss in vision-language models. *arXiv preprint arXiv:2509.11986*, 2025.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1118–1125. IEEE, 2018.

Robert McCarthy, Daniel CH Tan, Dominik Schmidt, Fernando Acero, Nathan Herr, Yilun Du, Thomas G Thuruthel, and Zhibin Li. Towards generalist robot learning from internet video: A survey. *Journal of Artificial Intelligence Research*, 83, 2025.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, et al. Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

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. *arXiv preprint arXiv:2304.07193*, 2023.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya 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)*, pp. 6892–6903. IEEE, 2024.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv preprint arXiv:2407.06581*, 2024.

Oleh Rybkin, Karl Pertsch, Konstantinos G Derpanis, Kostas Daniilidis, and Andrew Jaegle. Learning what you can do before doing anything. *arXiv preprint arXiv:1806.09655*, 2018.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023.

Mong Yuan Sim, Wei Emma Zhang, Xiang Dai, and Biaoyan Fang. Can VLMs actually see and read? a survey on modality collapse in vision-language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24452–24470, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1256. URL https://aclanthology.org/2025.findings-acl.1256/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. *arXiv preprint arXiv:1905.13566*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025b.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Hongxu Yin, Sifei Liu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025c.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001*, 2025.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *arXiv preprint arXiv:2506.15691*, 2025.

Qinqing Zheng, Mikael Henaff, Brandon Amos, and Aditya Grover. Semi-supervised offline reinforcement learning with action-free trajectories. In *International conference on machine learning*, pp. 42339–42362. PMLR, 2023.

A Related Work

Latent action learning. Recent studies (Nikulin et al., 2025; Klepach et al., 2025; Zhang et al., 2025) have highlighted the failure of LAMs in the presence of action-correlated distractors. Both Nikulin et al. (2025) and Zhang et al. (2025) proposed incorporating supervision from a small number of true actions during LAM training to help identify controllable changes. While this approach is effective, it is not generalizable, since in many domains, such as egocentric human videos (Yang et al., 2025c), true actions cannot be obtained in a practical manner. In contrast, we propose leveraging promptable representations from VLMs as an unsupervised approach to separate controllable changes from noise. We empirically demonstrate that, without any supervision, this enables LAMs to recover the performance in the presence of distractors (see Figure 1).

The work most closely related to ours is UniVLA (Bu et al., 2025), which also focuses on task-centric latent actions. UniVLA uses DINOv2 features, which are intended to provide priors that better capture task-relevant information. In addition, it proposes a two-stage LAM training pipeline that incorporates language task instruction embeddings as inputs to the LAM in the first stage. However, UniVLA design has important limitations. While conditioning the LAM on task instructions may help separate task-relevant videos from noise at a high level, it does not provide the per-step learning signal required to accurately recover low-level ground-truth actions. As a result, in the single-task learning setting the task instruction remains a constant vector, and UniVLA effectively reduces to LAPO, leading to noisy latent actions. In contrast, we provide the task instruction to the VLM, and while the instruction itself is constant, the VLM produces different language-conditioned representations for each observation, which LAM can use as prediction targets (see Figure 2). Thus, our approach is simpler, does not require multiple stages, provides clean targets at each step, and remains effective in the single-task regime, significantly outperforming UniVLA (see Section 6) and DINOv2 (see Section 5).

Promptable representations. Recently, Chen et al. (2024a) proposed using VLM embeddings generated with task-specific prompts to extract improved state representations, which enhance sample efficiency in online reinforcement learning. Our work is heavily inspired by this approach, and we closely follow their methodology for extracting promptable representations. Our contribution, however, is a new setting and novel empirical insights. Results of Chen et al. (2024a) do not guarantee the minimality of representations and, as a consequence, success in latent action learning. In contrast,

our work shows that VLMs differ substantially in their zero-shot suitability for LAMs, and that selecting an appropriate VLM can make LAMs effective in the presence of distractors, increasing success rates by up to sixfold, a result that has not been documented in prior work.

