An Interactive Agent Foundation Model

Zane Durante [*1 2 §] , **Bidipta Sarkar** [*1 2 §] , **Ran Gong** [*2 3 §] , **Rohan Taori** [1 2 §] , **Yusuke Noda** [2] , **Paul Tang** [1] , **Ehsan Adeli** [1] , **Shrinidhi Kowshika Lakshmikanth** [1] , **Kevin Schulman** [1] , **Arnold Milstein** [1] , **Demetri Terzopoulos** [3] , **Ade Famoti** [2] , **Noboru Kuno** [2] , **Ashley Llorens** [2] , **Hoi Vo** [2 †] , **Katsu Ikeuchi** [2 †] , **Li Fei-Fei** [1 †] , **Jianfeng Gao** [2 †] , **Naoki Wake** [*2 ►] , **Qiuyuan Huang** [*2 ►]

Figure 1. Overview of an Agent AI system that can perceive and act in different domains and applications. Agent AI is emerging as a promising avenue toward Artificial General Intelligence (AGI). Our model represents an initial step in the development of a model that is highly capable of human-level reasoning across many tasks and levels of granularity.

Abstract

The development of artificial intelligence systems is transitioning from creating static, task-specific models to dynamic, agent-based systems capable of performing well in a wide range of applications. We propose an **Interactive Agent Foundation Model** that uses a novel multi-task agent training paradigm for training AI agents across a wide range of domains, datasets, and tasks. Our training paradigm unifies diverse pre-training strategies, including visual masked auto-encoders, language modeling, and next-action prediction, enabling a versatile and adaptable AI framework. We demonstrate the performance of our framework across three separate domains—Robotics, Gaming AI, and Healthcare. Our model demonstrates its ability to generate meaningful and contextually relevant outputs in each area. The strength of our approach lies in its generality, leveraging a variety of data sources such as robotics sequences, gameplay data, large-scale video datasets, and textual information for effective multimodal and multi-task learning. Our approach provides a promising avenue for developing generalist, action-taking, multimodal systems.

1. Introduction

The development of AI systems that can not only gather useful sensory information, but also interact with their environments in meaningful ways has been a long-time goal for AI researchers. One key advantage of developing generalist AI systems is that of training a single neural model across many tasks and data modalities, an approach which is highly scalable via data, compute, and model parameters (Reed et al., 2022). With recent significant advances surrounding general-purpose foundation models (Bommasani et al., 2021), the AI community has a new set of tools for developing generalist, action-taking AI systems en route to artificial general intelligence. Despite their impressive results across various AI benchmarks, large foundation models frequently hallucinate the presence of objects and actions in scenes and infer factually incorrect information (Rawte et al., 2023; Peng et al., 2023). We posit that one of the key reasons why these foundation models hallucinate is due to their lack of grounding in the environments in which they are trained (e.g., large-scale internet data instead of physical or virtual environments). Furthermore, the dominant

[*]Equal Contribution. [►]Project Lead. [†] Equal Advisor.
[§] Work done while interning or researching part-time at Microsoft Research, Redmond. [1]Stanford University; [2]Microsoft Research, Redmond; [3]University of California, Los Angeles.

approach for building multimodal systems is to leverage frozen pre-trained foundation models for each modality and to train smaller layers that allow for cross-modal information passing (Alayrac et al., 2022; Li et al., 2022; 2023d; Dai et al., 2023; Liu et al., 2023). Since the visual- and language-specific submodules are not tuned during multimodal training, any hallucination errors in the submodules will likely be present in the resulting multimodal system. Additionally, lack of cross-modal pre-training could make grounding information across modalities challenging.

Towards such a generalist model that is grounded and pre-trained within physical or virtual environments, we propose a unified pre-training framework for handling text, visual data, and actions as input. We treat each input type as separate tokens and pre-train our model to predict masked tokens across all three modalities. Our approach uses pre-trained language models and pre-trained visual-language models to effectively initialize our model with pre-trained submodules, which we jointly train in our unified framework. We call our approach and resulting model an **Interactive Agent Foundation Model**, due to its ability to *interact* with humans and its environment, as well as its visual-language understanding ability as shown in Figure 1.

In this paper, we show that a 277M parameter model[1] that is jointly pre-trained across 13.4 M video frames from several distinct domains and data sources can effectively engage in interactive multi-modal settings using text, video, images, dialogue, captioning, visual question answering, and embodied actions within four disparate virtual environments. In order to effectively evaluate the broad range of capabilities and generalization abilities of our model, we show results across distinct domains: (1) Robotics, (2) Gaming AI, and (3) Healthcare. Despite using domain-specific visual inputs, text descriptions, and action-spaces, our model is effectively able to generalize across all three domains. To facilitate research in this discipline, we plan to release our code and models publicly.

2. Related Work

2.1. Foundation Models

A large number of works have sought to develop general-purpose foundation models based on large-scale pre-training on broad-scale internet data from a variety of sources (Bommasani et al., 2021). Within the field of Natural Language Processing, this generally consists of larger proprietary LLMs (Wang et al., 2022) such as the GPT-series (Brown et al., 2020; Min et al., 2022), or smaller open-source models such as the LLaMA series (Touvron et al., 2023), or instruction-tuned variants such as Alpaca (Taori et al., 2023) and Vicuna (Zheng et al., 2023). Within the field of computer vision, strategies such as masked auto-encoders (He et al., 2022) and contrastive learning (Radford et al., 2021) are two popular methods for self-supervised learning.

[1]We are currently developing an even larger model.

2.2. Multimodal Understanding

Recently, many multimodal models have been developed that seek to learn a relatively small number of parameters to connect large pre-trained visual encoders and language model decoders (that are generally frozen) with representative models including Flamingo (Alayrac et al., 2022), the BLIP-series (Li et al., 2022; 2023d; Dai et al., 2023), and LLaVA (Liu et al., 2023). These models are generally trained using the standard language modeling cross-entropy loss on large-scale internet data consisting of visual-text pairs, using a source of data similar to that used to train contrastive dual encoder models (Radford et al., 2021; Bain et al., 2021; Sun et al., 2023b). Unlike most previous work, we explore training models to predict visual tokens and action tokens in addition to language tokens and explicitly train our model for agentic tasks.

2.3. Agent-Based AI

Agent-based AI is distinguished from traditional AI by its need to generate dynamic behaviors that are grounded in an understanding of environmental contexts. Recent research has focused on employing advanced large foundation models to create Agent-based AI systems, as shown in (Durante et al., 2024). In the field of robotics, for instance, recent studies have highlighted the potential of LLM/VLMs in enhancing multimodal interactions between robots, environments, and humans. This applies to both manipulation (Jiang et al., 2022; Brohan et al., 2023; 2022; Li et al., 2023e; Ahn et al., 2022; Shah et al., 2023b; Li et al., 2023c; Wake et al., 2023a; Gong et al., 2023a) and navigation (Gadre et al., 2023; Dorbala et al., 2023; Cai et al., 2023; Shah et al., 2023a; Zhou et al., 2023; Dorbala et al., 2022; Liang et al., 2023; Huang et al., 2023). Additionally, significant advances in reinforcement learning have improved agent policy training on top of VLM/LLMs. Key advancements have been made in areas such as reward design (Yu et al., 2023; Katara et al., 2023; Ma et al., 2023), efficient data collection (Kumar et al., 2023; Du et al., 2023), and the management of long-horizon steps (Xu et al., 2023; Sun et al., 2023a; Li et al., 2023a; Parakh et al., 2023; Wake et al., 2023b). Similarly to robotics, gaming agents require an understanding of visual scenes and textual instructions/feedback (Puig et al., 2023; Li et al., 2021; Srivastava et al., 2022; Gong et al., 2023b). Agent-AI in the context of healthcare has focused on the text-based interaction between humans by utilizing the capabilities of LLM/VLMs. Representative applications include diagnostic assistance (Lee et al., 2023; Li et al., 2023b), knowledge retrieval (Peng et al., 2023; Guu et al., 2020), and remote monitoring (Amjad et al., 2023).

