Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos

Bowen Baker[*†]
bowen@openai.com

Ilge Akkaya[*†]
ilge@openai.com

Peter Zhokhov[*†]
peterz@openai.com

Joost Huizinga[*†]
joost@openai.com

Jie Tang[*†]
jietang@openai.com

Adrien Ecoffet[*†]
adrien@openai.com

Brandon Houghton[*†]
brandon@openai.com

Raul Sampedro[*†]
raulsamg@gmail.com

Jeff Clune[*†‡]
jclune@gmail.com

Abstract

Pretraining on noisy, internet-scale datasets has been heavily studied as a technique for training models with broad, general capabilities for text, images, and other modalities.[1–6] However, for many sequential decision domains such as robotics, video games, and computer use, publicly available data does not contain the labels required to train behavioral priors in the same way. We extend the internet-scale pretraining paradigm to sequential decision domains through semi-supervised imitation learning wherein agents learn to act by watching online unlabeled videos. Specifically, we show that with a small amount of labeled data we can train an inverse dynamics model accurate enough to label a huge unlabeled source of online data – here, online videos of people playing Minecraft – from which we can then train a general behavioral prior. Despite using the native human interface (mouse and keyboard at 20Hz), we show that this behavioral prior has nontrivial zero-shot capabilities and that it can be fine-tuned, with both imitation learning and reinforcement learning, to hard-exploration tasks that are impossible to learn from scratch via reinforcement learning. For many tasks our models exhibit human-level performance, and we are the first to report computer agents that can craft diamond tools, which can take proficient humans upwards of 20 minutes (24,000 environment actions) of gameplay to accomplish.

1 Introduction

Work in recent years has demonstrated the efficacy of pretraining large and general foundation models[7] on noisy internet-scale datasets for use in downstream tasks in natural language[1–4] and computer vision.[5,6,8] For sequential decision domains (e.g. robotics, game playing, and computer usage) where agents must repeatedly act within an environment, a wealth of data also exists on the web; however, most of this data is in the form of *unlabeled* video (i.e. without the actions taken at each frame), making it much less straightforward to train a behavioral prior in these domains than it is in e.g. natural language. In a few rare settings, such as Chess, Go, and StarCraft, there

[*] This was a large effort by a dedicated team. Each author made huge contributions on many fronts over long time periods. All members were full time on the project for over six months. BB, IA, PZ, and JC were on the original VPT project team and were thus involved for even longer (over a year). Aside from those original team members, author order is random. It was also randomized between IA and PZ.

[†] OpenAI

[‡] University of British Columbia

already exist large datasets with action labels from various online platforms that researchers have used for imitation learning.[9,10] When large labeled datasets do not exist, the canonical strategy for training capable agents is reinforcement learning (RL),[11] which can be sample inefficient and expensive for hard-exploration problems.[12–18] Many virtual tasks, e.g. navigating websites, using Photoshop, booking flights, etc., can be very hard to learn with RL and do not have large, commonly available sources of labeled data.[19,20] In this paper, we seek to extend the paradigm of training large, general-purpose foundation models to sequential decision domains by utilizing freely available internet-scale unlabeled video datasets with a simple semi-supervised imitation learning method. We call this method Video PreTraining (VPT) and demonstrate its efficacy in the domain of Minecraft.

Existing semi-supervised imitation learning methods aim to learn with few or no explicit action labels; however, they generally rely on the policy's ability to explore the environment throughout training, making them susceptible to exploration bottlenecks.[21–25] Furthermore, most prior semi-supervised imitation learning work was tested in the relatively low data regime; because we experiment with *far* more data (∼70k hours of unlabeled video), we hypothesize that we can achieve good performance with a much simpler method, a trend that has proven true for pretraining in other modalities such as text.[1] In particular, given a large but unlabeled dataset, we propose generating pseudo-labels by gathering a small amount of labeled data to train an inverse dynamics model (IDM) that predicts the action taken at each timestep in a video. Behavioral cloning (BC) can require a large amount of data because the model must learn to infer intent and the distribution over future behaviors from only past observations. In contrast, the inverse dynamics modeling task is simpler because it is *non-causal*, meaning it can look at both past and future frames to infer actions. In most settings, environment mechanics are far simpler than the breadth of human behavior that can take place within the environment, suggesting that non-causal IDMs could require far less data to train than causal BC models. Using pseudo-labels generated from the IDM, we then train a model to mimic the distribution of behavior in the previously unlabeled dataset with standard behavioral cloning at scale, which does not require any model rollouts and thus does not suffer from any potential exploration bottlenecks in the environment. Finally, we show we can fine-tune this model to downstream tasks with either behavioral cloning or reinforcement learning.

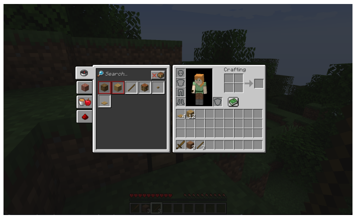

Figure 1: Example Minecraft crafting GUI. Agents use the mouse and keyboard to navigate menus and drag and drop items.

