IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED DECEMBER, 2024
IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

MotIF: Motion Instruction Fine-tuning

Minyoung Hwang, Joey Hejna, Dorsa Sadigh, Yonatan Bisk

***Abstract*—While success in many robotics tasks can be determined by only observing the final state and how it differs from the initial state – e.g., if an apple is picked up – many tasks require observing the full motion of the robot to correctly determine success. For example, brushing hair requires repeated strokes that correspond to the contours and type of hair. Prior works often use off-the-shelf vision-language models (VLMs) as success detectors; however, when success depends on the full trajectory, VLMs struggle to make correct judgments for two reasons. First, modern VLMs often use single frames, and thus cannot capture changes over a full trajectory. Second, even if we provide state-of-the-art VLMs with an input of multiple frames, they still fail to correctly detect success due to a lack of robot data. Our key idea is to fine-tune VLMs using abstract representations that are able to capture trajectory-level information such as the path the robot takes by overlaying keypoint trajectories on the final image. We propose motion instruction fine-tuning (MotIF), a method that fine-tunes VLMs using the aforementioned abstract representations to semantically ground the robot's behavior in the environment. To benchmark and fine-tune VLMs for robotic motion understanding, we introduce the MotIF-1K dataset containing 653 human and 369 robot demonstrations across 13 task categories with motion descriptions. MotIF assesses the success of robot motion given task and motion instructions. Our model significantly outperforms state-of-the-art API-based single-frame VLMs and video LMs by at least twice in F1 score with high precision and recall, generalizing across unseen motions, tasks, and environments. Finally, we demonstrate practical applications of MotIF in ranking trajectories on how they align with task and motion descriptions. Dataset, code, and checkpoints are in https://motif-1k.github.io/**

***Index Terms*—Intention Recognition, Data Sets for Robot Learning, Semantic Scene Understanding**

I. Introduction

MEASURING success in robotics has focused primarily on *what* robots should do, not *how* they should do it. Concretely, *what* is determined by the *final state* of an object, robot, or end-effector [1], [2]. However, not all trajectories that achieve the same final state are *equally successful*. When transporting a fragile object, a path through safer terrain could be considered *more successful* than a shorter yet riskier route (Fig. 1a). Similarly, in the presence of humans a robot's actions when navigating, holding objects, or brushing human hair (Fig. 1 b-d) can cause surprise, discomfort, or pain, making such motions *less successful*.

Success detectors play an important role in robot learning since they evaluate whether or not a robot has completed a task. However, most overlook the importance of "*how*" the task is accomplished, focusing on the initial and final states of the trajectory [2], [3]. This simplification fails to account for tasks that fundamentally require evaluating the entire trajectory to assess success. As we incorporate robots into everyday scenarios, the manner in which they complete tasks will become increasingly important given the context of a scene and its semantic grounding (*e.g.*, avoid collision). Therefore, a more holistic approach to success detection is needed that considers both the task and how the agent should move to complete it.

While modern vision-language models (VLMs) have recently been used as promising tools for success detection [2], [3], they are unable to capture complex notions of how a task is completed for two reasons. First, the majority of VLMs are designed to reason over single images, while success detection in robotics is inherently sequential. Second, even models trained on multiple frames, like video LMs, struggle to recognize fine-grained motion due to a lack of training data. To bridge this gap, we explore how the choice of abstract motion representations, such as visualizing trajectories, affects the performance of both VLMs and video LMs. We propose a trajectory based visual motion representation which overlays a robot's past trajectory on the current or final frame, capturing both the path shape and its semantic connections to the environment. This approach leverages the world knowledge encoded in VLMs and refines it to assess robotic behaviors more effectively.

We propose **motion instruction fine-tuning**, a method that fine-tunes pre-trained VLMs to equip the capability to distinguish nuanced robotic motions with different shapes and semantic groundings. Using the aforementioned trajectory representation, we query our model to output a binary value indicating whether the motion is *correct* (1) or *incorrect* (0). To do so, we collect the **MotIF-1K** dataset, due to limited availability of robot data with diverse semantically grounded motions. We find that co-training mostly on human data with limited robot data enables transfer to robotic motion understanding effectively. **MotIF-1K** contains a variety of motions with 653 human and 369 robot demonstrations across 13 task categories, offering extensive coverage of both the *what* and the nuanced *how* of motion, complete with detailed annotations. It identifies common types of motions featuring varying degrees of semantic grounding, such as the robot's relationship with objects or humans in the environment. The dataset also captures diverse path shapes, in terms of directionality, concavity, and oscillation. For instance, paths in Fig. 1a differ in terms of semantic grounding, where it might be undesirable for a robot to pass over the grass. Fig. 1d describes how straight and curly hairs require different brushing techniques. Notably, MotIF-1K includes subtle motions that are often indistinguishable solely by their start and end states (Fig. 4).

MotIF, a motion discriminator developed by fine-tuning on MotIF-1K, shows further improved success detection on

Manuscript received: August, 26, 2024; Revised November, 22, 2024; Accepted December, 20, 2024. This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor Tamim Asfour and Reviewers' comments. This research was supported in part by Other Transaction award HR00112490375 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program.

