Video PreTraining Search-Based Behavioral Cloning

Federico Malato*
University of Eastern Finland
fmalato@uef.fi

Florian Leopold*
University of Bielefeld
fleopold@techfak.uni-bielefeld.de

Amogh Raut
Indian Institute of Technology

Ville Hautamäki
University of Eastern Finland

Andrew Melnik
University of Bielefeld

Abstract

We formulate the control problem as a search problem over a dataset of expert trajectories. For the current latent state of the agent interacting with the Minecraft environment, we perform a proximity search over MineRL-dataset expert trajectories in the latent representation of a Video PreTraining (VPT) model. Then the agent copies actions from the expert trajectory until the state representations of the agent and the expert trajectory diverge. Then the proximity search is repeated. Our approach can effectively recover meaningful trajectories and shows very human-like behavior of a Minecraft agent. We applied our method to the MineRL BASALT Challenge 2022, where it ranked top of the leaderboard at the end of Round 1.

Methodology

In our approach, we relay on expert's demonstration and use them to reshape the control problem as a search problem over a latent space of partial trajectories (referred as *situations*. In our approach we assume that:

- Similar situations require similar solutions or actions.
- A situation can be represented in a latent space.
- The situations latent space is a metric space. Therefore, we can assess numerical similarity between two *situations*, one from ongoing interaction with the environment and the other is from the dataset.

Search-based BC aims to reproduce an expert's behavior with high fidelity, by copying a number of subsequent actions coming from the expert's past experience. We refer to these partial trajectories as ”situations”. We define a *situation* as a set $\{(o_\tau, a_\tau)\}_{\tau=t}^{t+\Delta t}$ of observation-action pairs coming from an expert's trajectory, that is, a number of subsequent steps in the dataset used for BC training. Following from this, we can assume:

- Every situation in the dataset was somehow overcome.
- Due to the expert's optimality assumption, each situation has been addressed in an *optimal* way.

Therefore, finding a match between two situations ideally ensures that an optimal solution to it has already been provided. Thus, every drifting from it (i.e. anything that is not perfectly copying those actions) results in either an equally valid or worse solution.

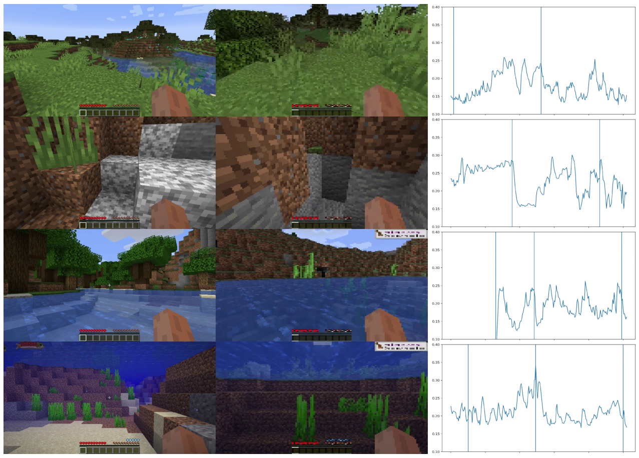

Figure 1: Four examples of divergence visualization. (**Left**) Observed frame from the environment. (**Middle**) Current reference frame from the dataset. (**Right**) Divergence history graph.

Video PreTraining (VPT) model Our approach uses pretrained VPT model (Baker et al. 2022) for encoding a situation in a latent space. The model uses a convolutional neural network (CNN) that encodes each image into a vector. Vectors of past 128 images and the current image pass to four transformer blocks. Two output heads after the transformer blocks take the discard embedding output tokens of past 128 frames and takes the embedding token of the current frame to predict actions. One head predicts a discrete action (one out of a combination of 8000? actions). The other head predicts a computer mouse control discretized into 11x11=121 regions. The architecture is shown in Figure 2.

Search-based BC In our approach, we encode the expert's trajectories through a pretrained VPT architecture. Additionally, we store the corresponding action for each frame. Thus, we obtain a latent space populated by situations that have already been addressed by an expert, along with their assumed optimal solution. Whenever a new observation is sampled from a test environment, we encode it using the

[*] Equal contribution.

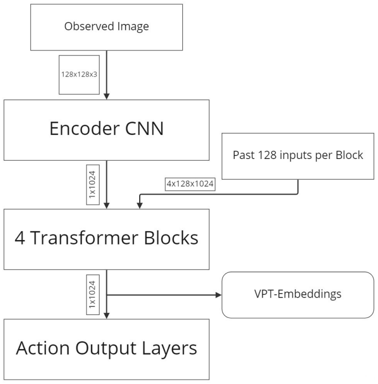

Figure 2: VPT architecture.

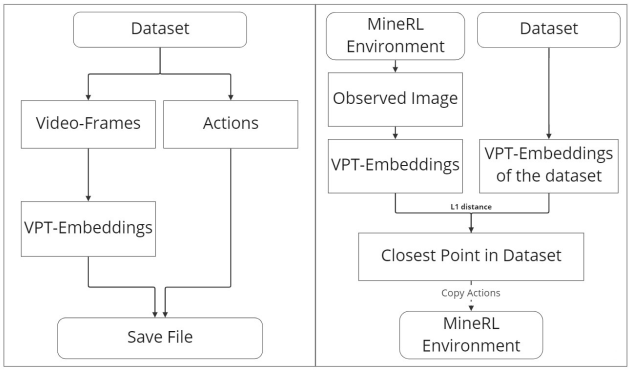

Figure 3: Schematic of our approach. (**Left**) Training procedure. (**Right**) Evaluation loop.

same network. Then, we perform a search step in the situations latent space, looking for the most similar observation. Once that a situation has been selected, we copy the actions that the expert has used to address it.

