An Interactive Agent Foundation Model

Zane Durante[1,2†] Ran Gong[1,3†] Bidipta Sarkar[1,2†] Noaki Wake[1] Rohan Taori[2]
Paul Tang[2] Shrinidhi K. Lakshmikanth[2] Kevin Schulman[2] Arnold Milstein[2] Hoi Vo[4]
Ehsan Adeli[2] Demetri Terzopoulos[3] Li Fei-Fei[2] Jianfeng Gao[1]
[1]Microsoft Research, Redmond [2]Stanford University [3]UCLA [4] Microsoft Gaming

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 imitation learning, 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 [49]. With recent significant advances surrounding general-purpose foundation models [5], 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 [45, 48]. 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 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 [2, 11, 32, 34, 37]. 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/or actions as input. We treat each input type as separate tokens and pre-train a system to predict masked tokens across all three modalities. Our approach leveraeges 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 the environment across a wide range of domains and tasks.

In this paper, we show that a 277M parameter model that is jointly pre-trained across 13.4 M video frames from several distinct domains and data sources can be effectively adapted for interactive multi-modal settings using text, video, images, dialogue, captioning, visual question answering, and embodied actions across five disparate environments. In order to effectively evaluate the broad range of capabilities and generalization abilities of our model, we show results across three distinct domains: robotics, gaming, and healthcare. Despite using domain-specific visual inputs, text descriptions, and action-spaces, our model is

[†]Equal contribution. Work done while interning or researching part-time at Microsoft Research, Redmond. Corresponding: durante@stanford.edu

effectively able to generalize across all three domains. Importantly, we also find that our agent pre-training framework produces effective visual encoders that can generalize well in novel domains. To facilitate research in this area, we release our training code at the following URL.

2. Agent Foundation Model

Our proposed framework is shown in Figure 1. By combining visual perception, action understanding, and linguistic reasoning skills, our model offers the potential to endow robots with a more intuitive understanding of their surroundings and better contextual interactions. Our framework focuses on developing a unified pretraining methodology that can incorporate data sources containing only actions, agent states, images, videos, and language data or any combination thereof. Due to the flexibility of data inputs, our model benefits from increased adaptability across a variety of downstream tasks (e.g., video understanding, temporal reasoning, action prediction, interaction with human feedback, etc.). Finally, by using a relatively small transformer decoder (only 125M parameters) and a joint image and video encoder, we reduce our overall model size which can be useful for edge deployments or in limited computing scenarios such as robotics, gaming, and healthcare.

2.1. Model Architecture

To effectively initialize our model to handle text, visual, and action tokens as input, we initialize our architecture with two pre-trained submodules. First, we use CLIP ViT-B/16 from Radford *et al*. [47] to initialize our visual encoder, denoted E_θ, and initialize our action and language decoder model, F_ϕ, from OPT-125M [65]. Given a video input, our encoder uses temporal embeddings and Gated Temporal Attention blocks that only activate for multi-frame inputs (shown in Figure 2). We jointly train our visual encoder on two objectives: masked auto-encoding (MAE) on the visual inputs and next token prediction for the language and action targets. Our initial experiments revealed that CLIP's standard learned positional embeddings hindered convergence during pre-training on the MAE objective, leading us to use sinusoidal positional embeddings instead.

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. Since we are using relatively small checkpoints, we are able to jointly train our entire model during pre-training. This is in contrast to most previous visual-language models that largely rely upon frozen submodules and/or seek to learn an adaptation network for cross-modal alignment [2, 32, 37]. We show our general process for tokenization, cross-modal information sharing, and token prediction in Figure 3.