Minyoung Hwang is with Massachusetts Institute of Technology, Cambridge, MA 02139 USA(myhwang@mit.edu). Joey Hejna is with Stanford University, Stanford, CA 94305 USA(jhejna@stanford.edu). Dorsa Sadigh is with Google Deepmind, Mountain View, CA 94043 USA, and Stanford University, Stanford, CA 94305 USA(dorsa@cs.stanford.edu). Yonatan Bisk is with Carnegie Mellon University, Pittsburgh, PA 15213 USA (ybisk@andrew.cmu.edu).
Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

(a) Outdoor Navigation (b) Indoor Navigation (c) Object-Interactive Manipulation (d) User-Interactive Manipulation

Fig. 1: Different robotic motions for various tasks. For each task, we visualize two different motions (path 1 and 2) from real robot demonstrations, where the trajectories share the same initial and final states. Most existing success detectors ignore intermediate states, thereby cannot distinguish them.

nuanced robot motions. We evaluate MotIF on the test split of MotIF-1K and demonstrate generalization to unseen motions, tasks, and environments. We significantly outperform state-of-the-art (SoTA) VLMs (*e.g.* GPT-4o, GPT-4V, and Gemini-1.5 Pro) with both single and multi-frame (video) input, with at least twice higher in both precision and F1, while maintaining high recall. Additionally, we demonstrate using MotIF on using the model output to rank real robot trajectories from a planner, outperforming SoTA VLMs by at least 20.6% higher win rate.

II. Related Work

With the recent development of large language models (LLMs) and VLMs, foundation models have been used to understand the environment and critique agent behaviors. Additionally, the increasing use of visual observations in robotics has brought attention to motion-centric visual representations.

Success Detection Using Foundation Models. [4], [5], [6], [7], [8] use LLMs to generate reward functions that evaluate agent behaviors. However, as LLMs cannot understand how a robot's actions are visually grounded in a scene, such methods require everything including the state of the environment to be translated into language. VLMs have been used to understand and evaluate agent behaviors given visual and language information. [9] use a VLM critic for general vision-language tasks, and closed API-based models have been used as behavior critics in robotics, even if inaccurate [10], [3]. Prior work [11], [10], [12], [13] have also used VLMs as zero-shot reward models for training downstream policies. [2] trains a VLM success detector for evaluating what was achieved from the robot, but does not consider *"how"* the agent solves the task. Other studies [14], [15], [16] train VLMs to be physically or spatially grounded, but focus on static environment understanding, not dynamic motions. [17] uses joint states to generate motion descriptions, but is restricted to short horizon, non-grounded motions from a set vocabulary (e.g., move arm up, rotate arm right). In contrast, we fine-tune VLMs to understand and evaluate grounded motions (e.g., make a detour to the left of the table) and consider more complicated, long-horizon, object or human interactive motions.

Motion-Centric Visual Representations. A growing interest in motion-centric visual representations in robotics has led to recent work in egocentric trajectory representation [18], [19] and point tracking using optical flow [20], [21] or learned predictors [22], [23], [24]. Despite the recent development of video language models (video LMs) [25], [26], [27], [28], [29], [30], they often struggle to capture fine-grained motion nuances and are computationally intensive. Prior works summarize trajectory or actionable choices in a single image frame by visualizing waypoints [18], keypoints in the environment [31], [32], or desirable paths [19], [33]; however, the focus of these works is mostly on conditioning the policy on these representations to improve policy learning rather than evaluation of nuanced motions and behaviors. We focus on using visual representations on top of exocentric views, which provide comprehensive environment context, to analyze motion understanding.

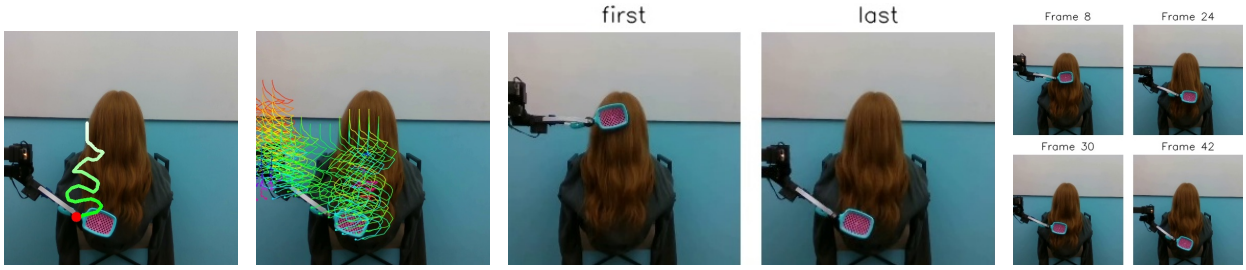
III. Motion Instruction Fine-tuning (MotIF)

In this section we first broaden the definition of success by including motion as a core component. We then discuss why pre-trained models are insufficient for motion-based success detection in robotics. Finally, we introduce MotIF for fine-tuning VLMs to be motion-aware success detectors.

A. Problem Statement

Success detection has been an integral part of recent robotics literature. Typically, success detection is defined as a binary function of the final state conditioned on the task, $y = f(o_T|\text{task}) \in \{0, 1\}$ [2], where y is the binary success label and o_T is the image observation of the agent and the environment at the final time step T. This restrictive assumption prevents success detectors from criticizing *how* a task is completed. For many tasks like collision-aware navigation, the final state does not provide sufficient information to capture the robot's interaction with objects or humans in the environment. Such tasks are often described by both their objective (i.e. bring me lemonade) and their execution (i.e. avoid going over the grass). In these scenarios, success cannot be determined by just the final state. Perhaps the simplest approach is modeling the entire trajectory, $y = f(o_1, \ldots, o_T|\text{objective}, \text{execution})$; however, this is computationally costly and often redundant. We instead propose using an abstract visualization of the trajectory $\tau = (o_1, \ldots, o_T)$. Notably, $I(\tau)$ outputs a 2D representation of the full trajectory overlaid on the last image o_T.