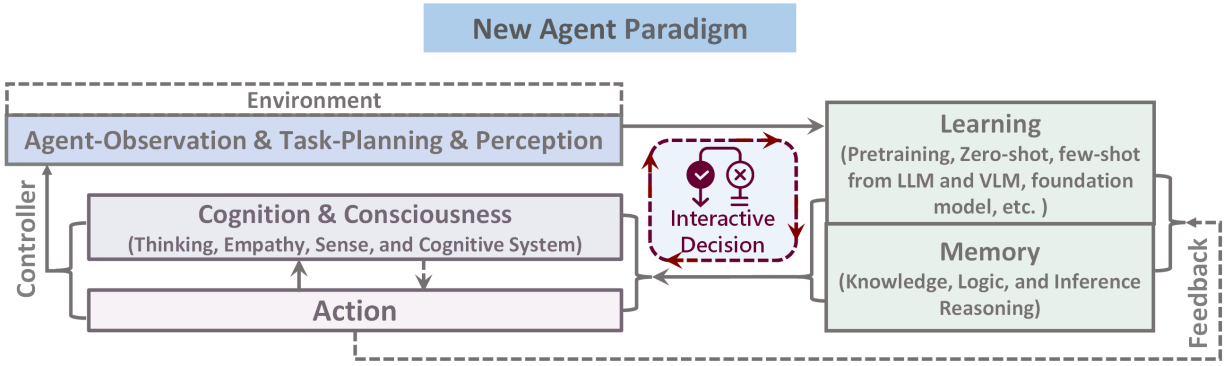


Figure 2. We propose an Agent AI paradigm for supporting interactive multi-modal generalist agent systems. There are 5 main modules as shown: (1) Agent in Environment and Perception with task-planning and observation, (2) Agent learning, (3) Memory, (4) Action, and (5) Cognition and Consciousness (we use "consciousness" to imply a degree of awareness of an agent's state and surroundings). A key difference between our approach and some previous interactive strategies is that, after training, the agent's action will directly impact task planning, as the agent does not need to receive feedback from the environment to plan its next actions.

3. Agent Paradigm

Recent advancements in AI technology have been remarkable, enabling a reasonable understanding of linguistic and visual information acquired in open-world environments. At this pivotal historical juncture, public interest in embodied agent technology is shifting from research confined to simulations and controlled environments to practical applications in highly uncertain environments. For example, consider a scenario where a robot, upon being unboxed, can instantly start communicating with non-expert humans and swiftly adapt to performing household tasks in the home environment. In this section, we define a new paradigm for embodied agents to position our proposed Interactive Agent Foundation Model within the context of this new paradigm.

We define the embodied agent paradigm as *"any intelligent agent capable of autonomously taking suitable and seamless action based on sensory input, whether in the physical world or in a virtual or mixed-reality environment representing the physical world"* (Figure 2). Importantly, an embodied agent is conceptualized as a member of a **collaborative system**, where it communicates with humans with its vision-language capabilities and employs a vast set of actions based on the humans' needs. In this manner, embodied agents are expected to mitigate cumbersome tasks in virtual reality and the physical world.

We believe such a system of embodied agents requires at least three key components:

1. **Perception** that is multi-sensory with fine granularity. Like humans, multi-sensory perception is crucial for agents to understand their environment, such as gaming environments, to accomplish various tasks. In particular, visual perception is useful for agents that can parse the visual world (e.g., images, videos, gameplay).

2. **Planning** for navigation and manipulation. Planning is important for long-range tasks, such as navigating in a robotics environment and conducting sophisticated tasks. Meanwhile, planning should be grounded on good perception and interaction abilities to ensure plans can be realized in an environment.

3. **Interaction** with humans and environments. Many tasks require multiple rounds of interactions between AI and humans or the environment. Enabling fluent interactions between them would improve the effectiveness and efficiency of completing tasks for AI.

In light of these principles, our proposed **Interactive Agent Foundation Model** represents preliminary research that focuses on these critical aspects, aiming to develop an embodied agent that functions as a practical assistance system. For an overview of our goals for developing an embodied agent, see Figure 2.

Achieving an embodied agent is not easy, especially considering the complex dynamics of systems with multi-modal observations in the physical world. Despite the advancement of recent LLM/VLMs, many challenges must be addressed, including but not limited to: 1) unstructured environments, where current visual inputs affect both high-level and low-level actions of the embodied agent given the same goal instruction; 2) open sets of objects, which require the agent's decision-making module to use common sense knowledge that is hard to encode manually; 3) natural language interactions, which require the agent to understand and operate on more than just template-based commands, but also a context of goals, constraints, and partial plans expressed in everyday language. To enable a more comprehensive approach to these complex challenges, the inclusion of researchers and practitioners from a broader range of fields is critical.

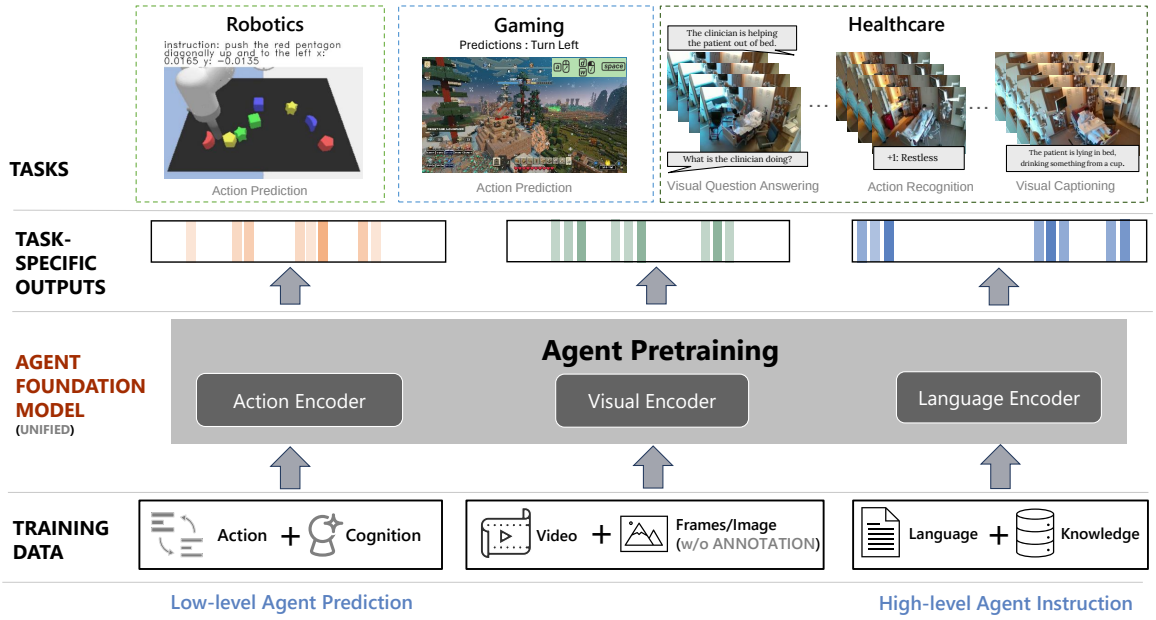


Figure 3. Overview of our Interactive Agent framework. Our foundation model is designed to process multi-modal information that conveys various levels of abstraction. This approach facilitates a comprehensive understanding of the context and environment, thus ensuring that actions are coherent. By training on a variety of task domains and applications, we develop a versatile foundation model that can be fine-tuned for executing optimal actions in a variety of contexts, paving the way towards generally intelligent agents.

4. Agent Foundation Model

Our proposed framework is shown in Figure 3. By synergistically combining visual perception with linguistic understanding, our models offer the potential to endow robots with a more intuitive understanding of their surroundings and better contextual reasoning. Our current work focuses on developing a joint image and video encoder and aligning this joint encoder to existing foundation models. This has several notable benefits: firstly, it allows for the use of both action, image, and video with language datasets for pre-training. Secondly, it increases the capabilities of the model across a variety of downstream tasks (e.g., video understanding, temporal reasoning, action prediction, interaction with human feedback, etc.). Finally, by using a joint encoder, we can reduce the overall model size (instead of using two separate encoders), which can be useful for edge deployments or in limited computing scenarios such as robotics, gaming, and interactive healthcare tasks.

