LEARNING TO ACT WITHOUT ACTIONS

Dominik Schmidt*
University College London

Minqi Jiang
FAIR at Meta AI

ABSTRACT

Pre-training large models on vast amounts of web data has proven to be an effective approach for obtaining powerful, general models in several domains, including language and vision. However, this paradigm has not yet taken hold in deep reinforcement learning (RL). This gap is due to the fact that the most abundant form of embodied behavioral data on the web consists of videos, which do not include the action labels required by existing methods for training policies from offline data. We introduce Latent Action Policies from Observation (LAPO), a method to infer latent actions and, consequently, latent-action policies purely from action-free demonstrations. Our experiments on challenging procedurally-generated environments show that LAPO can act as an effective pre-training method to obtain RL policies that can then be rapidly fine-tuned to expert-level performance. Our approach serves as a key stepping stone to enabling the pre-training of powerful, generalist RL models on the vast amounts of action-free demonstrations readily available on the web.

1 INTRODUCTION

Training on web-scale data has shown to be an effective approach for obtaining powerful models with broad capabilities in domains including language and vision (Radford et al., 2019; Caron et al., 2021). Much recent work has thus sought to apply the same paradigm in deep reinforcement learning (RL, Sutton & Barto, 2018), in order to learn generalist policies from large amounts of web data (Baker et al., 2022; Reed et al., 2022). However, common methods for learning policies from offline demonstrations, such as *imitation learning* (Pomerleau, 1988; Ross & Bagnell, 2010) and *offline RL* (Kumar et al., 2020; Levine et al., 2020), generally require data to include action or reward labels, which are missing from observational data, such as videos found on the web.

In this work we introduce Latent Action Policies from Observation (LAPO), a method for learning useful, pre-trained policies from purely observational data. LAPO is founded on the key insights that (1) some notion of a *latent action* that explains each environment transition can still be inferred from observations alone, and (2) given such inferred latent actions per transition, a *latent-action policy* can be obtained using standard imitation learning methods. Our experiments provide strong evidence that such latent policies accurately capture the observed behavior of the expert by showing that the latent policy can be rapidly finetuned to recover expert-level performance in the observed tasks. This fact makes the approach underlying LAPO a significant stepping stone toward pre-training general, rapidly-adaptable policies on massive action-free demonstration datasets, such as the vast quantities of videos available on the web.

Crucially, LAPO learns to infer latent actions, and consequently, obtain latent action policies in a fully unsupervised manner. LAPO is similar to prior work (Torabi et al., 2018; Schmeckpeper et al., 2020; Baker et al., 2022; Zheng et al., 2023) in that we first train an *inverse dynamics model* (IDM) that predicts the action taken between two consecutive observations, and then use this IDM to add action labels to a large action-free dataset. However, unlike these prior works, which rely on some amount of ground-truth action-labelled data for training the IDM, LAPO does not make use of any labels and infers latent action information purely from observed environment dynamics. To do so, the IDM in LAPO learns to predict *latent actions* rather than true actions. These latent actions take the form of a learned representation that explains an observed transition. In order to learn this representation, LAPO makes use of a simple unsupervised objective that seeks to establish predictive

*Correspondence to dominik.schmidt.22@ucl.ac.uk.

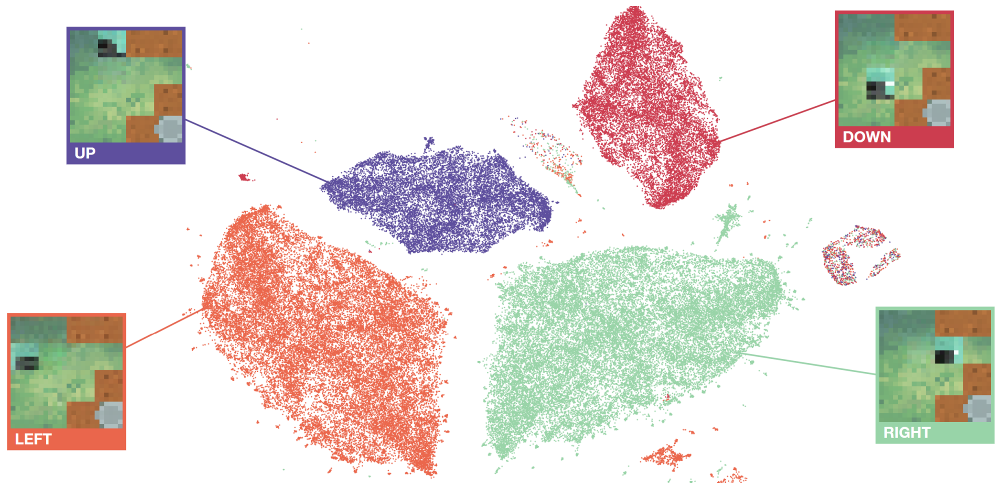

Figure 1: UMAP projection of the learned latent action space for `Miner` alongside sample next-state predictions generated by the FDM for each cluster of latent actions. Each point represents a continuous (i.e. pre-quantization) latent action generated by the IDM for a transition in the observation-only dataset. Each point is color-coded by the true action taken by the agent at that transition. For clarity, `NOOP` actions are omitted. The structure of the latent action space is highly interpretable and closely corresponds to the true action space, even though no ground-truth action labels were used during training.

consistency between the IDM and a separate *forward dynamics model* (FDM). The FDM is trained on transitions consisting of two consecutive observations (o_t, o_{t+1}) to predict the future observation o_{t+1}, given the past observation o_t and a latent action. The latter is generated by the IDM after observing both past and future observations. Thus, unlike the FDM which sees only the past, the IDM has access to both the past and future and learns to pass useful information about the future to the FDM through the latent action. By making the latent action an information bottleneck (Tishby et al., 2000), we prevent the IDM from simply forwarding the entire future observation to the FDM and instead force it to learn a highly compressed encoding of state transitions. We refer to this encoding as a latent action, as it captures the visible effects of the agent's true actions.

Our experiments in Section 6 evaluate the potential of LAPO as a method for pre-training useful RL policies from action-free demonstrations. First, we train a latent IDM via LAPO on large expert-level action-free offline datasets for each of the 16 games of the Procgen Benchmark (Cobbe et al., 2019; 2020). We observe that the structure of the learned latent action spaces are highly interpretable, often exhibiting clusters that closely align with the true action space (see Figure 1). This is a remarkable result, as our approach uses zero ground-truth action labels and instead recovers all action information purely from observed dynamics. Next, we leverage the learned latent representation for downstream online RL tasks. For this, we first use the IDM to assign latent action labels to each transition in the same observation-only dataset on which it was trained. We then perform behavior cloning on the resulting action-observation dataset to learn a latent-action policy, which seeks to imitate the latent actions predicted by the IDM. Finally, we seek to transfer the latent action policy to the true action space, in order to deploy and evaluate it in the online environment. We show that simply fine-tuning the last layers of the latent policy in the original environment with a general-purpose RL algorithm allows the policy to rapidly adapt to the true action space and recover (or at times exceed) expert performance. Further, we show that we can alternatively train a latent to true action decoder on a small labeled dataset (on the order of a few hundred transitions). This action decoder, when composed with the latent action policy, attains significantly better performance than training PPO from scratch with no need for environment interaction or reward labels. This indicates that latent policies produced by LAPO capture meaningful behavior in the latent action space.