Suppose a task instruction T and motion description M corresponds to the robot's trajectory. Our goal is to assess success on robot motions given task specifications describing *how* the task should be done. Doing so requires assessing motion or path shape and, if the task requires object or human interaction, semantically grounding the robot's motion in the environment. Thus, we define the set of motions based on two criteria: *path shape* and *semantic grounding* (see Fig. 5). First, motions are distinguished based on properties of *path*



(a) Single Keypoint **(b)** Optical Flow **(c)** 2-frame Storyboard **(d)** 4-frame

Fig. 2: Visual Motion Representations. We explore three visual motion representations: (a) single keypoint tracking, (b) optical flow, and (c-d) multi-frame storyboard. For single keypoint tracking, temporal changes are shown with color gradient from white to green, ending with a red circle. For optical flow, we visualize the flow of all keypoints with rainbow colors. We sample N keyframes for N-frame storyboard.

shapes such as directions of translations, rotations, oscillations, repeated motions, and the convexity of curves. Second, *semantic grounding* in the environment involves understanding the context of the scene and the robot's interaction with the environment. For instance, moving over or making a detour towards an object implies the robot is aware of that instance. Similarly, we consider the relative distance and orientation of the agent with respect to the key objects in the scene (see examples in project page). In the following sections, we develop a VLM that acts as a success detector $y = f(o_1, \ldots, o_T | T, M)$.

B. Visual Motion Representations

Can foundation models be used for success detection? While VLMs have demonstrated a strong understanding of physical and causal commonsense reasoning [34], [35] and semantic grounding [15], [14], [36], they often work with static images as inputs, and cannot reason about sequential inputs necessary for dynamic tasks. Understanding motion requires not only isolating the most meaningful aspects of the scene but also identifying which changes that occurred due to the robot's motion are semantically relevant to the task. One naive solution is to pass multiple frames into the model as a storyboard, which can perform poorly due to lower resolution images (see Fig. 2 c-d and the example of GPT-4o in project page). On the other hand, video LMs can predict over multiple frames, but can still struggle to understand fine-grained details over time and require more compute than single-frame models, making them more difficult to use in real-time. Instead of using raw image frame(s) as input, we can look to prior work [19], [33] which show that VLMs can effectively leverage diagrams or abstract representations on top of image observations.

Representing a robot's motion in a single image. To improve motion recognition performance, we explore what representations effectively capture a robot's motion. To construct diagrams of robotic motions, we overlay a robot's trajectory on the image observation as shown in Fig. 2 a-b. One solution is to detect K keypoints $\{(x_0^1, y_0^1), \ldots, (x_0^K, y_0^K)\}$ in the initial frame I_0, and track the movement of each keypoint until the final frame I_T (see Fig. 2b). Here, x_t^k and y_t^k denote the x and y coordinates of the k^{th} keypoint in a 2D image observation at timestep t, respectively. The detected visual traces, i.e., optical flow, represent the apparent motions of the robot and how the environment changes accordingly. While this solution helps a single image representation contain the information of multiple keypoints, the full optical flow with trajectories of all keypoints may obscure a large portion of the background and important objects in the scene. Additionally, visualizing many keypoints often results in indistinguishable or overlapping trajectories, which may create visual clutter and reduce the clarity of the motion.

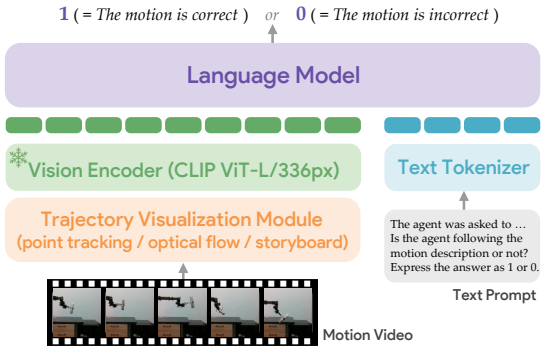


Fig. 3: Network Architecture. Given a visual motion representation of a robot's trajectory and its corresponding task and motion specifications, our model outputs a binary value indicating whether the motion is correct (1) or incorrect (0).

Therefore, the proposed method, MotIF, visualizes the trajectory of the most representative keypoint (see Fig. 2a), as a simplified yet more interpretable visual motion representation. We call the keypoint as point of interest, where a point of interest is typically chosen as a point on the end effector's surface by human annotators. Compared to prior work [19] that visualizes the center of mass of the end effector, our approach ensures that the selected keypoint is visually recognizable and not occluded. Details on labeling the point of interest and visualizing its trajectory are provided in Section IV.

C. Fine-Tuning VLMs

While we can directly pass a single image visualizing the robot's trajectory into a model zero-shot, non-fine-tuned models struggle to understand complex robotic motions. Section V empirically shows that existing state-of-the-art VLMs often infer undesirable motions as correct (false positives) or successful motions as incorrect (false negatives). Prior works [14], [15] have shown promising results fine-tuning VLMs to improve their grounding capabilities and understanding of a scene. Similarly, we also fine-tune VLMs with the different representations proposed earlier.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Model Architecture. Fig. 3 shows the network architecture of our model, which is based on LLaVA-1.5 7B [37]. Given a trajectory representation $I(\tau)$ as an image and a text prompt consisting of a task instruction T and motion description M followed by the question *"Is the agent following the motion description or not? Express the answer as 1 or 0."*, our model outputs a binary success prediction y. y indicates whether the motion is *correct* (1) or *incorrect* (0) with respect to the task and motion specification (T, M). During fine-tuning, only the language model is updated, while the visual encoder is frozen.

