OBJECT-CENTRIC LATENT ACTION LEARNING

Albina Klepach* AIRI
Alexander Nikulin AIRI, MIPT
Ilya Zisman AIRI, Skoltech
Denis Tarasov AIRI, ETH Zürich

Alexander Derevyagin AIRI
Andrei Polubarov AIRI, Skoltech
Nikita Lyubaykin AIRI, Innopolis University

Vladislav Kurenkov AIRI, Innopolis University

ABSTRACT

Leveraging vast amounts of internet video data for Embodied AI is currently bottle-necked by the lack of action annotations and the presence of action-correlated distractors. We propose a novel object-centric latent action learning approach, based on VideoSaur and LAPO, that employs self-supervised decomposition of scenes into object representations and annotates video data with proxy-action labels. This method effectively disentangles causal agent-object interactions from irrelevant background noise and reduces the performance degradation of latent action learning approaches caused by distractors. Our preliminary experiments with the Distracting Control Suite show that latent action pretraining based on object decompositions improves the quality of inferred latent actions by $\mathbf{x2.7}$ and efficiency of downstream fine-tuning with a small set of labeled actions, increasing return by $\mathbf{x2.6}$ on average.

1 INTRODUCTION

In recent years, the scaling of model and data sizes has led to the creation of powerful and general foundation models (Bommasani et al., 2021) that have enabled many breakthroughs in understanding and generation of natural language (Achiam et al., 2023; Brown et al., 2020) and images (Dehghani et al., 2023; Radford et al., 2021). On the other hand, the field of embodied AI has generally remained behind in terms of generalization and emergent abilities (Guruprasad et al., 2024), being mostly limited by the lack of diverse data for pre-training (Lin et al., 2024). The vast amount of video data on the Internet, covering a wide variety of human-related activities, can potentially fulfill the current data needs (McCarthy et al., 2024). Unfortunately, videos cannot be used directly due to the lack of action labels, which is necessary for imitation and reinforcement learning algorithms.

In order to compensate for the lack of action labels, approaches based on Latent Action Models (LAM) (Schmidt & Jiang, 2024; Ye et al., 2024; Cui et al., 2024; Bruce et al., 2024), aim to infer latent proxy-actions between consecutive observations. Such actions can be later used for imitation learning to obtain behavioral policy prior from large unlabeled datasets. A significant challenge in this endeavor is the presence of action-correlated distractors—dynamic backgrounds, incidental object motions, camera shifts, and other nuisances—that falsely correlate with agent actions, and may lead models to overfit to non-causal patterns (Wang et al., 2024; Misra et al., 2024b; McCarthy et al., 2024). Existing methods for learning latent actions from videos, such as Latent Action Pretraining (LAPA) (Ye et al., 2024), often assume curated, distractor-free datasets or rely on costly annotations. Although effective in controlled settings, this reliance on clean data limits their scalability and applicability in real-world scenarios.

In this preliminary work, we propose object-centric latent action learning in order to improve applicability to real-world data. By decomposing scenes into spatio-temporal object slots (Locatello et al., 2020), our method provides the structural priors needed to disentangle causal agent-object

*Correspondence to: klepach@airi.net. Work done by dunnolab.ai.

representations from distractors. Object-centric representations (Zadaianchuk et al., 2023) isolate entities into slots through self-supervised feature similarity losses, inherently filtering noise like static backgrounds or incidental motion. This enables Latent Action Models to focus on the dynamics of task-relevant objects while ignoring spurious correlations. Using the Distracting Control Suite (DCS) (Stone et al., 2021), we empirically demonstrate that latent action learning based on self-supervised object-centric decomposition from VideoSAUR (Zadaianchuk et al., 2023) improves the quality of latent actions by $\mathbf{x2.7}$ and downstream performance after fine-tuning with small amount of ground-truth actions by $\mathbf{x2.6}$ on average.

In a concurrent work to ours, Villar-Corrales & Behnke (2025) also explores the application of object-centric representations for latent action learning and video prediction, showing positive results in the robotic tabletop simulations. In contrast to our work, Villar-Corrales & Behnke (2025) does not explore the limitations of such an approach to latent action learning in the presence of distractors, similar to existing research. Moreover, instead of VideoSAUR (Zadaianchuk et al., 2023) employed in our work, their method uses SAVi (Kipf et al., 2021) for object-centric learning, which, as we show (see Appendix A) does not work well in more complex environments with real-world distractors.

2 BACKGROUND

Learning from observations. Learning solely from observations (LfO) has emerged recently (McCarthy et al., 2024) to mimic the success of large-scale pretraining in different domains, like text, as a pathway for scalable embodied foundation models. Early efforts like Video PreTraining (VPT) (Baker et al., 2022) demonstrated the potential of pretraining on internet-scale video data (e.g., Minecraft gameplay) to recover latent policies. However, VPT requires costly action labeling via human demonstrations, limiting its scalability. In Ghosh et al. (2023) the authors proposed modeling latent intentions, representations of various outcomes an agent might aim to achieve, to learn useful features from observations data.