In summary, we show that a simple unsupervised objective can be used to recover latent action information and learn latent policies purely from observed environment dynamics. Our experiments across challenging procedurally-generated environments show that the latent representations learned by our method capture much of the structure underlying the true action space. We show that the resulting latent-space policies can be rapidly adapted to recover expert-level performance in the true action space. We believe our approach can serve as a key stepping stone towards training powerful, generalist policies on web-scale video-based demonstrations.

2 Related work

Most similar to our approach, ILPO (Edwards et al., 2019) also aims to infer latent actions via a dynamics modelling objective. However, unlike LAPO, ILPO learns a latent policy rather than an IDM jointly with the FDM, and uses a discrete rather than a continuous latent action space. For each FDM update, ILPO identifies the discrete latent action that minimizes the next-state prediction error, and only optimizes the FDM for that specific action. In practice, this objective can lead to mode collapse, where a feedback loop causes only a small number of (or only one) discrete latent to ever be selected (Struckmeier & Kyrki, 2023). By using an IDM rather than a discrete latent policy, LAPO avoids enumerating all hypothetically possible transitions and instead learns the latent action corresponding to the actual observed transition. To train the policy, ILPO minimizes the difference between the true next state and the expectation of the next state under the policy. This loss is ill-conditioned, as even when it is minimized, the next-state predictions for individual latent actions do not have to align with the environment. Moreover, LAPO's use of continuous rather than finite discrete latents may allow better modeling of more complex environment dynamics, as any information useful for prediction can be captured in the latent action representation. Further, ILPO's use of discrete latent actions makes its computational complexity linear rather than constant in the size of the latent action space, due to the required enumeration of all actions.

In the model-based setting, FICC (Ye et al., 2023) pre-trains a world model from observation-only demonstrations. FICC thus seeks to distill dynamics from demonstrations, rather than behaviors, as done by LAPO. Moreover, while the cyclic consistency loss used by FICC closely resembles the LAPO objective, the adaptation phase of FICC must operate in reverse, mapping true actions to latent actions in order to finetune the world model. Unlike the approach taken by LAPO, this adaptation strategy cannot make direct use of online, ground-truth action labels, and requires continued training and application of a world model. In contrast, LAPO directly produces latent action policies imitating the expert behavior, which can be rapidly adapted online to expert-level performance. Outside of RL, Playable Video Generation (PVG, Menapace et al., 2021) similarly focuses on world model learning with a similar objective for the purpose of controllable video generation. Like ILPO, PVG uses a more limiting, small set of discrete latent actions.

A related set of semi-supervised approaches first train an IDM using a smaller dataset containing ground-truth action labels, and then use this IDM to label a larger action-free dataset, which can subsequently be used for behavior cloning. VPT (Baker et al., 2022) and ACO (Zhang et al., 2022) follow this approach, training an IDM on action-labeled data which is then used to generate pseudo labels for training policies on unlabeled footage gathered from the web. RLV (Schmeckpeper et al., 2020) uses an observation-only dataset within an online RL loop, where action and reward labels are provided by an IDM trained on action-labelled data and a hand-crafted reward function respectively. Similarly, BCO (Torabi et al., 2018) trains an IDM through environment interaction, then uses the IDM to label action-free demonstrations for behavior cloning. However, relying on interactions for labels can be inefficient, as finding useful labels online may itself be a difficult RL exploration problem. SS-ORL (Zheng et al., 2023) is similar to VPT but performs offline RL instead of imitation learning on the IDM-labelled data and requires reward-labels. Unlike these methods, LAPO avoids the need for significant amounts of action-labeled data to train an IDM, by directly inferring latent actions and latent policies which can easily be decoded into true actions.

LAPO differs from previous methods for *imitation learning from observation* (IfO, Torabi et al., 2019a;b; Yang et al., 2019), which typically require the imitating policy to have access to the true action space when training on action-free demonstrations. Crucially, unlike these prior approaches, LAPO does not require access to the ground-truth action space to learn from action-free demonstrations. Other methods leverage observation-only demonstrations for purposes other than learning useful behaviors. Intention-Conditioned Value Functions (ICVF Ghosh et al., 2023) uses a temporal difference learning objective based on observation-only demonstrations to learn state representations useful for downstream RL tasks. Aytar et al. (2018) propose to use expert demonstrations during online RL to guide the agent along the expert's trajectories. For this, they propose two auxiliary classification tasks for learning state representations based on observation-only demonstrations.

3 BACKGROUND

3.1 REINFORCEMENT LEARNING

Throughout this paper, we consider RL under partial observability, using the framework of the partially-observable Markov decision process (POMDP, Åström, 1965; Kaelbling et al., 1998). A POMDP consists of a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, and γ is the discount factor. At each timestep t, the RL agent receives an observation o_t derived from the state s_t, according to the observation function $\Omega : \mathcal{S} \mapsto \mathcal{O}$, and takes an action according to its policy $\pi(a_t|o_t)$, in order to maximize its expected discounted return, $\sum_{k=t}^{\infty} \gamma^{k-t} r_k$. The environment then transitions to the next state s_{t+1} according to the transition function, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$, and agent receives a reward r_t based on the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$. This work considers first learning a policy π from offline demonstrations, followed by further fine-tuning the policy online as the agent interacts with its environment.

3.2 LEARNING FROM OBSERVATIONS

Often, we have access to recordings of a performant policy, e.g. a human expert, performing a task of interest. When the dataset includes the action labels for transitions, a supervised learning approach called *behavior cloning* (BC, Pomerleau, 1988) can be used to train an RL policy to directly imitate the expert. Consider a dataset D of such expert trajectories within the environment, where each trajectory τ consists of a list of all transition tuples $(o_0, a_0, o_1), \ldots, (o_{|\tau|-1}, a_{|\tau|-1}, o_{|\tau|})$ in a complete episode within the environment. BC then trains a policy π_{BC} to imitate the expert by minimizing the cross-entropy loss between the policy's action distribution and the expert's action a^* for each observation in D: $\mathcal{L}_{\text{BC}} = -\frac{1}{|D|} \sum_{\tau \in D} \sum_{t=0}^{|\tau|} \log(\pi(a_t^*|o_t))$.

