Vision-Language Models Unlock Task-Centric Latent Actions

Alexander Nikulin
AIRI, MSU, MIPT
nikulin@airi.net

Ilya Zisman
AIRI, Skoltech, ISP RAS
zisman@airi.net

Albina Klepach
AIRI
klepach@airi.net

Denis Tarasov
AIRI, ETH Zürich
tarasov@airi.net

Alexander Derevyagin
AIRI, HSE
derevyagin@airi.net

Andrei Polubarov
AIRI, Skoltech, ISP RAS
polubarov@airi.net

Nikita Lyubaykin
AIRI, Innopolis University
lyubaykin@airi.net

Vladislav Kurenkov
AIRI, Innopolis University
kurenkov@airi.net

Abstract: Latent Action Models (LAMs) have rapidly gained traction as an important component in the pre-training pipelines of leading Vision-Language-Action models. 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 in unsupervised way. We use these representations as a targets during LAM training and benchmark a wide variety of popular VLMs. Our results show that simply asking VLMs to ignore distractors can substantially improve latent action quality, yielding up to a six-fold increase in downstream success rates on Distracting MetaWorld.

Keywords: CoRL, VLA, Latent Action Models

Ask and it will be given to you; seek and you will find;
knock and the door will be opened to you.
Matthew 7:7-8

1 Introduction

Latent action models [1, 2] have quickly become integral to the pre-training pipelines of leading Vision–Language–Action (VLA) systems [3, 4, 5, 6, 7]. By inferring compact, semantically meaningful latent action representations at scale, Latent Action Models (LAM) mitigate the scarcity of high-quality action-labeled data, giving a promise to unlock vast amounts of unlabeled videos [8]. Removing the data bottleneck facilitates further scaling in embodied AI and robotics; consequently, any improvements to LAMs can have outsized downstream impacts.

Unfortunately, most LAMs [1, 2, 9, 10] to date have been trained on relatively clean datasets, where changes between observations can be explained almost entirely by ground-truth actions—such as in static scenes with a single manipulator. In contrast, real-world data often contains numerous action-correlated distractors, including background human movement or other spurious correlations. As shown by Nikulin et al. [11], Zhang et al. [12], 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. While providing supervision via true actions can be effective [11], it is not scalable — especially in domains where these actions are impossible to obtain, such as egocentric human videos.

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 in any video. Wouldn't it also be convenient to simply ask LAM to focus on the relevant features, e.g. robotic arm, and ignore any other details? Inspired by the work of Chen et al. [13], Huang et al. [14] on promptable representations, we propose to utilize the common-sense reasoning abilities of modern Vision-Language Models (VLMs) 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.

In this work, we present our preliminary investigation on whether promptable representations produced by prompting VLMs to focus on task-specific details can serve as an effective target for latent action learning in the presence of distractors. Using Distracting MetaWorld as our main environment, we begin from a simple demonstration experiment, showing that limitations of LAM can be mitigated with the right target. We further conduct large-scale benchmarking of different VLMs to assess their effectiveness at providing promptable representations. Finally, using the best setup found, we demonstrate that without any supervision, promptable representations can significantly improve latent action quality and downstream performance, increasing success rate six-fold.

2 Background

Problem setting. We consider a setting of offline imitation learning from observation [15, 16], which closely matches the regime increasingly utilized by the field of embodied AI [8, 4, 3] (e.g. robotics). Our goal is to pre-train a policy $\pi(o|a)$, given a large dataset of expert trajectories $\mathcal{D} := \{(o_i^n)\}_{i=1}^{\tau}$, 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 [17, 11], while in our experiments, we consider a ratio as low as $< 1\%$.

Latent action models. Given the dataset of observations $\mathcal{D} := \{(o_i^n)\}_{i=1}^{\tau}$, latent action models (LAM) [18, 19, 1] try to infer latent actions z_t such that they are maximally predictive of observed transitions (o_t, o_{t+1}) while being minimal [1], i.e. describe changes only relevant to control. After pre-training, LAM is used to supply latent actions for imitation learning on unlabeled dataset to obtain useful behavioral priors. As a final stage, small decoder is trained to map from latent to ground-truth actions on a small number of labels.

Modern LAMs [6, 2, 20, 21, 9, 10] mostly follow the same high-level architecture introduced by LAPO [1], 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 $\mathcal{L}_{\text{MSE}} = \mathbb{E}_{(o_t, o_{t+1}) \sim \mathcal{D}} \left[\| f_{\text{FDM}}(f_{\text{IDM}}(\boldsymbol{o}_t, \boldsymbol{o}_{t+1}), \boldsymbol{o}_t) - \boldsymbol{o}_{t+1} \|^2 \right]$.

