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

Kapranova Elizaveta 211

Compare Reinforcement Learning and Imitation Learning

Reinforcement Learning

- Learns through interaction and feedback from the environment
- Aims to maximise cumulative rewards
- Uses rewards and penalties
- Does not need examples of optimal behaviour

Imitation Learning

- Learns by mimicking expert behaviour
- Aims to replicate expert behaviour
- Uses supervised learning from expert demonstrations
- Requires high-quality expert demonstrations

Why we choose Minecraft?

- One of the most popular games in the world, so there is a lot of online data
- It is open-ended sandbox game with extremely wide variety of potential things to do and it makes model more applicable to the real-world
- It was already used in RL problems, so there is some data

Structure of VPT

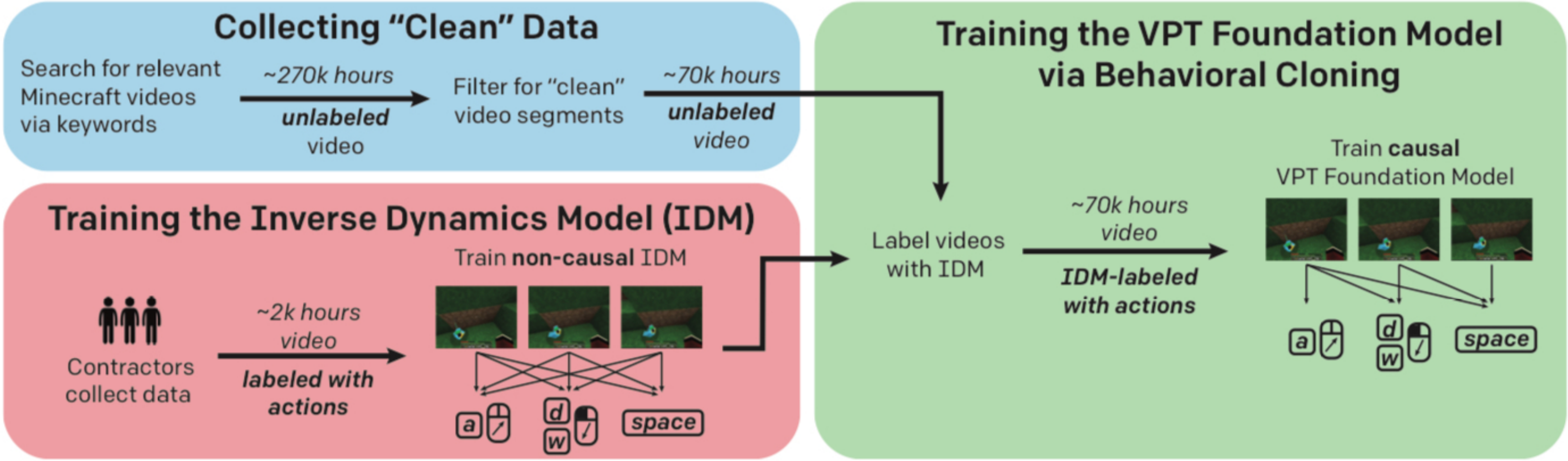


Figure 2: Video Pretraining (VPT) Method Overview.

Data filtering

- We have three labels
  - Minecraft Survival Mode - No Artifacts
  - Minecraft Survival Mode - with Artifacts
  - None of the Above

Data filtering

- RN50x64 ResNet CLIP Model (for obtaining embeddings for each frame)
- SVM using RBF (for classifying)
- Filter videos that consist of at least 80% “clean” frames
- Median filter (for extracting “clean” segments of duration at least 5s) - **web_clean** dataset
- **early_game** dataset - subset of web_clean that consist only of videos with the start of the game