4.1. Model Architecture

To effectively initialize our model to handle text, visual, and agent tokens as input, we initialize our architecture with two pre-trained submodules. First, we use CLIP ViT-B16 from (Radford et al., 2021) to initialize our visual encoder, denoted E_θ, and initialize our action and language model, F_ϕ, from OPT-125M (Zhang et al., 2022). We encode each frame in a video V_i as visual features $Z_i = E_\theta(V_i)$. We enable cross-modal information sharing by training an additional linear layer ℓ that transforms the embeddings of our visual encoder E_θ into the token embedding space of our transformer model F_ϕ. Thus, given a text prompt W and a single video frame V_i, we can obtain $\hat{A}$, a text token or action token prediction via $\hat{A} = F_\phi(W, \ell(E_\theta(V_i)))$. To incorporate prior time steps into our model, we also include the previous actions and visual frames as input during pre-training. For a given time step t, we predict $\hat{A}_t$ as

$$\hat{A}_t = F_\phi(W, \ell(E_\theta(V_1)), A_1, \ell(E_\theta(V_2)), A_2, \ldots, \ell(E_\theta(V_{t-1})), A_{t-1}, \ell((E_\theta(V_t))). \quad (1)$$

In practice, due to memory constraints, we only handle the previous M actions and frames, and update the previous V_i and A_i as a sliding window. In order to more effectively train our visual encoder to predict masked visual tokens, we use sinusoidal positional embeddings, as in (He et al., 2022) instead of the positional embeddings of CLIP. Since we are using relatively small checkpoints, we are able to jointly train our entire model during pre-training, unlike previous visual-language models that largely rely upon frozen submodules and seek to learn an adaptation network for cross-modal alignment (Alayrac et al., 2022; Li et al., 2022; Liu et al., 2023). We show our general process for formatting our input tokens in Figure 4, and describe our pre-training strategy in Section 4.2. For additional details, see Appendix A.

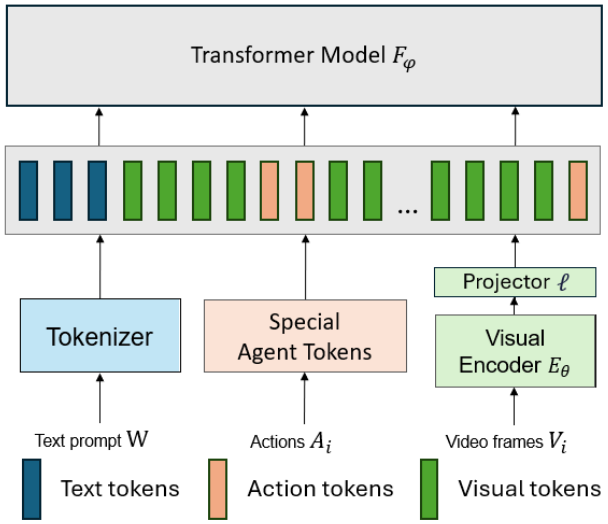


Figure 4. Our Unified Tokenization Framework. We propose a general pre-training strategy for predicting input tokens. For text tokens, we use the standard language modeling task with next token prediction. For actions, we expand the vocabulary of the language model to include special "agent" tokens that represent each of the actions available to the language model. Finally, we incorporate visual tokens into our framework by training a visual encoder to predict masked visual tokens.

4.2. Pre-Training Strategy

We pre-train our model on a wide range of robotics and gaming tasks, with each input sample containing text instructions, videos, and action tokens. We notate each sample as a sequence $S = (W, V_1, A_1, V_2, A_2, \ldots, V_T, A_T)$, where W is the sequence of tokens corresponding to the text instruction, V_i is the sequence of image patches corresponding to frame i, and A_i is the sequence of action tokens corresponding to the frame i of a video sequence of T frames. We denote w_j as the tokens of the text prompt W, and denote the parameters of our model as θ. For each sample, there are three components to the loss function: language modeling, masked image auto-encoding, and action modeling.

The language modeling loss is a standard causal language modeling loss to minimize the negative log likelihood of each token in the instruction conditioned on prior tokens. The language modeling loss for a particular sample S is

$$L_{lang}(S) = -\sum_{j=1}^{|W|} \log p_\theta(w_j | w_{<j}). \tag{2}$$

The masked image autoencoding loss is generated by randomly masking 75% of the image patches and calculating the mean-squared error between the reconstructed image and original image in pixel space for the masked image patches. The masked auto-encoder loss for a particular sample, S is:

$$L_{mae}(S) = \sum_{t=1}^{T} ||\textsc{u}(V_t) - \textsc{u}(D_\theta(E_\theta(\textsc{m}(V_t))))||_2^2, \tag{3}$$

where M randomly masks 75% of the image patches, U only selects the previously masked out features, and E_θ and D_θ are the encoder and decoder for the vision module, respectively.

Finally, the action modeling loss minimizes the negative log-likelihood of each action token conditioned on all prior information, including all text tokens, prior visual tokens, and prior action tokens. The action modeling loss for a particular sample S is:

$$L_{act}(S) = -\sum_{t=1}^{T} \sum_{i=1}^{|A_t|} \log p_\theta((a_t)_i | W, V_{\leq t}, A_{\leq t}, (a_t)_{<i}). \tag{4}$$

The full loss function for each sample combines the above components:

$$L(S) = \frac{L_{lang}(S) + L_{mae}(S) + L_{act}(S)}{|W| + \sum_{t=0}^{T}(|V_t| + |A_t|)}. \tag{5}$$

On robotics data, we only use $T = 4$ frames of video as input since the tasks are Markovian and therefore do not require long histories to accurately predict the next action. Our gaming data samples use $T = 9$ frames of video as input since an observation history is necessary for the partially-observable gaming tasks.

5. Tasks

We believe that a foundational model, trained in visual, language, and agent capabilities, leads to a powerful and general-purpose tool that significantly impacts a variety of interactive tasks. To evaluate the effectiveness of our approach, we applied the model to three major agent-AI scenarios, encompassing representative downstream tasks: 1) Robotics: human-machine manipulation in the physical world; 2) Gaming: human-machine embodiment in virtual reality; 3) Healthcare: augmented human-machine interaction in traditional multimodal tasks. For these tasks, the pre-trained model was fine-tuned with specific datasets. As a result, the model demonstrated reasonable and competitive performance in terms of action prediction, visual understanding, natural language-driven human-machine interactions, gaming, and hospital scene understanding. We outline the task definitions and specific datasets used below.

5.1. Robotics Tasks

For the robotics scenario, we tested the model on language-guided manipulation tasks. To this end, we selected two

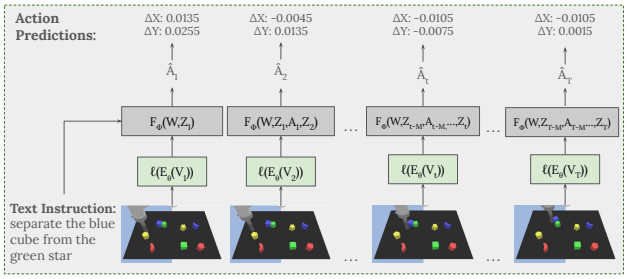


Figure 5. Our robotics and gaming pre-training pipeline. For simplicity, we use the same notation as in Sections 4.1 and 4.2; we represent our text instruction as W, input frames as V_t, our visual encoder and linear projection layer as E_θ and ℓ, respectively, our action and language transformer model as F_ϕ, and the predicted actions at time step t as $\hat{A}_t$.

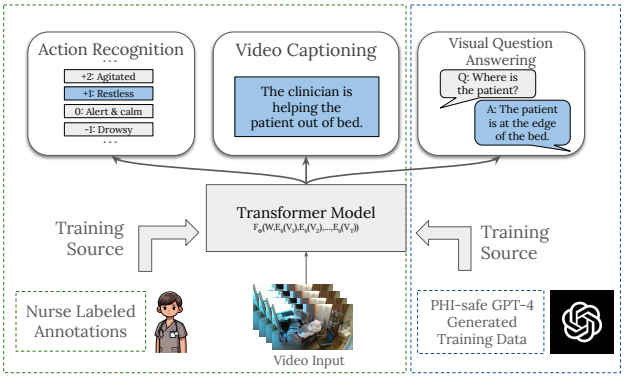


Figure 6. A High-level Overview of our Healthcare Tasks. We leveraged nurse-labeled annotations to train our multimodal agent on healthcare data. To adapt our model for visual question answering, we generated additional training data with GPT-4 using the PHI-safe process shown in Appendix B.

distinct robotics manipulation datasets: Language-Table (Lynch et al., 2023) and CALVIN (Mees et al., 2022). In the Language-table dataset, a robot gripper rearranged tabletop objects following language commands. The data were collected through teleoperation in a simulation, totaling 4.93 million frames. In the Calvin dataset, a 7-DOF robot manipulator performed manipulation tasks following relatively abstract instructions linked with a series of language commands. We utilized only the data containing language instructions, which amounted to 1.44 million frames. We chose these two datasets to gain insights into the model's performance across two dimensions: language-instruction abstraction and task-step length.

