

LP. Прохождение Minecraft с помощью Video PreTraining #1

Айбек Мырзатай227

Майнкрафт

Minecraft - это бесконечная во все стороны открытая sandbox игра с огромными возможностями, деревом технологии, и с рандомно генерируемым миром. Идеальный полигон для поведенческого machine learning'a.



Что не так с RL?

Hard Exploration задача - это задача где очень сложно достичь вознаграждения, потому что возможности слишком широкие и коварные. Minecraft относится к hard exploration задаче.

Reinforcement Learning фундаментально плохо работает для hard-exploration задач, потому что машина не способна перебрать всевозможные комбинации playout-ов и наткнутся на осмысленный фидбэк.

RL в одиночку едва ли справляется с тем, чтобы скрафтить палки или верстак. Наша модель будет в состоянии сделать алмазную кирку!!

Мы воспользуемся другой парадигмой обучения, называемой Imitation Learning, а также тем преимуществом, что школьники по всему миру собрали нам огромный датасет примеров того как проходится игра.

Постановка задачи

В классической постановке задачи поведенческого machine learning'a у нас есть среда, состояния, действия, а также агент и его политика. В нашем случае

Среда - это рандомная процедурно генерируемая сессия игры в майнкрафт

Состояние - это положение пикселей на экране

Действие - это нажатие клавиш на клавиатуре и движение мышки

Политика - целевая функция из предыдущих состояний в следующее действие, которую мы обучаем. В нашем случае, это VPT model.

Каждую секунду мы испытываем 20 фреймов — или же 20 различных состояний и 20 возможностей для различных действий. Наш робот будет взаимодействовать с игрой через нативный человеческий интерфейс — клавиатура и мышка.



Сбор датасета

В качестве данных мы используем 70 тысяч часов видео с интернета. Это огромный массив данных, однако же его недостаток - это то, что он не отмеченный.

Inverse Dynamics Model (IDM) - это нейронная сеть $p_{IDM}(a_t | o_1, \dots, o_T)$, которая обучается по видео предсказывать какие действия были приняты в каждом состоянии. Ключевое отличие от политики в том, что IDM может предсказывать действия на основе в том числе будущих состояний.

Архитектура IDM - это сверточная сеть, слой ResNet, и слой residual transformer'a.

Чтобы обучить IDM авторы статьи попросили коллег покатать майнкрафт на 1962 часов. Они записывали не только видео, но и инпут клавиатуры с мышкой (здесь и дальше **заказные данные**). IDM обучается на заказных данных, а затем помогает отметить неотмеченные 70 тысяч часов данных.

Collecting “Clean” Data

Search for relevant
Minecraft videos
via keywords

$\sim 270k$ hours
unlabeled
video

Filter for “clean”
video segments

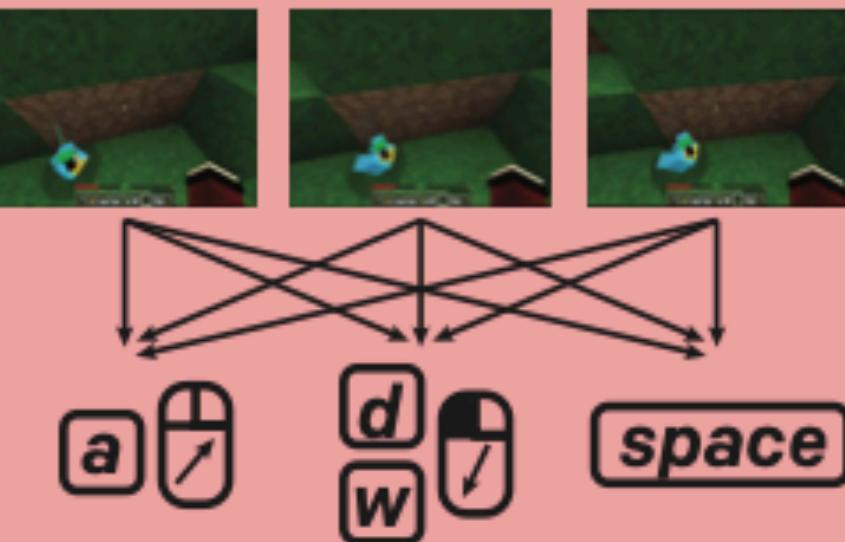
$\sim 70k$ hours
unlabeled
video

Training the Inverse Dynamics Model (IDM)

Contractors
collect data

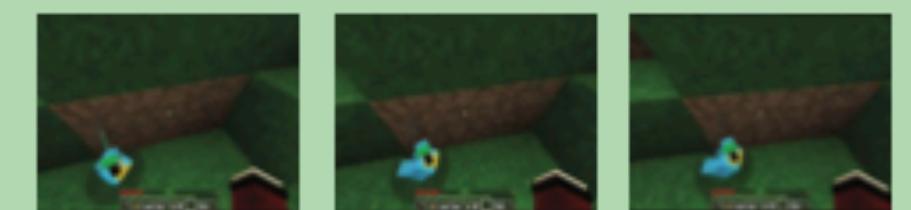
$\sim 2k$ hours
video
*labeled with
actions*

Train non-causal IDM



Training the VPT Foundation Model via Behavioral Cloning

Train **causal**
VPT Foundation Model



$\sim 70k$ hours
video
*IDM-labeled
with actions*

Label videos
with IDM

Behavioral Cloning

Мы тренируем базовую модель π_θ с помощью стандартного behavioral cloning, т.е. максимизируем правдободие действий выбранных IDM

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

На этом этапе модель крафтит палки, собирает цветочки и создает из них красители, охотится за дикими животными, убивает зомби, собирает ягоды и грибы и кушает их, находит деревни и собирает оттуда редкие айтемы. Модель также умеет перемещаться по неровным местностям, плавает, может прыгать и ставить под себя блок.