Unless a perfect match is found, the sampled situation will typically drift away from our current situation as time passes. Therefore, at each timestep we recompute the similarity between the current observation and the last selected situation. Whenever the numerical difference between the two overcomes an empirically defined threshold, a new search over the latent space is performed and a new situation is selected as most resemblant one. If the threshold is never crossed, a new search is performed after a predefined number of maximum timesteps. This search process is repeated until an episode ends.

To determine the most similar situation, we compute the L1 distance between latent representations. This comes with a major advantage: since VPT is trained on predicting the actions, the features of our latent space will be highly focused on actions. Hence the L1 distance between two points of the space is highly dependent on the observed actions being done in the past steps and only secondarily on actual image features. Therefore, if the copied actions can not actually be performed (i.e. there are physical constraints that prevent the agent from performing the actions), the L1 distance between the current latents and the reference situation will increase rapidly. This allows our agent to recover quite fast from faulty situations.

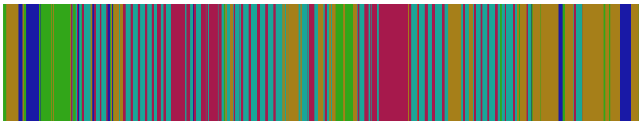

Figure 4: Cluster band visualization of an example episode in the FindCave dataset. KMeans clustering, 5 clusters, episode length 1000 steps.

Environment	First step	Second step	Last step
FindCave	0.132	0.228	0.244
VillageAnimalPen	0.126	0.209	0.217
MakeWaterfall	0.115	0.238	0.244
BuildVillageHouse	0.137	0.237	0.305

Table 1: Average L1 distance value between current state and reference situation (from left to right) right after a new search, on the following step and right before a new search.

Our approach is illustrated in Figure 3. We refer to the generation of the latent space as the ”training” procedure of our agent, even though it does not correspond to a proper training. Rather, it is a preprocessing step needed to ensure some prior knowledge to our agent.

Experiments

We have performed our experiments using the MineRL environment (Guss et al. 2019). Specifically, we have followed the tracks defined in the MineRL BASALT competition (Shah et al. 2021) 2022. In the competition, an agent must solve some high level tasks like finding a cave, building a village house, or making a waterfall from a water bucket. To do this, it is allowed to use a number of demonstrations coming from human contractors. No reward is provided to the agent. Additionally, the agent should show human-like behavior while completing the tasks.

Results

Our early stage results show promising behavior. As a fact stating that our approach produces respectable results over a BC baseline, we state that at current time our agent is ranked first in the competition’s leaderboard page. Moreover, the agent produces visually appealing and human-like solutions to the vast majority of addressed situations. Due to the copy actions feature, our agent mimics human behavior more closely than the BC baseline provided by the organizers of the competition.

Despite showing great promise, our agent is far from being perfect. In our study, we currently identify two major problems. Mitigating those limitations will be the focus of our further studies. At present time, we will only offer their descriptions and some temporary solutions to those. Please be noted that our solutions are not final, nor the descriptions constitute final evidence for the problems we identified.

Making sense of actions

Since our approach is based on a similarity search over latent representations of the current observation, the actions that are copied are not filled with semantics. To better describe this, consider the case where an agent is looking all the way up. If the next search states that the most similar situation is one that features a series of ”look up” actions, the agent will execute those actions regardless of the fact that the camera has already reached its limit on the y-axis.

To solve this, we should ensure that the current latent representation of the current state and the one of the current frame are somewhat similar (i.e. the search is not too dependent on past frames). When this happens, the agent might experience some discrepancies between the selected actions and their actual efficacy. Nonetheless, the similarity threshold helps in mitigating this: given the dependency of latent representations over actions, whenever a bad search is performed, we are reasonably sure that a new search will be performed quite soon.

Warmup phase

As previously stated, the latent representations encapsulate both present and past information. Therefore, in situations where there is no past, e.g. at the very beginning of an episode, the embedding might result far away from the true one. To mitigate this, we allow our agent to ”warm up”, by keeping it still for the first second of a new episode. This way, the agent can gather some history and produce a much higher quality representation of the current state. Using the warmup phase can be vital whenever the agent faces a dangerous situation at the beginning of an episode, e.g. when spawning close to a lava pit.

Additionally, we have performed some quantitative measurements of the L1 distance in three particularly interesting situations: the first and the second observations after a new search, and the last observation before a new search. The results are reported in Table 1. For three out of four task, we found that the average on-switch L1 distance is much lower than the others. We hypothesize that after a new search, the agent might find a very good match for the current observation. Then, when the second observation comes through, the trajectory rapidly diverges from the reference one. Finally, we noticed that the divergence increases again right before a new search is performed, even though such raise is not as promiment. In the ”Build village house” task, though, we assessed a noticeable increase of this last measure. We hypothesize that this difference is the result of having more un-performable actions cases than in any other environment.

References

[Baker et al. 2022] Baker, B.; Akkaya, I.; Zhokhov, P.; Huizinga, J.; Tang, J.; Ecoffet, A.; Houghton, B.; Sampedro, R.; and Clune, J. 2022. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv:2206.11795*.

[Guss et al. 2019] Guss, W. H.; Houghton, B.; Topin, N.; Wang, P.; Codel, C.; Veloso, M.; and Salakhutdinov, R. 2019. MinerL: A large-scale dataset of minecraft demonstrations. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2019-August.

[Shah et al. 2021] Shah, R.; Wild, C.; Wang, S. H.; Alex, N.; Houghton, B.; Guss, W. H.; Mohanty, S. P.; Kanervisto, A.; Milani, S.; Topin, N.; Abbeel, P.; Russell, S.; and Dragan, A. D. 2021. The minerl BASALT competition on learning from human feedback. *CoRR* abs/2107.01969.