Can Foundation Models Perform Zero-Shot Task Specification For Robot Manipulation?

Abstract

Task specification is at the core of programming autonomous robots. A low-effort modality for task specification is critical for engagement of non-expert end-users and ultimate adoption of personalized robot agents. A widely studied approach to task specification is through goals, using either compact state vectors or goal images from the same robot scene. The former is hard to interpret for non-experts and necessitates detailed state estimation and scene understanding. The latter requires the generation of desired goal image, which often requires a human to complete the task, defeating the purpose of having autonomous robots. In this work, we explore alternate and more general forms of goal specification that are expected to be easier for humans to specify and use such as images obtained from the internet, hand sketches that provide a visual description of the desired task, or simple language descriptions. As a preliminary step towards this, we investigate the capabilities of large scale pre-trained models (foundation models) for zero-shot goal specification, and find promising results in a collection of simulated robot manipulation tasks and real-world datasets.

Keywords: Goal-conditioned RL, Visual RL, Robot Learning

1. Introduction

Robots are gradually entering our homes to help automate aspects of our everyday life. End users of such technologies are likely to have personalized requirements and preferences. A low-effort and intuitive modality of communication is needed to allow non-expert users to program robots and customize them to user needs. This requires robots to be equipped with a form of "common sense" that is grounded in human-centric experiences and understanding of the world. Recent advances in computer vision (Krizhevsky et al., 2012) and natural language processing (NLP) (Devlin et al., 2019; Radford et al., 2019) have enabled machine learning models to make sense of images and text using large-scale internet datasets, that comprise in part, data generated by humans in human-centric environments. Of particular interest are "foundation models" (Bommasani et al., 2021) – deep neural network models trained on massive internet datasets – that have powered impressive advances in downstream vision and NLP tasks. In this backdrop, our work explores if such foundation models can also enable task specification for robotics and advance embodied intelligence.

Traditional approaches to program robots, both at training and deployment times, involve the use of reward functions, task demonstrations, or goal specification. Among these, goal specification is particularly appealing from an end user viewpoint since goals are easier to provide compared to full demonstrations and also require minimal expertise compared to scripting detailed reward functions that are prone to exploitation (Amodei et al., 2016). Existing goal-conditioned approaches use either compact state space goal vectors or RGB images from the same robot scene. The former is often hard to interpret and requires detailed state estimation, which is difficult outside lab settings. The latter forces a human expert to perform the task to generate the necessary goal image, thereby

© 2022 .

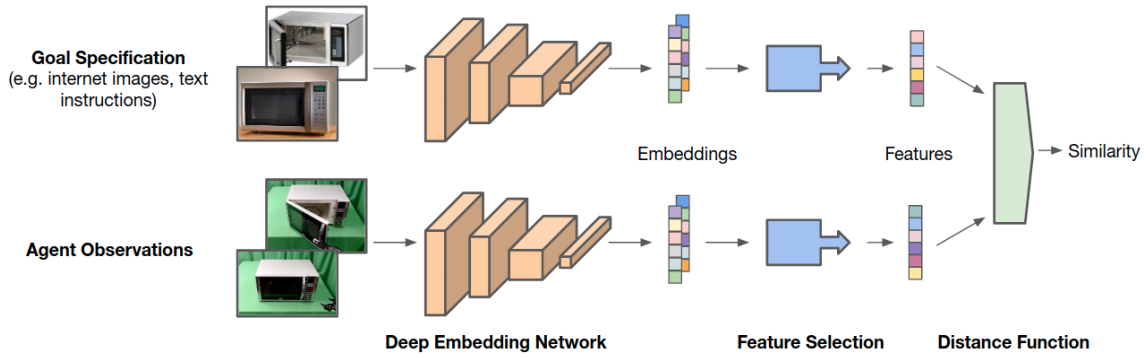

Figure 1: Overview of our proposed **Ze**ro **S**hot **T**ask-specification (ZeST) framework. The observation and goal specifications are first embedded (e.g. using a foundation model). We compute features of these embedding, and subsequently measure the similarity of these features. A high similarity between agent observation and goal implies the specification has been achieved.

defeating the purpose of an autonomous robot. We study two alternate forms of goal conditioning to overcome the aforementioned difficulties.