Subsequent work shifted to Latent Action Policies (LAPO) (Schmidt & Jiang, 2024), a combination of forward-dynamics model (FDM) $f_{\text{FDM}}(\boldsymbol{o}_{t+1}|\boldsymbol{o}_t, \boldsymbol{a}_t)$, which predict future states from current observations, and inverse-dynamics model (IDM) $f_{\text{IDM}}(\boldsymbol{a}_t|\boldsymbol{o}_t, \boldsymbol{o}_{t+1})$, which infer latent actions from state transitions. FDM and IDM are jointly learned to minimize the next state prediction and further used to label the trajectory of observations with latent actions. As Schmidt & Jiang (2024) show, obtained latent actions can recover ground true actions, however, they assume distractor-free environments, a brittle assumption for real-world video data.

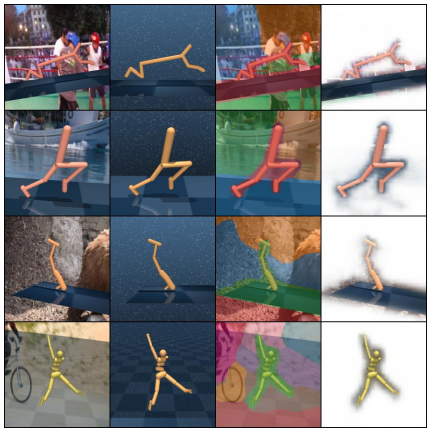

Figure 1: Examples of slot decoder masks on the Distraction Control Suite utilized in our Object-Centric Latent Action Learning pipeline. From top to bottom, the rows correspond to different tasks: cheetah-run, walker-run, hopper-hop, humanoid-walk. From left to right: the distracted observation (background video, color, and camera position variations), the non-distracted observation, the mixture of slot decoder masks obtained after object-centric pretraining, and the main object slot decoder mask selected after object-centric pretraining.

Learning in noisy setting. Recent work by Wang et al. (2023) argues that optimal World Models should suppress such distractors by isolating controllable, reward-relevant factors. There are papers (Efroni et al., 2022) proving the discovery of true latent states from offline trajectories of observations and actions. Real-world videos inherently contain action-correlated distractors: environmental dynamics (e.g., moving backgrounds, camera jitter) that spuriously correlate with agent actions. However, existing LfO methods lack mechanisms to disentangle distractors (Misra et al., 2024a), leading to overfitting in noisy settings.

Object-Centric Pretraining for Videos. Object-centric learning aims to decompose visual scenes into structured, entity-level representations that capture independent objects, i.e. slots. During learning, slots compete with each other in describing the image. Such procedure is named

Slot Attention and firstly it was suggested for images (Locatello et al., 2020). It was later enhanced in SAVi (Kipf et al., 2022) and STEVE Singh et al. (2022) to work with videos by iteratively applying slot attention to consecutive frames to ensure stable temporal order of slots through time. However, naive SlotAttention (Locatello et al., 2020) approaches usually do not scale well to real-world videos. VideoSAUR (Zadaianchuk et al., 2023) utilizes a large-scale pretrained DIVOv2 (Oquab et al., 2023) features, which helps to capture deep concepts instead of low-level RGB and introduces a temporal feature similarity loss, which encodes semantic and temporal correlations between image patches. Such changes bias the model toward discovering moving objects while filtering static distractors.

3 METHOD

Object-Centric Representation Learning. We employ the VideoSAUR (Zadaianchuk et al., 2023) to decompose input video frames into spatio-temporal object slots. This self-supervised model isolates individual entities within a scene, providing structured representations that are less susceptible to background noise and incidental motion. In the end of this step we obtain an encoder F^s, which directly maps a trajectory of observations $\boldsymbol{O}$ into the trajectory of slots $\boldsymbol{S}$. The number of slots K for each observation o_t in $\boldsymbol{S_t}$ is hyperparameter, fixed at the beginning of the training. The resulting representation $\boldsymbol{S}_t$ of the observation $\boldsymbol{o}_i$ is a combination of slot vectors $\boldsymbol{s}_t^k$, i.e. $F^s(\boldsymbol{O}) = \boldsymbol{S}$. Due to having a transformer-based slot decoder, slots $\boldsymbol{s}_t^k$ can be projected to initial observation $\boldsymbol{o}_t$ utilizing attention maps as alpha masks, to obtain object masks $\boldsymbol{m}_t^k$. Further we will denote them as masks (see Figures 1, 4 and 5 for a visualization).

Slot Selection. From the generated object slots, we identify and select those relevant to the agent's interactions. We plan to address the automatic selection of control-related slots in the future. In the DCS environments, the control-related objects are the main agent (cheetah, walker, hopper or humanoid) and the floor. Depending on the number of slots and the environment, they can arise in the same or different slots, so if needed, slots can be combined during this stage by concatenation over selected slots or taking mean over selected masks. In the current pipeline slot selection among slot vectors $\boldsymbol{s}_{ij}^*$ is performed based on the corresponding masks $\boldsymbol{m}_{ij}^*$. Due to fixed slots initialization (see details in Appendix D) it can be done only once for the whole dataset.