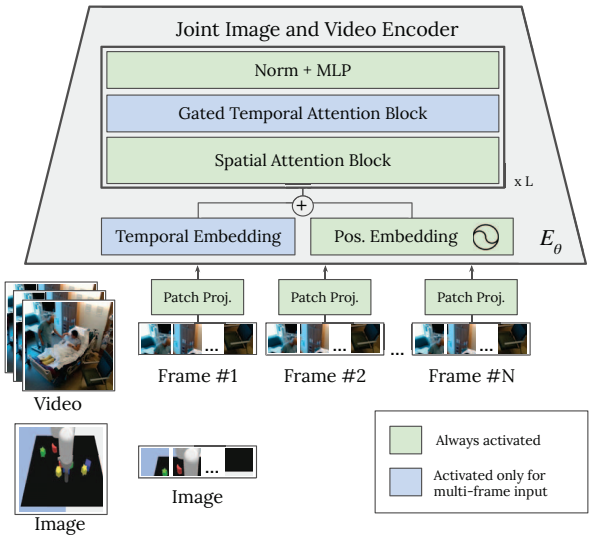

Figure 2. Our Visual Encoder, E_θ. We use a lightweight gated temporal attention mechanism that activates only for multi-frame inputs, allowing us to use most of the model parameters for both image and video inputs while still enabling cross-frame temporal attention within the visual encoder when desired. We use sinusoidal positional embeddings for effective MAE reconstruction.

2.2. 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. Note that each action token sequence A_i contains information about both the internal state of the agent and the embodied actions taken at frame i. We denote w_j as the tokens of the text prompt W, and denote the parameters of

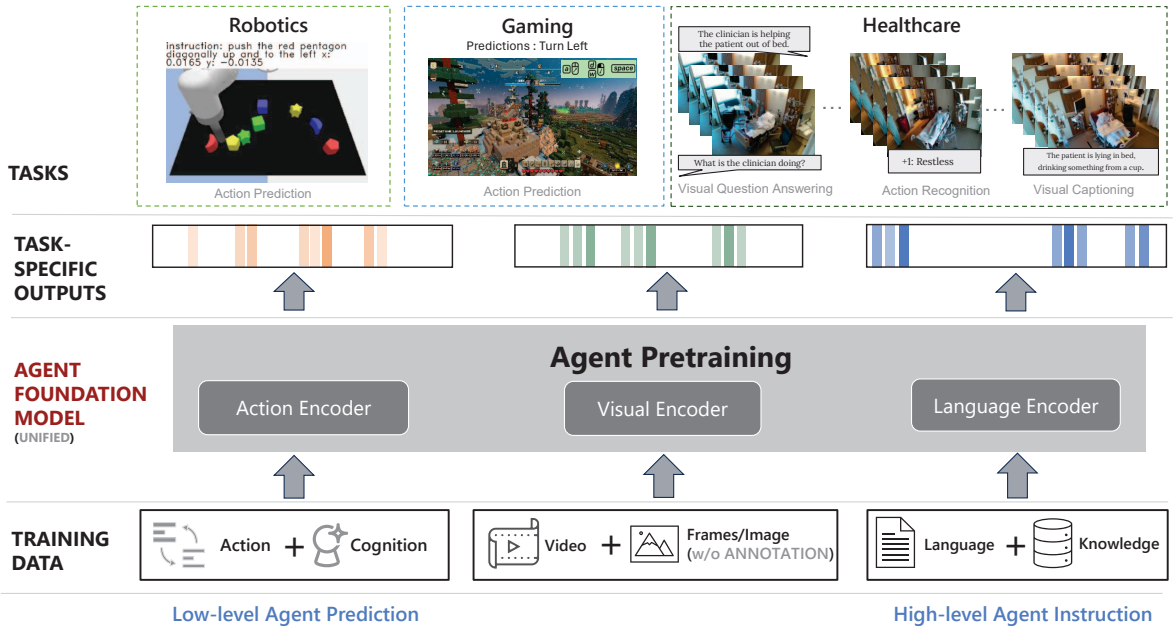

Figure 1. Overview of our Interactive Agent framework. Our foundation model is designed to process multi-modal information that conveys various levels of abstraction. This approach facilitates a comprehensive understanding of the context and environment, thus ensuring that actions are coherent. By training 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.

our language and action decoder model as ϕ and the parameters of our MAE encoder-decoder as θ. For each sample, there are three components to the loss: 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:

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

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:

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

where M randomly masks 75% of the image patches, U only selects the 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:

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

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.

