



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

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton
(NeurIPS, 2022)

박도연
dypark@cau.ac.kr

2025.11.03

Contents

- I. Background
- II. Introduction
- III. Methods
- IV. Results
- V. Discussion and Conclusion

Background

■ Imitation Learning

- 목표: 시연 데이터 $D = \{(o_t, a_t)\}$ 에서 관측 o_t 가 주어졌을 때 전문가의 행동 분포를 모사하는 정책 $\pi_\theta(a|o)$ 를 학습.
- 정책의 인과성(causality): 환경에서 정책을 실행하려면, 현재 시점(t) 및 과거 시점의 관찰 결과만을 기반으로 행동을 예측해야 함

$$\pi \sim p(a_t|o_1, \dots, o_t) \approx p_{expert}(a_t|o_1, \dots, o_t)$$

- a_t : 시점 t 에서의 행동 (action)
- o_1, \dots, o_t : 시점 1부터 t 까지의 관찰 결과 (observations)
- π : 학습된 정책 (policy)

■ Behavioral Cloning (BC)

- 감독학습으로 $o \rightarrow a$ 지도. 라벨이 충분할 때 강력
- 장점: 쉽고 빠르며, 보상 없이 학습 가능(탐색 비용 0)
- 단점: covariate shift로 compounding error(오류 누적) 발생
- Causal task

Introduction

- 문제 배경

- 최근 연구들은 자연어와 컴퓨터 비전 분야에서 인터넷 규모의 노이즈가 포함된 데이터셋으로 대규모 범용 foundation model을 사전학습하여 다운스트림 작업에 활용하는 것이 효과적임을 보여줌
- 하지만 **sequential decision domain**들(e.g. robotics, game playing, and computer usage)에 대한 영역에서의 인터넷 규모의 데이터셋들은, 시퀀스 단위의 정보를 포함하고 있지 않음
- 웹에도 방대한 양의 데이터가 존재하지만, 대부분은 프레임은 단위 행동 정보가 없는 **unlabeled video**이다.
 - Labeled data: imitation learning
 - E.g. Chess, Go, StarCraft ...
 - Unlabeled data: Reinforcement Learning (RL)
 - Sample inefficient
 - Exploration의 어려움

Introduction

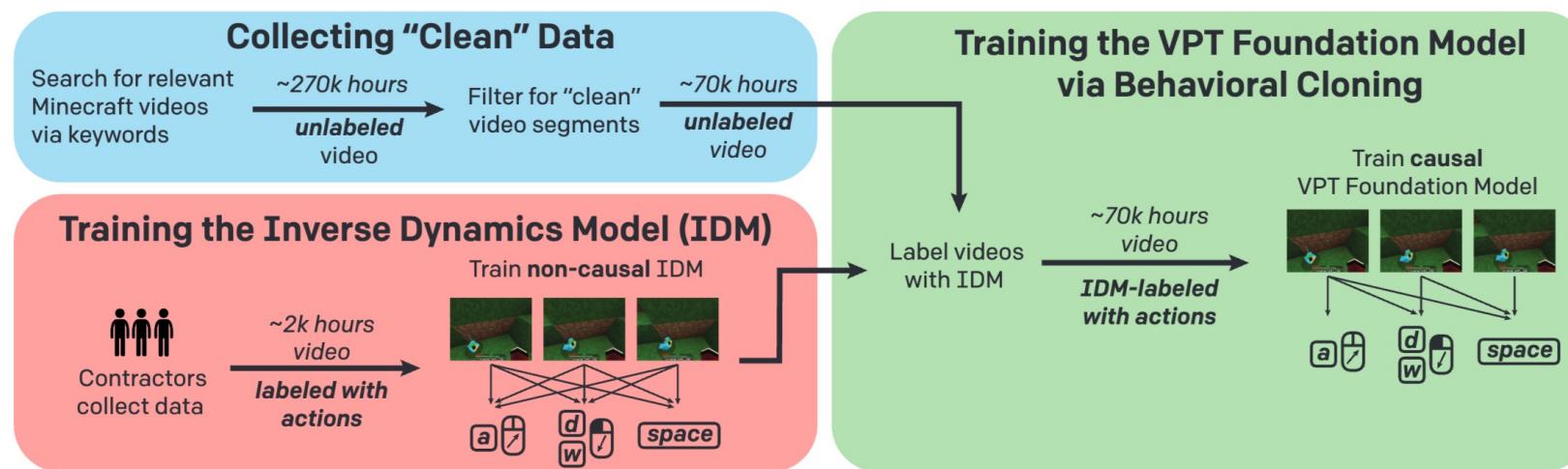
- Minecraft
 - One of the most actively played games in the world and thus has a wealth of commonly available video data online
 - Fairly open-ended sandbox game with an extremely wide variety of potential things to do, build, and collect, making our results more applicable to real-world applications such as computer usage, which also tends to be varied and open-ended
 - Already garnered interest by the RL community as a research domain due to its complexity and correspondingly difficult exploration challenges.