Latent Action Modeling. Utilizing the selected object-centric representations, we train a latent action model inspired by the LAPO (Schmidt & Jiang, 2024). The inverse-dynamics model $\boldsymbol{z}_t \sim f^s_{\text{IDM}}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_{t+1})$ and the forward-dynamics model $\hat{\boldsymbol{s}}_{t+q} \sim f^s_{\text{FDM}}(\cdot|\boldsymbol{s}_t, \boldsymbol{z}_t)$ are trained to reconstruct the trajectory of slots $||\hat{\boldsymbol{s}}_{t+1} - \boldsymbol{s}_{t+1}||^2$ (or masks). We denote this as *lapo-slots* and, accordingly, *lapo-masks* for masks representations. Thus, lapo-slots reconstruct next observations in latent space, while lapo-masks reconstruct images, which were filtered based on selected slots masks (see Figures 1 and 5).

Behavior Cloning and Finetuning. The inferred latent actions serve as proxies for actual action labels. We train a behavior cloning (BC) agent to predict these latent actions, using the same dataset as for latent action learning. To evaluate the pre-training effectiveness, as a final stage, we finetune the resulting agent on a limited set of trajectories with ground-truth action labels, in line with (Schmidt & Jiang, 2024; Ye et al., 2024). The scores of the BC agent depending on number of the labeled trajectories are presented on the Figure 3.

4 EXPERIMENTS

Task formulation. To address the robustness to action-correlated distractors, a critical challenge in learning from raw video data, we evaluate on Distracting Control Suite (DCS) (Stone et al., 2021) environment. DSC is an extension of DM Control with three types of real-world visual noise: (1) dynamic backgrounds: 60 diverse videos from DAVIS 2017 (Pont-Tuset et al., 2018), simulating realistic environmental clutter; (2) color variations: hue/saturation shifts that preserve object semantics but degrade low-level features; (3) camera perturbations: randomized pose adjustments mimicking handheld camera noise. The scale of color and camera variations is 0.1. As for the tasks: we selected 4 (in the order of increasing complexity): cheetah-run, walker-run, hopper-hop, humanoid-walk. We trained expert policy optimization agents on trained on privileged state information and collected the dataset with observation-action tuples. Importantly, the level of distractions is tuned so the behavior

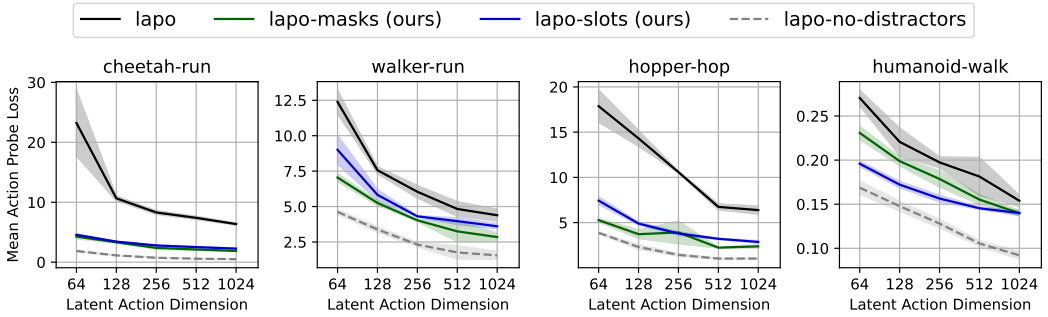

Figure 2: Mean action probes MSE for varying dimensions of latent actions. TL;DR: Object-centric learning improves action probes for all tasks, especially in cheetah-run and hopper-hop. The plots are arranged from left to right in order of increasing complexity of correspoding task. The curves represent the following approaches: *lapo*: baseline latent action pretraining (LAPO) (Schmidt & Jiang, 2024) trained on the Distraction Control Suite (DCS); *lapo-no-distractors*: LAPO trained on non-distracted observation data; *lapo-slots* and *lapo-masks*: models following the object-centric latent action pretraining pipeline described in Section 3. The result are averaged among 3 random seeds.

Table 1: Mean action probes MSE for different tasks. TL;DR: LAPO's performance drops by x5.3 in the presence of distractors. Object-centric learning, using slots and masks, reduces this gap, with notable improvements in tasks like cheetah-run and hopper-hop. The rows represent corresponding tasks from the Distraction Control Suite (DSC), with the last row showing the average action probe MSE across all tasks. The columns represent the following approaches: *lapo*: baseline latent action pretraining (LAPO) (Schmidt & Jiang, 2024) trained on the DCS; *lapo-no-distractors*: LAPO trained on non-distracted observation data; *lapo-slots* and *lapo-masks*: models following the object-centric latent action pretraining pipeline described in Section 3. Each value represents the average MSE across 5 different latent action dimensions and 3 random seeds.

Task	**lapo**	**lapo-masks (ours)**		**lapo-slots (ours)**		**lapo-no-distractors**	
cheetah-run	11.2 ± 6.9	2.8 ± 1.0	x4.0	3.1 ± 0.9	x3.6	1.0 ± 0.6	x12.0
walker-run	7.0 ± 3.2	4.5 ± 1.7	x1.6	5.3 ± 2.2	x1.3	2.7 ± 1.3	x2.6
hopper-hop	11.2 ± 4.9	3.5 ± 1.2	x3.2	4.4 ± 1.8	x2.5	1.9 ± 1.2	x5.8
humanoid-walk	0.20 ± 0.04	0.18 ± 0.04	x1.1	0.16 ± 0.04	x1.3	0.13 ± 0.03	x1.6
average	7.4	2.7	x2.7	3.3	x2.5	1.4	x5.3

cloning (BC) agent, trained on privileged true actions, is able to achieve expert score. More details on dataset collection can be found in Appendix B.