3. Tasks

To evaluate the effectiveness of our approach, we applied our methodology to three distinct scenarios, encompassing representative downstream tasks and novel domains: (1) robotics, encompassing human-machine manipulation in the physical world; (2) gaming, allowing for interactive agents to be embodied in virtual reality; and (3) healthcare, an out of domain scenario where our methodology can

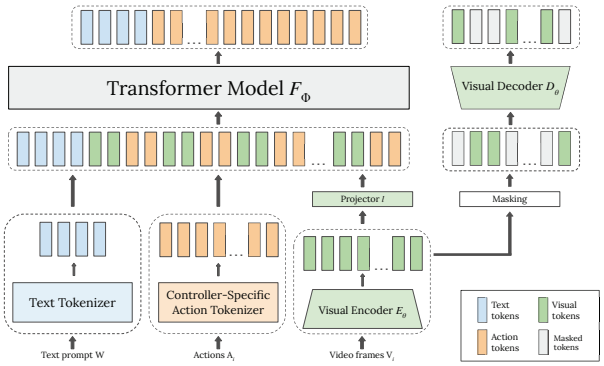

Figure 3. Our unified tokenization framework. We propose a general pre-training strategy for predicting input tokens. For text tokens, we use the standard language modeling task with next token prediction. For actions, we expand the vocabulary of the language model to include action and state tokens that represent each of the action and states possible for the agent across each specific domain. Finally, we incorporate visual tokens into our framework by training a visual encoder-decoder to predict masked visual tokens. F_ϕ is trained via causal masking over the entire input sequence S.

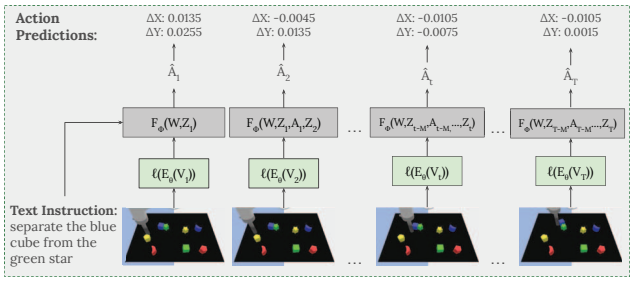

Figure 4. Our robotics and gaming pre-training pipeline. For consistency, we use the same notation as in Sections 2.1 and 2.2; we represent our text instruction as W, input frames as V_t, our visual encoder and linear projection layer as E_θ and ℓ, respectively, our action and language decoder model as F_ϕ, and the predicted actions at time step t as $\hat{A}_t$. In the figure above, we show an example prompt and action prediction set from Language Table, but note that our process is identical across all pre-training datasets.

be evaluated on relatively standard multimodal tasks. For each of these tasks, the pre-trained model was later 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.

3.1. Robotics Tasks

For the robotics scenario, we tested the model on language-guided manipulation tasks. To this end, we selected two distinct robotics manipulation datasets: Language-Table [39] and CALVIN [41]. 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.

3.2. Gaming Tasks

Our primary gaming dataset consists of the Minecraft demonstrations collected by contractors in [4]. 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.

3.3. 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 [51] 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 [25]. Both assessments are recorded frequently in the medical record and automated documentation has the potential to optimize caretaker time.