5.2. Gaming Tasks

Our primary gaming dataset consists of the Minecraft demonstrations collected by contractors in (Baker et al., 2022). In the original dataset, contractors were simply instructed to play Minecraft with no specific goal, and the dataset provided video gameplay synchronized with player actions and inventory metadata. However, since our architecture can leverage text instructions, we use GPT-4V to label videos with more specific instructions. Our prompt to GPT-4V also includes changes in the player's inventory over the video, which we found helped to reduce misclassifications of objects and actions in the video. In total, the Minecraft portion of our pre-training dataset consists of 4.7 million frames.

In addition to Minecraft, we also used a dataset of gameplay from Bleeding Edge, a team-base multiplayer game, which consists of video and synchronized player actions. Similarly, there are no specific instructions provided with the video, so we use GPT-4V to label the videos in our dataset. The Bleeding Edge portion of our pre-training dataset consists of 2.3 million frames across 7 different settings in the game.

5.3. Healthcare Tasks

In the healthcare domain we explored, our main dataset consisted of real-world recorded scenes from hospital ICU (intensive care unit) rooms using wall-mounted RGB cameras. Experienced ICU nurses generated captions of extracted 5-10 second video clips depicting common nursing activities in the ICU. We also included routine nursing documentation of important observations based on longer 5-30 minute windows, which included common clinical measures that assist with assessment and treatment of the patient's condition. For the analysis described in this paper, we focused on the RASS (Richmond Agitation-Sedation Scale) score used to assess the patient's state of agitation and sedation (Sessler et al., 2002) and the bed position to confirm that the head of the bed is at the proper angle to decrease the chance of acquiring a ventilator-associated pneumonia (Keeley, 2007). Both assessments are recorded frequently in the medical record and automated documentation has the potential to optimize caretaker time.

In order to fine-tune our model for human interactions in our ICU use case, we leveraged the nurse-provided video-clip captions and clinical documentation to have GPT-4 generate a synthetic video question-answer dataset that was used to expand the capabilities of our model after healthcare fine-tuning. A definite advantage of the GPT-4 generated derivative dataset is that it did not use any confidential patient data and consequently can be made publicly available to train any language-grounded clinical model. Figure 6 provides an overview of the healthcare tasks we evaluated: (1) video captioning, (2) video question answering, and (3) RASS score prediction (which we formulate as an activity recognition problem). For more information about our GPT-4 based question-answer generation procedure, see Appendix B.

6. Experiments

From a technical perspective, we are developing a generic artificial intelligence agent foundation model that can understand a wide array of input modalities and can produce coherent outputs and actions within a wide range of diverse interactive environments. In addition to evaluating our framework in these more specific domains, we evaluated the capabilities of our pre-training model on robotics manipulation, game playing, and interactive healthcare tasks. The details of the experimental setting and our main results are described in the following sub-sections.

6.1. Pre-training Experiments

To pre-train our model, we used the full training sets of Language Table, CALVIN, Minecraft, and Bleeding Edge, and trained for 100 epochs. We used a linear warmup cosine learning rate scheduler, with an initial learning rate of 0.0001. We initialized the vision component of our model with the CLIP base model with patch size 16, and initialized the language and action components with OPT-125M. We used 12 nodes of 16 V100 GPUs for 175 hours for all of our pre-training.

We added new action tokens corresponding to the actions used in our training set. All tasks include a token to indicate starting actions and a token to indicate ending actions. For Minecraft, there are additionally 23 button actions, and we discretized mouse actions to 100 bins along the x axis and 100 bins along the y axis. For Bleeding Edge, there are 11 button actions, and 2 joysticks. Each joystick has 256 possible values for rotation and 4 values for magnitude, resulting in a total of 520 joystick action tokens.

For robotics, we added new action tokens corresponding to valid actions in the environment, along with agent state tokens for proprioception. For all robotics data, we included a special action token to indicate the end of a trajectory. In Language Table, we included 21 binned actions for each of the x and y directions, representing the end effector translation target. We also included 21 binned state tokens representing the current end effector translation for each of the x and y directions, and an equal number of state tokens representing the previous robot action. In CALVIN, we included two actions for the gripper, indicating opening and closing, along with 21 actions for each of the six degrees of freedom of the end effector in the relative Cartesian displacement action space. We also included 21 binned states for each of the 14 attributes of the proprioceptive state, excluding the gripper action which has two states.

Our gaming dataset has 525,309 trajectories for Minecraft and 256,867 for Bleeding Edge, each consisting of 9 frames. Our robotics dataset consists of 1,233,659 trajectories for Language-Table and 360,566 for CALVIN, each consisting of 4 frames. Therefore, our total dataset consists of 13,416,484 frames. When sampling trajectories to train our model, we additionally added color jitter to each of the images, randomly scaling the brightness and saturation between 70% and 140%, and randomly shifting the hue by at most 0.05. We plot our pre-training loss in Figure 7.

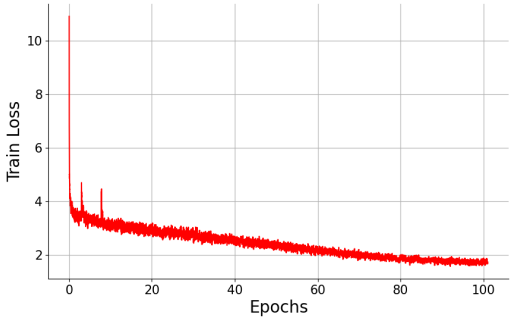


Figure 7. Plot of total pre-training loss over 100 epochs.

6.2. Robotics Experiments

The pre-trained model was fine-tuned for the Language-Table and CALVIN datasets and evaluated separately. For fine-tuning, we used the same pipeline as in pre-training, maintaining the original MAE and language-modeling loss functions, and the original vocabulary size. During fine-tuning, 50% of the image patches were masked, while no masking was involved in the evaluation.

6.2.1. Language-Table

In the Language-table dataset, we used data from a setup involving a total of 8 blocks, out of which 6 blocks were non-manipulated and unrelated to the tasks. This setup resulted in 181,020 trajectories. We split each trajectory into a series of 4 frames to fit our model architecture, resulting in 1,233,659 samples for fine-tuning. To investigate performance against different task characteristics, the model was evaluated on 5 different subtasks: 1) moving a block to another block; 2) moving a block relative to another block; 3) moving a block to an absolute position; 4) moving a block to a relative position; 5) separating two blocks. For each task, 50 trajectories were randomly sampled and evaluated three times, and the average success rate was computed. While the pre-trained model performed better than training from scratch (Table 1), our model was outperformed by other models such as (Brohan et al., 2023), which could be attributed to the fact that we used less data for pre-training, only using the human-teleoperated data in the Language-Table, CALVIN, and gaming datasets.

Table 1. Results for robotics fine-tuning across tasks on CALVIN and Language-Table, along with their corresponding evaluation metrics.

	CALVIN						LANGUAGE TABLE
MODEL	1 STEP	2 STEP	3 STEP	4 STEP	5 STEP	AVG LENS	SUCCESS RATE
MCIL	37.3	2.7	0.2	0.0	0.0	0.4	—
OURS (FROM SCRATCH)	20.6	0.8	0.0	0.0	0.0	0.214	40.0
OURS	**64.8**	**29.0**	**12.3**	**4.7**	**1.9**	**1.127**	**42.0**

Table 2. Performance metrics for gaming data. We report BLEU-4 scores for action prediction in Minecraft (abbreviated as MC), and Bleeding Edge (abbreviated as BE). We choose the last epoch for the pre-trained model and the epochs with the best validation score for the other models.

MODEL	MC (BLEU-4)↑	BE (BLEU-4)↑
OURS (FROM SCRATCH)	0.174	0.238
OURS (PRE-TRAIN ONLY)	0.170	0.249
OURS (PRE-TRAIN AND FINE-TUNED)	**0.272**	**0.411**

6.2.2. CALVIN

In the CALVIN dataset, each long-step trajectory was split into a series of 4 frames, resulting in 360,566 samples across 34 tasks for fine-tuning. To better capture the entire scene, the third-person view RGB camera was chosen as the source of image input from the available camera resources. For fine-tuning, we incorporated all available appearance settings, including the one used for testing, to enlarge the dataset, following the standard $ABCD \rightarrow D$ task definition. To evaluate the model performance with multiple steps, we computed the averaged success rate at each step, following the methodology described in the original CALVIN paper (Mees et al., 2022). Compared to Multi-context Imitation Learning (MCIL) (Lynch & Sermanet, 2021), our model shows better performance while only using 1% of the data (Table 1).

6.3. Gaming Experiments

For both gaming settings of Minecraft and Bleeding Edge, we evaluated our model's ability to predict actions given video frames and high-level instructions, along with its MAE reconstruction quality. Specifically, we used a held-out test dataset of 100 videos each, formatted in the same manner as our training data.