Limitations of latent action models. Recent studies highlighted LAM failure when action-correlated distractors are present [11, 22, 12]. While they can recover ground-truth actions when only controllable changes are present, real-world videos typically involve both controllable factors and exogenous noise (e.g., people moving in the background). In such cases, LAMs cannot disentangle the dynamics, leading latent actions to primarily capture noise, which makes them useless for imitation learning. Both Nikulin et al. [11], Zhang et al. [12] proposed providing supervision with a small number of true actions during LAM training to help identify controllable changes. While this solution is effective, it is not generalizable, as in many domains, such as egocentric human videos [23], it is not possible to obtain true actions in a reasonable way.

3 Experimental Setup

Environments and datasets. In contrast to Nikulin et al. [11], we use MetaWorld Multi-Task 10 [24] as our primary benchmark, as it provides greater realism than Distracting Control [25], while being lightweight enough to allow experimentation with VLMs under limited resources. We modify MetaWorld to include distracting dynamics videos in the background, using the same DAVIS videos as in Nikulin et al. [11]. We also move the default camera position farther back to include more of the background video in the observation, making latent action learning more challenging. See Figure 1 for a visualization.

We follow the standard three-stage pipeline [1, 2, 11]: (1) pre-train the LAM, (2) train BC on latent actions, and (3) train a decoder head on a small number of true-action labels. For each task, we collect 5k trajectories from the scripted experts provided by MetaWorld and up to 16 additional labeled trajectories for the final stage, which is less than 1% of the full datasets.

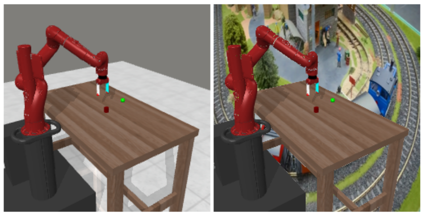


Figure 1: Visualization of observations with and without distractors in our modification of MetaWorld environment.

Evaluation. For evaluation, we follow standard metrics similar to Nikulin et al. [11]: linear probing and success rate. 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. This metric is also used for hyperparameter tuning, which may be impractical in real-world settings but allows us to estimate the upper-bound performance of each method for fair comparison.

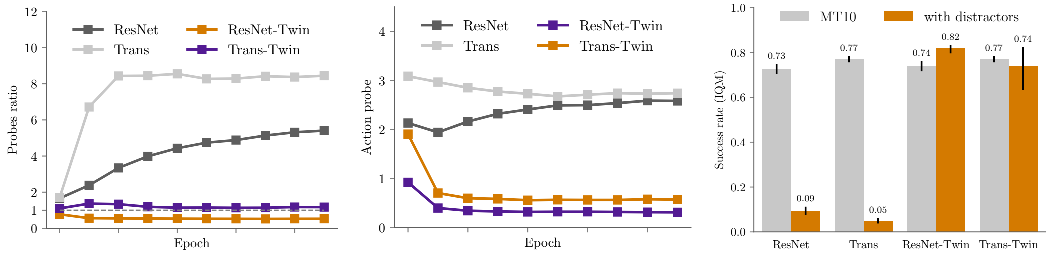
However, as Nikulin et al. [11] notes, linear probing has a key limitation: it can reveal whether true actions are present in the latent space, but it does not guarantee minimality, meaning that exogenous noise may still be encoded. To preserve this guarantee, we fix the latent action dimensionality to 128 for all methods, which at least allows us to rank quality under equal information bottleneck. Finally, to measure the true usefulness of latent actions, we evaluate the success rate in the environments after fine-tuning on true action labels.

Promptable representations. We follow the Chen et al. [13] 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. We obtain such representations from the last and next-to-last layers [13]. In contrast to the Chen et al. [13], Huang et al. [14] 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 embeddings or taking only the embedding of the last token from either prompt or the generated answer.

Latent action model architecture. We use the architecture proposed by Schmidt and Jiang [1], omitting action quantization, due to its harmful effect [11, 26, 27]. We use frame stacking, but only in IDM, while FDM uses only the current frame to predict the next, as in Chen et al. [9]. Other than that, in our main experiments, we do not use any improvements upon LAPO (if not explicitly stated otherwise), such as augmentations or multi-step predictions in FDM [11, 9, 2, 20], to remove possible confounders on latent action quality. When predicting in the latent space instead of images, we follow Nikulin et al. [11] and use multiple MLP blocks similar to those used in Transformers [28]. For action decoder head, we use a small three-layer MLP. See Appendix C for hyperparameters used.