Constructing Training Data. Effective fine-tuning requires a well-structured dataset that includes both positive and negative samples. Positive samples consist of trajectories paired with their corresponding task instructions and motion descriptions. This encourages the model to associate visual motion representations with their correct task specification. For the i^{th} trajectory τ_i in the training set we generate a single image I_i representing the trajectory based on the chosen visual motion representation. To construct positive samples for training, we pair a set of H images with their corresponding task instructions and motion descriptions, forming the set $\mathbf{S}^{+}_{\text{train}} = \{(I_1, T_1, M_1), \ldots, (I_H, T_H, M_H)\}$, where T_i and M_i are the task instruction and motion description for image I_i, respectively. To construct negative samples for image I_i, we choose the N_{neg} least similar motion descriptions. The similarity between motion descriptions is calculated using SentenceTransformer [38] embeddings. The set of negative samples is then constructed as $\mathbf{S}^{-}_{\text{train}} = \bigcup_{i=1}^{H}\{(I_i, T_i, M^{-}_{i,1}), \ldots, (I_i, T_i, M^{-}_{i,N_{\text{neg}}})\}$, where $M^{-}_{i,j}$ is the j^{th} least similar motion description to M_i. We set $N_{\text{neg}} = 10$ in our experiments. The full training dataset D is $\mathbf{S}^{+}_{\text{train}} \cup \mathbf{S}^{-}_{\text{train}}$.

Co-training with Human and Robot Data. Based on prior work [37], fine-tuning VLMs requires a huge amount of training data. Ideally, our dataset D would cover all motions that a robot would execute across a wide variety of tasks. Unfortunately, robot demonstrations are time-consuming and often difficult to collect, which makes scaling the size of the dataset hard. Moreover, robots' physical constraints can prevent transfer or generalization to unseen robots with different dynamics. On the other hand, human demonstrations have a high degree of freedom and are often easier and more intuitive to collect, especially when looking for a diversity of motion. Thus, we opt to train on a mixture of human and robot demonstrations, D_h and D_r respectively. By doing so, we facilitate easy data collection while also ensuring the downstream VLM is more robust to embodiment and motion types.

IV. MotIF-1K Dataset

To benchmark and improve VLMs and Video LMs for motion understanding, we release the MotIF-1K dataset containing 653 human and 369 robot demonstrations across 13 tasks. As in Fig. 1 and Fig. 4, each task has demonstrations for 2 to 5 distinct motions which vary during the intermediate steps of a trajectory. For instance, a motion's path shape and semantic relationship with nearby objects may be different across demonstrations. This captures the diversity of *how* a task can be achieved, reflecting the nuanced and complex motions present in real-world scenarios. For instance, when shaking a boba drink, one person might use vigorous vertical movements, while another might use careful side-to-side movements to avoid spillage after inserting a straw. Moving a cup near a laptop via the shortest path is preferred if the cup is empty, but a detour is necessary if the cup contains water. Motion diversity is essential in grounded settings where motions need to adapt to varying environmental contexts. Fig. 4 shows examples of diverse motions. By collecting a diverse set of context-dependent motions with different intermediate trajectories, we ensure that our dataset challenges VLMs to consider the full trajectory for success detection.

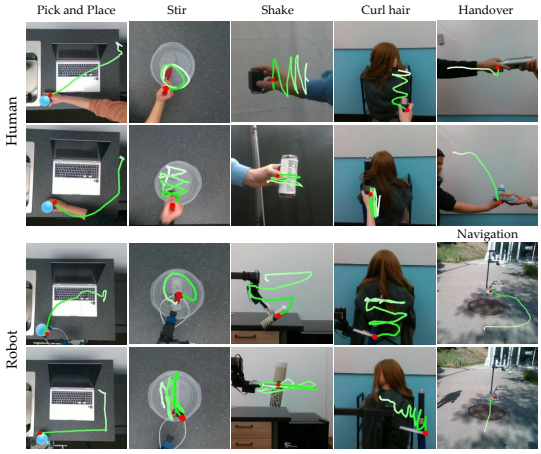


Fig. 4: Trajectory Visualizations. We visualize two different motions for solving the same task with the same embodiment.

A. Collecting Human and Robot Demonstrations

In this section, we explain how we collect human and robot demonstrations. Visualizations are in Fig. 4. Our human demonstrations are collected by six people to ensure ample variation in motion. For robot data, a single expert teleoperates a Stretch robot with a Meta Quest 3 VR controller for manipulation tasks and a gamepad for navigation tasks. We record the agent's joint states and image observations with a fixed exocentric RGBD camera for visual consistency. For the pick and place, stir, shake, brush hair, and tidy hair tasks we collect trajectories in two orthogonal camera views to support future 3D motion understanding using multiview images. In this paper, we treat observations from different camera viewpoints as separate trajectories and focus on effectively representing the agent's motion in each frame. Section IV-B explains how we annotate the trajectories with task instructions and fine-grained motion descriptions.