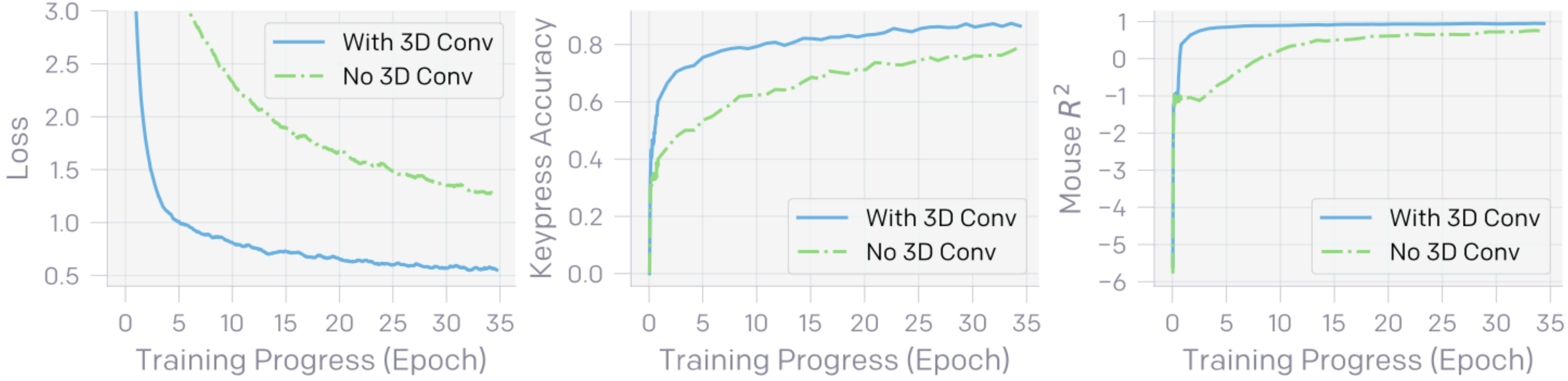
Contractor Data

- Collect as many units of wood as possible, using only wooden or stone tools (**treechop** task)
- Start new world every 30 minutes
- Build basic house in 10 minutes using only dirt, wood, sand and either wooden or stone tools (**contractor_house** dataset)