Training details. To conduct the experiments, 4 models were trained: *lapo*, *lapo-no-distractors*, *lapo-slots* and *lapo-masks*. The baseline model is LAPO, which is trained on observations with distractors, following the Schmidt & Jiang (2024) procedure (*lapo* in the figures). We use it as a baseline to demonstrate the currently existing limitations of latent action pretraining. We also trained LAPO on data without distractors as our second baseline to illustrate the performance gap caused by distractors (*lapo-no-distractors* in the figures). LAPO-slots and LAPO-masks are the models that follow the object-centric latent action pretraining training pipeline described in Section 3 (respectively, *lapo-slots* and *lapo-masks* in the figures). All models were trained on the same datasets. More details on training can be found in Appendix C

4.1 Latent Action Quality

To evaluate the quality of obtained latent actions we employed linear probing of latent actions against ground-truth expert actions. Such probe quantifies how well latent actions capture causal dynamics.

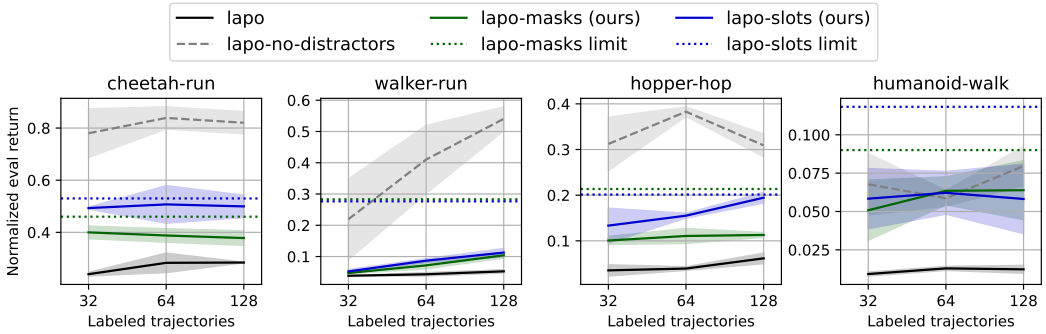

Figure 3: Normalized evaluation returns of the BC agent trained on latent actions for varying numbers of fine-tuning labeled trajectories. TL;DR: Object-centric learning improves evaluation returns for all tasks, showing high sample efficiency on tasks like cheetah-run and hopper-hop. The plots are arranged from left to right in order of increasing task complexity. The curves represent the following approaches: *lapo*: baseline latent action pretraining (LAPO) (Schmidt & Jiang, 2024) trained on the Distraction Control Suite (DCS); *lapo-no-distractors*: LAPO trained on non-distracted observation data; *lapo-slots* and *lapo-masks*: models following the object-centric latent action pretraining pipeline described in Section 3. The horizontal lines labeled *lapo-slots limit* and *lapo-masks limit* denote the scores of the corresponding BC agents fine-tuned on the entire training dataset of 5k trajectories. The values are averaged across three random seeds. The BC agent trained with access to the full dataset of ground-truth actions would return a score of 1 for each task.

The quantitative results of the mean action probes are presented in Figure 2 and in Table 1. The results are averaged among 3 seeds.

Object-slots reduce the gap caused by the distractors. The experiments showed a significant $\mathbf{x5.3}$ gap between the quality of latent actions on non-distracted and distracted observations. On all tasks the gap is reduced by the integration of object-centric leaning at least twice on slots or masks. The best improvements are observed on cheetah-run, 78% on slots and 82% on masks, and hopper-hop, 73% on slots and 83% on masks. Moreover, on the average, object-centric learning reduced mean action probes by $\mathbf{x2.7}$ on masks and $\mathbf{x2.5}$ on slots.

4.2 Downstream Performance

To evaluate the quality of the resulting agents we measure episodic return of a behavior cloning agent, trained on the obtained latent actions and fine-tuned on 32–128 trajectories (1k transitions) with ground-truth actions. Small size of the action-labeled sample is critical for real-world deployment. Each agent is evaluated over 25 episodes in the environment. The scores on the Figure 3 are normalized by the performance of BC trained on the full datasets with all action labels revealed (see Table 3 in the Appendix). The scores on the Table 2 are scaled by 100. The quantitative results of evaluation retuns are presented on the Figure 3 and in the Table 2. The result are averaged among 3 random seeds.

LAPO struggles in the presence of distractors. The gap between LAPO trained in distracted and non-distracted setting is $\mathbf{x3.9}$ among all tasks, which shows the importance a technique, that can filter out the distraction.

Object-slots reduce the gap caused by distractors. Following the results in the Section 4.1 slots and masks outperform baseline lapo for all 4 tasks: by at least $\mathbf{x2.1}$ for slots ($\mathbf{x2.6}$ on the average), and at least $\mathbf{x1.7}$ for masks ($\mathbf{x2.0}$ on the average). The maximal improvement is observed on humanoid-walk: $\mathbf{x7}$ for both slots and masks, while improvement between non-distracted and distracted lapo is $\mathbf{x4.5}$.