After demonstrations are collected, we preprocess the image observations using three different visual motion representation methods (see Fig. 2):

- **optical flow** [20], [21]: visualizing the trajectories of all visible keypoints with rainbow colors. For each keypoint, its trajectory is drawn with a single color.
- **single keypoint tracking (MotIF)**: visualizing a trajectory of a single keypoint. For single point tracking on human data, we use mediapipe [39] and track the center of the hand pose. For robot data, we annotate 2D keypoints to identify point of interests in the initial frame of each episode, which

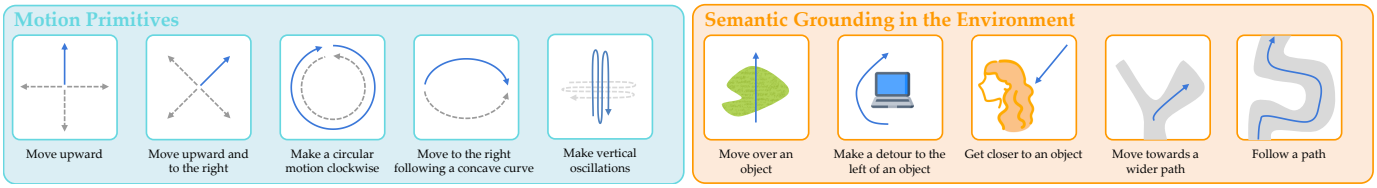


Fig. 5: Motion Diversity. 10 canonical motions are described with blue arrows and their corresponding descriptions. In the blue box, motion primitives are categorized based on the shape of the ideal path. Gray dashed arrows denote variants of the blue arrow, lying in the same category. The orange box shows motions that involve grounding in the environment, where the relationship between the robot and an instance in the environment is considered.

is either a keypoint on the manipulated object or the robot's end effector. Then, we choose the keypoint nearest to the point of interest. For both human and robot data, temporal changes are shown with color gradient from white to green, ending with a red circle.

- N**-frame storyboard** ($N = 2, 4, 9$): sampling N keyframes and stacking those frames into a single image. We use K-means clustering on the image embeddings of all frames to sample keyframes that are sufficiently different in latent space. Frame indices are annotated above each frame image.

B. Grounded Motion Annotations

In this section, we use "agent" to refer to either human or robot demonstrating the task. We use human annotators to label the motion in each video. While automated motion labeling using proprioception has previously been used, it is only applicable to short-horizon motions (<10 frames), whereas our dataset contains long-horizon motions (>300frames). Compared to previous datasets which do not capture any information about *how the motions are grounded in the environment or to the user*, we consider motion diversity in two different axes: (1) path shape, and (2) semantic grounding in the environment.

Path Shape. As illustrated in Fig. 5, we first set a vocabulary of motion primitives in terms of path shape; direction and convexity of translation (e.g., move upward, follow a convex curve), direction of rotation (e.g., make a circular motion clockwise), and oscillatory movement (e.g., move up and down). Motion often consists of multiple motion primitives with different path shapes, such as moving right then downward, or moving right while oscillating vertically. When annotating such motions in language, we first list the primitives in the temporal order and prioritize the dominant ones among those that happen simultaneously.

References to Objects and Humans. Another important aspect of grounded motions are references to objects or humans in the scene. The orange box in Fig. 5 illustrates five common examples of motions in terms of semantic grounding. These motions gain specific meaning in relation to objects or humans in the environment. Based on the task specification regarding an object (e.g., avoid damaging the laptop) or a user (e.g., focus brushing the bottom part of the hair that is tangled), an agent's motion can be distinguished accordingly. For instance, a motion of shaking salt over food is defined not only by the path shape of the shaking movement but also by the spatial relationship between the agent and the food. We use a set vocabulary (e.g., move over, make a detour, get closer/farther, follow path) to annotate the grounded motions in language.

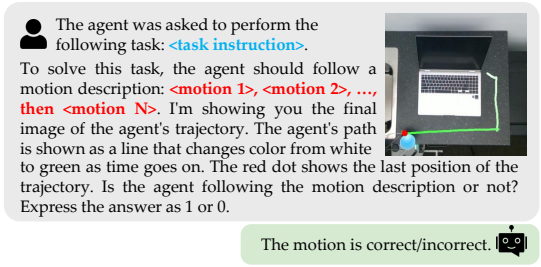


Fig. 6: Text Prompt Template shows how a motion composed of multiple primitives is described in the text input of a VLM.

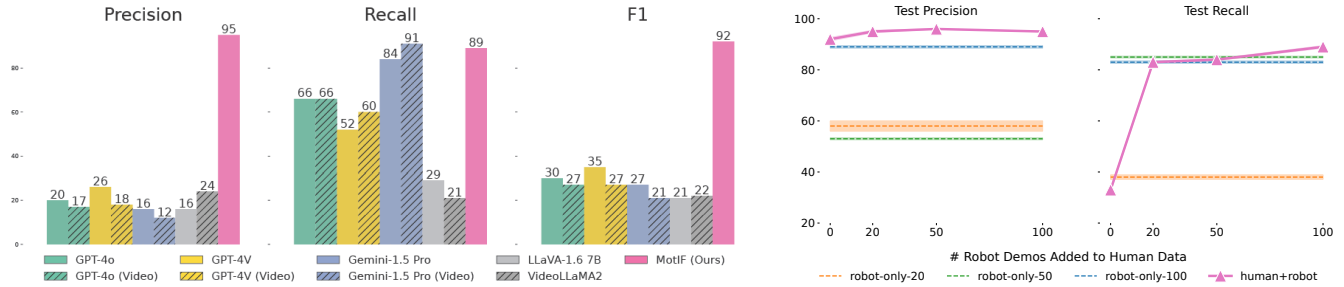
The final motion descriptions in MotIF-1K are constructed as combinations of descriptions in terms of path shape and references to objects and humans. With the annotated task instructions and motion descriptions, we construct text prompts that could be used to describe the agent's motion. Using this text prompt and the image representation of the trajectory as inputs, we train VLMs to predict the alignment between the text and image (see Section III-C). For the example in Fig. 6, the task instruction is "pick up the cup and place it to the lower left of the laptop" and the motion description could be "move downward and farther from the laptop, then move to the left". Here, two motion primitives, "move downward" and "move to the left", are combined with a grounded motion annotation,"move farther from the laptop". Given the text prompt and trajectory representation $I(\tau)$ in Fig. 6, our fine-tuned VLM outputs a binary value indicating whether the motion is correct or not.