4 The Importance of Right Target

We begin with a demonstration experiment to show that the limitation of LAMs in the presence of distractors arises entirely from the poor target used in the forward dynamics model (FDM), rather than from any flaw in the overall idea or architecture. By LAM construction, latent actions are optimized



(a) w/ to wo/ distractors probe ratio

(b) Probes w/ distractors

(c) Success rate on MT10

Figure 2: 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. We use three random seeds and report IQM and 95%-CI based on stratified bootstrapping, following the Agarwal et al. [29]. See Section 4 for details.

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 recover the ground-truth actions despite distractors in the input observations to IDM and FDM? If not, the idea with promptable representations would be impractical.

To answer these questions we construct a special dataset with twin observations for each task: during data collection we render and save same observation with and without distractors. Next, during training 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 exogenous noise (see Figure 1). To show that existing limitations are agnostic to the architecture of FDM and IDM, we explore both ResNet [1] and spatio-temporal transformer [6, 2] backbones.

Results. First of all, as can be seen in Figure 3, we confirm that in our domain simply adding distractors results in complete degradation of latent actions quality regardless of backbone used. This subsequently leads to almost zero success rate after fine-tuning on true actions (see Figure 2c), which does not happen without distractors. Ideally, probes should be close to each other, as real underlying actions are identical between both settings.

Next, in Figures 2a and 2b we show the effect of using perfect targets during LAPO training (with -Twin postfix). To better illustrate the trend, in Figure 2a we report the ratio of probes with and without distractors for each method. With the ideal target probes immediately drop to the level of LAPO without distractors, and ratio converges to one. To our surprise, it is in fact possible to get even better result, as LAPO-ResNet achieves ratio below one, i.e. outperforming LAPO-ResNet without distrators. We attribute this to the implicit augmentation effect of distractors. Finally, improvement in latent action quality directly results in leveling success rates (see Figure 2c).

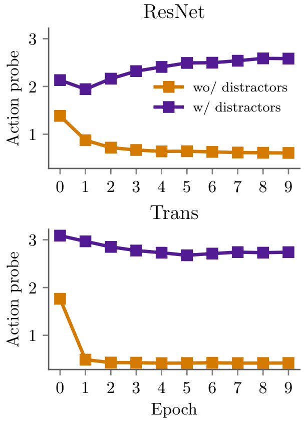
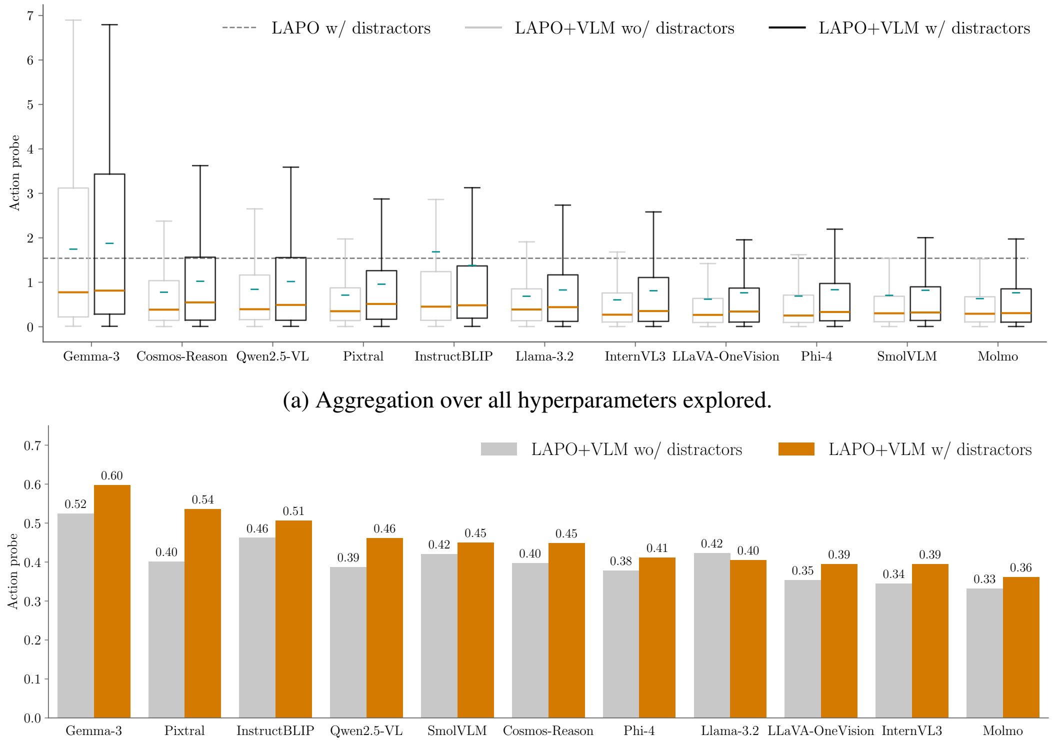


Figure 3: Baseline LAPO action probes on MT10. Averaged over 3 random seeds.