B LIMITATIONS

Segmentation, while simple, is not enough. The concept of extracting VLM embeddings with the hope that they will filter out distractors may initially seem strange. If the goal is to filter out distractors, would not it be more straightforward to simply segment the relevant parts and train LAPO directly in image space using masks? In fact, our benchmark includes VLMs capable of segmentation, such as Sa2VA (Yuan et al., 2025), and we even utilize such prompts (see Table 6), yet we still rely on embeddings instead of masks. While segmentation is appealing, it does not address the fundamental problem. Consider a scenario with a robotic arm and varying lighting conditions. Even if we segment the arm, we will still get changes in our observations that are not related to the actual actions, such as color shifts and reflections on the arm. The same issue arises with camera movement and changes in perspective. The key, therefore, is to work in a semantic latent space, which is where the common-sense reasoning capabilities of VLMs become crucial.

On the choice of MetaWorld benchmark. One notable limitation of our study is its small scale, as we rely on MetaWorld as our primary benchmark and do not extend our analysis to large VLAs and datasets, such as Open-X (O'Neill et al., 2024). However, this choice is deliberate for two reasons. First, while MetaWorld is simple, with distractors, it is difficult enough to completely break traditional LAMs and to distinguish different VLMs in terms of the promptable representations' quality (as we demonstrate in Section 5). As an early exploration, it was crucial to expand in variety (e.g., exploring more VLMs) within our limited resources. We hope that our analysis provides practitioners with valuable insights into the available options. Second, encoding entire datasets is both expensive and time-consuming, as it involves inference with large VLMs (e.g., 8B parameters) and generating answers. For our 5k trajectory datasets, the process can quickly exceed 24 hours, let alone for truly large datasets. Since this is purely inference and gradients are not required, the process can be significantly accelerated, for example, using vLLM (Kwon et al., 2023). However, we have left this as future work.

C HARDER DISTRACTORS

We provide a preliminary experiments with additional distractors beyond background videos. To make latent action learing harder, we added a change in lighting throughout the episode (see Figure 14), in addition to the videos. We chose one task (button-press-topdown). We used 3 random seeds and full datasets with 5k trajectories. We summarize results in a Table 5. Promptable representations still deliver a significant improvement in success rate in the presence of distractors.

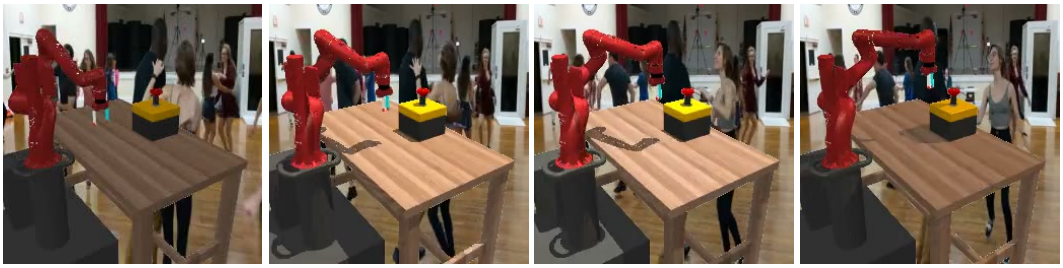


Figure 14: Visualization of a lighting change throughout the episode.

D COMPUTE RESOURCES

All experiments were performed on servers with a single H100 GPU with 80 GB of GPU memory, 12 CPU cores, and 244 GB of RAM. All metrics for the experiments were logged using the Weights &

Table 5: Success rates of LAPO and LAPO+Molmo with and without distractors.

Method	Success rate	
	Without distractors	**With distractors**
LAPO	0.99	0.21
LAPO+Molmo	0.96	**0.85**

Biases platform and will be released. Overall, the paper Weights & Biases project had 46,455 tracked experiments at the time of submission and used an estimated ~ 714 days of GPU compute in total.