For easy tasks slots show higher sample efficiency. Horizontal lines in Figure 3 denote the evaluation score of an average corresponding BC agent finetuned on the whole training dataset of trajec-

Table 2: Normalized evaluation returns of the BC agent trained on latent actions for different tasks. TL;DR: LAPO's performance drops by x3.9 in the presence of distractors. Object-centric learning, using slots and masks, reduces this gap. The rows represent corresponding tasks from the Distraction Control Suite (DSC), with the last row showing the average action probe MSE across all tasks. The columns represent the following approaches: *lapo*: baseline latent action pretraining (LAPO) (Schmidt & Jiang, 2024) trained on the Distraction Control Suite (DCS); *lapo-no-distractors*: LAPO trained on non-distracted observation data; *lapo-slots* and *lapo-masks*: models following the object-centric latent action pretraining pipeline described in Section 3. Each value represents the average MSE across three different amounts of fine-tuning labeled trajectories and three random seeds. The values are scaled by 100, so the BC agent trained with access to the full dataset of ground-truth actions would achieve a score of 100 for each task.

Task	**lapo**	**lapo-masks (ours)**		**lapo-slots (ours)**		**lapo-no-distractors**	
cheetah-run	23.0 ± 6.0	39.0 ± 1.0	x1.7	50 ± 1.0	x2.2	69.0 ± 14.0	x3.1
walker-run	4.0 ± 0.7	7.4 ± 2.8	x1.8	8.4 ± 3.0	x2.1	22.0 ± 19.0	x5.5
hopper-hop	3.2 ± 1.5	10.8 ± 0.6	x3.4	16 ± 3.0	x5.0	25.0 ± 9.0	x7.7
humanoid-walk	0.8 ± 0.4	5.9 ± 0.7	x7.0	6.0 ± 0.2	x7.0	3.8 ± 3.1	x4.5
average	7.7	16	x2.0	20	x2.6	30	x3.9

tories. The figure shows that for cheetah-run only 32 trajectories and 128 trajectories for hopper-hop are enough to achieve this limit of possible evaluation score. This explicitly shows the applicability of the object-centric pretraining for LAM in real-world scenario of limited access to labeled datasets.

Slots perfrom better than masks. Even though Figure 2 shows that latent action quality for masks is slightly better than for slots (2.7 for masks versus 3.3 for slots), evaluation in the environment shows that slots were able to produce latent actions more amenable to pretraining, resulting in 20 % higher returns, than masks (16 for masks versus 20 for slots).

5 Conclusion & Limitations

Our preliminary results demonstrate that object-centric representations significantly mitigate the impact of distractors for learning latent actions from videos. By disentangling scene elements into meaningful, interpretable slots, latent actions focus on causal dynamics rather than spurious correlations. This could provide a strong inductive bias for more effective Latent Action Models in noisy environments.

However, challenges remain. Although object slots reduce noise, the selection of task-relevant slots is still ambiguous. Without explicit supervision, models may focus on irrelevant elements. This highlights a fundamental limitation in current object-centric methods: the difficulty in dynamically attending to the 'right' objects in varying tasks and environments. Another limitation is the dependency on the quality of the training data. Cleaner, well-curated datasets naturally lead to more robust representations, whereas large-scale uncurated video data necessitate additional mechanisms for filtering out noise. This presents a trade-off between data volume, model complexity, and data cleaning efforts. Although curated data simplifies training, it limits generalization and scalability. However, noisier datasets require more sophisticated models, but unlock broader generalization capabilities for embodied AI.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act

by watching unlabeled online videos. *ArXiv*, abs/2206.11795, 2022. URL https://api.semanticscholar.org/CorpusID:249953673. 2

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

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. 1

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. 1

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023. 1

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provable rl with exogenous distractors via multistep inverse dynamics, 2022. URL https://arxiv.org/abs/2110.08847. 2

Dibya Ghosh, Chethan Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions, 2023. URL https://arxiv.org/abs/2304.04782. 2

Pranav Guruprasad, Harshvardhan Sikka, Jaewoo Song, Yangyue Wang, and Paul Pu Liang. Benchmarking vision, language, & action models on robotic learning tasks. *arXiv preprint arXiv:2411.05821*, 2024. 1

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. 2

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video, 2022. URL https://arxiv.org/abs/2111.12594. 3, 9

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024. 1

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020. URL https://arxiv.org/abs/2006.15055. 1, 3

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. *arXiv preprint arXiv:2404.19664*, 2024. 1, 2

Dipendra Misra, Akanksha Saran, Tengyang Xie, Alex Lamb, and John Langford. Towards principled representation learning from videos for reinforcement learning, 2024a. URL https://arxiv.org/abs/2403.13765. 2

Dipendra Misra, Akanksha Saran, Tengyang Xie, Alex Lamb, and John Langford. Towards principled representation learning from videos for reinforcement learning. *arXiv preprint arXiv:2403.13765*, 2024b. 1

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 3, 10

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018. URL https://arxiv.org/abs/1704.00675. 3

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. 1

Dominik Schmidt and Minqi Jiang. Learning to act without actions, 2024. URL https://arxiv.org/abs/2312.10812. 1, 2, 3, 4, 5, 6

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos, 2022. URL https://arxiv.org/abs/2205.14065. 3, 9

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite – a challenging benchmark for reinforcement learning from pixels, 2021. URL https://arxiv.org/abs/2101.02722. 2, 3