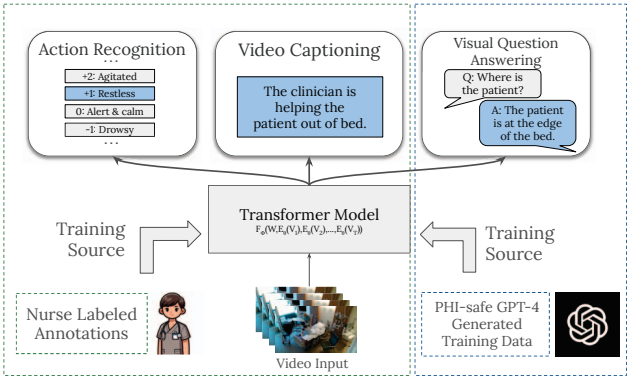

Figure 5. A High-level Overview of our Healthcare Tasks. We leveraged nurse-labeled annotations to train our multimodal agent on healthcare data. To adapt our model for visual question answering, we generated additional training data with GPT-4 using the PHI-safe process shown in the Appendix.

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 5 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 details about our GPT-4 based question-answer generation procedure, see the Appendix.

4. Experiments

To effectively evaluate the ability of our pre-trained model to serve as a foundation model for interactions, we ground our evaluation in real-world domains with diverse action sets, visual data, and tasks. We also choose healthcare as an illustrative out of domain setting, as no healthcare data was used during pre-training. The details of the experimental settings are described in the following sub-sections.

4.1. Pre-training Setup

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 1e-4. We used 12 nodes of 16 V100 GPUs for 175 hours for pre-training.

We extended the vocabulary of our transformer decoder by adding novel action tokens corresponding to the actions and agent states found in our joint pre-training dataset. 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 pretraining 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 6.

4.2. Robotics Experiments

Our final pre-trained checkpoint 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.

4.2.1. 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)

	CALVIN						LANGUAGE TABLE
MODEL	1 STEP	2 STEP	3 STEP	4 STEP	5 STEP	AVG LENS	SUCCESS RATE
RT-1 [6]	84.4	61.7	43.8	32.3	22.7	2.45	**74.0**
GR-2 [10]	98.6	96.1	93.1	90.1	85.9	4.64	—
MDT [50]	98.6	95.8	91.6	86.2	80.1	4.52	—
ROBOFLAMINGO [35]	96.4	89.6	82.4	74.0	66.0	4.08	—
MCIL [38]	28.2	2.5	0.3	0.0	0.0	0.31	—
3D-VLA [67]	44.7	16.3	8.1	1.6	0	-	
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 1. Results for robotics fine-tuning across tasks on CALVIN and Language-Table, along with their corresponding evaluation metrics. For Calvin, our method, MCIL and 3D-VLA only use a static camera. Other methods use both the static camera and the wrist camera.

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 gains were not particularly substantial (2% absolute improvement). Furthermore, our model was outperformed by other models such as RT-1 [6] which used significantly more robotics data for pre-training.

4.2.2. CALVIN

The CALVIN dataset comprises a much more challenging dataset than Language Table. 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 given in CALVIN [41]). 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 pape. When compared to Compared to Multi-context Imitation Learning (MCIL) [38], our model shows better performance while only using 1% of the data (Table 1). Furthermore, we significantly outperform 3D-VLA [66], the only other method evaluated in the same setting (third-person camera only). We also note that our agent pre-training significantly improved our model's performance for CALVIN (e.g., $>$ 43% absolute improvement in the 1 step setting).

4.3. 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.

In order to effectively evaluate the quality of generated outputs, we report the BLEU-4 scores of actions in Table 3. We chose BLEU-4 scores in order to evaluate the quality of model outputs when given the ground truth and without requiring access to a simulator. 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 Table 2 and the Appendix.

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

Table 3. Performance metrics for gaming data. We report BLEU-4 scores for action prediction in Minecraft (abbreviated as MC), and Bleeding Edge (abbreviated as BE). Cross-domain pre-training results in similar performance to domain-specific fine-tuning without pre-training. However, best performance is achieved with both.

4.4. Healthcare Experiments

For the experiments on the 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 4.1.