E ADDITIONAL FIGURES

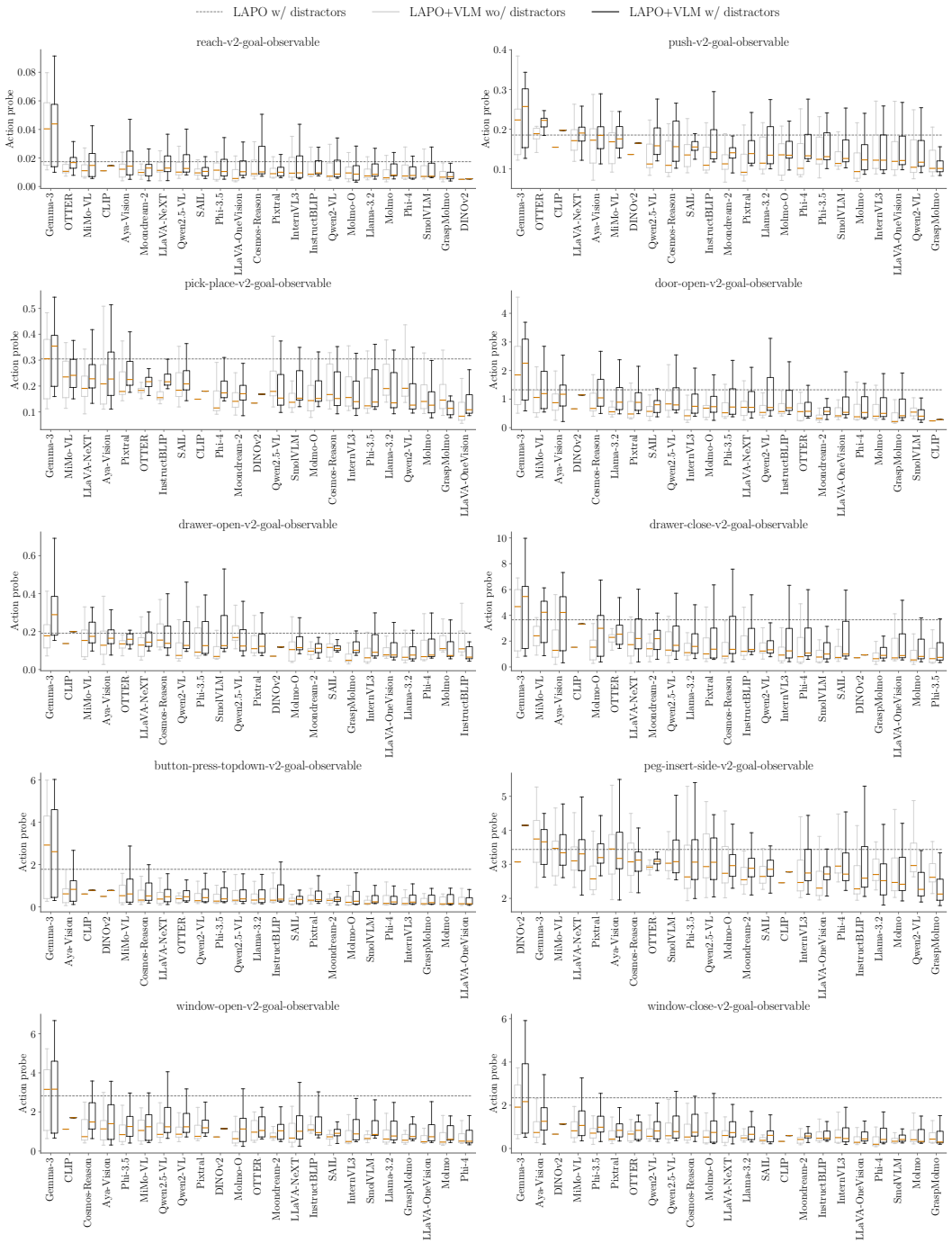


Figure 15: Aggregation over all hyperparameters for each task in MT10.