Angel Villar-Corrales and Sven Behnke. Playslot: Learning inverse latent dynamics for controllable object-centric video prediction and planning, 2025. URL https://arxiv.org/abs/2502.07600. 2

Tongzhou Wang, Simon S. Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself, 2023. URL https://arxiv.org/abs/2206.15477. 2

Yucen Wang, Shenghua Wan, Le Gan, Shuai Feng, and De-Chuan Zhan. Ad3: Implicit action is the key for world models to distinguish the diverse visual distractors. *arXiv preprint arXiv:2403.09976*, 2024. 1

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 10

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos, 2024. URL https://arxiv.org/abs/2410.11758. 1, 3

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities, 2023. URL https://arxiv.org/abs/2306.04829. 2, 3, 10

A Comparing STEVE and VideoSAUR slot projections

We started from STEVE (Singh et al., 2022) model, as one of widely used and promising approaches. However, STEVE was not able to select the hopper on the images. You can compare STEVE's and VideoSAUR's slot projections on Figure 4. We also note that STEVE is mostly identical to SAVi (Kipf et al., 2022), differentiated only by the fact that it uses a transformer-based decoder instead of a pixel-mixture decoder.

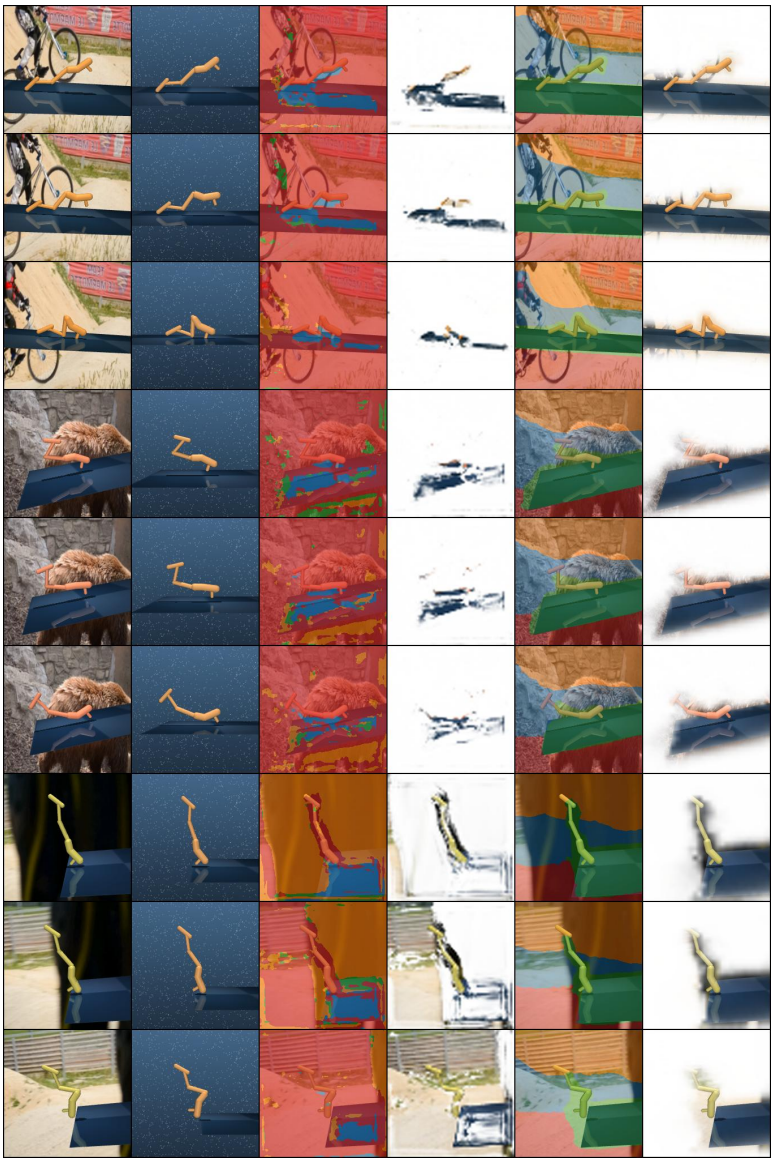

Figure 4: Examples of slot projections on DCS hopper-hop. From left to right: clean image without distractors, image with distractors, STEVE's slots projections, the main object's slot projection by STEVE, VideoSAUR's slot projections, the main object's slot by VideoSAUR

B Dataset collection

The datasets were collected via experts policies trained on Disctracting Control Suite via PPO (for cheetah-run, walker-run and hopper-hop) and SAC (for humanoid-walk). The scores of the experts

are presented on the Table 3. The final dataset of transitions for each tasks consists of 5k trajectories (1k transitions each). The observations in the dataset have height and width 64px.

Table 3: Comparison of the Performance of Algorithms. *Expert* denotes the policy used to collect the dataset trained on privileged information about minimal state of the observation, PPO (for cheetah-run, walker-run and hopper-hop) and SAC (for humanoid-walk). *BC Vanilla* denotes the scores of behavior cloning agents (BC) trained on full expert dataset to imitate expert policy on the privileged for our method true action labels and non-distracted observations. *BC* denotes the scores of BC agents trained on full expert dataset to imitate expert policy on the distracted observations and privileged for our method true action labels.