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

Introduction

- VPT(Video Pretraining) Method Overview
 - Goal: 인터넷의 unlabeled video를 활용해 sequential decision agent 학습



Methods

▪ Collecting “Clean” Data

1. 마인크래프트 플레이 비디오 세트 큐레이션 + 메타데이터 기반 필터링 ~ 27만 시간 unlabeled 비디오 확보
2. Contractor들이 이미지 세트 N=8800에 레이블 지정
3. 이를 바탕으로한 SVM 훈련

minecraft survival longplay
minecraft gameplay no webcam
minecraft gameplay survival mode
minecraft survival tutorial
minecraft survival guide
minecraft survival let's play
minecraft survival for beginners
minecraft beginners guide
ultimate minecraft starter guide
minecraft survival guide 1.16
minecraft how to start a new survival world
minecraft survival fresh start
minecraft survival let's play episode 1
let's play minecraft episode 1
minecraft survival 101
minecraft survival learning to play
how to play minecraft survival
how to play minecraft
minecraft survival basic
minecraft survival for noobs
minecraft survival for dummies
how to play minecraft for beginners
minecraft survival tutorial series
minecraft survival new world
minecraft survival a new beginning
minecraft survival episodio 1
minecraft survival episode 1
minecraft survival 1. bölüm
i made a new minecraft survival world

Table 1: Search terms used for generating the initial web dataset.



Figure 10: Amazon Mechanical Turk worker interface showing an example labeling task

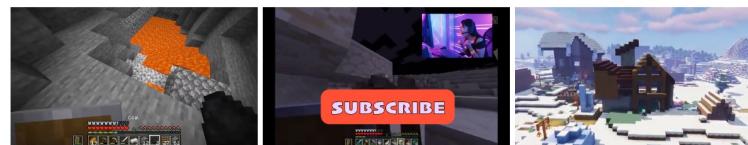
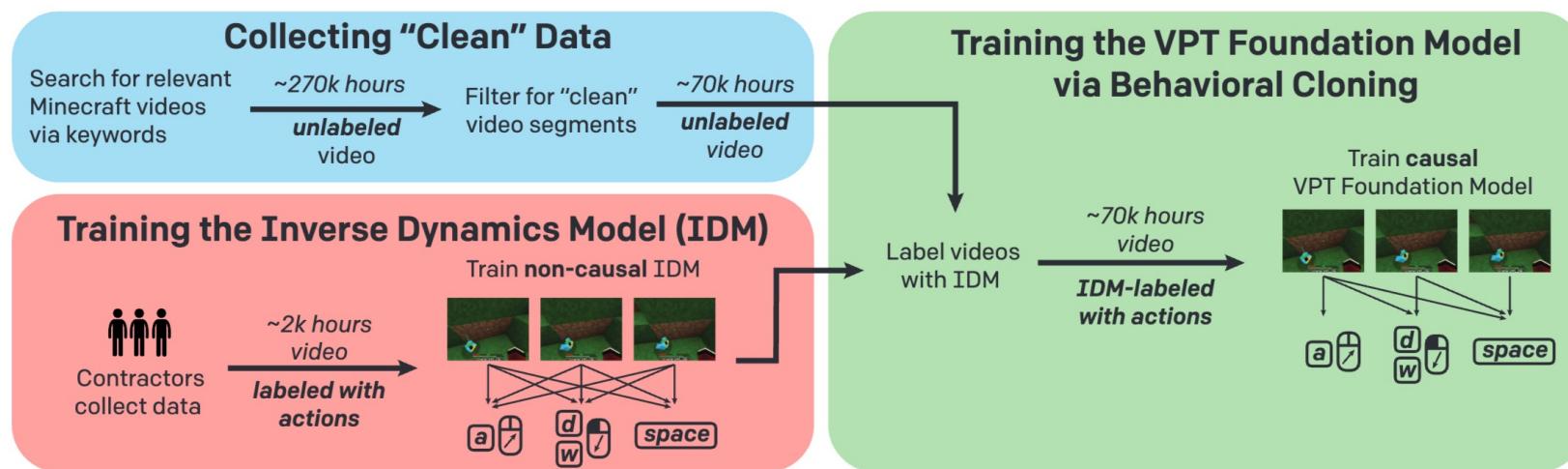