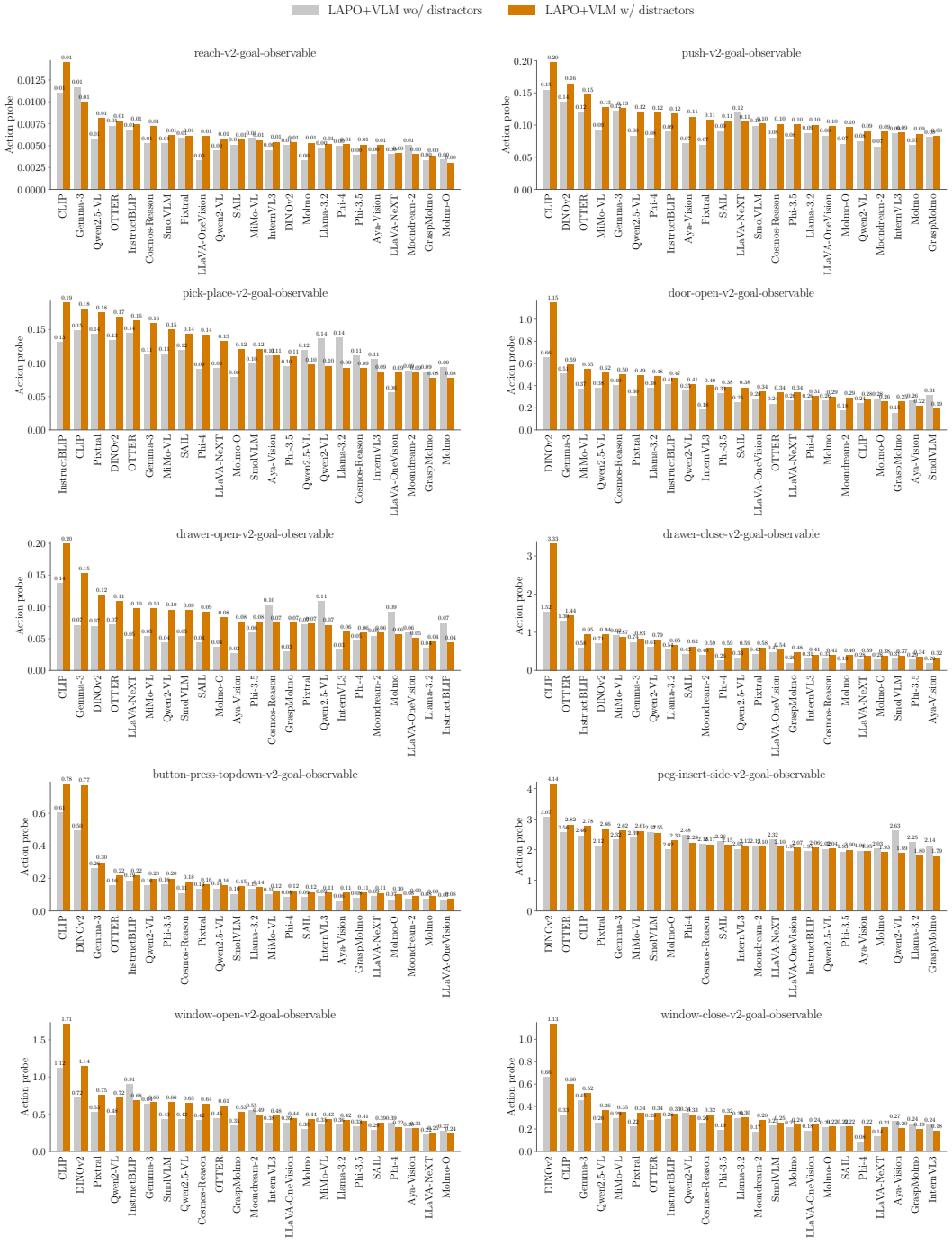


Figure 16: Probe values for best hyperparameters for each task in MT10.