Unfortunately, most demonstration data in the wild, e.g. videos, do not contain action labels. In this case, the demonstration data simply consists of a continuous stream of observations taking the form $(o_0, o_1, \ldots, o_{|\tau|})$. This setting, sometimes called *imitation learning from observations* (IfO, Torabi et al., 2019a), poses a more difficult (and practical) challenge. IfO methods often seek to learn a model that predicts the missing action labels, typically trained on a separate dataset with ground-truth actions. Once labeled this way, the previously action-free dataset can then be used for BC. In this work, we likewise convert the IfO problem into the BC problem. However, instead of relying on access to a dataset with ground-truth action labels, our method directly infers *latent actions* z_t that explain each observed transition (o_t, o_{t+1}), with which we train *latent action policies*, $\tilde{\pi}(z_t|o_t)$.

3.3 DYNAMICS MODELS

LAPO employs two kinds of dynamics models: The first is the *inverse dynamics model*, $p_{\text{IDM}}(a_t|o_t, o_{t+1})$, which predicts which action a_t was taken by the agent between consecutive observations o_t and o_{t+1}. An IDM can be used to label a sequence of observations with the corresponding sequence of actions. The second is the *forward dynamics model*, $p_{\text{FDM}}(o_{t+1}|o_t, a_t)$, which predicts the next observation o_{t+1} given the previous observation o_t and the action a_t taken by the agent after observing o_t. The FDM can be used as an approximation of the environment's transition function. In this work, the IDM and FDM are deterministic and implemented as deep neural networks. Unlike a standard IDM, the IDM used by LAPO predicts continuous latent actions, $z_t \in \mathcal{Z}$, where $\mathcal{Z} = \mathbb{R}^n$.

3.4 VECTOR-QUANTIZATION

Vector-quantization (Gray, 1984, VQ) is a method for learning discrete features by quantizing an underlying continuous representation. This has been shown particularly useful in deep learning, where VQ enables learning discrete representations while allowing gradients to flow through the quantization step. VQ has been used effectively in many domains including vision (van den Oord et al., 2017; Ramesh et al., 2021b; Huh et al., 2023), audio (Dhariwal et al., 2020; Zeghidour et al., 2022), and model-based RL (Micheli et al., 2023). To quantize a continuous vector $\mathbf{z}$, VQ maintains a codebook $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ and maps $\mathbf{z}$ to the closest codebook vector $\mathbf{c}_i$. The straight-through gradient estimator is used to pass gradients through the quantization step (Bengio et al., 2013).

4 Latent Action Policies from Observation

4.1 Learning a Latent Action Representation

We now describe our approach for learning a latent IDM via a forward dynamics modelling objective (see Fig. 2). First, to learn about the action at time t, we sample a sequence of observations $o_{t-k}, \ldots, o_t, o_{t+1}$ from our observation-only dataset. Here, o_t and o_{t+1} are the observations before and after the action of interest is taken and observations $o_{t-k}, \ldots, o_{t-1}$ are additional context controlled by hyperparameter $k \geq 0$. The IDM then predicts the latent action z_t based on the full sequence of observations.

$$z_t \sim p_{\text{IDM}}(\cdot | o_{t-k}, \ldots, o_t, o_{t+1})$$

Next, the FDM predicts the post-transition observation o_{t+1} based only on the past observations $o_{t-k}, \ldots, o_t$ and the latent action z_t.

$$\hat{o}_{t+1} \sim p_{\text{FDM}}(\cdot | o_{t-k}, \ldots, o_t, z_t)$$

Both models are trained jointly via gradient descent to minimize the next state prediction error $||\hat{o}_{t+1} - o_{t+1}||^2$.

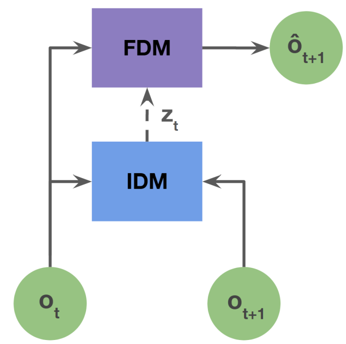

Figure 2: LAPO architecture. Both IDM and FDM observe o_t, but only the IDM observes o_{t+1}. To enable accurate predictions of o_{t+1}, the IDM must pass useful transition information through the quantized information bottleneck z_t to the FDM.

This method, as described so far, would likely result in the IDM learning to simply copy o_{t+1} into z_t, and the FDM learning to forward z_t as is. To remedy this, we make the latent action an information bottleneck. This forces the IDM to compress any information to be passed to the FDM. Since both the FDM and IDM have access to past observations but only the IDM observes the post-transition observation o_{t+1}, the IDM is learns to encode only the *difference* between o_{t+1} and o_t into z_t, rather than full information about o_{t+1}. Naturally, one highly efficient encoding of the differences of two consecutive observations, at least in deterministic environments, is simply the agent's true action. Our hypothesis is thus, that forcing the IDM to heavily compress transition information as described, may allow us to learn a latent action representation with a structure closely corresponding to the true action space.

As both the IDM and FDM have access to past observations, the learned latent actions may become conditional on these observations. Intuitively, this means that some latent action z could correspond to different true actions when performed in different states. While this is not necessarily an issue, biasing the method toward simpler representations is likely preferrable (Solmonoff, 1964; Schmidhuber, 1997; Hutter, 2003). Consequently, we apply vector quantization (VQ) to each latent action before passing it to the FDM, thus forcing the IDM to reuse the limited number of discrete latents across different parts of the state-space, leading to disentangled representations.

4.2 Behavior Cloning a Latent Action Policy

Using the trained latent IDM, we now seek to obtain a policy. For this, we initialize a latent policy $\pi\colon \mathcal{O} \to \mathcal{Z}$ and perform behavior cloning on the same observation-only dataset that the IDM was trained on, with the required action labels generated by the IDM. This is done via gradient descent with respect to policy parameters on the loss $||\pi(o_t) - z_t||^2$ where $z_t \sim p_{\text{IDM}}(\cdot | o_{t-k}, \ldots, o_t, o_{t+1})$.

4.3 Decoding Latent Actions

The policy π, obtained via BC, produces actions in a continuous latent space. We consider two ways to adapt π to produce outputs in the true action space of the online RL environment.

Online RL policy fine-tuning When given access to the online RL environment, we can simply finetune the latent action policy in the environment using RL. This approach, which we refer to as online RL-fine-tuning, treats the latent behavior cloning step as a pretraining phase. During RL fine-tuning, the policy only needs to be adapted slightly to output actions in the true action space, e.g. by replacing the latent action policy head with a true action policy head. Moveover, the performance of the policy can potentially be improved beyond that of the data-generating policy. In particular, when the data was generated by a mixture of policies of different skill levels or pursuing different goals (as is the case with data from the web), this step can extract the data-generating policies more closely aligned with the reward function used for fine-tuning. Our experiments make use of a single data-generating expert, which can exhibit a similar diversity of outcomes across trajectories.