Figure 11: (Left) Sample image for Class 1: Minecraft Survival Mode - No Artifacts. (Middle) Sample image for Class 2: Minecraft Survival Mode - with Artifacts - Image contains annotations and picture-in-picture of the narrator. (Right) Sample image for Class 3: None of the Above - Image is missing the hotbar as well as health and armor bars, indicating that it was not captured during survival mode gameplay

CLIP Model Specification	RN50x64 (see text)
CLIP Input Image Resolution	448x448x3
CLIP Embedding Feature Length	1024
SVM Parameters	
Kernel	rbf
C	20
Gamma	scale
Sample Size	
Class 1	2200
Class 2	2200
Class 3	4400

Table 2: Feature Extraction Details and SVM Configuration. The parameters are for the SVM implementation in Scikit-learn⁶⁸.

Methods



Methods

- Training the Inverse Dynamics Model(IDM)
 - 소량의 라벨된 작업자 데이터를 수집하여 inverse dynamics model을 학습
 - Inverse Dynamics
 - 이전 상태 s_t 에서 주어진 다음 상태 s_{t+1} 로 가기 위한 행동 a_t 추론
 - $\hat{a} \leftarrow f_{inv}(o_t, o_{t+1})$
 - **Non-causal**(앞/ 뒤 프레임 모두 사용): action 추론 쉬움
 - 훨씬 적은 라벨로 좋은 pseudo-label 생성 가능 (Figure 9)



Figure 9: Zero-shot performance of BC models trained from scratch on the `earlygame_keyword` dataset labeled with IDMs that were trained on increasing amounts of contractor data.

Methods

- Training the Inverse Dynamics Model(IDM)
 - 각 프레임의 행동(마우스, 키) 예측
 - Action Space: keypresses, mouse movements, clicks 등 인간 플레이어가 직접 사용할 수 있는 거의 모든 행동 포함
 - 마우스 움직임의 이산화(Foveated Binning): 마우스 움직임은 연속적인 값이 아닌, 각 축(X 및 Y)을 따라 초점 영역별로(foveated binning) 이산화된 11개의 '빈(bin)'으로 구현

Action	Human action	Description
forward	W key	Move forward.
back	S key	Move backward.
left	A key	Strafe left.
right	D key	Strafe right.
jump	space key	Jump.
inventory	E key	Open or close inventory and the 2x2 crafting grid.
sneak	shift key	Move carefully in current direction of motion. In the GUI it acts as a modifier key: when used with attack it moves item from/to the inventory to/from the hotbar, and when used with craft it crafts the maximum number of items possible instead of just 1.
sprint	ctrl key	Move fast in the current direction of motion.
attack	left mouse button	Attack; In GUI, pick up the stack of items or place the stack of items in a GUI cell; when used as a double click (attack - no attack - attack sequence), collect all items of the same kind present in inventory as a single stack.
use	right mouse button	Place the item currently held or use the block the player is looking at. In GUI, pick up the stack of items or place a single item from a stack held by mouse.
drop	Q key	Drop a single item from the stack of items the player is currently holding. If the player presses ctrl-Q then it drops the entire stack. In the GUI, the same thing happens except to the item the mouse is hovering over.
hotbar. [1-9]	keys 1 – 9	Switch active item to the one in a given hotbar cell.

Table 3: Binary actions included in the action space. <https://minecraft.fandom.com/wiki/Controls> has more detailed descriptions of each action.