Overall, we confirm that the right target is key to unlock latent action learning in the presence of distractors. Although these experiments may seem obvious in hindsight, they allow us to convey a key empirical observation about latent action learning, one that provides the same intuition that originally led us to explore promptable representations.



(a) Aggregation over all hyperparameters explored.

(b) Aggregation over best hyperparameters.

Figure 4: Benchmarking the effectiveness of promptable representations provided by different VLMs for latent action learning on all tasks from MT10. Results aggregated over $\sim$ 14k experiments. 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 5.

5 The Promise of Promptable Representations

Our main hypothesis is that VLMs, due to their common-sense reasoning abilities, can serve as an effective unsupervised way of obtaining clean observation representations, which would disentangle controllable features from the noise. As we demonstrated in the previous section, it would be enough to unlock latent action learning in the presence of distractors.

We have no doubt that most modern VLMs would easily identify the robotic arm location in the image (like Figure 1) and describe it in detail, even in the presence of background noise. However, the ability to generate valid text does not necessarily imply that the underlying embeddings are suitable for our purposes. For a representation to serve as an effective target for LAM, it should (1) contain task-centric visual information, (2) be minimal by filtering out visual details irrelevant to the prompt, and (3) remain consistent across dynamics to mimic changes caused by real actions. Unfortunately, current VLMs are known to struggle with visual focus [30, 31] and pixel-level understanding [32, 33]. Given these limitations, we begin by benchmarking a wide variety of modern VLMs to assess their viability, conducting $\sim$ 14k experiments in total. Based on this benchmark, we then select the most effective VLM along with the best hyperparameters (e.g., prompt, aggregation strategy, and others).

Proper way to evaluate VLMs via small scale experiments. Conducting large scale VLMs evaluation on the full datasets would be prohibitively expensive. Chen et al. [13] proposed assessing prompts via linear probing on small datasets, for example by asking whether task-relevant entities are present in the image and measuring probe accuracy. While feasible, this approach is suboptimal in our setting. Probing representations to predict real actions may help rank prompts for a single VLM, but it cannot reliably compare across multiple VLMs or hyperparameters, since probing does not capture the minimality of representations, an essential property for LAMs. Instead, we directly train LAPO+VLM on a small subset of trajectories, e.g. 64 instead of full 5k, and measure

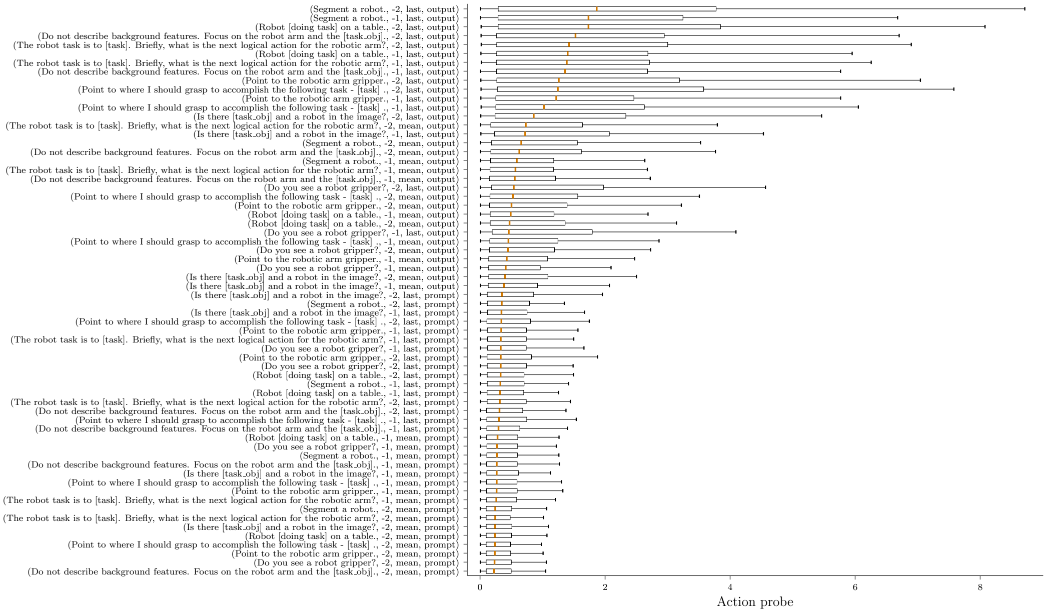


Figure 5: Action probe rankings across all explored hyperparameter combinations. Reported values are averaged over all VLMs, tasks, and settings (with and without distractors).

the resulting latent action quality. We validated that hyperparameter rankings obtained in this way transfer reasonably well to the full dataset, although probes can have different values.