Offline action decoding If only offline, action-labeled transitions are available, we can train a small decoder network $d\colon \mathcal{Z} \to \mathcal{A}$, mapping the predicted latent action for each labeled transition to the ground-truth action. This decoder can be composed with the latent policy and the resulting decoded latent policy $d \circ \pi$ can be deployed in the online environment.

5 Experimental Setting

Our experiments center on the Procgen Benchmark (Cobbe et al., 2020), as it features a wide variety of tasks that present different challenges for our method. Compared to other classic benchmarks such as Atari (Bellemare et al., 2013), the procedural generation underlying Procgen introduces a difficult generalization problem and results in greater visual complexity which makes dynamics modeling at pixel-level challenging. Moreover, several Procgen environments feature partial observability, which along with stochasticity, presents issues for methods attempting to infer actions purely from observation by making it ambiguous which parts of an environment transition are due to the agent and which are due to stochastic effects or unobserved information. Our observation-only dataset consists of approximately 8M frames sampled from an expert policy that was trained with PPO for 50M frames. The use of this synthetic dataset, rather than video data from the web, allows us to better evaluate our method, as we can directly access the expert policy's true actions, as well as metadata such as episodic returns for evaluation purposes.

We use the IMPALA-CNN introduced by Espeholt et al. (2018) to implement both our policy and IDM with a 4x channel multiplier as used by Cobbe et al. (2020), and U-Net (Ronneberger et al., 2015) based on a ResNet backbone (He et al., 2016) with approximately 8M parameters for the FDM. In the offline action decoding setting, the decoder is a fully-connected network with hidden sizes $(128, 128)$. We use an EMA-based update (Polyak & Juditsky, 1992) for the vector quantization embedding. We use a single observation of additional pre-transition context, i.e. $k = 1$.

We use Proximal Policy Optimization (PPO, Schulman et al., 2017) for online fine-tuning and initialize the policy with the weights of the latent policy, except for the last layer, which is re-initialized with output dimensionality $|\mathcal{A}|$ instead of $|\mathcal{Z}|$. We keep all convolutional layers frozen and only fine-tune the actor and policy heads as well as the previous fully-connected layer. We found that a much larger learning rate of 0.01 can be stably used when only fine-tuning these three layers. We similarly tuned the learning rate for training the full network from scratch, but found no improvement compared to the original value of 5e-4. Other hyperparameters are given in Appendix A.4.

6 Results and Discussion

6.1 Online RL policy fine-tuning

We now discuss our results for fine-tuning the latent policy in the online environment using RL. Recall, that this is the third stage of our approach, after first training a latent IDM and then obtaining a latent policy via behavior cloning. In this experiment, we investigate how LAPO's pretraining step is able to speed up online RL compared to learning a policy from scratch. Since our work focuses on the setting where large amounts of observation-only data are freely available, but environment interaction is limited or expensive, we run PPO for 4M steps rather than the 25M steps suggested by Cobbe et al. (2020). Experiments are performed separately per environment. As can be seen in Figure 3, fine-tuning policies pretrained with LAPO recovers expert-level performance within only 4M frames, while PPO from scratch reaches only 44% of expert performance in the same period. We highlight that in 8 of 16 games, we are able to reach 80% of expert performance within only 300k frames, and in 3 of 16 games, our approach exceeds expert performance within 4M frames. We also provide results for an ablation of our method that does not apply vector-quantization to latent actions, showing that VQ is an important component of our method.

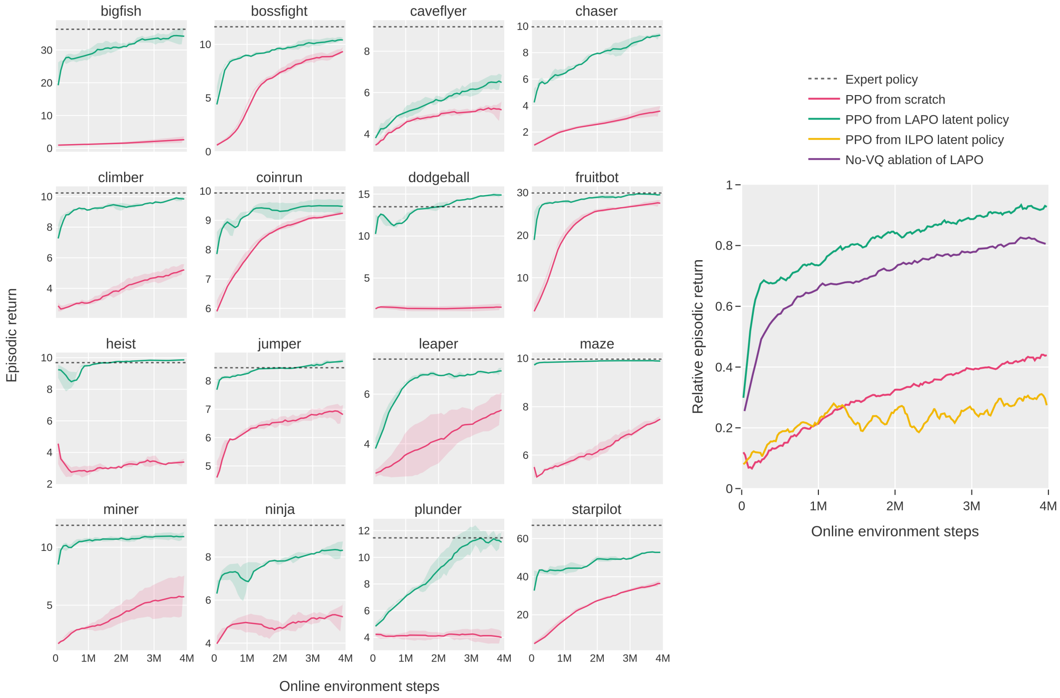

Figure 3: Left: Mean test returns over the course of training for PPO fine-tuning of a pretrained LAPO policy and PPO from scratch (averaged across 3 seeds). Right: Mean test returns relative to per-environment expert policies averaged across all 16 Procgen environments. Error bars indicate standard deviation across seeds.

We use ILPO as our primary baseline, however we found that its policy immediately collapses in several Procgen tasks. In tasks where it does not collapse, online fine-tuning of the ILPO policy did not perform better than learning from scratch. While Edwards et al. (2019) do provide positive results for the task `CoinRun` (which we reproduce in Appendix A.5), their experiments target only a single level of the environment. We thus hypothesize that ILPO's policy collapse is in part due to the modeling challenge posed by the more visually diverse levels in the fully procedurally-generated setting. By using an IDM rather than a policy plus a high-dimensional, continuous latent action space, LAPO can better model stochasticity (such as visual noise) and unobserved information (such as off-screen parts of the game). In contrast, it is likely difficult for ILPO to model the full distribution of possible transitions using only a small set of discrete latents.