Figure 13: Relative camera angle or mouse movement in pixels vs. action bin. The same binning is used for both X and Y coordinates. The binning is foveated, meaning that binning is more fine-grained for smaller movements and more coarse-grained for larger movements. There are 11 bins for each axis (X and Y). The center of each bin (indicated with green circles) is used when un-discretizing movements (that is, when converting from an action expressed as a bin to a camera angle or mouse movement).

Methods

- Training the Inverse Dynamics Model(IDM)

- 각 프레임의 행동(마우스, 키) 예측
- Architecture
 - Input: 연속된 128개의 정규화된 이미지(128x128x3) 프레임
 - 3-D Convolution Layer
 - ResNet Image Processing Network
 - Flattening and Dense Layers
 - Non-Causal Residual Transformer Blocks
 - Final Action Prediction Heads

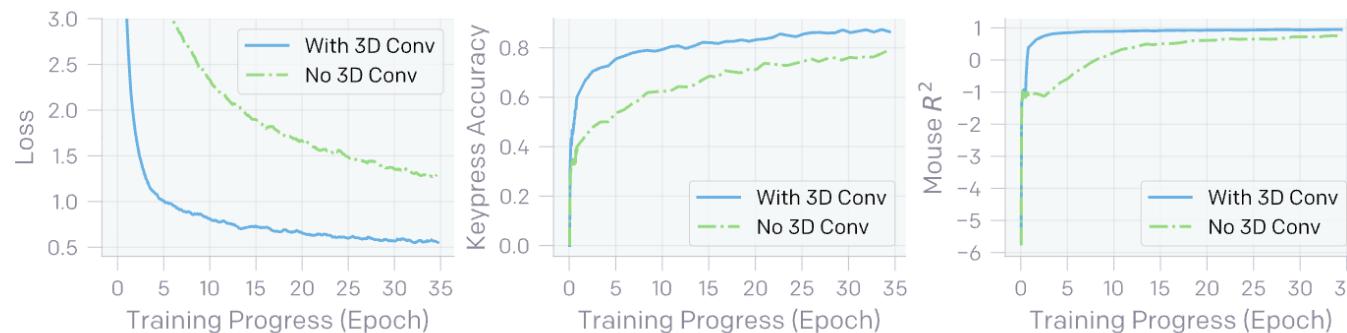


Figure 14: Effect of 3-D Convolution in the IDM Architecture.

Methods

- Training the Inverse Dynamics Model(IDM)

- Loss Function: 각 독립적인 행동 예측 손실들의 합으로 전체 손실 계산
 - 각 독립적인 손실은 예측된 행동이 실제 행동일 negative log-likelihood 를 최소화

$$L(\theta) = - \sum_{t=1}^T \log p_{IDM}(a_t | o_1, o_2, \dots, o_T; \theta)$$

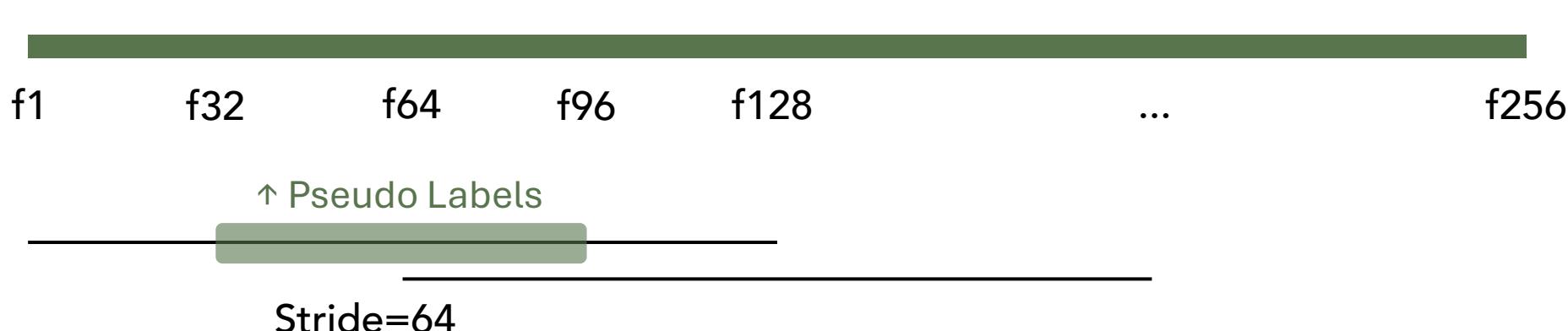
- $L(\theta)$: 모델 파라미터 θ 에 대한 전체 손실
 - $p_{IDM}(a_t | o_1, \dots, o_T; \theta)$: IDM이 주어진 관찰 o_1, \dots, o_T 를 바탕으로 시점 t 에서 행동 a_t 를 예측할 확률
 - T : 비디오 시퀀스의 길이