The first approach involves goals or instructions in natural language, such as *"open the microwave"*. To be successful, the robot must understand grounding and semantics – it must be capable of understanding what a microwave looks like and also be capable of differentiating between open and closed microwaves. The second approach involves using goal images from a different scene, such as images from the internet. For example, to instruct a robot to open a microwave, we can provide the robot an image of an open microwave obtained from the internet. To be successful, the robot must have domain adaptation capabilities and form correspondences between the state of the microwave in the goal image and current scene. The goal of our work is to study if foundation models are useful for such types of task specification, particularly in the zero-shot regime where they have been very successful in traditional vision and NLP tasks.

Our Contributions: In this work, (1) We introduce a *framework* for studying foundation models for zero-shot task specification (ZeST). See Figure 1 for an overview. (2) We evaluate the effectiveness of ZeST for enabling zero-shot *policy execution* through a set of goal selection tasks; and (3) We evaluate ZeST for *policy learning* in offline reinforcement learning. We find that ZeST is quite effective in zero-shot goal-selection and provides a 14-fold increase in performance over a random guessing baseline. In offline RL, we find that using ZeST scores as a proxy for the reward function enables the learning of policies that perform better than a behavior cloning baseline.

2. Background and Related Work

We consider environments that take the form of a **High-Dimensional Markov Decision Process (MDPs)**, described by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{X}, \mathcal{A}, P, R, d_0, \gamma \rangle$. Informally, this setting has been widely studied and was characterized more formally by Du et al. (2019) as "Block MDPs". Here, $\mathcal{S}$ denotes a compact state space that is not directly observable by the agent, $\mathcal{X}$ denotes the high-dimensional observation space that contains sufficient information to uniquely recover the underlying state, and $\mathcal{A}$ is the action space. For example, the observation space can be (multi-view) camera images while the state space can be object poses. The underlying transition dynamics and reward function are described in the state space as $P(s'|s, a)$ and $R(s)$. Additionally, d_0 is the starting state distribution and $\gamma \in [0, 1)$ is the discount factor. Since the observation can be mapped uniquely

to the state, an equivalent MDP can be constructed in the observation space as well. A trajectory $\tau = \{(X_0, a_0), (X_1, a_1), ..., (X_H, a_H)\}$ is a sequence of observation-action pairs of length H. A policy, $\pi(a_t|X_t)$ maps from observations to a probability distribution over actions. The objective is to learn a policy that maximizes the long term reward, i.e. $\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)\right]$ where the expectation is under $s_0 \sim d_0$, $a_t \sim \pi(\cdot|X_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$.

Goal-Conditioned Policies In the standard RL formulation, an agent learns a policy for a single task (reward function). Goal-conditioning allows the agent to perform multiple tasks by conditioning on different goals from a goal space $G \in \mathcal{G}$. The policy $\pi(a_t|X_t, G)$ and reward function $R(s_t, G)$ can be made a function of the goal, thereby enabling multi-task learning. We can interpret goal-conditioning as constructing a broader MDP with an augmented state and observation spaces $\mathcal{S} \cup \mathcal{G}$ and $\mathcal{X} \cup \mathcal{G}$. The idea of goal-conditioning has been extensively studied. Prior work has considered goals either in the compact state space (Kaelbling, 1993; Andrychowicz et al., 2017; Fu et al., 2018; Gupta et al., 2019), or in high-dimensional image space, but from the same scene as the agent (Nair et al., 2018; Nair and Finn, 2019; Singh et al., 2019; Johns, 2021). Compact state vectors are hard to interpret, depend on the scene, and require detailed state estimation. Image based goals are human-interpretable, but requires another agent (typically a human) to perform the task first to generate the image for conditioning. In this work, we propose the use of off-domain images and/or language instructions to reduce the task specification burden on users. Our method can also be adapted to work with existing goal-conditioned learning algorithms.

Foundation Models and Applications A major advancement in modern deep learning is the emergence of representation and transfer learning. Models trained on generic datasets from the internet are capable of learning representations that transfer successfully to a plethora of downstream applications. This observation has been widespread since the seminal works of AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016). Recent advances in self-supervised learning have led to further advancements in visual (He et al., 2020), language (Devlin et al., 2019; Radford et al., 2019), and multi-modal (Radford et al., 2021) representation models that have widely impacted downstream applications in vision (Tian et al., 2020; Conde and Turgutlu, 2021; Patashnik et al., 2021; Frans et al., 2021) and NLP (Lin et al., 2021; Bugliarello et al., 2021; Izacard and Grave, 2020), and for this reason have been touted as "foundation models" (Bommasani et al., 2021). However, the use of foundation models for control and embodied intelligence is relatively new and under-explored. CLIPort (Shridhar et al., 2021) combines CLIP embedding with transporter network to learn language-conditioned robot manipulation policies from demonstrations. The concurrent works of Khandelwal et al. (2021) and Parisi et al. (2022), study the use CLIP and other self-supervised representation networks as a perception module for control tasks and observe they outperform traditional ImageNet-pretrained backbones.