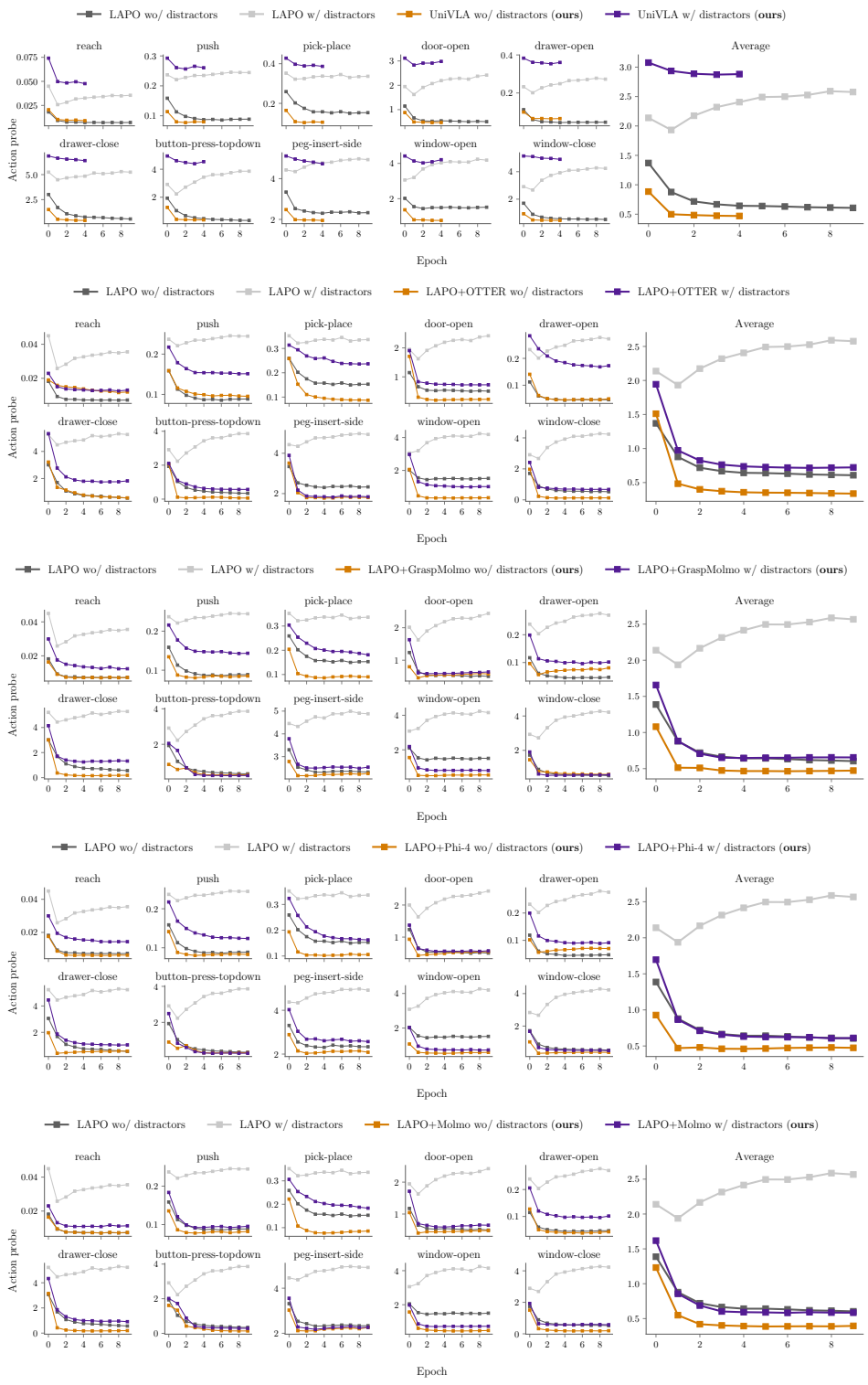


Figure 17: Action probes comparison for LAPO and LAPO+VLMs on full datasets for all tasks in MT10. Results are averaged over three random seeds. LAPO+VLM significantly improves upon LAPO in terms of the latent actions quality, and without any supervision with true actions closes the gap with LAPO without distractors. While all VLMs bring improvements, Molmo achieve best results overall. For resulting success rates see Figure 12