Methods

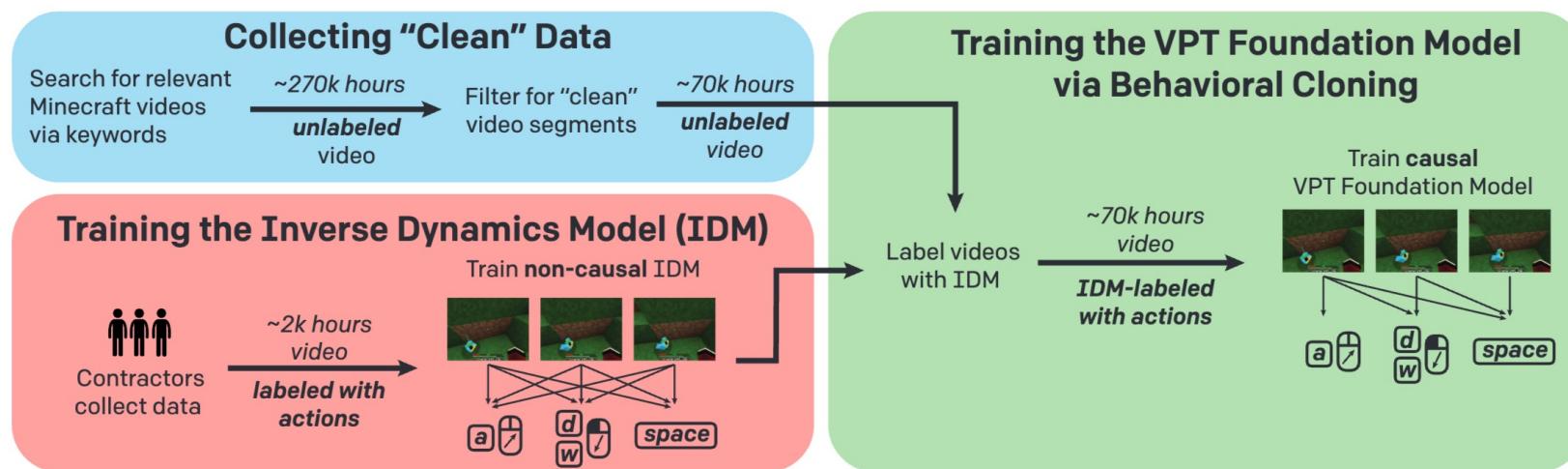
- Training the Inverse Dynamics Model(IDM)

- Generating Pseudo Labels with the IDM

1. IDM은 128개의 연속적인 이미지 프레임으로 구성된 비디오 시퀀스를 입력받아, 이 시퀀스 내의 모든 128개 프레임에 대한 행동을 동시에 예측하도록 훈련됨
2. 하지만 비디오 클립의 양쪽 끝 프레임의 경우, 충분한 미래 프레임 정보가 없으므로 IDM의 예측이 실질적으로 인과적(causal)이 될 수 있음. 이는 IDM의 비인과적 장점을 제대로 활용하지 못하게 만듬
3. 슬라이딩 윈도우(sliding window)의 적용
 1. 구체적으로, 64프레임의 스트라이드(stride)를 사용하여 비디오 전체에 IDM을 적용하되, 각 윈도우에서 IDM이 예측한 행동 중 **중앙에 위치한 64개 프레임(예: 128프레임 중 32번째부터 96번째 프레임)**에 대한 예측만을 의사 레이블로 사용
 2. 이 방식을 통해 각 의사 레이블 예측이 항상 충분한 과거 및 미래 시간 정보를 활용하여 이루어지도록 보장하며, 비디오의 처음과 마지막 프레임을 제외하고는 클립의 경계 부분 예측을 사용하지 않음