6.2 Offline action decoding

Next, we consider an offline approach to training a latent action decoder. Here, we simply train a decoder network that maps latent actions to true actions using a small amount of action-labeled demonstration data. We then compose the decoder with the latent policy and evaluate this decoded latent policy in the environment. As we will see in Section 6.3, the latent and true action spaces share similar structure. This similarity allows an effective decoder to be trained on only a miniscule amount of action-labeled data. Figure 4 shows that an offline decoder trained on less than 256 labeled transitions achieves the same performance as a policy trained from scratch for 4M steps with PPO. However, we observe that with increasing dataset size, performance eventually plateaus below the level of RL online fine-tuning. This is

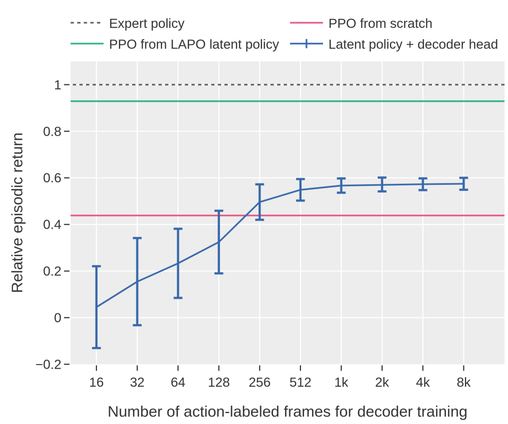

Figure 4: Offline decoding performance vs. # labeled transitions (Mean and std across 3 seeds).

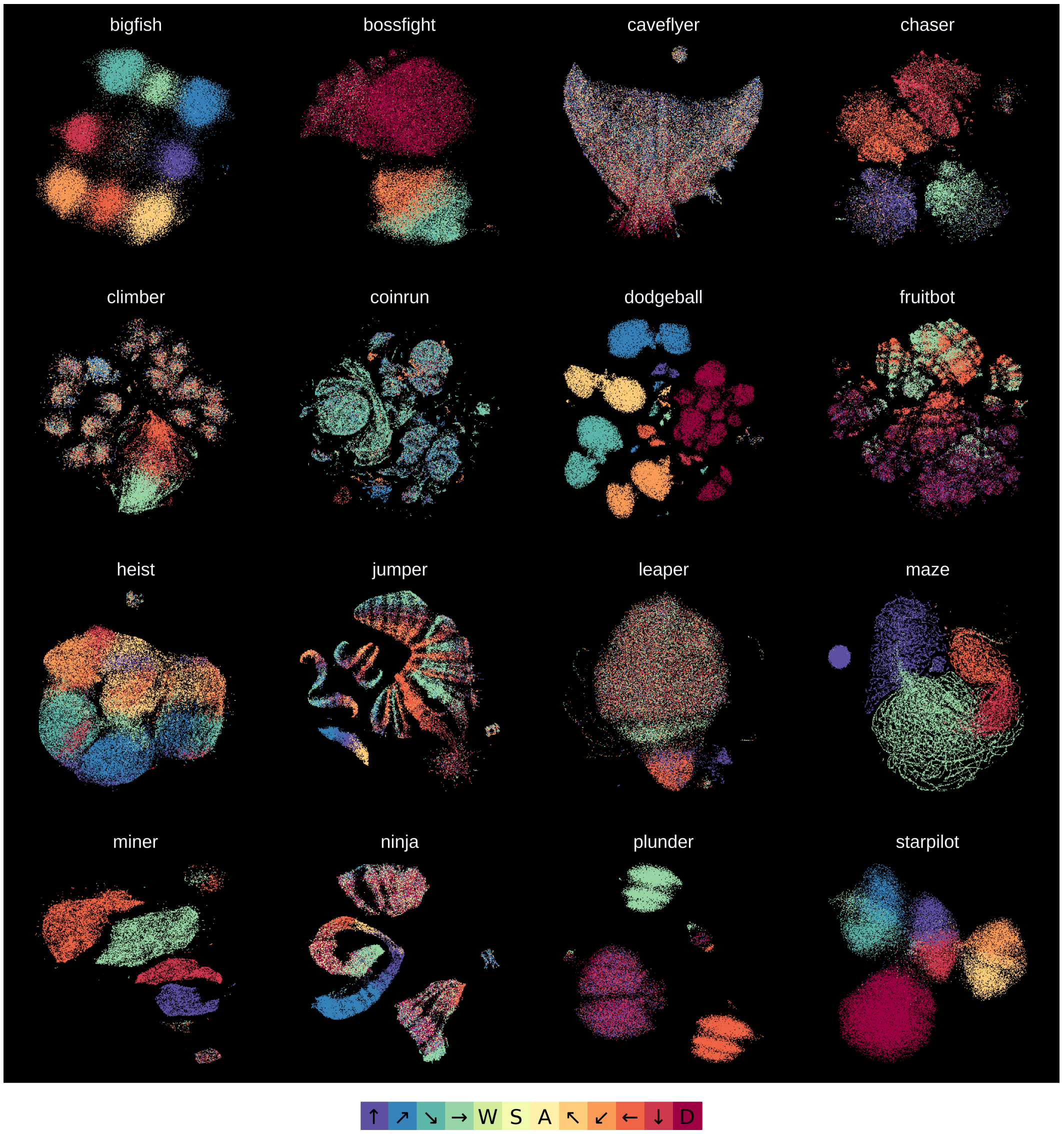

Figure 5: UMAP projection of the learned latent action space for all 16 procgen games. Each point represents the continuous (pre-quantization) latent action generated by the IDM for a transition in the observation-only dataset. Each point is color-coded by the true action taken by the agent at that transition (true action labels are only for visualization, not used for training). Arrows in the legend correspond to movement directions. `NOOP` actions are omitted for clarity.

likely because, as previously noted, latent actions are not necessarily state-invariant. A decoder that only has access to a latent action, but not the state in which that action is being taken, is not always able to successfully decode it. We provide per-environment results for offline decoding in Appendix A.1.

