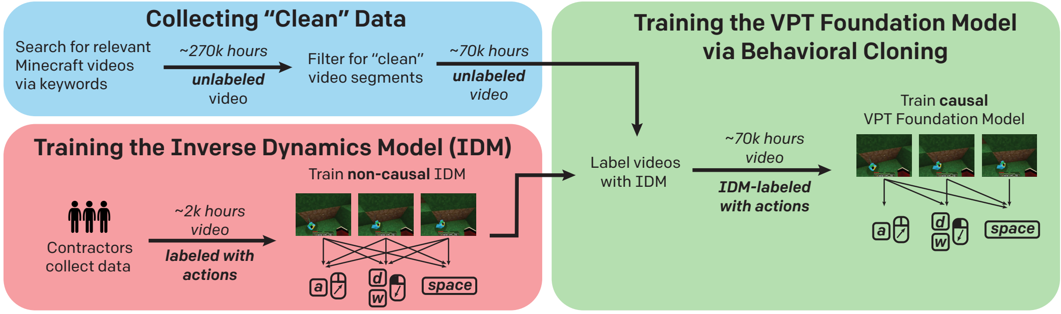
Collecting "Clean" Data
Search for relevant Minecraft videos via keywords
~270k hours unlabeled video
Filter for "clean" video segments
~70k hours unlabeled video
Training the Inverse Dynamics Model (IDM)
Contractors collect data
~2k hours video labeled with actions
Train non-causal IDM
a d w space
Training the VPT Foundation Model via Behavioral Cloning
Label videos with IDM
~70k hours video IDM-labeled with actions
Train causal VPT Foundation Model
a d w space

Figure 2: Video Pretraining (VPT) Method Overview.

3 Methods

Inverse Dynamics Models (IDM) VPT, illustrated in Figure 2, requires we first collect a small amount of labeled contractor data with which to train an inverse dynamics model $p_{\text{IDM}}(a_t|o_{1\ldots T})$, which seeks to minimize the negative log-likelihood of an action at timestep t given a trajectory of T observations $o_t \ : \ t \in [1 \ldots T]$. In contrast to an imitation learning policy, the IDM can be non-causal, meaning its prediction for a_t can be a function of both past and *future events*, i.e. $o_{t'>t}$. Compared to the behavioral cloning objective of modeling the distribution of human intent given past frames only, we hypothesize that inverting environment dynamics is easier and more data efficient to learn. Indeed, Sec. 4.1 will show that the IDM objective is much easier to learn, and furthermore Sec. 4.6 will show that with very little labeled data (as few as 100 hours) we can train a fairly accurate IDM. This IDM can be used to label online videos, providing the large amount of data required for the harder task of behavioral cloning. See appendices D and B for IDM training and data collection details.

Data Filtering We gather a large dataset of Minecraft videos by searching the web for related keywords (Appendix A). Online videos often (1) include overlaid artifacts, such as a video feed of the player's face, channel logos, watermarks, etc., (2) are collected from platforms other than a computer with different gameplay, or (3) are from different game modes, e.g. in Minecraft we only want "survival mode" where players start from scratch and must gather or craft all their items. We call data "clean" if it does not contain visual artifacts and is from survival mode, and call all other data "unclean." With enough data, a large enough model, and enough training compute, a BC model trained on both unclean and clean videos would likely still perform well in a clean Minecraft environment. However, for simplicity and training compute efficiency, we choose to filter out unclean segments of video (note that a video may contain both clean and unclean segments). We do this by training a model to filter out unclean segments using a small dataset (8800) of images sampled from online videos labeled by contractors as clean or unclean (Appendix A.2).

VPT Foundation Model We train a foundation model with standard behavioral cloning, i.e. minimizing the negative log-likelihood of actions predicted by the IDM on clean data. For a particular trajectory of length T we minimize

$$\min_{\theta} \sum_{t \in [1 \ldots T]} -\log \pi_\theta(a_t|o_1, \ldots, o_t), \text{ where } \ a_t \sim p_{\text{IDM}}(a_t|o_1, \ldots, o_t, \ldots, o_T) \tag{1}$$

As we will see in the following sections, this model exhibits nontrivial zero-shot behavior and can be fine-tuned with both imitation learning and RL to perform even more complex skills.

4 Results

4.1 Performance of the Inverse Dynamics Model

The IDM architecture is comprised primarily of a temporal convolution layer, a ResNet[62] image processing stack, and residual unmasked attention layers, from which the IDM simultaneously predicts keypresses and mouse movements (see Appendix D for IDM architecture and training details). A key hypothesis behind our work is that IDMs can be trained with a relatively small amount of labeled data. While more data improves both mouse movement and keypress predictions, our best