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 output the text tokens of the corresponding caption or answer and report the average perplexity across both settings. We frame RASS score prediction as a 10-way activ-

Task	Text instruction	Start frame	Predicted Action	Ground Truth Action
Minecraft	the player is using an iron_sword to attack and kill pigs in a forest...		[STARTACTION] [attack] [ENDOFACTION]	[STARTACTION] [attack] [ENDOFACTION]
Bleeding Edge	the player is controlling a red robot ... fighting other characters		[STARTACTION] [lockon][meleeattack] [lrot162] [lmag4] [ENDOFACTION]	[STARTACTION] [lockon][meleeattack] [lrot160] [lmag4] [ENDOFACTION]

Table 2. Examples of actions predicted by our fine-tuned models given text instructions and a corresponding video frame for Minecraft (above) and Bleeding Edge (below). More examples are presented in the Appendix.

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**

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 the model's text generation abilities in new domain settings.

ity classification problem, and train a separate classification head for our visual encoder. We use the video-level setting for our visual encoder with 9 frames as input, as described in the Appendix.

To evaluate the effectiveness of our pre-training framework, we compared the performance of our model against three baselines that also leverage CLIP and OPT for initialization. First, we compared against an *frozen* baseline that uses the same pre-trained models, kept frozen, while fine-tuning either a single linear layer for cross modal information passing (similar to the alignment stage of LLaVA [37]) or for linear probe classification. Second, we compared against a *unfrozen* 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 *from scratch* baseline that uses our same joint image-video encoder architecture 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 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 hyperparameter search.

5. Analysis and Discussion

On the importance of visual encoder training. We found that our pre-trained visual encoder outperforms CLIP on all action-oriented tasks, but performs worse for natural language generation. The key differences between our visual encoder and CLIP are: (1) the usage of sinusoidal positional embeddings, and (2) the joint MAE and next token prediction pre-training objectives that are propagated to the visual encoder. Our findings are consistent with Kim *et al*. [26], which also found that off-the-shelf visual encoders like CLIP provide poor features for robotic manipulation tasks and require gradient propagation to be effective and Xiao *et al*. [62], which found MAE to be an effective visual pre-training method for robotic control tasks.

Learning to see via interaction. When evaluated in a new domain, we found that our agent pre-training provided a powerful visual encoder that could effectively recognize actions occurring in videos even when used without the transformer decoder F_θ (Table 4). This is encouraging, and suggests that **generalist visual encoders can be learned by jointly observing and interacting with the environment**. This aligns with Gibson's ecological theory of perception [17], which posits that perception is inherently tied to action, and that interacting with the environment allows individuals to directly pick up more relevant and meaningful information, improving perceptual accuracy and efficacy.

When is agent pre-training most helpful? Our agent pre-training strategy improved performance for all action prediction and action recognition tasks when compared to training from scratch, as shown in Table 1, Table 3, and Table 4. However, agent pre-training was especially important in more complex environments like CALVIN, Minecraft, and Bleeding Edge, where relative performance improved by up to 215%, 56%, and 73% respectively. In contrast, the

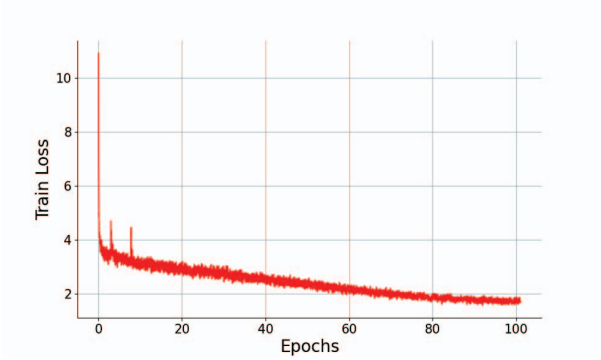

Figure 6. A plot of total pre-training loss over 100 epochs. Loss spikes are due to the MAE component of the loss. A figure showing each loss component separately can be found in the Appendix.

simpler Language Table setting only saw a relative performance improvement of 5%. We note that in all domains, best results were achieved by first pre-training across all environments and then fine-tuning in each environment separately, and that performance of our model when only pre-trained across all environments was often similar to that of our model when only fine-tuned on one environment (without pre-training), as shown in Table 3.