- Starting from new world and empty inventory craft a diamond pickaxe in 20 minutes (**obtain_diamond_pickaxe** dataset)

Inverse Dynamic Model (IDM)

A non-causal model

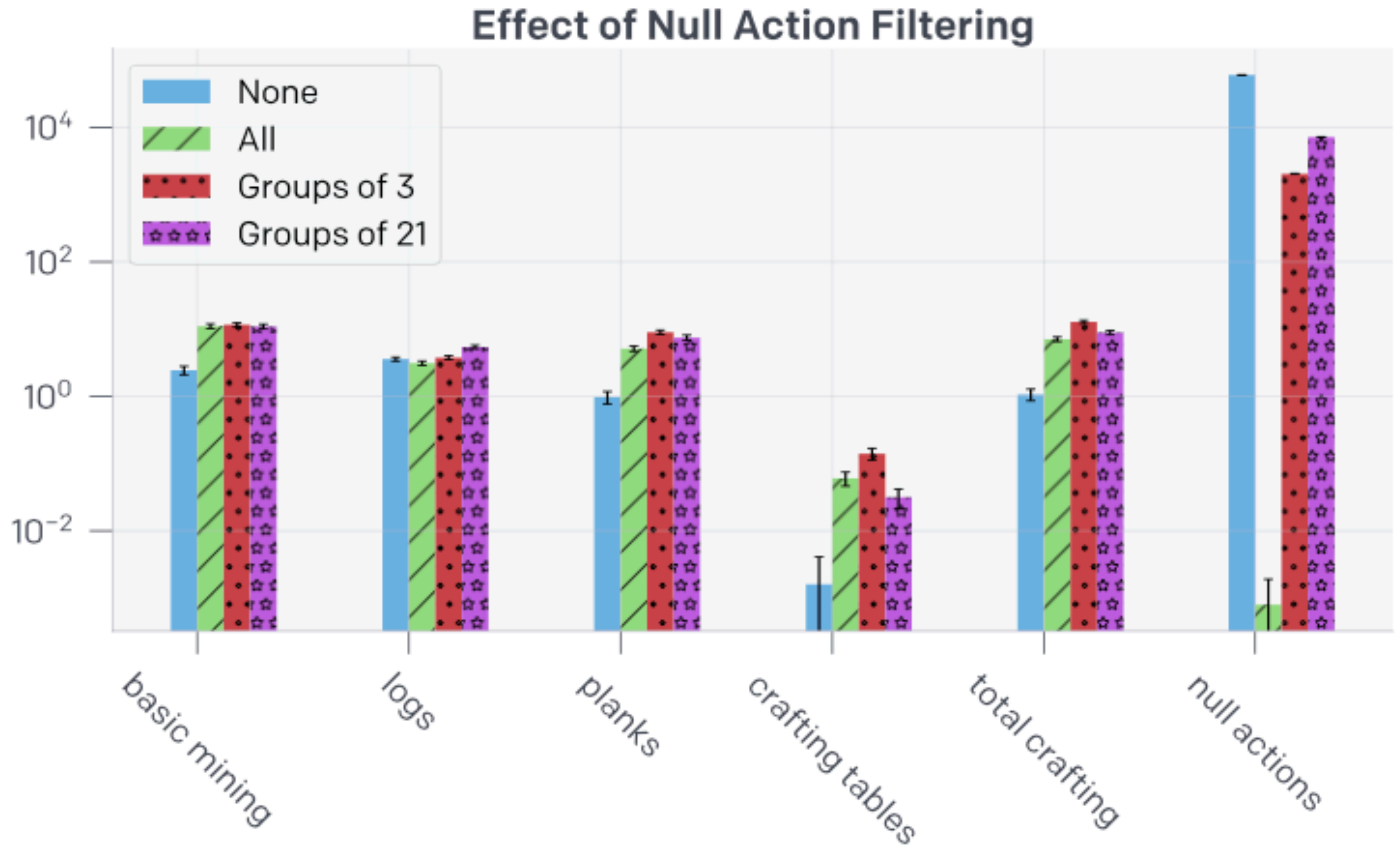
- Input: 128 consecutive image frames with dimensions 128x128x3
- There are 0.5B trainable weights
- First layer - 3-D convolution (a very important layer)
- ResNet
- 4 non-causal (unmasked) transformer block