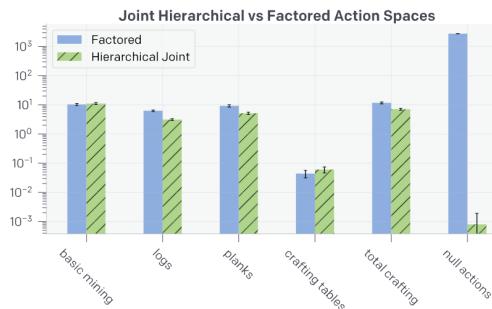
Methods



Methods

▪ Foundation Model Behavioral Cloning

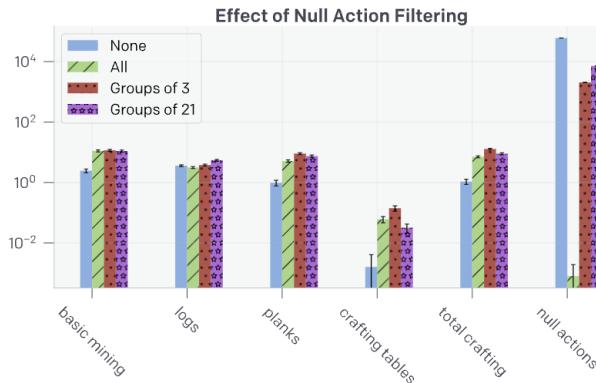
- Behavioral prior (causal): IDM이 불린 pseudo-label을 정답으로 삼아 표준 Behavioral Cloning(BC)으로 학습
- Data: web_clean(~70k h): “clean” dataset
- Label: IDM이 프레임별로 추정한 행동 $a_t \sim p_{IDM}(a_t|o_{1:T})$
- Architecture
 - Input: 연속된 128개의 정규화된 이미지 프레임 ($128 \times 128 \times 3$)
 - 3-D Convolution Layer 없음
 - ResNet Image Processing Network
 - Flattening & Frame-wise Dense Layers
 - Causal Residual Transformer Blocks
 - 마스킹된 self-attention(언어모델 방식)
 - Transformer-XL 스타일 메모리: 같은 비디오의 이전 배치 key/value에 attend, 상대적 위치 임베딩
- Final Action Prediction Heads
 - 키 on/off(2-class) + 마우스 가로/세로 11-class (IDM과 동일; 실험에 따라 factored/계층형 조인트 변형 가능)



Methods

▪ Foundation Model Behavioral Cloning

- Behavioral prior (causal): IDM이 불인 pseudo-label을 정답으로 삼아 표준 Behavioral Cloning(BC)으로 학습
- Null Action Filtering
 - 초기 BC모델은 35% 이상이 null action을 취하였음
 - Dataset에서 null action이 있는 1/ 3/ 21 프레임 제거
 - 일반적으로 Null action filtering시 성공률이 전반적으로 올라갔다.
 - 여러 방식 중, 3프레임 이상 연속 null만 필터링하는 방식이 가장 좋은 성능이었음



Results

- Performance of the Inverse Dynamics Model
 - IDM이 얼마나 적은 라벨 데이터로 잘 학습하는가?
 - 목적
 - Minecraft 비디오에서 사람의 키보드 & 마우스 행동을 자동 라벨링 하기 위해 작은 양의 라벨 데이터로 행동을 추론하는 IDM 훈련
 - 핵심 아이디어
 - BC(Behavior Cloning)로 바로 사람 행동을 배우는 건 라벨링 비용 큼
 - 대신 IDM으로 소량의 라벨 데이터만 학습
 - 그 IDM으로 수십만 시간 비디오를 자동 라벨링
 - 학습 결과
 - Keypress accuracy: 90.6%
 - Mouse movement R^2 : 97% (거의 완벽 재현)
 - 적은 데이터에서도 성능 빠르게 상승 (데이터 효율 매우 좋음)