V. Experiments

In this section we seek to answer the following questions: 1) How does MotIF compare to start-of-the-art models? 2) How important is robot data in understanding motion? and finally 3) What is the effect of visual motion representation? We compare our approach with state-of-the-art models and assess the benefits of co-training on human and robot data. We investigate the impact of different visual motion representations. All models are evaluated on the test split of MotIF-1K. See the project page for results in the validation split, and visualizations of trajectories with MotIF outputs.

Baselines. We evaluate against GPT-4o, GPT-4V [25], and Gemini-1.5 Pro [26] as state-of-the-art API-based baselines. We also compare to the best performing pre-trained open LLaVA [37] models with various sizes (7B, 13B, 34B). To evaluate different visual motion representations, we compare our proposed single point tracking to full optical flow, N-frame storyboard ($N = 2, 4, 9$), and a single-frame image. All learned

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Comparison with SoTA off-the-shelf VLMs and Video LMs

(b) Co-training on Human and Robot Data

Fig. 7: Performance on MotIF-1K Test Split. (a) shows that our models outperforms state-of-the-art (SoTA) off-the-shelf models in the test split. For reference, we also compare against the best-performing open-source baselines (LLaVA-1.6 7B for single-frame and VideoLLaMA2 for video LM) based on their F1 scores. (b) Performance improves with more robot demonstrations. Dashed lines indicate performance with robot-only data. Performance of the model trained only with human data is shown as human+robot with #robot=0 (the leftmost pink triangle). For simplicity, all metrics are in percentage. Error bars show standard deviation across five random seeds (12109, 20709, 42266, 73487, 84501).

single-frame baselines are fine-tuned from the LLaVA-1.5 7B model [37], [40], which was pre-trained on the Vicuna visual question answering (VQA) dataset [41]. For video LM baselines, we choose to fine-tune VideoLLaMA2 7B [27] since its pretrained model shows the highest F1 among multiple video LMs [27], [29], [30], [28] on the MotIF-1K test split.

Training Details. We train on all human and 100 robot demonstrations from MotIF-1K. For each demonstration, we construct one positive and 10 negative samples of image-text pairs. When fine-tuning the VLMs, we freeze the weights of the pre-trained visual encoder, CLIP ViT-L/14 [42] with an input size of 336px by 336px. We fine-tune the projection layer and the language model in the VLM using low-rank adaptation (LoRA) with cross-entropy loss for 30 epochs with a learning rate of $5e^{-5}$ and a batch size of 32. Video LM baselines are also fine-tuned using LoRA, with a learning rate of $2e^{-5}$ and a batch size of 16. We use a single A100 GPU for fine-tuning.

Evaluation Set and Metrics. All learned models output a binary label indicating whether or not the agent's motion in the image(s) align with the given task and motion descriptions. For off-the-shelf models, we convert language responses into their corresponding labels. We evaluate models on the validation and test split of MotIF-1K containing 129 and 134 robot demonstrations, respectively. The test split contains a set of unseen trajectories which vary in camera viewpoints, motions, tasks, and environment in comparison to the training and validation split. Similar to the training data, we construct one positive and 10 negative samples per demonstration. For each experiment, we evaluate the performance of models using precision $(= \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}))$, recall $(= \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}))$[1], and F1 $= 2/(1/\mathrm{precision} + 1/\mathrm{recall})$. These metrics show the reliability and robustness of model outputs. High precision minimizes false positives, and high recall ensures most valid motions are identified. For all evaluations, we measure standard deviation on performance with five random seeds with temperature 0.5.

How does MotIF compare to state-of-the-art VLMs? Our model is developed by fine-tuning a 7B LLaVA model with the MotIF-1K dataset. While LLaVA is pre-trained on general VQA tasks answering questions given a static image, we fine-tune the model using our proposed trajectory motion representation. As shown in Fig. 7a, MotIF outperforms GPT-4o, GPT-4V, Gemini-1.5 Pro, and LLaVA in all metrics in the test split, by at least 162.9% in F1 and 265.4% in precision. Even comparing with video LMs (GPT-4o, GPT-4V, Gemini-1.5 Pro, and VideoLLaMA2) with 8 uniformly sampled frames as input, MotIF shows significantly higher precision and F1 while achieving 89% recall. We include qualitative results in the project page that show our model robustly works in data out of training domain, such as unseen tasks and environments.