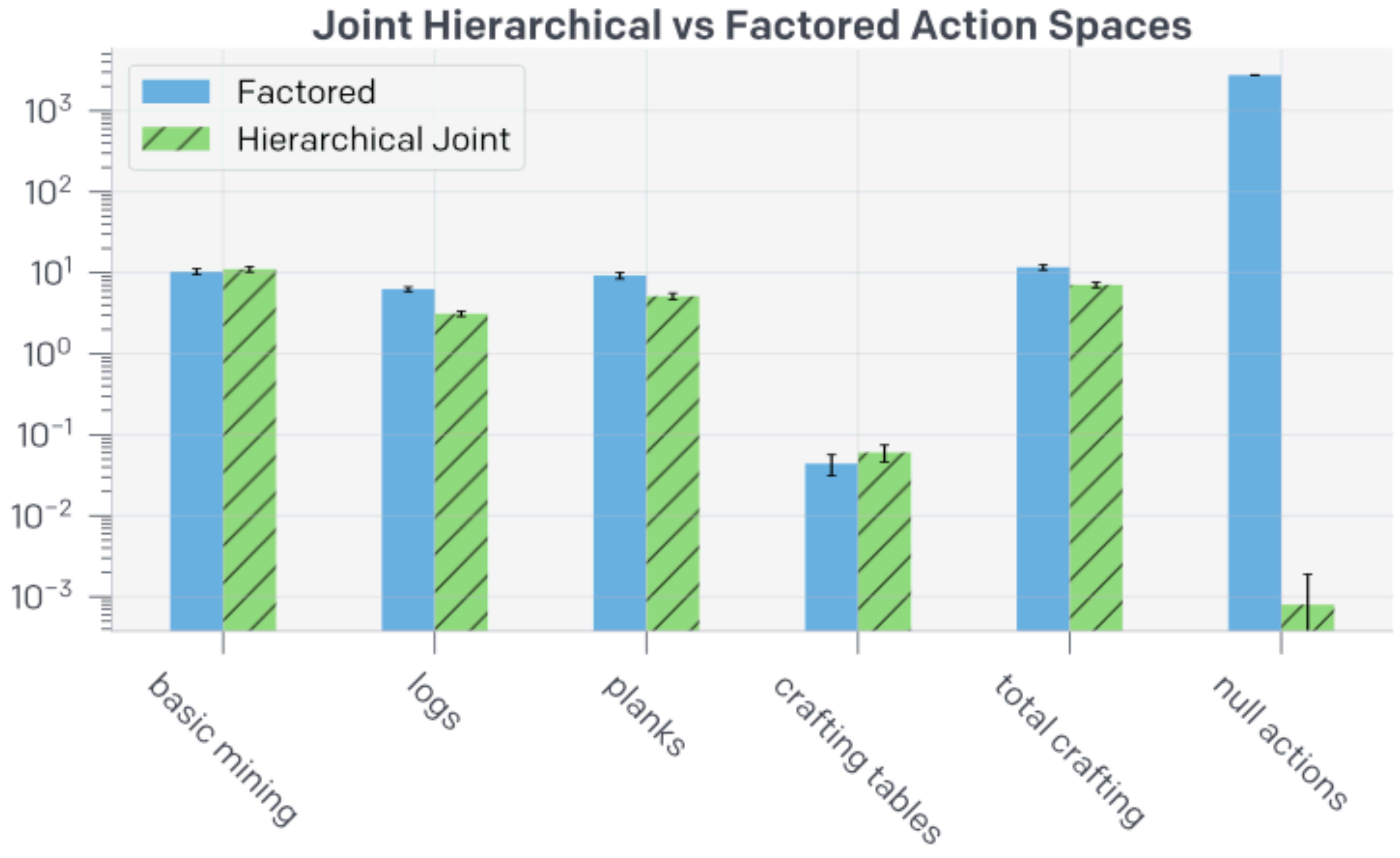


Foundation Model Behavioural Cloning

A causal model

- Remove from IDM the first layer and make it causal
- Add null action filtering - the best approach is to remove only groups of 3 or more null actions
- Use Joint Hierarchical Action Space - not all combinations of keypresses and mouse movements are valid





Foundation Model Behavioural Cloning

A causal model

This is standard behavioural cloning (minimising the negative log-likelihood)