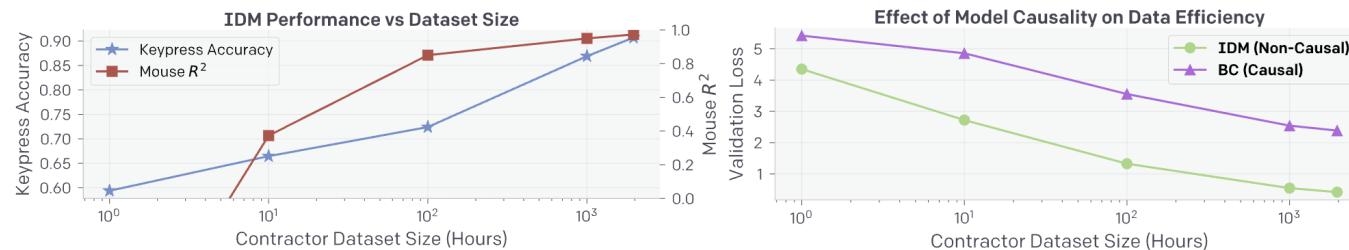


Figure 3: (Left) IDM keypress accuracy and mouse movement R^2 (explained variance⁶¹) as a function of dataset size. (Right) IDM vs. behavioral cloning data efficiency.

Results

- VPT Foundation Model Training and Zero-Shot Performance

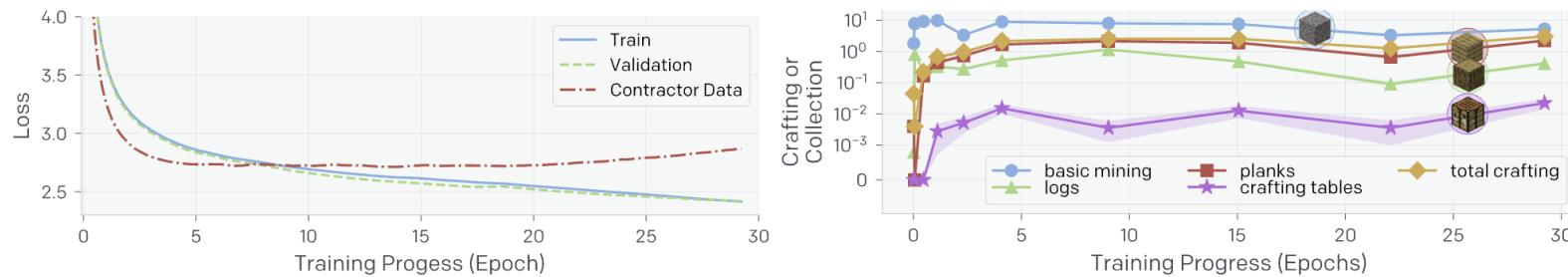


Figure 4: **(Left)** Training and validation loss on the `web_clean` internet dataset with IDM pseudo-labels, and loss on the main IDM contractor dataset, which has ground-truth labels but is out-of-distribution (see text). **(Right)** Amount a given item was collected per episode averaged over 2500 60-minute survival episodes as a function of training epoch, shaded with the standard error of the mean. Basic mining refers to collection of dirt, gravel, or sand (all materials that can be gathered without tools). Logs are obtained by repeatedly hitting trees for three seconds, a difficult feat for an RL agent to achieve as we show in Sec. 4.4. Planks can be crafted from logs, and crafting tables crafted from planks. Crafting requires using in-game crafting GUIs, and proficient humans take a median of 50 seconds (970 consecutive actions) to make a crafting table.