Task	**Expert**	**BC no-distraction**	**BC**
cheetah-run	838	840	823
walker-run	740	735	749
hopper-hop	307	300	253
humanoid-walk	617	601	428

C TRAINING DETAILS

All experiments were run in H100 GPU. The models are trained in bfloat16 precision. Training duration is shown in Table 5

Object-centric learning pretraining codebased was adopted from Videosaur Zadaianchuk et al. (2023). It utilizes DINOv2 (Oquab et al., 2023) pretrained encoder vit-base-patch8-224-dino from Timm (Wightman, 2019) models hub. The images in the dataset are upscaled for dino encoder up to 224px. The hyperparameters for object-centric pretraining can be found in Table 6

Latent action learning model for images and object-centric masks is formed from IDM and FDM models based on a ResNet-like CNN encoder and decoder. The hyperparameters for latent action learning from images (used for lapo, lapo-masks) can be found in Table 8

Latent action learning model for object-centric slots is formed from IDM and FDM based on 3-layer MLP blocks with residual connections and GeLU activations to effectively process vector representations. The hyperparameters for latent action learning from representations (used for lapo-slots) can be found in Table 8

Table 4: Amount of parameters for different models. The rows represent the following approaches: *ocp*: denotes the number of parameters of the object-centric model; *lapo-slots*: denotes the number of parameters for latent action learning from vector representations; utilizing object-centric representations from a precollected dataset; *lapo*: denotes the number of parameters for latent action learning from images; *lapo-masks*: denotes the number of parameters for latent action learning from images, utilizing object-centric masks from a precollected dataset (using the same model as *lapo*); *bc*: Denotes the number of parameters of the behavior cloning agent trained on latent actions.

Method	**Number of parameters**
ocp (total)	92,149,776
ocp (trainable)	6,343,440
lapo-slots	89,186,432
lapo-masks	211,847,849
lapo	211,847,849
bc	107,541,504

Table 5: Average training duration of the methods. The row represent the following approaches: *ocp* denotes the time spent on object-centric pretraining stage which is common for both slots and masks; *lapo-masks* and *lapo-slots* denote the time spent on training latent action model, reading object-centric representations from a precollected dataset; *lapo* denote the time spent on training latent action model, *ocp + lapo-masks* and *ocp + lapo-slots* denote the time for full pipeline of object-centric latent action learning

Method	**Training Duration**
ocp	$\sim$ 6 h 35 m
lapo-slots	$\sim$ 2 h 29 m
lapo-masks	$\sim$ 7 h 38 m
lapo	$\sim$ 7 h 38 m
ocp + lapo-slots	$\sim$ 9 h 4 m
ocp + lapo-masks	$\sim$ 14 h 13 m
bc + finetuning	$\sim$ 3 h 2 m

D FIXED INITIALIZATION FOR SLOT STABILITY

To mitigate slot permutation variance across predictions, we introduce a fixed slot initialization scheme that learns deterministic initial slot vectors while preserving robustness. Unlike standard Gaussian initialization, which samples slots stochastically at each step, our approach learns per-slot parameters (mean $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and variance $\boldsymbol{\sigma}_k \in \mathbb{R}^d$) during training. During training, we inject controlled noise scaled by the learned variance into the slot initializations, acting as a regularizer to encourage robust feature disentanglement. At inference, slots are initialized deterministically using the learned means, ensuring consistent slot-object assignments. This hybrid strategy bridges the gap between training stability and inference consistency: the noise-augmented training phase prevents overfitting to fixed initializations, while the deterministic inference phase enables efficient object-wise slot selection via decoder masks as visual priors.

$$\texttt{Train:} \quad \boldsymbol{s}_k^{(\text{init})} = \boldsymbol{\mu}_k + \boldsymbol{\sigma}_k \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \texttt{Inference:} \quad \boldsymbol{s}_k^{(\text{inference})} = \boldsymbol{\mu}_k.$$