Results. We summarize our benchmarking results in Figure 4 and provide full per-hyperparameter rankings in Figure 5. 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 1 in Appendix B). 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, amounting to roughly 14k experiments in total. The full list of VLMs, including exact model names, sizes, and prompt templates, is provided in Appendix B.

As can be seen in Figure 4a, overall all VLMs provide some degree of improvement over LAPO in terms of the median action probe. However, some of them, especially Molmo [34], are generally preferable and have lower variance, indicating higher robustness to different hyperparameters. In Figure 4b we visualize ranking by aggregating best scores for each task. While this changes ranking a bit, we still observe that Gemma-3 [35] is the worst and Molmo [34] is consistently the best. Based on Figure 5, we observe that in general, promptable representations aggregated by averaging next-to-last layer prompt embeddings perform the best. Ironically[1], 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.

This brings us to a striking conclusion that state-of-the-art VLMs do not necessarily provide better promptable representations. For example, InstructBLIP [36] outperforms both Gemma-3 [35] and Pixtral [37], despite being considerably older. Furthermore, Cosmos-Reason [38] results indicate that explicit fine-tuning on robotics data is not sufficient to guarantee improved representations. We believe that our results, besides relevance to latent action learning, reveal a large blind spot in how VLMs are currently evaluated, with critical implications for robotics and Vision-Language Action (VLA) models.

6 Promptable Representations Unlock Task-Centric Latent Actions

Based on the benchmark results (see Figure 4), we selected Molmo as our primary VLM for further experiments. Although it achieved the lowest median action probe on the small datasets, it remains

[1] This result is the main inspiration for the paper epigraph.

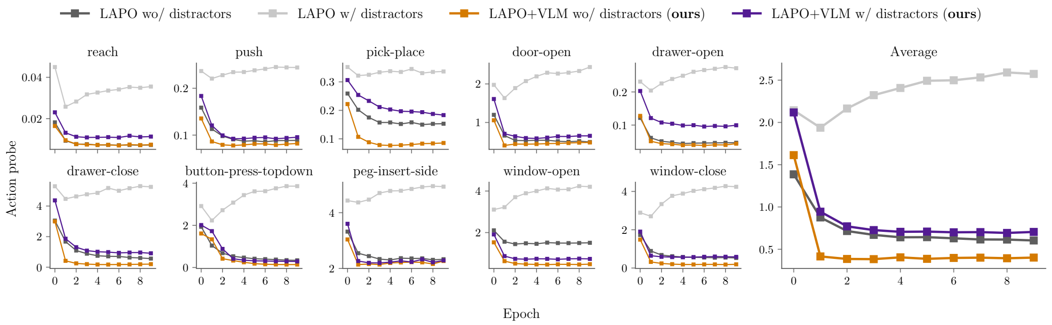


Figure 6: Action probes comparison for LAPO and LAPO+VLM (Molmo) on full datasets for all tasks in MT10. Results are averaged over three random seeds. As can be seen, LAPO+VLM significantly improves upon LAPO in terms of the latent actions quality, and without any supervision closes the gap with LAPO without distractors. For resulting success rates see Figure 7.

necessary to validate whether this performance transfers to the full 5k datasets and yields improved success rates, as this is not guaranteed [11]. We chose the best hyperparameters for Molmo and trained LAPO+VLM on the full datasets, using three random seeds. As this work is preliminary, we currently report results only with a ResNet backbone.

Results. We present the resulting action probes for each task in Figure 6 and final success rates after fine-tuning on 16 trajectories with real actions in Figure 7. LAPO+VLM achieves a substantial improvement in latent action quality, both with and without distractors. With distractors, it nearly closes the gap to LAPO trained without distractors, while without distractors it even slightly outperforms it. Crucially, this improvement carries over to downstream performance: success rates increase by a factor of six in the presence of distractors, while remaining unchanged without them. These results confirm the viability of promptable representations as a clean target for latent action modeling under distracting conditions.

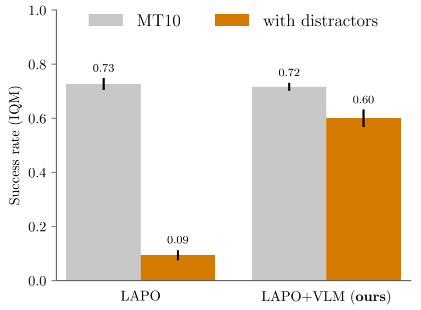


Figure 7: Success rate on MT10 for LAPO and LAPO+VLM (Molmo), which uses promptable representations. We use three random seeds and report IQM and 95%-CI based on stratified bootstrapping, following the Agarwal et al. [29].

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 for the next generation of Vision-Language-Action models.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] D. Schmidt and M. Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023.

[2] 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.