How does robot data impact performance? Fig. 7b shows the positive transfer from human to robot data, by co-training on full human data with 653 trajectories and an increasing number of robot demonstrations. Simply adding 20 robot demos improved performance significantly in recall by 151.5% and slightly in precision by 3.3%. With 50 and 100 robot demos, performance improves in all evaluation metrics. Interestingly, co-training on human data and 20 robot demos outperforms training solely on 20 robot demos by 63.8% and training on 100 robot demos by 6.7%. This implies that human data can be used to learn representations of grounded motions which can transfer to robots, despite the large embodiment gap. However, some robot data is still necessary; training only on human data shows very poor performance getting 33% recall which is worse than random guess. This might be because fine-tuning exclusively on human data specializes the model towards the human domain, which may cause performance decrease on robot data. Adding even a small amount of robot data significantly improves performance by encouraging the model to learn more generalized, embodiment-agnostic representations, demonstrating the crucial role of robot data in achieving robust performance.

How does motion representation affect performance? Table I shows the performance in MotIF-1K test split and compares different visual motion representations trained on all human data and 100 robot trajectories used in the co-training experiment. Among the four motion representation methods for a single-frame VLM, MotIF (Last Frame + Traj) shows the highest F1 of 92%. While all methods show similarly high precision scores, MotIF achieves 89% recall, which is significantly higher than the single-frame baselines. Using the last frame without any motion representation obtains the lowest recall of 60%. Using full optical flow shows poor

[1]TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative

Model	Motion Representations	Precision	Recall	F1 Score
Single Frame VLMs	-	**97** ± 0.26	60 ± 0.87	74 ± 0.74
	Optical Flow	**97** ± 0.45	68 ± 0.27	80 ± 0.33
	Storyboard ($N = 4$)	94 ± 0.20	76 ± 0.73	84 ± 0.46
	Last Frame + Traj (Ours)	95 ± 0.14	**89** ± 0.52	**92** ± 0.30
Video LMs	8 Frames [43]	88 ± 0.80	81 ± 0.28	85 ± 0.45
	8 Frames + Traj [43]	91 ± 0.52	**89** ± 0.28	90 ± 0.25

TABLE I: Motion Representation Comparison on MotIF-1K Test Split. Among all methods, single-frame with trajectory drawing demonstrates the highest F1 score. Our approach identifies valid motions effectively and generalizes better than baselines. Among the three storyboard representations ($N = 2, 4, 9$), we report the performance of the model that performs the best in the test split ($N = 4$). Video LM baselines use 8 uniformly sampled frames per trajectory. The bottom row shows the performance of using video with trajectory overlaid.

generalization performance in the test split, performing worse than MotIF. Perhaps the simplest baseline, inputting storyboard that aggregates multiple frames in a single image, also fails to outperform MotIF. Performance degradation with multi-frame and optical flow is likely due to reduced image quality from either reduced image resolution or visual clutter from optical flow that covers relevant information. Fine-tuning an 8-frame video LM [27] improves recall and F1 over the single-frame VLM without any motion representation. Still, the video LM's performance is worse than ours; single-frame VLM with trajectory visualization. Once we add trajectory visualization on the video frames, the video LM's performance improves, reaching near parity with MotIF. Results on both single-frame VLMs and video LMs demonstrate the effectiveness of our motion representation. However, training and running inference on a video LM is much more expensive than on a single-frame VLM, implying that our model is not only the best performing, but also more cost-effective than video LMs.

VI. Discussion

Summary. Task specification in robotics often goes beyond simply stating *what* the objective is, and additionally consists of *how* a task should be done. As a step in this direction, we introduce a dataset, MotIF-1K, alongside a unique representation method and training technique, MotIF. Our findings demonstrate that our system can effectively provide assessments of nuanced robotic actions. The MotIF-1K dataset consisting of human and robot trajectories, captures the diverse ways in which tasks can be executed. We propose MotIF that uses this data to fine-tune open-source VLMs to detect a more nuanced notion of success. Our results demonstrate that MotIF can effectively assess success over these nuanced motions by leveraging a simple visual motion representation; overlaying the robot's trajectory on the image observation, outperforming all closed and open sourced VLMs and video LMs.

Failure Cases. MotIF works best when the instructions are similar to those in the training dataset. Paraphrased instructions, especially metaphorical or evocative expressions such as "gently move, drawing an elegant curve" instead of verbose and precise expressions such as "move to the right, following a concave curve" have induced ambiguity in the model, often leading to inaccurate success detection. We expect more advanced pretrained language models could handle this in the future.

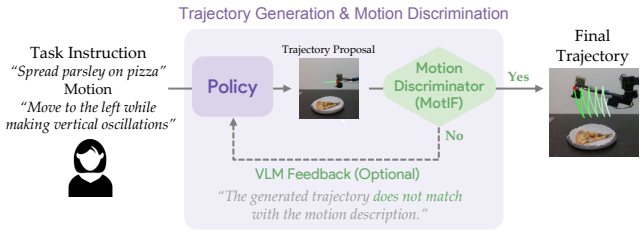
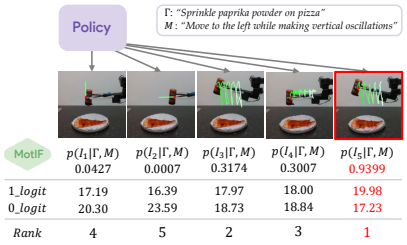


Fig. 8: Refining and Terminating Robot Planning. MotIF can close the loop of any existing open-loop controlled system by determining success in executing proper motion and giving this as a feedback to the system. We use an LLM as the policy that generates the sequence of the robot joint states.