We report the BLEU-4 scores of actions in Table 2. We compare our pre-trained baseline to fine-tuning on task-specific data initialized from our pre-trained model and a version initialized from CLIP and OPT. We find that both fine-tuned models over-fit to the training data within 5 epochs, so we report the BLEU-4 test scores from the checkpoints with the highest validation score. We find that fine-tuning our pre-trained model is significantly more effective than training from scratch for both gaming domains, highlighting the importance of our diverse pre-training mixture. We also show a visualization of predicted actions from our fine-tuned model compared to the validation ground-truth in Appendix E.

6.4. Healthcare Experiments

For our experiments on our healthcare dataset, we evaluated our model's ability on three separate downstream tasks: video captioning, visual question answering, and activity recognition in the form of RASS score prediction. We used the final checkpoint from our pre-training run as described in Section 6.1.

Healthcare Setting For visual question-answering, we use the question as the text prompt W, and use the fixed text prompt "A video of" for video captioning. We train our model to the corresponding text tokens of the caption or answer and report the average perplexity across both settings. We frame RASS score prediction as a 10-way activity classification problem, and train a separate classification head for our model. We use the video-level setting for our visual encoder with 9 frames as input, as described in Appendix A. To evaluate the effectiveness of our pre-training framework, we compared the performance of our model against three baselines that leverage CLIP and OPT for initialization. First, we compared against a *frozen* baseline that uses the same pre-trained models, kept frozen, while fine-tuning a single linear layer for cross modal information passing, similar to (Liu et al., 2023). Second, we compared against a *joint* baseline that uses the same pre-trained models but fine-tunes them jointly along with the linear layer. For both of these baselines, we encode frames with CLIP individually and concatenate the frame-level embeddings. Third, we compared against a baseline of our same architecture, that makes use of our video-level encoder and is initialized from CLIP and OPT, but does not use any large-scale agent pre-training. We show our performance against the proposed baselines in Table 4. For all results, we train for 20 epochs on 4 16GB V100 GPUs with a fixed learning rate of 4e-5 and report results on a held-out evaluation set. For fair comparison, we do not perform any additional hyperparameter search.

7. Ablations and Analysis

Pretraining Loss Curves: We plot our combined pre-training loss across 100 epochs in Figure 7, and show individual components of the loss function in Appendix C.

Text instruction	Start frame	Predicted Action	Ground Truth Action
the player is digging and placing dirt blocks to terraform the terrain around their house...		[STARTACTION] [attack] [CAMERAX0] [CAMERAY-1] [ENDOFACTION]	[STARTACTION] [attack] [ENDOFACTION]

Table 3. We show 5 demonstrations from a held-out Minecraft dataset. In addition to the high level instruction, we show the low-level predicted actions and ground truth actions. We truncate the instructions to show only the parts relevant to the current frames. The most common errors are slight differences in camera movements and occasionally performing unnecessary actions. Note that sometimes the ground truth values are not the only valid actions; for instance, the fourth example predicts that the player will click the bottle, which happens a few frames later in the ground truth trajectory.

Table 4. Performance on healthcare text generation and RASS score action recognition, along with the corresponding evaluation metrics. Agent pre-training on robotics and gaming data improves performance for action recognition, but does not improve text generation abilities.

MODEL	PERPLEXITY ↓	RASS ACC ↑
CLIP + OPT (FROZEN)	**93.3**	55.4
CLIP + OPT (UNFROZEN)	102.7	92.6
OURS (FROM SCRATCH)	100.0	70.3
OURS (AGENT PRE-TRAINED)	106.3	**95.7**

Comparisons with GPT-4V: In Figure 10, we show how our model has the ability to output low-level action predictions, while GPT-4V is unable to consistently output low-level controls. While our model is able to output precise movements and actions, GPT-4V only outputs high-level instruction.

Effects of Agent Pre-Training: In Table 2 and Table 4, we demonstrate the effectiveness of our agent pre-training strategy compared to training from scratch and training against an equivalent visual-language baseline. In particular, we show that a commonly used approach for fine-tuning visual-language models by using frozen visual encoders, similar to LLaVA (Liu et al., 2023) or Mini-GPT-4 (Zhu et al., 2023), performs worse than joint fine-tuning for action recognition on our healthcare dataset. Furthermore, our agent pre-training boosts performance for action prediction across all gaming and robotics datasets.

8. Conclusion

We introduced an Interactive Agent Foundation Model designed to take text, action, and visual inputs. We found that by pre-training on a mixture of robotics and gaming data, our model is effective in modeling actions across a variety of domains, even showing positive transfer when fine-tuning in unseen domains such as healthcare. The generality of our framework allows it to be broadly applicable across decision-making settings, unlocking new possibilities for generalist agents in multimodal systems.

9. Impact Statement

This paper presents the initial steps on making interactive agents possible through an Interactive Agent Foundation Model. We do not foresee negative societal consequences from presenting and open-sourcing our current work. In particular, the main output of our model is domain-specific actions, such as button inputs for gaming data, making the downstream applications of our model different from those of standard LLMs and VLMs.

In the domain of robotics, we wish to emphasize that our model should not be deployed on real robots without more training and additional safety filters.

In the domain of gaming, downstream applications of our foundation model may have some societal consequences. Smarter, more realistic AI characters could lead to more immersive worlds, which can increase players' enjoyment in games, but may also lead to social withdrawal if not used appropriately. Specifically, more realistic AI characters could potentially lead to video game addiction and players anthropomorphising artificial players. We encourage game developers who build AI agents using our models to mitigate these potential harms by encouraging social interactions between human players and applying appropriate content filters to AI agents.

In the domain of healthcare, we emphasize that our models are not official medical devices and have not gone through rigorous testing in live settings. We strongly discourage using our models for self-prescription. Even as our models improve in future iterations, we strongly encourage keeping a medical practitioner in the loop to ensure that unsafe actions are avoided. As our models continue to develop, we believe that they will be useful to caretakers, especially by automatically forming drafts of documentation and notifying caretakers when patients may need urgent attention.

Finally, we note that the capabilities of agent AI models may significantly change at scale. As we scale our model in terms of architecture, compute, and training data, we

will actively monitor its capabilities before releasing new versions publicly.

Acknowledgements

We are especially grateful to Desney Tan, Peter Lee, Doug Burger, Ryen White, Ece Kamar, John Langford, Jonathan Carlson and Microsoft's Office of the CTO (OCTO) for their advice, enormous support, and encouragement. We appreciate the Microsoft gaming team, Microsoft X-box team, Microsoft 343 team, Kareem Choudhry, Haiyan Zhang, Spencer Perreault, Dave Bignell, Katja Hofmann, Sam Devlin, Shanzheng Tan, and Raluca Georgescu for the gaming data collection and sharing. We thank Bill Dolan, Nebojsa Jojic, Sudha Rao, Adrian Brown, Andrzej Banburski-Fahey, and Jianwei Yang for their early insightful discussions and help with the gaming aspects of our project. We appreciate Kiran Muthabatulla and the MSR Central Engineering (CE) team for their discussion and feedback for the project. The authors gratefully acknowledge the Microsoft HoloLens team, Microsoft Mesh team, and Antonio Criminisi for their generous provision of equipment and project discussions. Finally, we would like to express our genuine appreciation for Jim Jernigan, Ben Huntley, Oleg Losinets, the Microsoft AOAI team, and the GCR team for their Azure-OpenAI endpoint support and their pointers to the literature.

We would also like to thank our colleagues from Stanford's Partnership in AI-assisted Care, who helped inform the medical applications explored in this work. In particular, we would like to thank Amit Kaushal and Roger Bohn for their clinical expertise and guidance. Additionally, we greatly appreciate Zelun Luo, David Dai, and Dev Dash for their participation as actors for our hospital dataset.

This research was supported by Microsoft Research Project Green 2024, Microsoft Research Project Fair 2023, Stanford University, University of California at Los Angeles, MSR Accelerator team, and the Microsoft OCTO team.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Amjad, A., Kordel, P., and Fernandes, G. A review on innovation in healthcare sector (telehealth) through artificial intelligence. *Sustainability*, 15(8):6655, 2023.

Bain, M., Nagrani, A., Varol, G., and Zisserman, A. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1728–1738, 2021.

Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cai, W., Huang, S., Cheng, G., Long, Y., Gao, P., Sun, C., and Dong, H. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. *arXiv preprint arXiv:2309.10309*, 2023.

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

Dorbala, V. S., Sigurdsson, G., Piramuthu, R., Thomason, J., and Sukhatme, G. S. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022.