[3] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[4] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[5] Y. Zhong, F. Bai, S. Cai, X. Huang, Z. Chen, X. Zhang, Y. Wang, S. Guo, T. Guan, K. N. Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

[6] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[7] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

[8] R. McCarthy, D. C. Tan, D. Schmidt, F. Acero, N. Herr, Y. Du, T. G. Thuruthel, and Z. Li. Towards generalist robot learning from internet video: A survey. *Journal of Artificial Intelligence Research*, 83, 2025.

[9] 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.

[10] S. Gao, S. Zhou, Y. Du, J. Zhang, and C. Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

[11] A. Nikulin, I. Zisman, D. Tarasov, N. Lyubaykin, A. Polubarov, I. Kiselev, and V. Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

[12] C. Zhang, T. Pearce, P. Zhang, K. Wang, X. Chen, W. Shen, L. Zhao, and J. Bian. What do latent action models actually learn? *arXiv preprint arXiv:2506.15691*, 2025.

[13] W. Chen, O. Mees, A. Kumar, and S. Levine. Vision-language models provide promptable representations for reinforcement learning. *arXiv preprint arXiv:2402.02651*, 2024.

[14] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel. Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.

[15] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[16] F. Torabi, G. Warnell, and P. Stone. Recent advances in imitation learning from observation. *arXiv preprint arXiv:1905.13566*, 2019.

[17] Q. Zheng, M. Henaff, B. Amos, and A. Grover. Semi-supervised offline reinforcement learning with action-free trajectories. In *International conference on machine learning*, pages 42339–42362. PMLR, 2023.

[18] O. Rybkin, K. Pertsch, K. G. Derpanis, K. Daniilidis, and A. Jaegle. Learning what you can do before doing anything. *arXiv preprint arXiv:1806.09655*, 2018.

[19] A. Edwards, H. Sahni, Y. Schroecker, and C. Isbell. Imitating latent policies from observation. In *International conference on machine learning*, pages 1755–1763. PMLR, 2019.

[20] Z. J. Cui, H. Pan, A. Iyer, S. Haldar, and L. Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *arXiv preprint arXiv:2409.12192*, 2024.

[21] Y. Chen, Y. Ge, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 2024.

[22] A. Klepach, A. Nikulin, I. Zisman, D. Tarasov, A. Derevyagin, A. Polubarov, N. Lyubaykin, and V. Kurenkov. Object-centric latent action learning. *arXiv preprint arXiv:2502.09680*, 2025.

[23] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, H. Yin, S. Liu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[24] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[25] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

[26] 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.

[27] J. Yang, Y. Shi, H. Zhu, M. Liu, K. Ma, Y. Wang, G. Wu, T. He, and L. Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

[30] P. Rahmanzadehgervi, L. Bolton, M. R. Taesiri, and A. T. Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv preprint arXiv:2407.06581*, 2024.

[31] M. Y. Sim, W. E. Zhang, X. Dai, and B. Fang. Can VLMs actually see and read? a survey on modality collapse in vision-language models. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 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/.

[32] C. Gou, A. Felemban, F. F. Khan, D. Zhu, J. Cai, H. Rezatofighi, and M. Elhoseiny. How well can vision language models see image details? *arXiv preprint arXiv:2408.03940*, 2024.

[33] Y. Dahou, N. D. Huynh, P. H. Le-Khac, W. R. Para, A. Singh, and S. Narayan. Vision-language models can't see the obvious. *arXiv preprint arXiv:2507.04741*, 2025.

[34] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. 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*, pages 91–104, 2025.

[35] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[36] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[37] P. Agrawal, S. Antoniak, E. B. Hanna, B. Bout, D. Chaplot, J. Chudnovsky, D. Costa, B. De Monicault, S. Garg, T. Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.

[38] A. Azzolini, J. Bai, H. Brandon, J. Cao, P. Chattopadhyay, H. Chen, J. Chu, Y. Cui, J. Diamond, Y. Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