6.3 Inspecting the Latent Action Space

To better understand what LAPO learns, we generate a UMAP projection (McInnes et al., 2018) of the latent action space for each of the 16 Procgen environments, shown in Figure 5. For most games, the structure of the latent action space is highly interpretable, with distinct clusters of latent actions closely corresponding to the true discrete actions. We also observe that the structure of the latent space varies greatly across environments. In `BigFish`, `Chaser`, `Dodgeball`, `Maze`, `Miner`, `Plunder`, and `StarPilot`, latent actions aggregate together in well-defined clusters aligned with the true actions. In other environments, including `Climber`, `CoinRun`, `FruitBot`, `Jumper` and `Ninja`, the latent action space exhibits more fragmented structure. This split roughly corresponds to environments with higher or lower degrees of partial observability. In the latter group of environments, the pixel observation shows only part of the level, cropped locally to the player character. Thus, when the player is moving, most observations will show parts of the level that were hidden in previous observations. In order for the FDM to accurately predict the next observation, the IDM needs to encode this off-screen information into the latent action, which we hypothesize leads to more complex structures in the latent space. However, as demonstrated in Section 6.1, the latent-space policy still performs well downstream even on environments with greater degrees of partial observability. Moreso, we find that vector quantization has a large impact in simplifying the structure of the latent action space. We generate the same UMAP projection for an ablation of our method without vector quantization in Appendix A.2. We note that for several environments, including those with and without partial observability, latent action clusters corresponding to distinct true actions are at times duplicated. For example, in `BigFish`, each true action sees four latent action clusters—likely reflecting how taking the same true action in different parts of the state-space can have differing effects. These results strongly support our hypothesis that vector quantization leads to simpler latent representations with less fragmentation within actions.

6.4 Limitations

A few factors can adversely impact the performance of LAPO. First, actions that have a delayed effect in observations will be predicted to take place with the same delay, i.e. the latent policy actually models the visible effects of an action, not the action itself. Nevertheless, in most environments, actions that have any impact on the state of the environment will elicit some degree of immediate change in the observation. Moreover, delayed actions can be partially addressed by extending the IDM and FDM architecture to consider multiple timesteps into the past and future, e.g. by using a Transformer-based architecture (Vaswani et al., 2017). Second, significant stochasticity can make it difficult for the IDM to compress the useful bits of information among the noise, degrading the quality of the latent representation. This issue can potentially be mitigated by training on much larger datasets. Lastly, training on much larger datasets—as would be required for modeling more complex domains like web-scale video—would require scaling up the model architecture, which introduces new challenges in balancing the strength of the FDM and the capacity of latent actions representations, as is often the case in autoencoding architectures (Chen et al., 2016).

7 Conclusion

This work introduced LAPO, a method for training policies over a learned latent action space, inferred from purely observational data. Unlike prior work on imitation learning from observation, LAPO does not rely on access to the true action space or a predefined set of discrete latent actions to learn a useful, pretrained policy. Instead, LAPO learns a latent action space end-to-end, by optimizing an unsupervised objective based on predictive consistency between an inverse and a forward dynamics model. Vector quantization of the continuous latent actions induces an information bottleneck that forces the quantized actions to capture state-invariant transition information. Across all 16 games of the challenging Procgen Benchmark, we show that this approach can learn latent action spaces that reflect the structure of the true action spaces, despite LAPO never having access to the latter. We then demonstrate that latent action policies, obtained through behavior cloning of latent actions, can serve as useful pretrained models that can be rapidly adapted to recover or even exceed the performance of the expert that generated the original action-free dataset.

Our results thus suggest that LAPO can serve as a useful approach for obtaining rapidly-adaptable pretrained policies from web-scale action-free demonstration data, e.g. in the form of videos. We believe LAPO could serve as an important stepping stone to unlocking the web-scale unsupervised pretraining paradigm that has proven effective in other domains like language and vision (Brown et al., 2020; Radford et al., 2021; Ramesh et al., 2021a). Toward this goal, we are excited about future work aimed at scaling up the dynamics models and decoder to more powerful architectures that can consider multiple timesteps and richer observations. Such models may enable efficient learning of generalist latent action policies in complex multi-task settings, all within a single LAPO instance, while enabling more effective generalization to tasks unseen during training.

REFERENCES

Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 2935–2945, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/35309226eb45ec366ca86a4329a2b7c3-Abstract.html.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): learning to act by watching unlabeled online videos. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9c7008aff45b5d8f0973b23e1a22ada0-Abstract-Conference.html.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL http://arxiv.org/abs/1308.3432.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations*, 2016.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056. PMLR, 2020. URL http://proceedings.mlr.press/v119/cobbe20a.html.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *CoRR*, abs/2005.00341, 2020. URL https://arxiv.org/abs/2005.00341.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell Jr. Imitating latent policies from observation. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1755–1763. PMLR, 2019. URL http://proceedings.mlr.press/v97/edwards19a.html.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1406–1415. PMLR, 2018. URL http://proceedings.mlr.press/v80/espeholt18a.html.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11321–11339. PMLR, 2023. URL https://proceedings.mlr.press/v202/ghosh23a.html.

R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984. doi: 10.1109/MASSP.1984.1162229.

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.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14096–14113. PMLR, 2023. URL https://proceedings.mlr.press/v202/huh23a.html.

Marcus Hutter. On the existence and convergence of computable universal priors. In *International Conference on Algorithmic Learning Theory*, pp. 298–312. Springer, 2003.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Willi Menapace, Stéphane Lathuilière, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 10061–10070. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00993. URL https://openaccess.thecvf.com/content/CVPR2021/html/Menapace_Playable_Video_Generation_CVPR_2021_paper.html.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=vhFu1Acb0xb.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pp. 305–313. Morgan Kaufmann, 1988.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021a.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 2021b. URL `http://proceedings.mlr.press/v139/ramesh21a.html`.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Trans. Mach. Learn. Res.*, 2022, 2022. URL `https://openreview.net/forum?id=1ikK0kHjvj`.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. In Jens Kober, Fabio Ramos, and Claire J. Tomlin (eds.), *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pp. 339–354. PMLR, 2020. URL `https://proceedings.mlr.press/v155/schmeckpeper21a.html`.

Jürgen Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL `http://arxiv.org/abs/1707.06347`.

RJ Solmonoff. A formal theory of inductive inference. i. *II Information and Control*, 7:224–254, 1964.

Oliver Struckmeier and Ville Kyrki. Preventing mode collapse when imitating latent policies from observations, 2023. URL `https://openreview.net/forum?id=Mf9fQ0OgMzo`.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 4950–4957. ijcai.org, 2018. doi: 10.24963/ijcai.2018/687. URL `https://doi.org/10.24963/ijcai.2018/687`.

Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 6325–6331. ijcai.org, 2019a. doi: 10.24963/ijcai.2019/882. URL https://doi.org/10.24963/ijcai.2019/882.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *ICML Workshop on Imitation, Intent, and Interaction (I3)*, 2019b.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

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.

Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. *Advances in neural information processing systems*, 32, 2019.

Weirui Ye, Yunsheng Zhang, Pieter Abbeel, and Yang Gao. Become a proficient player with limited data through watching pure videos. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=Sy-o2N0hF4f.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30: 495–507, 2022. doi: 10.1109/TASLP.2021.3129994. URL https://doi.org/10.1109/TASLP.2021.3129994.

Qihang Zhang, Zhenghao Peng, and Bolei Zhou. Learning to drive by watching youtube videos: Action-conditioned contrastive policy pretraining. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, volume 13686 of *Lecture Notes in Computer Science*, pp. 111–128. Springer, 2022. doi: 10.1007/978-3-031-19809-0_7. URL https://doi.org/10.1007/978-3-031-19809-0_7.