E EXAMPLES OF VIDEOSAUR SLOT PROJECTIONS

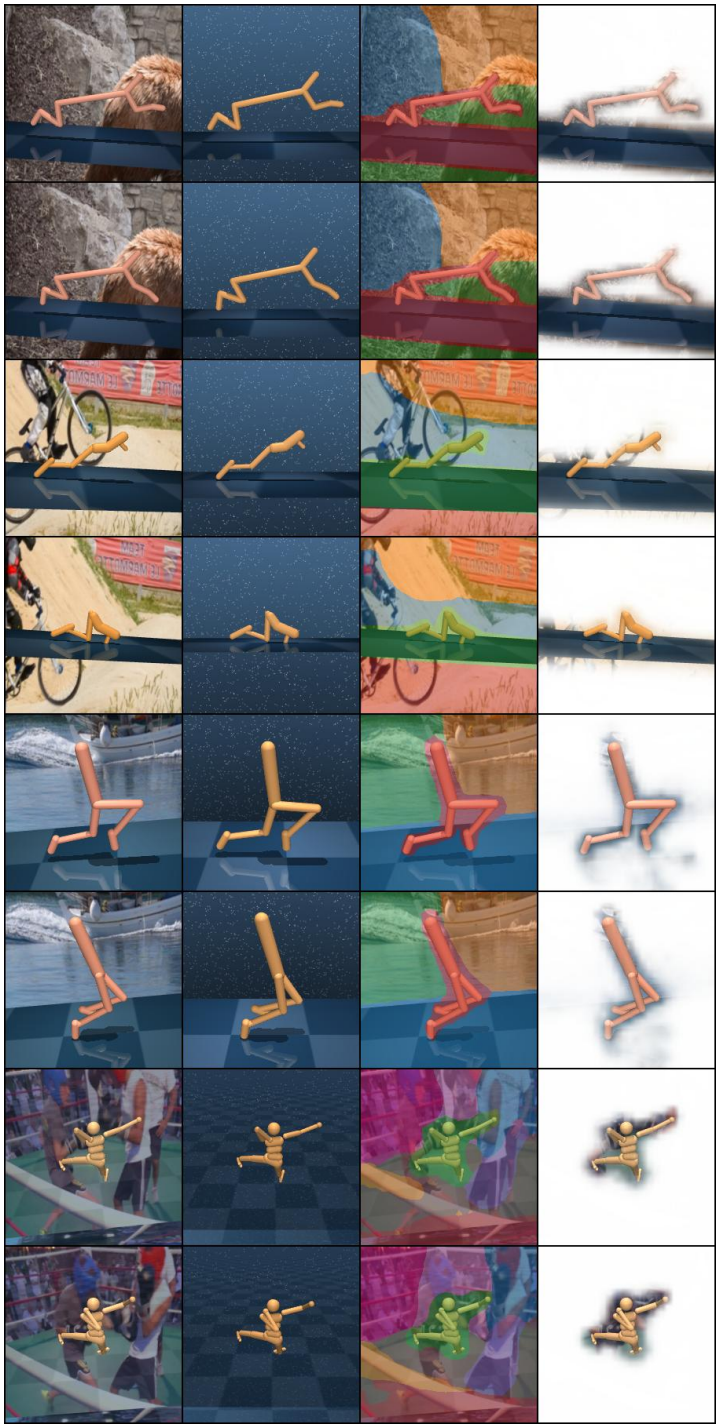

Figure 5: Examples of VideoSAUR slot projections on DCS for 4 tasks: (from upper to lower) cheetah-run, hopper-hop, walker-run, humanoid-walk. From left to right: distracted observation, clean observation, observation with segments of slot projections, slot projection (we call it "mask") of the main object

Table 6: Object-centric pretraining hyperparams. Number of slots varies across tasks: 4 for cheetah-run, walker-run, hopper-hop and 8 for humanoid-walk

Hyperparameter	**Value**
Episode Length	3
Image Size Dataset	64
Image Size Resize	224
Max Steps	100000
Number of Slots	4
Batch Size	256
Warmup Steps	2500
Weight Decay	0
Max Video Length	1000
Gradient Clip Value	0.05
Slot Dimension	128
Vision Transformer Model	vit_base_patch8_224_dino
Feature Dimension	768
Number of Patches	784
Batch Size per GPU	128
Total Batch Size	128
Similarity Temperature	0.075
Similarity Weight	0.1
Base Learning Rate	0.0001
Learning Rate	0.0003
Learning Rate Scheduler	exp_decay_with_warmup
Warmup Steps	2500
Decay Steps	100000

Table 7: Hyperparameters for latent action learning from vector representations (used for *lapo-slots*).

Hyperparameter	**Value**
Latent Action Learning	
Batch Size	8192
Hidden Dimension	1024
Number of Epochs	30
Frame Stack	1
Weight Decay	0
Learning Rate	0.00005
Warmup Epochs	3
Future Observation Offset	10
Latent Action Dimension	8192
BC	
Dropout	0
Use Augmentation	False
Evaluation Seed	0
Batch Size	512
Number of Epochs	10
Frame Stack	3
Encoder Deep	False
Weight Decay	0
Encoder Scale	32
Evaluation Episodes	5
Learning Rate	0.0001
Warmup Epochs	0
Encoder Number of Residual Blocks	2
BC finetuning	
Use Augmentation	False
Batch Size	512
Hidden Dimension	256
Weight Decay	0
Evaluation Episodes	25
Learning Rate	0.0003
Total Updates	2500
Warmup Epochs	0

Table 8: Hyperparameters for latent action learning from images (used for *lapo*, *lapo-masks*).

Hyperparameter	**Value**
Latent Action Learning	
Batch Size	512
Number of Epochs	10
Frame Stack	3
Encoder Deep	False
Weight Decay	0
Encoder Scale	6
Learning Rate	0.00005
Warmup Epochs	3
Future Observation Offset	10
Latent Action Dimension	1024
Encoder Number of Residual Blocks	2
BC	
Dropout	0
Use Augmentation	False
Evaluation Seed	0
Batch Size	512
Number of Epochs	10
Frame Stack	3
Encoder Deep	False
Weight Decay	0
Encoder Scale	32
Evaluation Episodes	5
Learning Rate	0.0001
Warmup Epochs	0
Encoder Number of Residual Blocks	2
BC finetuning	
Use Augmentation	False
Evaluation Seed	0
Batch Size	512
Hidden Dimension	256
Weight Decay	0
Evaluation Episodes	25
Learning Rate	0.0003
Total Updates	2500
Warmup Epochs	0