A Additional Figures

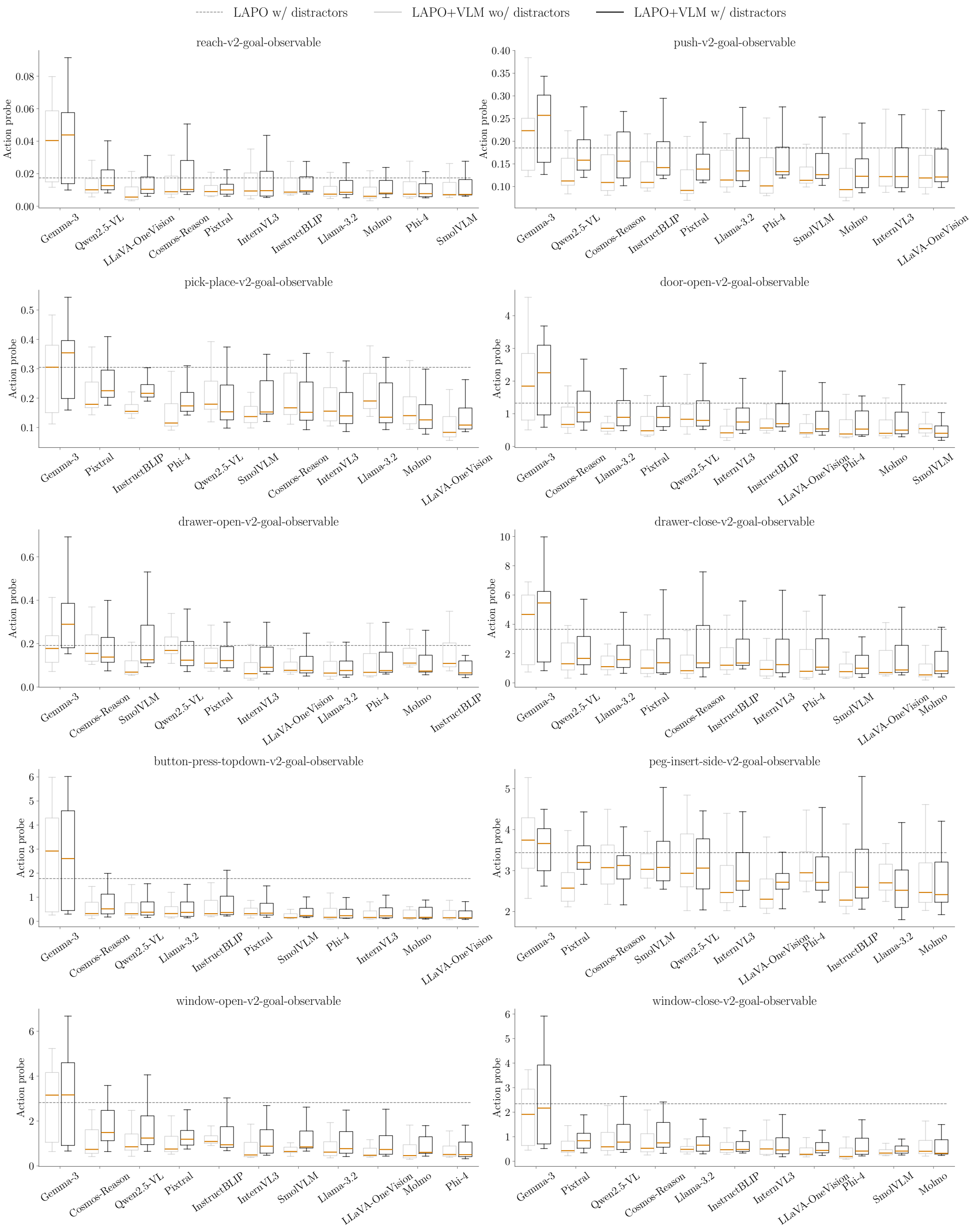


Figure 8: Aggregation over all hyperparameters for each task in MT10.