3. ZeST: Using Foundation Models for Task Specification

As outlined in Section 2, current approaches to goal-conditioning are limited due to requirement of state-vector based goals or goal images from the same scene as agent. In this work, we introduce ZeST, a framework for studying more generic and intuitive forms of goal-specification like off-domain images (e.g. those found on the internet) and language instructions. To succeed with this type of goal specification, the agent must capture "common sense" that is grounded in human-centric experiences and understanding of the world. Motivated by the advancement in foundation

models (Bommasani et al., 2021) that were trained using massive amounts of data from the internet, which contain in part data generated by humans in human-centric environments, we specifically study if foundation models can enable these generic forms of goal-conditioning. In this work, we also focus on **zero-shot** goal-specification, where the agent must be capable of performing tasks when simply presented with a user-specified (off-domain) goal. We do not assume any human annotations or supervised pairings between prior experiences of the agent and goal specifications, which is representative of real deployment scenarios. Moreover, foundation models have demonstrated impressive zero-shot learning results (Radford et al., 2021; Brown et al., 2020).

At its core, the ZeST framework provides a way to measure the "similarity" between the image observation of the agent and a user-specified goal, with high similarity implying the goal specification is satisfied. The most similar image in the replay buffer of the agent can be used for goal-conditioning. The similarity can also be used as a proxy for reward signal in case of policy learning. While the approach of measuring similarity between agent's observation and goal has been explored in prior work, for instance by using a learned classifier (Pinto and Gupta, 2016; Fu et al., 2018; Eysenbach et al., 2021), they have all required goal images from the same scene as the agent. The overall ZeST framework is described in Figure 1, and consists of three main components: (1) embedding network; (2) feature selector; (3) feature distance function. Each of the three modules can be instantiated according to properties and constraints of the downstream task.

For the **embedding network**, we use pre-trained foundation models for embedding both the goal specification and agent observation. While any foundation model can be used, in our experimental evaluation, we focus on ImageNet-supervised ResNet50 (He et al., 2016), ImageNet-trained MoCo (He et al., 2020), and CLIP (Radford et al., 2021). Since CLIP is a multi-modal embedding network, it also provides a mechanism for specifying tasks through language descriptions. Details of the three candidate embedding networks can be found in the Appendix. The **feature selection** module further featurizes the embedding. In this work, we consider two simple featurizations: 1) raw features that use the raw embeddings themselves, and 2) delta features that use the difference in embeddings between the desired and initial configurations, and analogously for the agent, the difference in embeddings between current observation and initial observations. The raw features are useful to directly compare the specified goal with the current observation. With delta features,

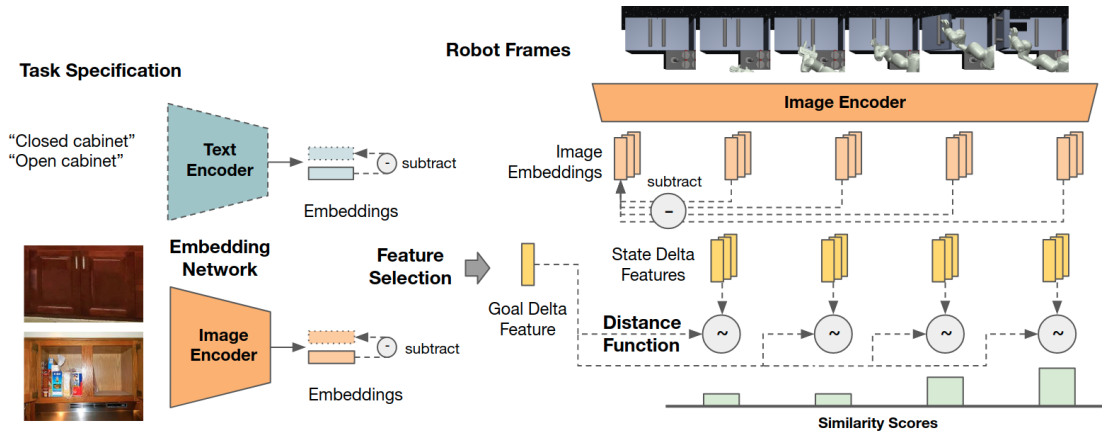

Figure 2: An instantiation of the ZeST framework with delta features. We observe that similarity of the observation (top row robot frames) with the task specification (open cabinet) increases as the robot executes a successful trajectory.