Qinqing Zheng, Mikael Henaff, Brandon Amos, and Aditya Grover. Semi-supervised offline reinforcement learning with action-free trajectories. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42339–42362. PMLR, 2023. URL https://proceedings.mlr.press/v202/zheng23b.html.

A Appendix

A.1 Offline decoding results

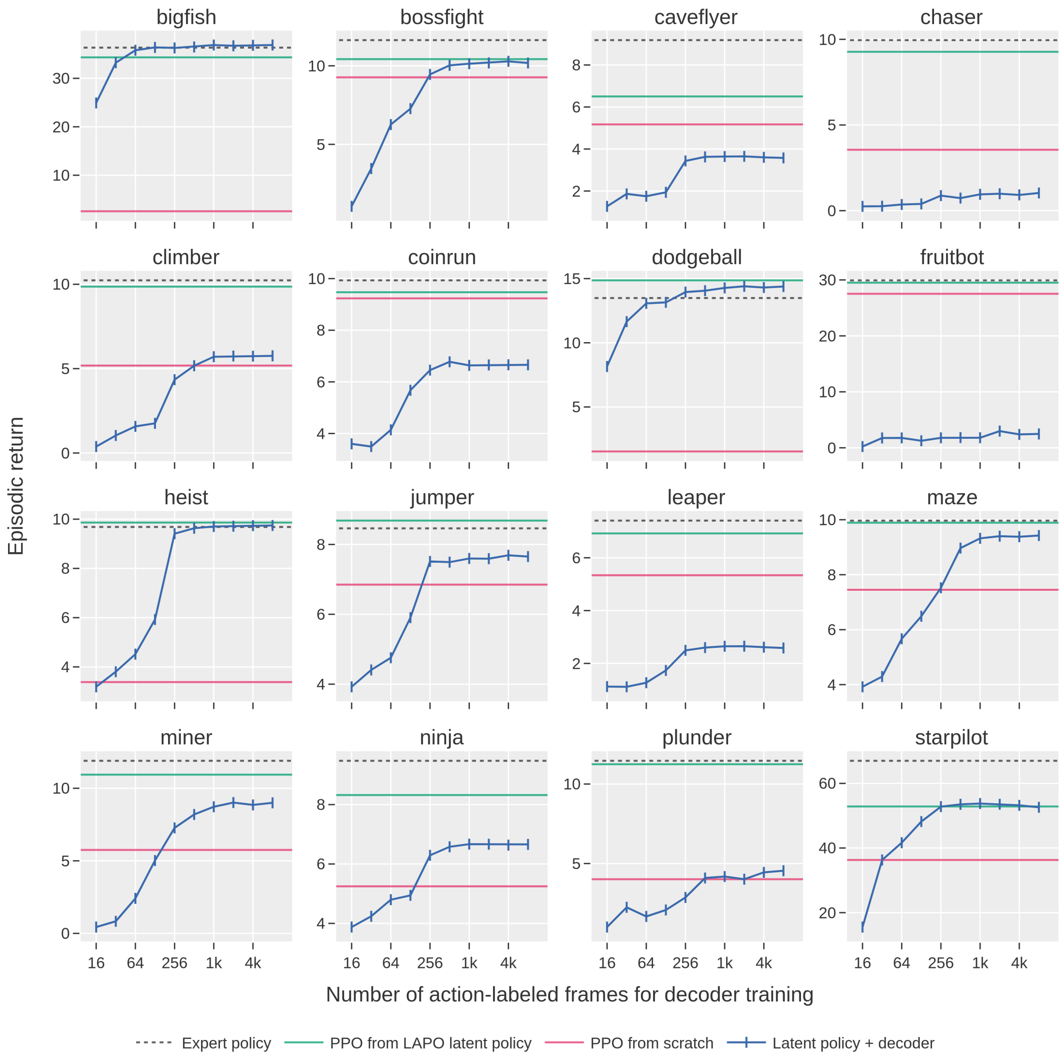

Figure 6: Test performance of the latent policy combined with a latent action decoder that is trained on an action-labelled offline dataset of a certain size, consisting of (z_t, a_t) tuples. An effective decoder can be trained with only a few hundred samples although performance generally plateaus before reaching the performance of our proposed online PPO fine-tuning approach.