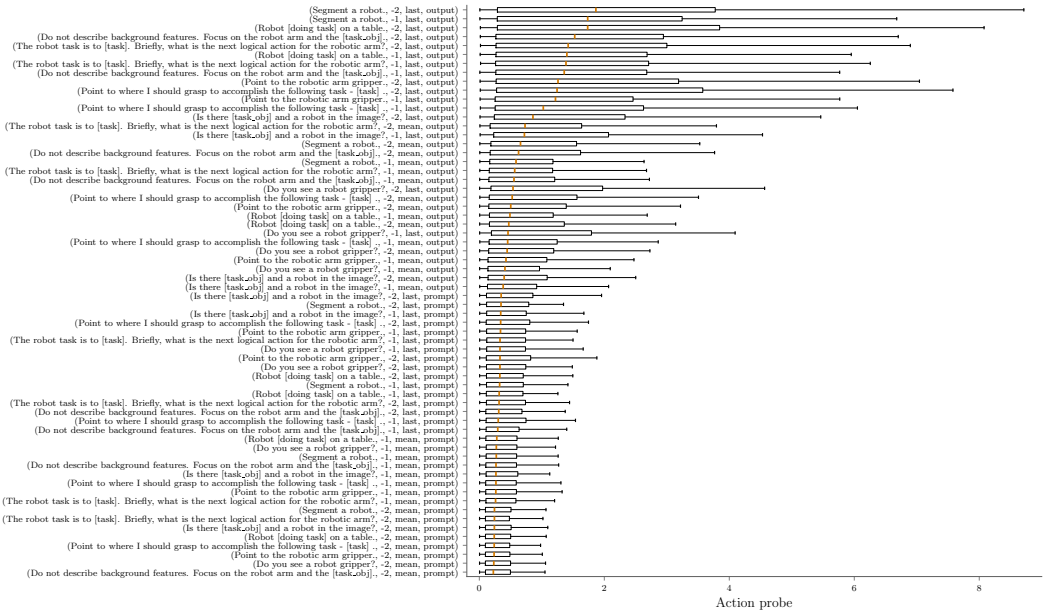


Figure 18: Action probe rankings across all explored hyperparameter combinations. Smaller probe is better. Values are averaged over all VLMs, tasks and settings, e.g. with and without distractors. Hyperparameters in order: prompt, layer, reduction type and source of the embeddings. Feel free to zoom in!

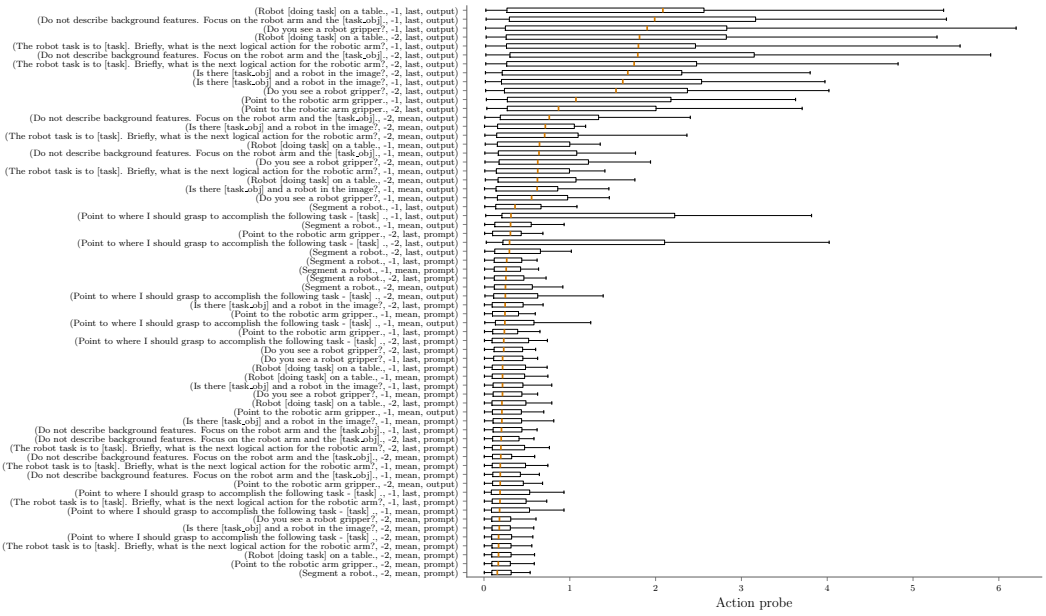


Figure 19: Action probes ranking for all combinations of hyperparameters explored for Molmo VLM. Values are averaged over all tasks and settings, e.g. with and without distractors.

F VISION-LANGUAGE MODELS DETAILS

Table 6: Prompt templates used in our experiments. We adapt them to each task by inserting information relevant to the task. All VLMs explored share the same prompts per task.

Prompt
The robot task is to [task]. Briefly, what is the next logical action for the robotic arm?
Do not describe background features. Focus on the robot arm and the [task-obj].
Do you see a robot gripper?
Is there [task-obj] and a robot in the image?
Robot [doing task] on a table.
Point to the robotic arm gripper.
Point to where I should grasp to accomplish the following task - [task].
Segment a robot.

Table 7: Exact HuggingFace IDs for all VLMs we used. We shortened their names in Figures to save some space.