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

Performance of IDM

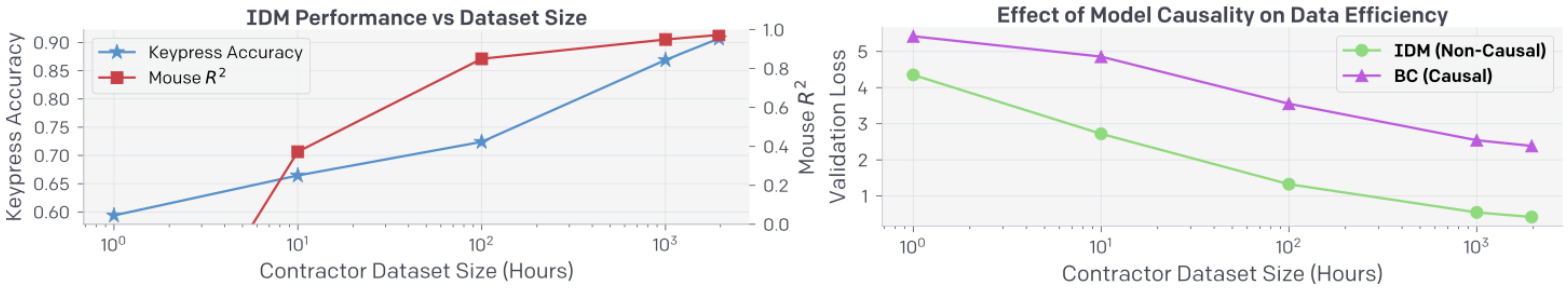
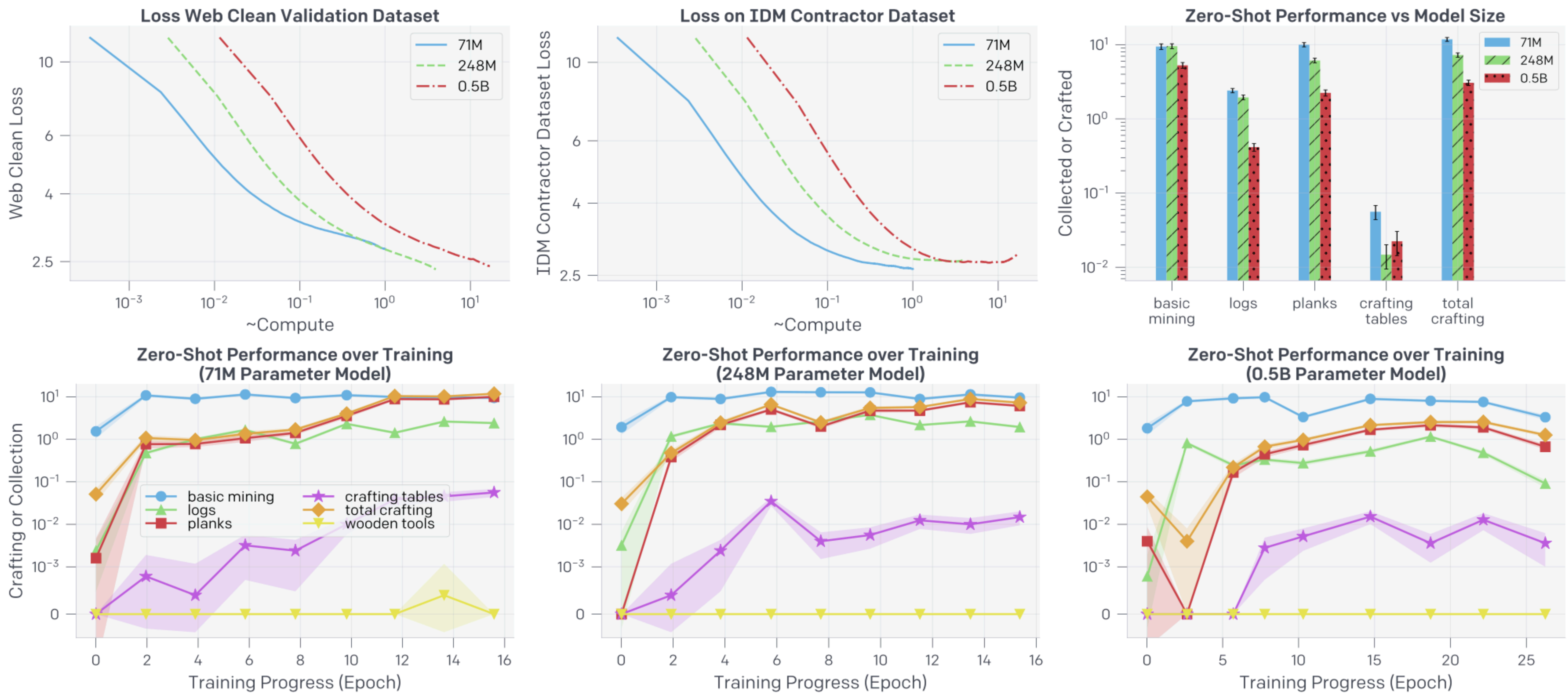


Figure 3: **(Left)** IDM keypress accuracy and mouse movement R^2 (explained variance[61]) as a function of dataset size. **(Right)** IDM vs. behavioral cloning data efficiency.