Pretraining Loss Stability. We find that although our combined loss is relatively stable during pre-training, there are two loss spikes, as shown in Figure 6. These spikes are caused by the MAE loss. Importantly, He *et al*. [21] observed that as model size increases, MAEs can become challenging to train due to loss instabilities. We believe that similar scaling issues could arise for frameworks such as ours that incorporate next token prediction and MAEs.

6. Related Work

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 [5]. Within the field of Natural Language Processing, this generally consists of larger proprietary LLMs [61] such as the GPT-series [8, 42], or smaller open-source models such as the LLaMA series [58], or instruction-tuned variants such as Alpaca [57] and Vicuna [68]. Within the field of computer vision, strategies such as masked auto-encoders [21] and contrastive learning [47] are two popular methods for self-supervised learning.

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 [2], the BLIP-series [11, 32, 34], and LLaVA [37]. 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 [3, 47, 56]. 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.

Agent-Based AI Recent research has focused on employing advanced large foundation models to create Agent-based AI systems, as shown in Durante *et al*. [15]. 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 [1, 6, 7, 18, 23, 33, 35, 43, 53, 59] and navigation [9, 12, 13, 16, 22, 36, 52, 69]. 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 [24, 40, 64], efficient data collection [14, 27], and the management of long-horizon steps [29, 44, 55, 60, 63]. Similarly to robotics, gaming agents require an understanding of visual scenes and textual instructions/feedback [19, 30, 46, 54]. 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 [28, 31] and knowledge retrieval [20, 45].

Vision-Language-Action Models More recently, a concurrent line of work has emerged that endows VLMs with action capabilities. These Vision Language Action Models (VLAs) use a similar setup to VLMs, and expand the language vocabulary to include discretized action tokens. Exemplar methods include RT-2 [7], OpenVLA [26], and 3D-VLA [66]. In comparison, our method not only uses discretized action tokens, but also introduces controller- and domain-specific state tokens, enabling multi-controller and multi-domain pre-training. Moreover, our method can exploit vision-only data for pre-training and can create adaptable pre-trained vision encoders by jointly observing and interacting with the environment.

7. Conclusion

We introduced a framework for learning Interactive Agent Foundation Models designed to take text, action, and/or visual inputs during pre-training. 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 perception, reasoning, and decision-making settings, facilitating the creation of generalist, multimodal agents.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 8

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 2, 8

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 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*, pages 1728–1738, 2021. 8

[4] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022. 4

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1, 8

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 6, 8

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 8

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 8

[9] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. *arXiv preprint arXiv:2309.10309*, 2023. 8

[10] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 6

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 8

[12] Vishnu Sashank Dorbala, Gunnar Sigurdsson, Robinson Piramuthu, Jesse Thomason, and Gaurav S Sukhatme. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022. 8

[13] Vishnu Sashank Dorbala, James F Mullen Jr, and Dinesh Manocha. Can an embodied agent find your" cat-shaped mug"? llm-based zero-shot object navigation. *arXiv preprint arXiv:2303.03480*, 2023. 8

[14] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 8

[15] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*, 2024. 8

[16] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. 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*, pages 23171–23181, 2023. 8

[17] James J Gibson. The ecological approach to visual perception. *Houghton Mifflin*, 1979. 7

[18] Ran Gong, Xiaofeng Gao, Qiaozi Gao, Suhaila Shakiah, Govind Thattai, and Gaurav S Sukhatme. Lemma: Learning language-conditioned multi-robot manipulation. *IEEE Robotics and Automation Letters*, 2023. 8

[19] Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023. 8

[20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020. 8

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *CVPR*, 2022. 8

[22] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 8

[23] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv*, 2022. 8

[24] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. *arXiv preprint arXiv:2310.18308*, 2023. 8