Dorbala, V. S., Mullen Jr, J. F., and Manocha, D. Can an embodied agent find your" cat-shaped mug"? llm-based zero-shot object navigation. *arXiv preprint arXiv:2303.03480*, 2023.

Du, Y., Yang, M., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J. B., et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

Durante, Z., Huang, Q., Wake, N., Gong, R., Park, J. S., Sarkar, B., Taori, R., Noda, Y., Terzopoulos, D., Choi, Y., et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*, 2024.

Gadre, S. Y., Wortsman, M., Ilharco, G., Schmidt, L., and Song, S. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171–23181, 2023.

Gong, R., Gao, X., Gao, Q., Shakiah, S., Thattai, G., and Sukhatme, G. S. Lemma: Learning language-conditioned multi-robot manipulation. *IEEE Robotics and Automation Letters*, 2023a.

Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S.-C., Terzopoulos, D., Fei-Fei, L., et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023b.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *CVPR*, 2022.

Huang, C., Mees, O., Zeng, A., and Burgard, W. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615. IEEE, 2023.

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. Vima: General robot manipulation with multimodal prompts. *arXiv*, 2022.

Katara, P., Xian, Z., and Fragkiadaki, K. Gen2sim: Scaling up robot learning in simulation with generative models. *arXiv preprint arXiv:2310.18308*, 2023.

Keeley, L. Reducing the risk of ventilator-acquired pneumonia through head of bed elevation. *Nursing in critical care*, 12(6):287–294, 2007.

Kumar, K. N., Essa, I., and Ha, S. Words into action: Learning diverse humanoid robot behaviors using language guided iterative motion refinement. *arXiv preprint arXiv:2310.06226*, 2023.

Lee, P., Bubeck, S., and Petro, J. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239, 2023.

Li, B., Wu, P., Abbeel, P., and Malik, J. Interactive task planning with language models. *arXiv preprint arXiv:2310.10645*, 2023a.

Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K., Gokmen, C., Dharan, G., Jain, T., et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., and Gao, J. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023b.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

Li, J., Gao, Q., Johnston, M., Gao, X., He, X., Shakiah, S., Shi, H., Ghanadan, R., and Wang, W. Y. Mastering robot manipulation with multimodal prompts through pretraining and multi-task fine-tuning. *arXiv preprint arXiv:2310.09676*, 2023c.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023d.

Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023e.

Liang, X., Ma, L., Guo, S., Han, J., Xu, H., Ma, S., and Liang, X. Mo-vln: A multi-task benchmark for open-set zero-shot vision-and-language navigation. *arXiv preprint arXiv:2306.10322*, 2023.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning, 2023.

Lynch, C. and Sermanet, P. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. URL https://arxiv.org/abs/2005.07648.

Lynch, C., Wahid, A., Tompson, J., Ding, T., Betker, J., Baruch, R., Armstrong, T., and Florence, P. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Parakh, M., Fong, A., Simeonov, A., Gupta, A., Chen, T., and Agrawal, P. Human-assisted continual robot learning with foundation models. *arXiv preprint arXiv:2309.14321*, 2023.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

Puig, X., Undersander, E., Szot, A., Cote, M. D., Yang, T.-Y., Partsey, R., Desai, R., Clegg, A. W., Hlavac, M., Min, S. Y., et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rawte, V., Sheth, A., and Das, A. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.

Sessler, C. N., Gosnell, M. S., Grap, M. J., Brophy, G. M., O'Neal, P. V., Keane, K. A., Tesoro, E. P., and Elswick, R. The richmond agitation–sedation scale: validity and reliability in adult intensive care unit patients. *American journal of respiratory and critical care medicine*, 166 (10):1338–1344, 2002.

Shah, D., Osiński, B., Levine, S., et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pp. 492–504. PMLR, 2023a.

Shah, R., Martín-Martín, R., and Zhu, Y. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023b.

Srivastava, S., Li, C., Lingelbach, M., Martín-Martín, R., Xia, F., Vainio, K. E., Lian, Z., Gokmen, C., Buch, S., Liu, K., et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pp. 477–490. PMLR, 2022.

Sun, J., Zhang, Q., Duan, Y., Jiang, X., Cheng, C., and Xu, R. Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning. *arXiv preprint arXiv:2309.11359*, 2023a.

Sun, Q., Fang, Y., Wu, L., Wang, X., and Cao, Y. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023b.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., and Ikeuchi, K. Gpt models meet robotic applications: Co-speech gesturing chat system. *arXiv preprint arXiv:2306.01741*, 2023a.

Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., and Ikeuchi, K. Chatgpt empowered long-step robot control in various environments: A case application. *IEEE Access*, 11:95060–95078, 2023b. doi: 10.1109/ACCESS.2023.3310935.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Xu, M., Huang, P., Yu, W., Liu, S., Zhang, X., Niu, Y., Zhang, T., Xia, F., Tan, J., and Zhao, D. Creative robot tool use with large language models. *arXiv preprint arXiv:2310.13065*, 2023.

Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K.-H., Arenas, M. G., Chiang, H.-T. L., Erez, T., Hasenclever, L., Humplik, J., et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Zhou, G., Hong, Y., and Wu, Q. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023.

Appendix

A. Architecture Details

To effectively handle images and video inputs jointly, we use a divided space-time attention similar to (Bain et al., 2021). We initialize our visual encoder from CLIP ViT-B16 (Radford et al., 2021), and learn temporal attention layers after each spatial attention layer. We further mask 75% of the image patches (using tubelet masking for videos) during training, and use a MAE-decoder similar to (He et al., 2022). Gaming and robotics use a frame-level visual encoder so that the agent is able to observe a continuous stream of tokens and act after every frame. For healthcare, we leverage the video understanding capabilities of our visual encoder since the tasks are video-level.

B. GPT-4 Prompting

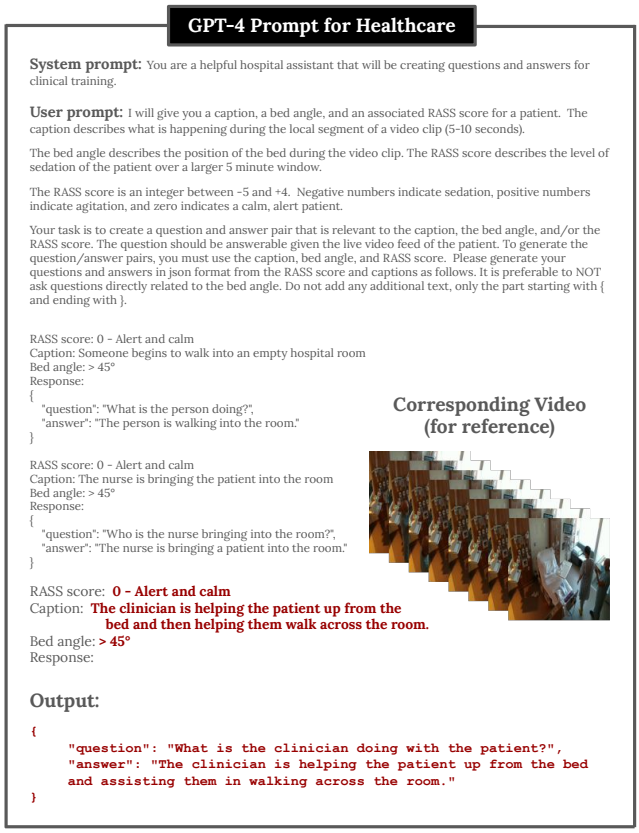


Figure 8. **Our PHI-safe GPT-4 Prompt for Generating Healthcare QA Examples.** By ensuring the usage of non-identifying video captions and documentation data, we prevent any identifiable patient data leakage to GPT-4 while simultaneously generating additional visual-language training data. For the particular example shown, we use a RASS score of "0 - Alert and calm", a caption of "The clinician is helping the patient up from the bed and then helping them walk across the room.", and a bed angle of "$> 45°$".

We show our GPT-4 Prompt for Healthcare Visual Question Answering generation in Figure 8, and our GPT-4V Prompt for gaming instruction generation in Figure 9.

C. Pre-training Loss Curves

We show all components of the loss function in Figure 11 and plot our combined pre-training loss across 100 epochs in Figure 7.

GPT-4-Vision

Prompt:

These are frames of a video of a Bleeding Edge player ordered from left to right and top to bottom as a grid. Give a simple, but precise description of what the player is doing in 1 sentence. Be specific about important items, entities, and actions. In your description do not mention specific frame numbers or the name of the game.