We chose to test our method in Minecraft because (a) it is one of the most actively played games in the world[26] and thus has a wealth of commonly available video data online, (b) it is a fairly open-ended sandbox game with an extremely wide variety of potential things to do, build, and collect, making our results more applicable to real-world applications such as computer usage, which also tends to be varied and open-ended, and (c) it has already garnered interest by the RL community as a research domain due to its complexity and correspondingly difficult exploration challenges.[27–31] In this work we use the native human interface for Minecraft so that we can (1) most accurately model the human behavior distribution and reduce domain shift between video data and the environment, (2) make data collection easier by allowing our human contractors to play the game without modification, and (3) eliminate the need to hand-engineer a custom interface for models to interact with the environment. This choice means that our models play at 20 frames per second and must use a mouse and keyboard interface to interact with human GUIs for crafting, smelting, trading, etc., including dragging items to specific slots or navigating the recipe book with the mouse cursor (Fig. 1). Compared to prior work in Minecraft that uses a lower frame rate and constructs crafting and attacking macros,[30,32–34] using the native human interface drastically increases the environment's exploration difficulty, making most simple tasks near impossible with RL from scratch. Even the simple task of gathering a single wooden log while already facing a tree takes 60 consecutive attack actions with the human interface, meaning the chance for a naive random policy to succeed is $\frac{1}{2}^{60}$. While this paper shows results in Minecraft only, the VPT method is general and could be applied to any domain.

In Section 4 we show that the VPT foundation model has nontrivial zero-shot performance, accomplishing tasks impossible to learn with RL alone, such as crafting planks and crafting tables (tasks requiring a human proficient in Minecraft a median of 50 seconds or ∼970 consecutive actions). Through fine-tuning with behavioral cloning to smaller datasets that target more specific behavior distributions, our agent is able to push even further into the technology tree, crafting stone tools (taking a human a median of 2.3 minutes or ∼2790 actions). Finally, fine-tuning via RL produces the most dramatic improvements: our agent is able to craft diamond tools, an unprecedented result in Minecraft made even more challenging by using the native human interface. This task requires a proficient human a median upwards of 20 minutes or ∼24000 actions. The main contributions of this work are (1) we are the first to show promising results applying semi-supervised imitation learning to extremely large, noisy, and freely available video datasets for sequential decision domains, (2) we show that such pretraining plus fine-tuning enables agents to solve tasks that were otherwise impossible to learn, (3) we show that labeled contractor data is far more efficiently used within the VPT method than it would be by directly training a foundation model from it and (4) we open source our contractor data, trained model weights, and Minecraft environment for future research into learning to act via semi-supervised imitation learning at scale.

2 Preliminaries and Related Work

Imitation learning methods[35–38] seek to construct a policy that accurately models the distribution of behavior in some dataset $D = \{(o_i, a_i)\},\ i \in \{1...N\}$ of action-observation pairs. In order to roll out these policies in an environment, they must be *causal*, meaning they condition on observations from the current timestep t and past timesteps only, i.e. $\pi \sim p(a_t|o_1...o_t)$. Imitation learning is simplest when demonstrations are labeled with corresponding actions. Imitating labeled trajectories has seen success in aerial vehicles,[39,40] self-driving cars,[41,42] board games,[9,43] and video games.[10,44]

When labeled demonstrations are not available, standard behavioral cloning will not work; however, there is a large body of work in imitating behavior from unlabeled demonstrations.[22] For instance, GAIL[23] constructs an adversarial objective incentivizing the trained policy to exhibit behaviors indistinguishable from those in the target dataset. Edwards et al.[45] propose to first learn a latent policy using unlabeled demonstrations and then map the learned latent actions to real actions with a small amount of environment interaction. Peng et al.[46] first use motion-capture methods to track agent positions in videos and then train RL agents to match these waypoints. Similarly, Behbahani et al.[47] and Aytar et al.[48] task a RL agent to match waypoints; however, they construct waypoints that are embeddings from unsupervised feature learning models. Pathak et al.[49] and Nair et al.[50] train goal conditioned policies to take actions that advance the current state towards expert-provided goal states expressed as high dimensional visual waypoints. Most similar to our own work, Torabi et al.[24] simultaneously train (1) an inverse dynamics model (IDM),[51] which aims to uncover the underlying action between timesteps given observations of past and future timesteps, e.g. $p_{\text{IDM}}(a_t|o_t, o_{t+1})$, and (2) a behavioral cloning (BC) model on trajectories of observations labeled with the IDM. Data to train the IDM is collected by rolling out the BC model in the target environment such that both models improve in tandem. However, at any point in training if there are sequences in the dataset that the IDM performs poorly on, it requires that the BC model perform those or similar sequences in order for the IDM to improve and correctly label them. Therefore, if the BC model does not explore efficiently, it could severely slow down learning. In order to avoid this potential issue we opted for a simpler two-stage approach: we first train an IDM on a small number of labeled trajectories collected from human contractors (they play the game as would normally as we record their keypresses and mouse movements). Because human contractors reach most relevant parts of the state space, we can hold the IDM fixed throughout BC training.

Compared to most previous work in semi-supervised imitation learning, we experiment in the much more complex and open-ended environment of Minecraft. Minecraft is a voxel-based 3D video game that, due its popularity and wide variety of mechanics, has attracted a vast amount of RL research.[27,28,30–34,52–60] A large body of work focuses on small, custom-made Minecraft worlds with tasks such as navigation,[53,60] block placing,[54,55] instruction following,[58,59] combat,[56] and others.[28,31,57] Work operating in the massive, randomly generated environments of Minecraft itself has included hill climbing,[52] automated curriculum learning[30] and, most closely related to the RL experiments presented in Sec. 4.4, diamond mining.[27,32–34] However, to the best of our knowledge, there is no published work that operates in the full, unmodified human action space, which includes drag-and-drop inventory management and item crafting.