[25] Libby Keeley. Reducing the risk of ventilator-acquired pneumonia through head of bed elevation. *Nursing in critical care*, 12(6):287–294, 2007. 4

[26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Fos-

ter, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 7, 8

[27] K Niranjan Kumar, Irfan Essa, and Sehoon Ha. Words into action: Learning diverse humanoid robot behaviors using language guided iterative motion refinement. *arXiv preprint arXiv:2310.06226*, 2023. 8

[28] Peter Lee, Sebastien Bubeck, and Joseph Petro. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239, 2023. 8

[29] Boyi Li, Philipp Wu, Pieter Abbeel, and Jitendra Malik. Interactive task planning with language models. *arXiv preprint arXiv:2310.10645*, 2023. 8

[30] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021. 8

[31] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023. 8

[32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 1, 2, 8

[33] Jiachen Li, Qiaozi Gao, Michael Johnston, Xiaofeng Gao, Xuehai He, Suhaila Shakiah, Hangjie Shi, Reza Ghanadan, and William Yang Wang. Mastering robot manipulation with multimodal prompts through pretraining and multi-task fine-tuning. *arXiv preprint arXiv:2310.09676*, 2023. 8

[34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 8

[35] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 6, 8

[36] Xiwen Liang, Liang Ma, Shanshan Guo, Jianhua Han, Hang Xu, Shikui Ma, and Xiaodan Liang. Mo-vln: A multi-task benchmark for open-set zero-shot vision-and-language navigation. *arXiv preprint arXiv:2306.10322*, 2023. 8

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2, 7, 8

[38] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. 6

[39] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023. 4

[40] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023. 8

[41] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022. 4, 6

[42] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022. 8

[43] Georgios Pantazopoulos, Malvina Nikandrou, Amit Parekh, Bhathiya Hemanthage, Arash Eshghi, Ioannis Konstas, Verena Rieser, Oliver Lemon, and Alessandro Suglia. Multitask multimodal prompted training for interactive embodied task completion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 768–789, Singapore, 2023. Association for Computational Linguistics. 8

[44] Meenal Parakh, Alisha Fong, Anthony Simeonov, Abhishek Gupta, Tao Chen, and Pulkit Agrawal. Human-assisted continual robot learning with foundation models. *arXiv preprint arXiv:2309.14321*, 2023. 8

[45] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023. 1, 8

[46] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. 8

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 8

[48] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023. 1

[49] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022. 1

[50] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 6

[51] Curtis N Sessler, Mark S Gosnell, Mary Jo Grap, Gretchen M Brophy, Pam V O'Neal, Kimberly A Keane, Eljim P Tesoro, and RK Elswick. 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. 4

[52] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pages 492–504. PMLR, 2023. 8

[53] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023. 8

[54] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022. 8

[55] Jingkai Sun, Qiang Zhang, Yiqun Duan, Xiaoyang Jiang, Chong Cheng, and Renjing Xu. Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning. *arXiv preprint arXiv:2309.11359*, 2023. 8

[56] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 8

[57] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 8

[58] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 8

[59] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt models meet robotic applications: Co-speech gesturing chat system. *arXiv preprint arXiv:2306.01741*, 2023. 8

[60] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Chatgpt empowered long-step robot control in various environments: A case application. *IEEE Access*, 11:95060–95078, 2023. 8

[61] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 8

[62] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 7

[63] Mengdi Xu, Peide Huang, Wenhao Yu, Shiqi Liu, Xilun Zhang, Yaru Niu, Tingnan Zhang, Fei Xia, Jie Tan, and Ding Zhao. Creative robot tool use with large language models. *arXiv preprint arXiv:2310.13065*, 2023. 8

[64] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023. 8

[65] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 2

[66] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. In *Forty-first International Conference on Machine Learning*. 6, 8

[67] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 6

[68] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 8

[69] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023. 8