Fine-tuning with Behavioral Cloning

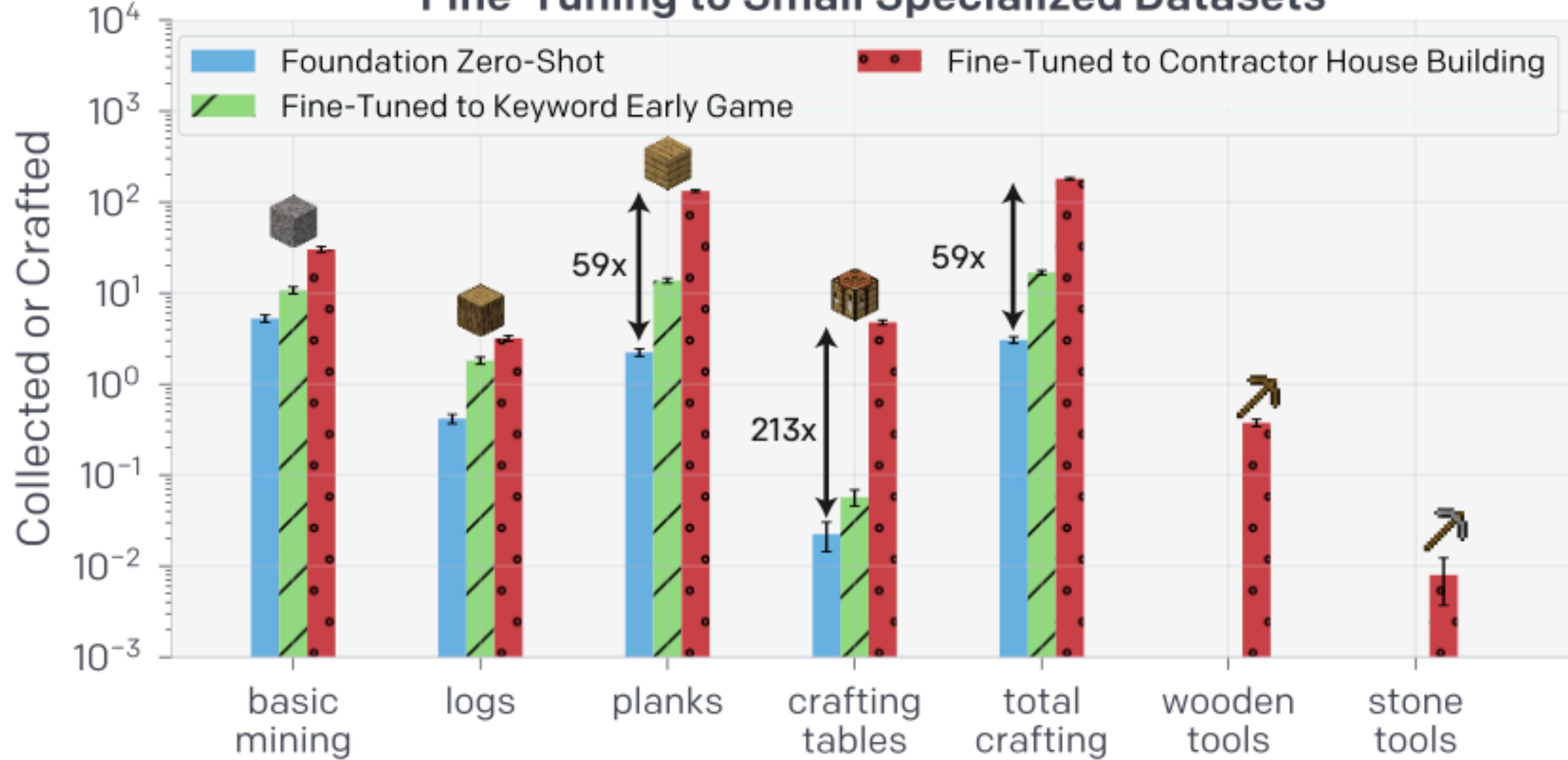
При том, что базовая VPT модель демонстрировала неплохой zero-shot перформанс, она не могла пойти дальше того, чтобы скрафтить верстак в дереве технологий.

Чтобы получить более внушительные результаты, зафайнтюним модель на следующие датасеты:

- contractor_house - коллеги авторов имеют 10 минут на то, чтобы построить дом из дерева, песка, и грязи
- early_game_keyword - датасет видосов по запросу “new world”, “let’s play episode 1”, и т.п.

Файнтюнинг на contractor_house дал дополнительные результаты: модель крафтит деревянные инструменты, добывает булыжник, крафтит каменные инструменты.

Fine-Tuning to Small Specialized Datasets



Fine-tuning with Reinforcement Learning



Мы файнтьюним модель с phasic policy gradient на ~1.3 миллионах сессий по 10 минут. Агент вознаграждается за приобретения важных айтемов из дерева технологий. Чем более продвинутый айтем, тем больше вознаграждение.

Одна из проблем RL - это *катастрофическое забывание*. RL может забывать и стирать свои скиллы, потому что они успевают реализоваться в playout-ах. Однако же это решается добавлением к функции потерь KL дивергенции между изначальной моделью и замороженной предобученной политикой.

Финальная трехфазовая модель майнит железную кирку и алмазы чаще чем люди в среднем за 10 минут, а также в 2.5% случаях крафтит алмазную кирку — беспрецедентный результат.