A.2 Latent space analysis for no-VQ ablation

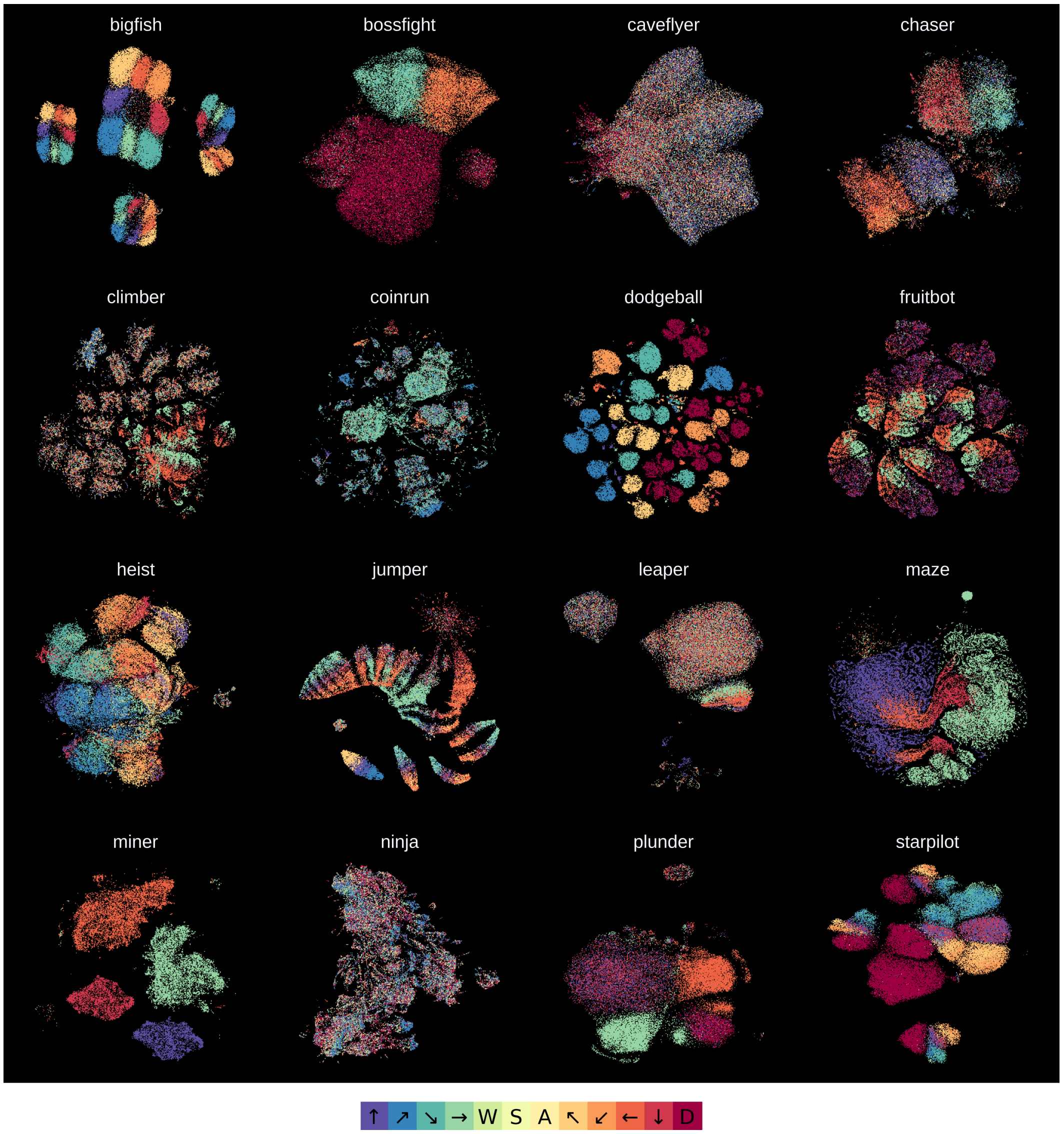

Figure 7: UMAP projection of the learned latent action space for all 16 procgen games, generated by IDMs trained without vector-quantization. Each point represents the continuous latent action generated by the IDM for a transition in the observation-only dataset. Each point is color-coded by the true action taken by the agent at that transition (true action labels are only used for visualization, not for training). Arrows in the legend correspond to movement directions. `NOOP` actions are omitted for clarity.

We note that although the No-VQ ablation performs worse in terms of online performance, its FDM loss was significantly lower. This indicates that VQ indeed acts as a bottleneck that constrains the amount of information passed from the IDM to the FDM. By forcing the IDM to compress transition information, this leads to a better latent representation for downstream policy learning.

A.3 Continuous control experiments

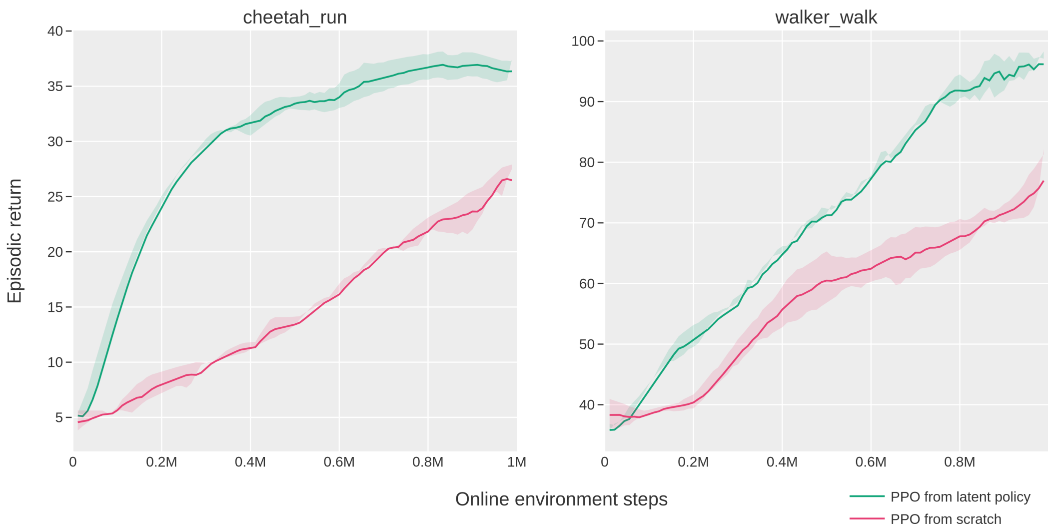

Figure 8: Mean test returns over the course of training for PPO fine-tuning of a pretrained LAPO policy compared to PPO from scratch (averaged across 2 seeds) on continuous control tasks. These experiments used the same hyperparameters for training the latent IDM and for behavior cloning as in Procgen experiments.

A.4 Hyperparameters

Table 1: Hyperparameters for the three stages of our method. Latent actions are 128-dimensional continuous vectors and are split and quantized into 8 discrete latents with 16-dimensional embeddings. We use Procgen distribution mode `easy`.

Stage	Parameter	Value
Latent IDM training	Pre-transition additional context k	1
	VQ # of codebooks	2
	VQ # of discrete latents per codebook	4
	VQ # of embeddings	64
	VQ embedding dimension	16
	VQ commitment cost	0.05
	VQ EMA decay	0.999
	Learning rate	2e-4
	Batch size	128
	Total update steps	70,000
	IMPALA-CNN channel multiplier	4
Latent behavior cloning	Learning rate	2e-4
	Batch size	128
	Total update steps	60,000
	IMPALA-CNN channel multiplier	4
RL online fine-tuning	Total environment interactions	4,000,000
	PPO hyperparameters	As in (Cobbe et al., 2020).

A.5 ILPO reproduction

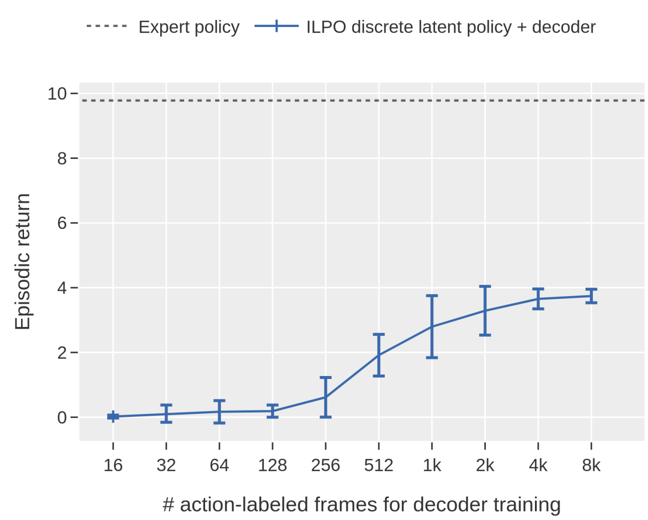

Figure 9: Reproduction of Figure 5.a from Edwards et al. (2019) for a single level from `coinrun`. When applied to data from the full distribution of levels, rather than just a single level, the ILPO FDM consistently collapsed in terms of which discrete latent achieves the minimum of $\mathcal{L}_{\text{min}}$, leading to collapse of the policy too. Results here are not exactly comparable to results from Edwards et al. (2019) since it is unknown which specific `coinrun` level was used by the authors.