Video input:

Output:

```
The player begins by running around the
map, passing through different
checkpoints and interacting with several
capture points, then fights against an
enemy player, and finally captures an
objective while being attacked by another
enemy.
```

Figure 9. Our GPT-4V prompt for games like Bleeding Edge that have 3rd person viewpoints and visually complex scenes. In order to input a large number of frames (48) to GPT-4V, we input the frames as a grid with frame numbers overlaid on each frame (as shown above).

D. Gaming Task Pipeline

We provide an example of our pipeline for a gaming task in Figure 12. Note the similarities to the robotics task in Figure 5 since both tasks require predicting an action given a text instruction and sequence of prior actions.

E. Example Outputs

We show examples of our model predicting actions on unseen, robotics simulation data in Table 5 and 6. We show example outputs for healthcare in Table 7, and show example outputs for gaming in Table 8 and 9.

Figure 10. When using GPT-4V to choose actions given a history of frames, we find that it gives reasonable high-level actions but does not choose precise low-level actions, highlighting the importance of our pre-trained model.

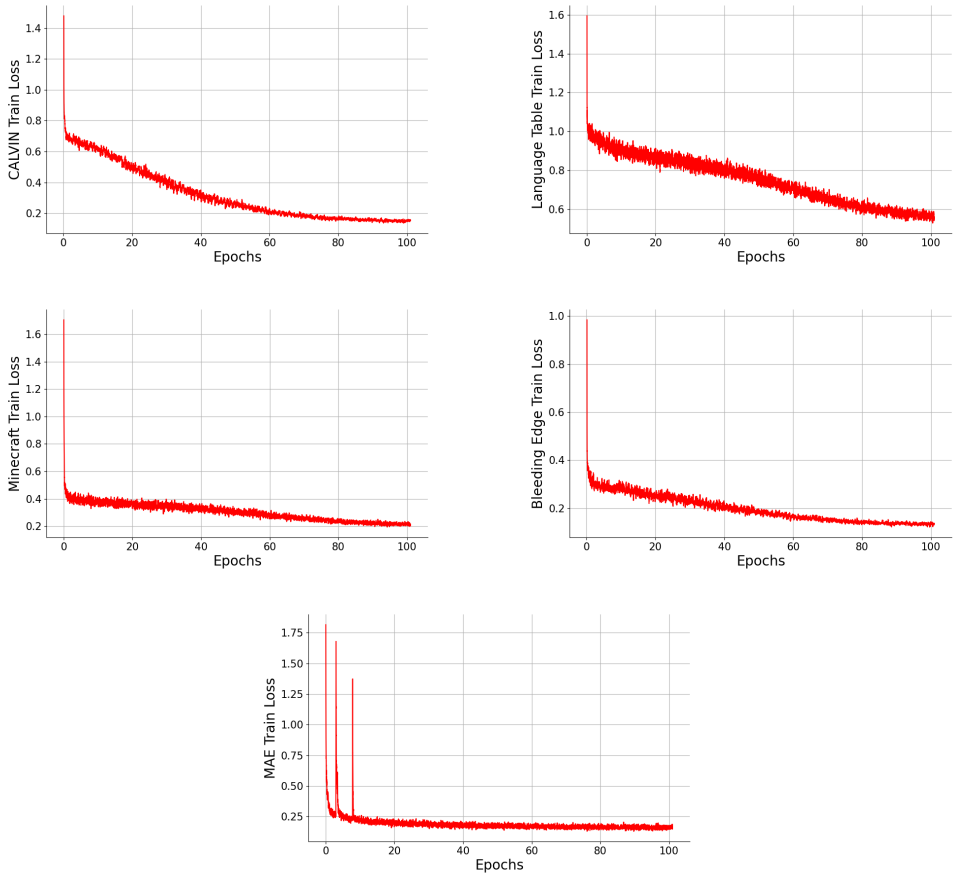


Figure 11. Plot of all components of the training loss over the 100 epochs of pre-training.

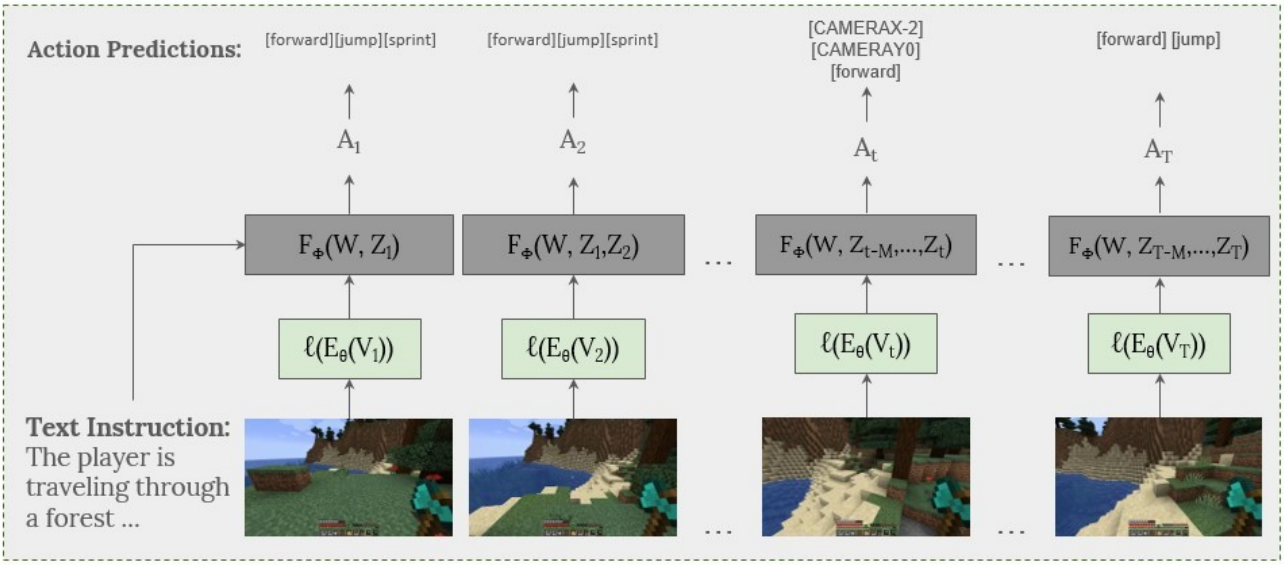


Figure 12. Our gaming pre-training pipeline. For simplicity, we use the same notation as in Sections 4.1 and 4.2; we represent our text instruction as W, input frames as V_t, our visual encoder and linear projection layer as E_θ and ℓ, respectively, our action and language transformer model as F_ϕ and the predicted actions at time step t as $\hat{A}_t$.

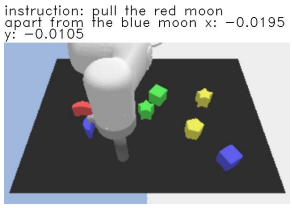
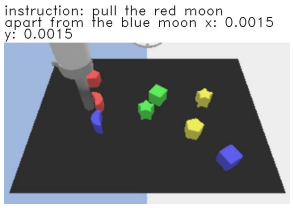
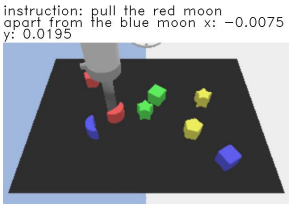
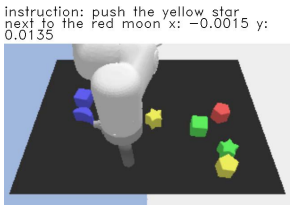
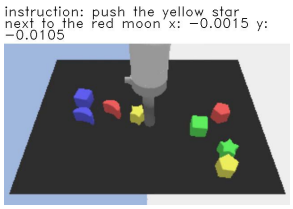
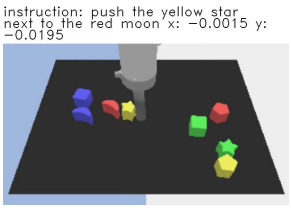
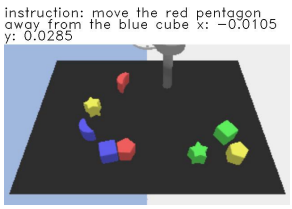
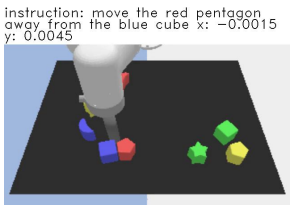
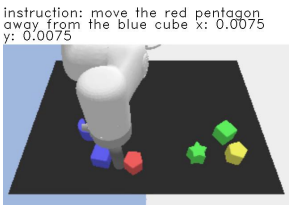
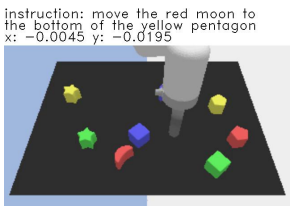
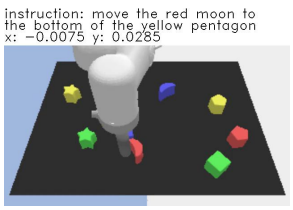
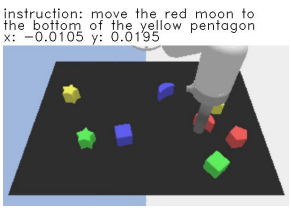
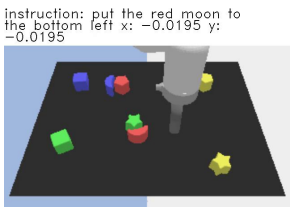
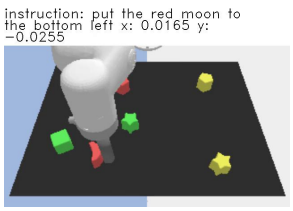
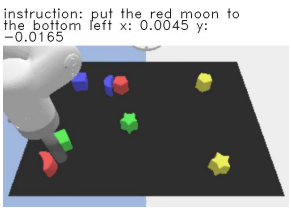
Text instruction	Start frame	Middle frame	End frame
Pull the red moon apart from the blue moon.	 →	 →	
Push the yellow start next to the red moon.	 →	 →	
Move the red pentagon away from the blue cube.	 →	 →	
Move the red moon to the bottom of the yellow pentagon.	 →	 →	
Pull the red moon to the bottom left.	 →	 →	