Compare with smaller IDM models



VPT Foundation Model Training and Zero-shot Performance

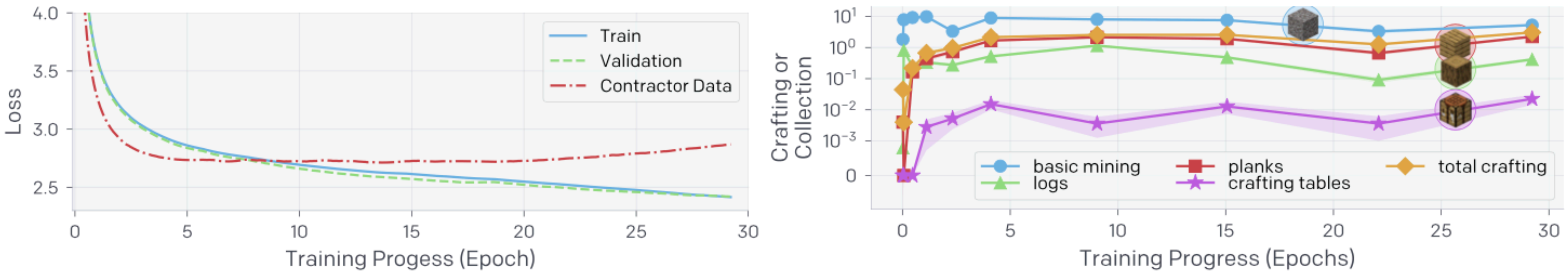
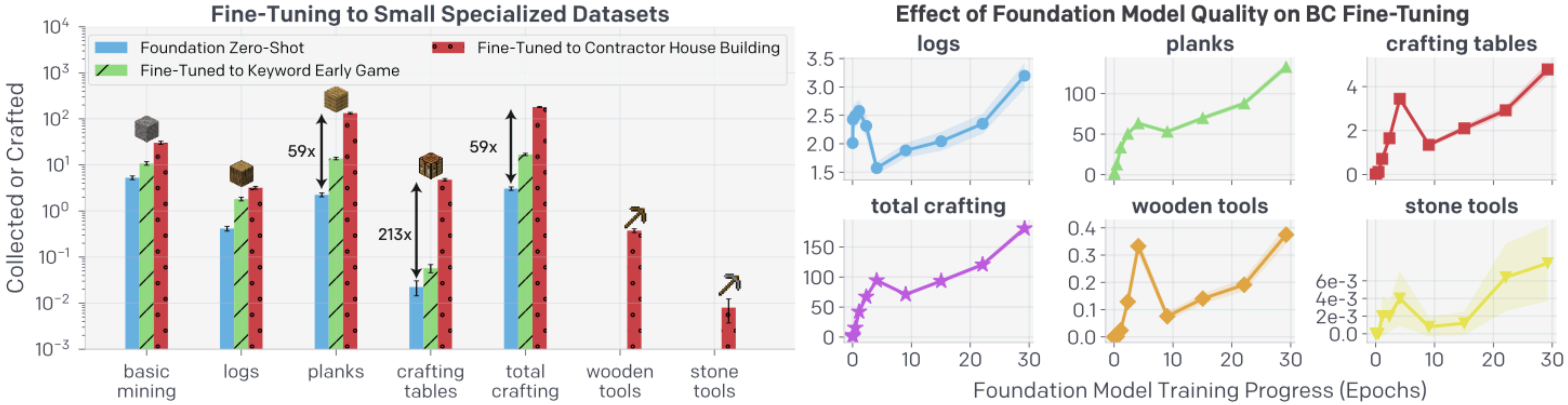


Figure 4: **(Left)** Training and validation loss on the `web_clean` internet dataset with IDM pseudo-labels, and loss on the main IDM contractor dataset, which has ground-truth labels but is out-of-distribution (see text). **(Right)** Amount a given item was collected per episode averaged over 2500 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.

Fine-Tuning with Behavioural Cloning

- We train another BC model on **contractor_house** and **early_game** datasets for improving ability to collect and craft "early game" items