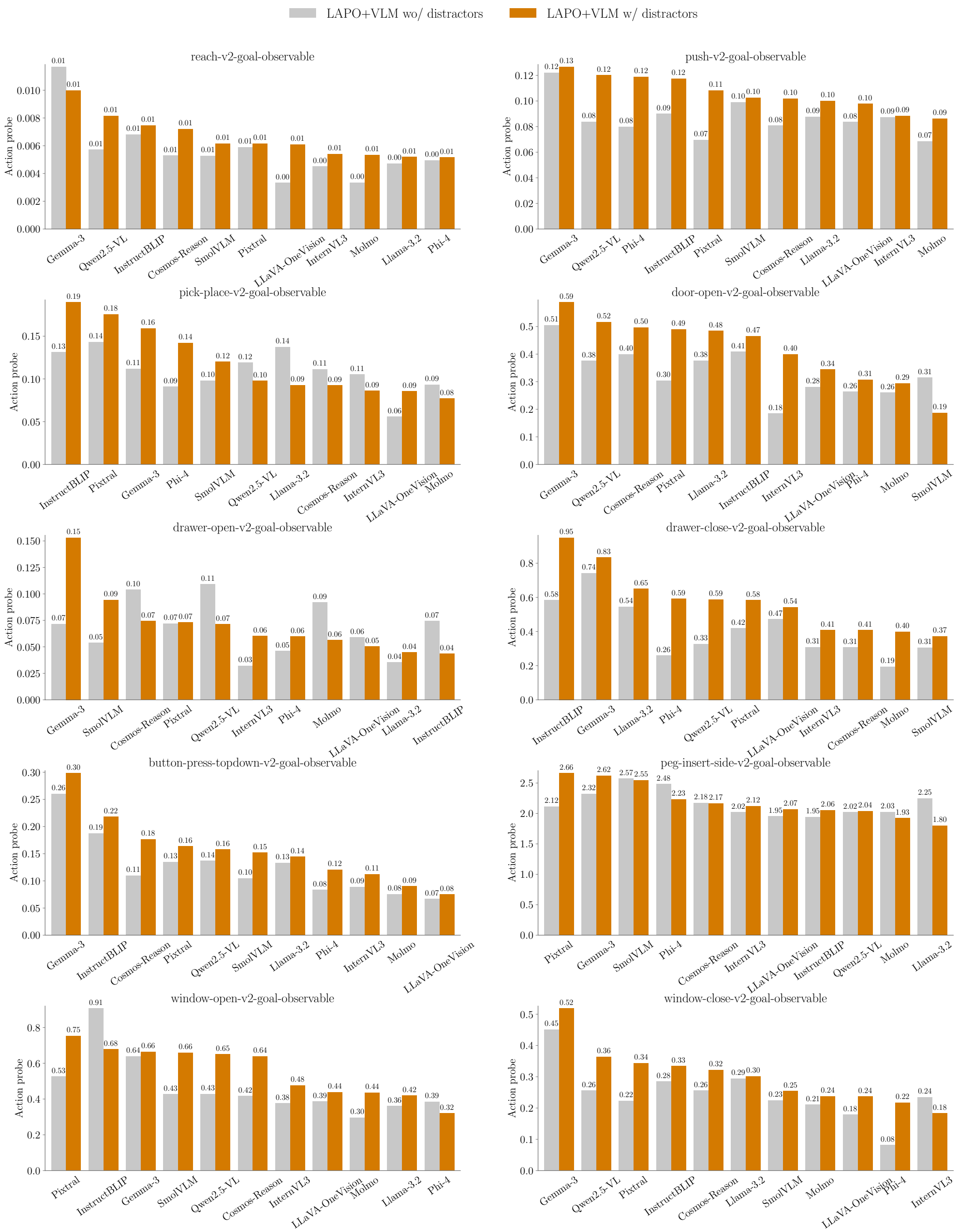


Figure 9: Probe values for best hyperparameters for each task in MT10.

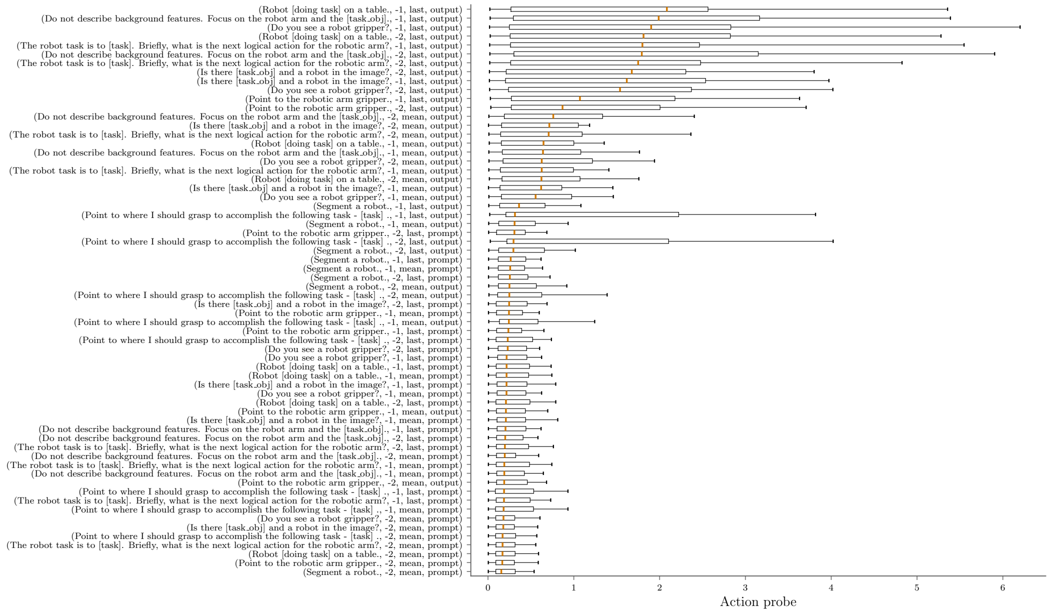


Figure 10: Action probes ranking for all combinations of hyperparameters explored for Molmo VLM. Values are averaged over all tasks and settings, e.g. with and without distractors.

B Vision-Language Models Details

Table 1: 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 2: 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

C Hyperparameters

Table 3: 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 4: 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 5: 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	200
	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]