Table 5. We show 5 unique demonstrations from Language Table, where our model successfully follows the text instruction. In addition to the high level instruction, we also show the low-level predicted actions of our agent above each frame.

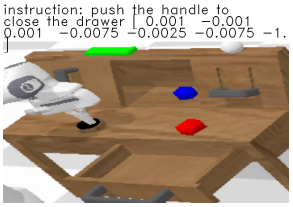
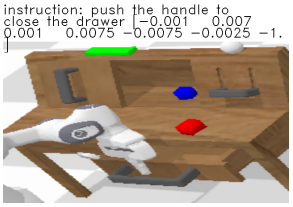
Text instruction	Start frame	Middle frame	End frame
Push the handle to close the drawer.	 →	 →	
Lift the red block from the sliding cabinet.	 →	 →	
Pull the handle to open the drawer.	 →	 →	
Push the sliding door to the left side.	 →	 →	
Push the sliding door to the right side.	 →	 →	

Table 6. We show 5 unique demonstrations from CALVIN, where our model successfully follows the text instruction. In addition to the high level instruction, we also show the low-level predicted actions of our agent above each frame.

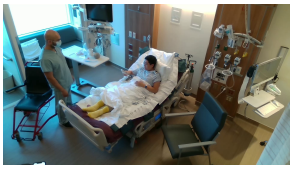
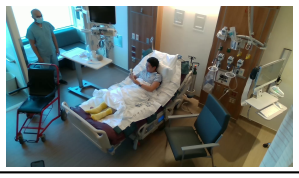
Task	Start frame	End frame	Model Output
Video Captioning			The patient is awake and calm. The patient is cooperative. The patient is alert
Video Question Answering			Q: Where is the patient? A: patient is in deep sedation. The patient likely requires assistance.
Action Recognition (RASS)			0 - Alert and calm
Video Captioning			The patient is awake and calm. They are speaking on the phone.

Table 7. We show 4 demonstrations of our agent model's outputs on a held-out Healthcare dataset that uses actors instead of actual patients. We demonstrate our model's outputs across 3 different tasks: video captioning, visual question answering, and RASS score prediction (action recognition). Due to the nature of our actor-collected example videos, the model predicts that the patient is awake and calm (RASS score of 0) for most video clips, despite only 60% of the training data containing RASS score of 0.

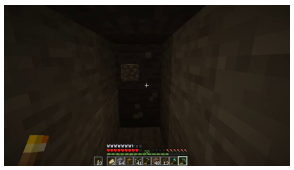
Text instruction	Start frame	Predicted Action	Ground Truth Action
the player is digging and placing dirt blocks to terraform the terrain around their house...		[STARTACTION] [attack] [CAMERAX0] [CAMERAY-1] [ENDOFACTION]	[STARTACTION] [attack] [ENDOFACTION]
the player is mining underground using a diamond pickaxe, gathering cobblestone, coal, iron ore...		[STARTACTION] [attack] [CAMERAX-3] [CAMERAY0] [ENDOFACTION]	[STARTACTION] [attack] [CAMERAX-3] [CAMERAY0] [ENDOFACTION]
the minecraft player is moving around a village ...		[STARTACTION] [forward] [sprint] [ENDOFACTION]	[STARTACTION] [forward] [sprint] [ENDOFACTION]
the player is using a brewing stand ...		[STARTACTION] [sneak] [use] [ENDOFACTION]	[STARTACTION] [sneak] [ENDOFACTION]
the player is ... terraforming by digging ...		[STARTACTION] [attack] [ENDOFACTION]	[STARTACTION] [attack] [ENDOFACTION]

Table 8. We show 5 demonstrations from a held-out Minecraft dataset. In addition to the high level instruction, we show the low-level predicted actions and ground truth actions. We truncate the instructions to show only the parts relevant to the current frames. The most common errors are slight differences in camera movements and occasionally performing unnecessary actions. Note that sometimes the ground truth values are not the only valid actions; for instance, the fourth example predicts that the player will click the bottle, which happens a few frames later in the ground truth trajectory.

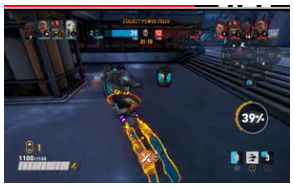
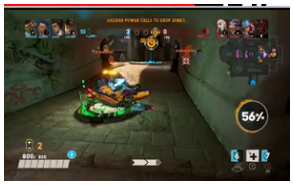
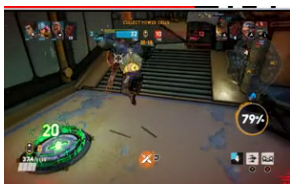
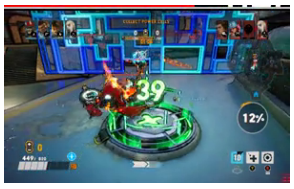
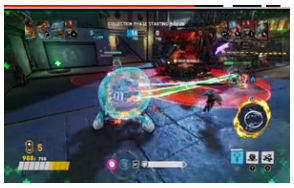
Text instruction	Start frame	Predicted Action	Ground Truth Action
the player is using a character with a sword to fight enemies and collect power cells ...		[STARTACTION] [lockon][meleeattack] [lrot214] [lmag4] [ENDOFACTION]	[STARTACTION] [lockon][meleeattack] [lrot213] [lmag4] [ENDOFACTION]
the player is riding a hoverboard-like vehicle ... avoiding or attacking enemy players ...		[STARTACTION] [lockon][meleeattack] [lrot204] [lmag4] [ENDOFACTION]	[STARTACTION] [lockon][meleeattack] [lrot201] [lmag4] [ENDOFACTION]
the player starts by descending some stairs towards an open area where they engage in combat with an enemy player ...		[STARTACTION] [jump] [lockon][specialability1] [lrot199] [lmag4] [ENDOFACTION]	[STARTACTION] [jump] [lockon][meleeattack] [lrot201] [lmag4] [ENDOFACTION]
the player ... captures an objective point while fighting off multiple opponents ...		[STARTACTION] [lockon][meleeattack] [lrot63] [lmag4] [ENDOFACTION]	[STARTACTION] [lockon][meleeattack] [lrot63] [lmag4] [ENDOFACTION]
a bleeding edge player is controlling a robot character with a sword ... engaging in combat with enemy players ...		[STARTACTION] [evade] [lrot236] [lmag4] [ENDOFACTION]	[STARTACTION] [evade] [lrot236] [lmag4] [ENDOFACTION]

Table 9. We show 5 unique demonstrations from a held-out Bleeding Edge dataset. In addition to the high level instruction, we show the low-level predicted actions and ground truth actions. We truncate the instructions to show only the parts relevant to the current frames. The most common errors are slight deviations from the precise value of the joysticks, which are naturally noisy. Some other errors include predicting the wrong type of attack, though this typically happens in situations where multiple attacks are still valid.