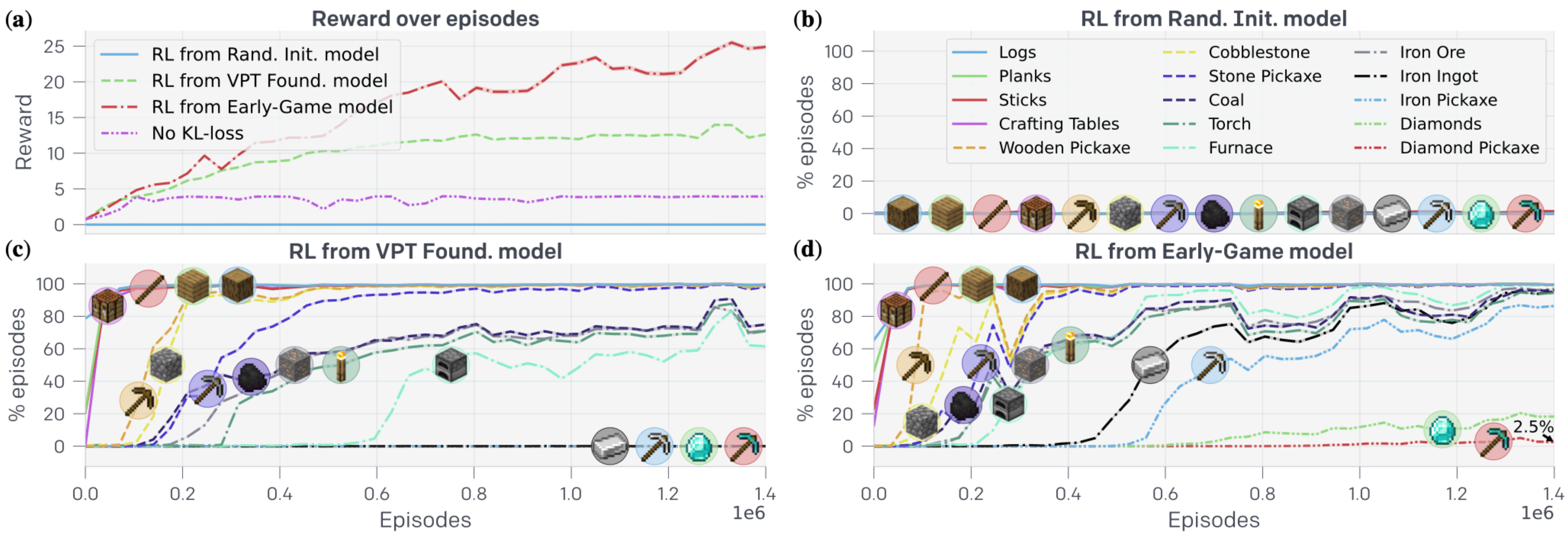


Fine-Tuning with Reinforcement Learning

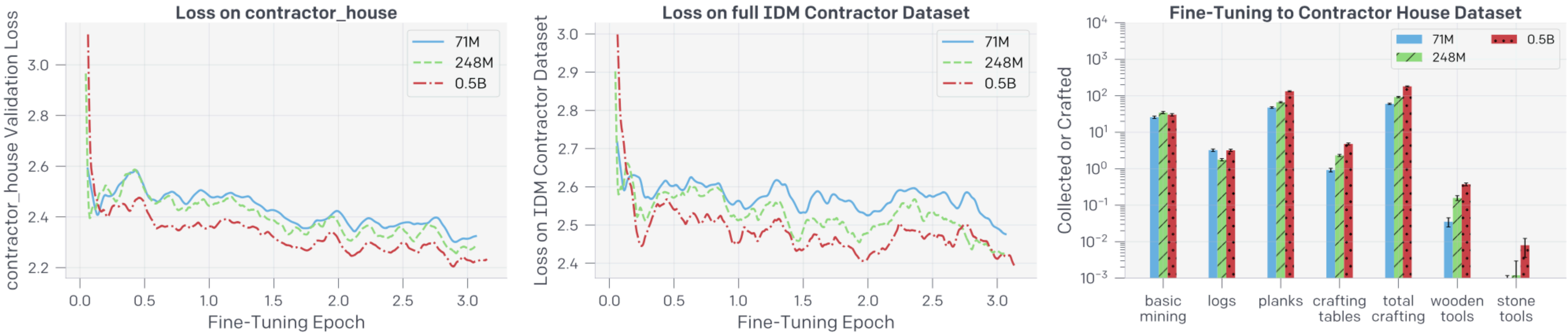
- Now we want to learn how to obtain diamond pickaxe in 10 minutes
- We use phasic policy gradient (PPG) and proximal policy optimisation (PPO)
- To prevent catastrophically forgetting we apply an auxiliary KL divergence loss

$$L_{klpt} = \rho \mathrm{KL}(\pi_{pt}, \pi_{\theta})$$

Fine-Tuning with Reinforcement Learning



Compare smaller IDM models with fine-tuning



We are first to report non-zero success rates on crafting a diamond pickaxe