Results

- Fine-Tuning with Behavioral Cloning

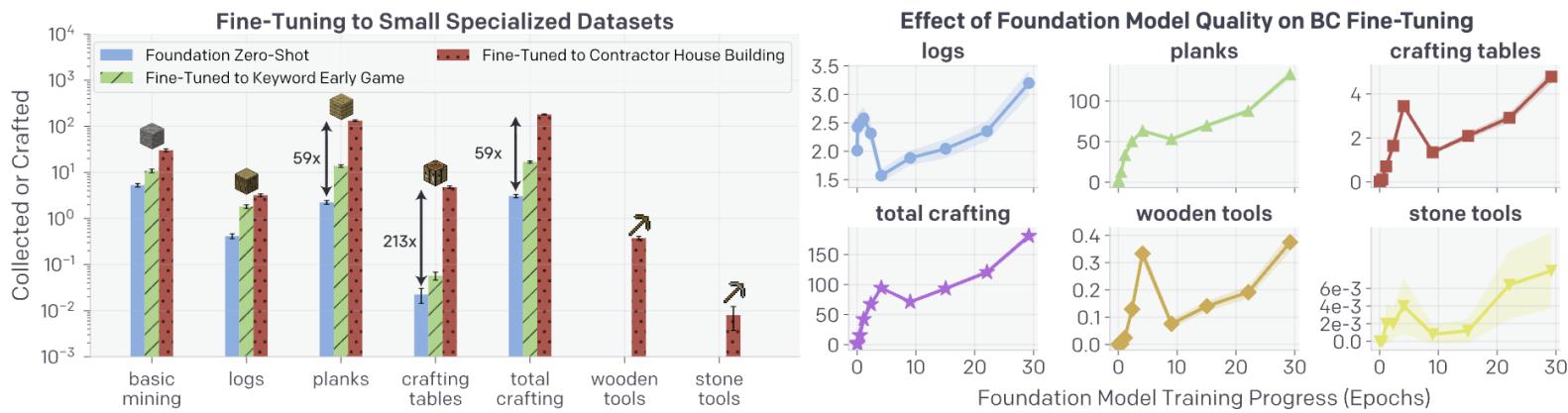


Figure 5: (Left) Collection and crafting rates for three policies: the zero-shot VPT foundation model, and the VPT foundation model BC fine-tuned to the `earlygame_keyword` or `contractor_house` datasets. BC fine-tuning to either dataset improves performance, including (for the `contractor_house` dataset) yielding wooden and stone tools. Proficient Minecraft players take a median of 1.2 minutes (1390 actions) to construct wooden tools and 2.3 minutes (2790 actions) to construct stone tools. (Right) Collection and crafting rates for VPT foundation model snapshots throughout training *after* they are BC fine-tuned to the `contractor_house` dataset. In general, crafting-related behaviors increase throughout foundation model training. Fig. 4 defines the other task terms (logs, planks, crafting tables, and total crafting).

Results

- Data Scaling Properties of the Foundation Model

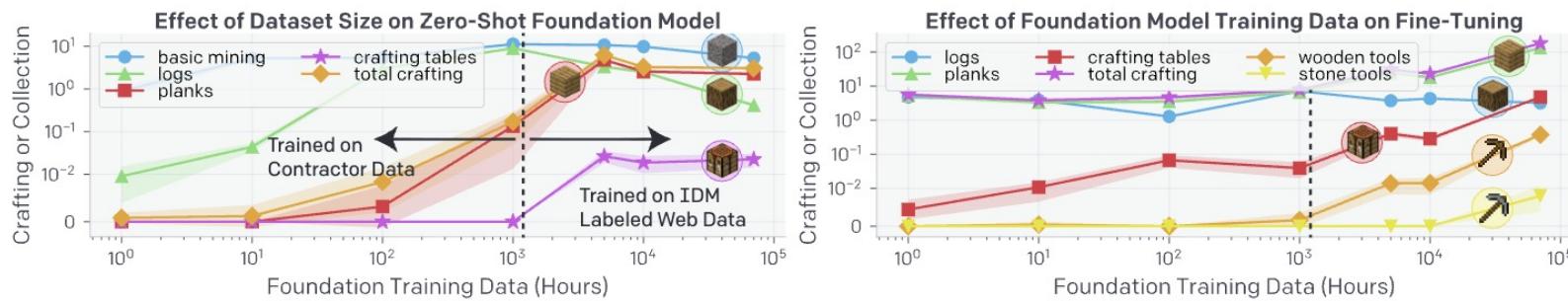


Figure 8: **(Left)** Zero-shot rollout performance of foundation models trained on varying amounts of data. Models to the left of the dashed black line (points $\leq 1k$ hours) were trained on contractor data (ground-truth labels), and models to the right were trained on IDM pseudo-labeled subsets of `web_clean`. Due to compute limitations, this analysis was performed with smaller (71 million parameter) models except for the final point, which is the 0.5 billion parameter VPT foundation model. **(Right)** The corresponding performance of each model *after* BC fine-tuning each model to the `contractor_house` dataset.

Discussion and Conclusion

- 웹의 unlabeled 영상 데이터를 행동 학습에 활용하는 새로운 방법 제시
- VPT는 기존 비디오 표현 학습 방식(생성/대조)과 달리 사전학습 단계부터 직접 행동을 학습할 수 있음
- 학습된 행동 priors는 RL 탐색을 획기적으로 강화 → 복잡한 장기 행동을 효율적으로 학습 가능

My Experiment

- VLA
- 3D vision