(a) Ranking Example

Model	Win Rate
GPT-4V	0.57
GPT-4o	0.65
Gemini	0.68
MotIF	**0.82**

(b) Win Rate

Fig. 9: Ranking Trajectories. We can use MotIF to rank trajectories. (a) $p(I_k|\Gamma, M)$ denotes how likely the motion in the k^{th} image corresponds to the given task instruction Γ and motion description M. (b) Win rate evaluates each model by measuring the prediction accuracy of pairwise rankings.

Applications of MotIF. MotIF can determine the success of trajectories generated by any policy evaluating if the motions align with the task instruction and motion description. This can provide a signal for when to terminate an episode or to further refine the policy. We include examples of the following uses of MotIF on LLM generated policies [44]: MotIF as a success detector (see App. Fig. 10 in project page), terminating or adapting robot policies using MotIF output as feedback (Fig. 8), and ranking trajectories (Fig. 9a). For refining and terminating robot planning, we implement code-as-policies [44] on pizza condiment spreading tasks using a real Stretch RE2 robot (see text prompts and qualitative results in project page). For quantitative analysis on ranking trajectories, we compare single-frame VLMs for inferring ranks on 45 pairs of trajectories. As shown in Fig. 9b, we measure win rate, the ratio of correctly ranked pairs among all pairs of trajectories. The table shows that MotIF infers the ranks most effectively among all methods.

Ablation Study. We conduct an ablation study by fine-tuning the vision encoder alongside the language model with a learning rate of $5e^{-5}$. This results in slightly worse performance (0.94 ± 0.0038 precision, 0.86 ± 0.0048 recall, and 0.90 ± 0.0040 F1) compared to our method, supporting the decision to keep the vision encoder frozen during fine-tuning.

Limitations and Future Directions. One limitation of MotIF is the dependency on 2D visual motion representations,

which might not capture all aspects of complex 3D motions. Future work could incorporate RGB-D and multi-view images, or 3D visual representations overlaid on the image for a more comprehensive understanding of 3D space using our dataset. Another direction is to use MotIF as a reward signal for reinforcement learning. MotIF can act as a motion discriminator when training RL policies that can potentially more accurately reflect user preferences and contextual appropriateness. Future work might create a VLM that outputs natural language responses, which could be used to not only evaluate binary success but also perform reasoning on robotic motions.

REFERENCES

[1] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv:2306.11565*, 2023.

[2] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. *arXiv:2303.07280*, 2023.

[3] Lin Guan, Yifan Zhou, Denis Liu, Yantian Zha, Heni Ben Amor, and Subbarao Kambhampati. " task success" is not enough: Investigating the use of video-language models as behavior critics for catching undesirable agent behaviors. *arXiv:2402.04210*, 2024.

[4] 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:2310.12931*, 2023.

[5] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. *arXiv:2303.00001*, 2023.

[6] Hengyuan Hu and Dorsa Sadigh. Language instructed reinforcement learning for human-ai coordination. In *ICML*, 2023.

[7] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. *Arxiv preprint arXiv:2306.08647*, 2023.

[8] Minyoung Hwang, Luca Weihs, Chanwoo Park, Kimin Lee, Aniruddha Kembhavi, and Kiana Ehsani. Promptable behaviors: Personalizing multi-objective rewards from human preferences. *arXiv:2312.09337*, 2023.

[9] Seongyun Lee, Seungone Kim, Sue Hyun Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. *arXiv:2401.06591*, 2024.

[10] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv:2402.03681*, 2024.

[11] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *NeurIPS*, 2023.

[12] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv:2310.12921*, 2023.

[13] Kate Baumli, Satinder Baveja, Feryal Behbahani, Harris Chan, Gheorghe Comanici, Sebastian Flennerhag, Maxime Gazeau, Kristian Holsheimer, Dan Horgan, Michael Laskin, et al. Vision-language models as a source of rewards. *arXiv:2312.09187*, 2023.

[14] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *arXiv:2309.02561*, 2023.

[15] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv:2401.12168*, 2024.

[16] Zawalski Michał, Chen William, Pertsch Karl, Mees Oier, Finn Chelsea, and Levine Sergey. Robotic control via embodied chain-of-thought reasoning. *arXiv:2407.08693*, 2024.

[17] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv*, 2024.

[18] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *CVPR*, 2021.

[19] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv:2311.01977*, 2023.

[20] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *NeurIPS*, 2022.

[21] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *CVPR*, 2023.

[22] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv:2405.01527*, 2024.

[23] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. *arXiv:2308.15975*, 2023.

[24] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv:2401.00025*, 2023.

[25] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

[26] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[27] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv:2406.07476*, 2024.

[28] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023.

[29] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv:2311.08046*, 2023.

[30] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024.

[31] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv:2406.10721*, 2024.

[32] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv:2403.03174*, 2024.

[33] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv:2402.07872*, 2024.

[34] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[35] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

[36] Minae Kwon, Hengyuan Hu, Vivek Myers, Siddharth Karamcheti, Anca Dragan, and Dorsa Sadigh. Toward grounded social reasoning. *arXiv:2306.08651*, 2023.

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.

[38] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on EMNLP*, 2019.

[39] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv:1906.08172*, 2019.

[40] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023.

[41] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *https://vicuna.lmsys.org*, 2023.

[42] 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 *ICML*, 2021.

[43] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

[44] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *ICRA*, 2023.