Name	**HuggingFace ID**
InstructBLIP	Salesforce/instructblip-vicuna-7b
Molmo	allenai/Molmo-7B-D-0924
Gemma-3	google/gemma-3-12b-it
Llama-3.2	unsloth/Llama-3.2-11B-Vision-Instruct
Qwen2.5-VL	Qwen/Qwen2.5-VL-7B-Instruct
InternVL3	OpenGVLab/InternVL3-8B
Cosmos-Reason	nvidia/Cosmos-Reason1-7B
Phi-4	microsoft/Phi-4-multimodal-instruct
LLaVA-OneVision	llava-hf/llava-onevision-qwen2-7b-ov-hf
SmolVLM	HuggingFaceTB/SmolVLM2-2.2B-Instruct
Pixtral	mistral-community/pixtral-12b
GraspMolmo	allenai/GraspMolmo
Qwen2-VL	Qwen/Qwen2-VL-7B-Instruct
Phi-3.5	Lexius/Phi-3.5-vision-instruct
LLaVA-NeXT	llava-hf/llava-v1.6-mistral-7b-hf
Aya-Vision	CohereLabs/aya-vision-8b
MiMo-VL	XiaomiMiMo/MiMo-VL-7B-RL
Molmo-O	allenai/Molmo-7B-O-0924
SAIL	ByteDance-Seed/SAIL-7B
Moondream-2	vikhyatk/moondream2
VLM2Vec	VLM2Vec/VLM2Vec-V2.0
UniME	DeepGlint-AI/UniME-LLaVA-1.6-7B
E5-V	royokong/e5-v
Janus-Pro	deepseek-community/Janus-Pro-7B
Sa2VA	ByteDance/Sa2VA-4B

G HYPERPARAMETERS

Table 8: LAPO-ResNet hyperparameters. Names are exactly follow the configuration files used in code.

Part	Parameter	Value
General	frame_stack	4
	probe_learning_rate	0.0003
	disable_distractors	True
	seed	0
	eval_seed	0
	eval_episodes	50
Latent action learning	latent_action_dim	128
	idm_encoder_scale	5
	idm_encoder_num_res_blocks	1
	idm_encoder_channels	[16, 16, 32, 32, 128, 128, 256]
	fdm_encoder_scale	1
	fdm_encoder_num_res_blocks	1
	fdm_encoder_channels	[16, 16, 32, 32, 128, 128, 256]
	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	1
	grad_norm	-
Latent behavior cloning	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	0
	encoder_scale	5
	encoder_num_res_blocks	1
	encoder_channels	[16, 16, 32, 32, 128, 128, 256]
Latent actions decoding	total_updates	100000
	batch_size	64
	learning_rate	0.001
	hidden_dim	128
	num_labeled_trajectories	[16, 8, 2, 4]

Table 9: LAPO-Trans hyperparameters. Names exactly follow the configuration files used in code.

Part	Parameter	Value
General	frame_stack	4
	probe_learning_rate	0.0003
	disable_distractors	True
	seed	0
	eval_seed	0
	eval_episodes	50
Latent action learning	latent_action_dim	128
	patch_size	32
	fdm_use_cross_attn	False
	idm_hidden_dim	896
	idm_num_layers	4
	idm_num_heads	16
	fdm_hidden_dim	256
	fdm_num_layers	4
	fdm_num_heads	8
	normalize_qk	False
	pre_norm	True
	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	1
	grad_norm	-
Latent behavior cloning	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	0
	encoder_scale	5
	encoder_num_res_blocks	1
	encoder_channels	[16, 16, 32, 32, 128, 128, 256]
Latent actions decoding	total_updates	100000
	batch_size	64
	learning_rate	0.001
	hidden_dim	128
	num_labeled_trajectories	[16, 8, 2, 4]

Table 10: LAPO+VLM hyperparameters. Names exactly follow the configuration files used in code.

Part	Parameter	Value
General	frame_stack	4
	probe_learning_rate	0.0003
	disable_distractors	True
	seed	0
	eval_seed	0
	eval_episodes	50
VLM (example)	type	molmo
	prompt	Point to the robotic arm gripper.
	layer	27
	target	output
	reduce_strategy	mean
Latent action learning	latent_action_dim	128
	idm_encoder_scale	5
	idm_encoder_num_res_blocks	1
	idm_encoder_channels	[16, 16, 32, 32, 128, 128, 256]
	fdm_hidden_dim	1024
	fdm_num_layers	4
	fdm_expand	4
	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	1
	grad_norm	-
Latent behavior cloning	num_epochs	10
	batch_size	64
	learning_rate	0.0001
	weight_decay	0.0
	warmup_epochs	0
	encoder_scale	5
	encoder_num_res_blocks	1
	encoder_channels	[16, 16, 32, 32, 128, 128, 256]
Latent actions decoding	total_updates	100000
	batch_size	64
	learning_rate	0.001
	hidden_dim	128
	num_labeled_trajectories	[16, 8, 2, 4]