the user could specify both the goal state and an initial/current state in order to indicate a desired change of an object's state. For example, a user would provide the images of both a closed microwave and open microwave, which serves to better specify the task by describing the high-level action of opening the microwave. For the **similarity metric**, we explore the use of cosine similarity and L2 distance, both of which have been used in prior work involving representation learning. In summary, the similarity between a pair of observations (X_t, X_0) and goal-specification (G_0, G_f) is written as:

$$\boldsymbol{\phi}^{\text{raw}}\left(X_t, G_f\right) := \boldsymbol{\alpha}\left(\psi(X_t), \psi(G_f)\right), \tag{1}$$

$$\boldsymbol{\phi}^{\text{delta}}\left((X_t, X_0), (G_f, G_0)\right) := \boldsymbol{\alpha}\left((\psi(X_t) - \psi(X_0)), (\psi(G_f) - \psi(G_0))\right) \tag{2}$$

where $\psi(\cdot)$ is the embedding function (foundation model) and $\boldsymbol{\alpha}$ is the similarity metric (cosine similarity, L2 distance etc.). Figure 2 illustrates an instantiation of ZeST with delta features.

We emphasize here that ZeST provides an abstraction to study generic forms of goal conditioning, and the use of various foundation models for the same. While we explore a number of design choices as outlined above and in the Appendix, there are many additional possibilities that all conform to our broad framework. An exhaustive study of all such combinations is beyond the scope of any single work, and we hope that our work inspires further work in the study of goal-conditioning and foundation models for embodied intelligence.

4. Experimental Tasks and Evaluation Metrics

In this section, we outline three different tasks that span both policy deployment and policy learning. For each task, we also outline different variations and evaluation metrics.

4.1. Goal Selection Task

Given a dataset of experience D, off-domain goal-specification (G_0, G_f), and initial/context observation X_0, the task for the agent is to select an observation $X_f \in D$ that satisfies the goal-specification in context of X_0. The dataset of experience, $D = \{\tau_1, \tau_2, \ldots \tau_N\}$, in general consists of observation-action trajectories. We remain agnostic to the source of dataset, which could have been obtained through teleoperation or autonomous execution various policies. The task for the agent is to find the image in the dataset that has maximum similarity to the goal-specification in the context of initial image X_0. Let μ denote the number of observations in the dataset that satisfy the goal specification. We evaluate the performance according to two metrics:

1. **Top-N Success Rate:** We rank each observation in the dataset according to the similarity measure, and compute the fraction of top $N \leq \mu$ ranked samples that satisfies the goal specification. This metric allows us to study if a high-similarity can be used to retrieve observations close to the goal, and ultimately enable downstream goal-conditioned policy execution.

2. **Dataset Total Variation:** While top-N success rate allows us to evaluate model prediction for high-similarity observations, it is agnostic to how the model behaves for low-similarity observations (i.e. non-goal states). Towards obtaining a broad evaluation metric for similarity scores across the entire dataset, we consider the dataset total variation measure. Let $\boldsymbol{\phi_i}$ denote the similarity score of observation X_i, and let $I(X_i)$ be an indicator function to denote

whether X_i satisfies the goal specification. Then, the dataset total variation is given by:

$$\text{DTV} := \frac{1}{\|D\|} \sum_{X_i \in D} \left| \frac{1}{\mu} I(X_i) - \frac{\boldsymbol{\phi_i}}{\sum_j \boldsymbol{\phi_j}} \right| \tag{3}$$

In our experiments, we average the results over multiple choices of X_0 and goal-specification (G_0, G_f) to quantify the overall performance of the similarity measure.

4.2. High-Level Action Selection Task

The agent is presented with a dataset of experience $D = \{\tau_1, \tau_2, \ldots \tau_N\}$, where each trajectory consists of an observation-action sequence. The agent is also presented with goal specification (G_0, G_f). Let $\Delta g = \psi(G_f) - \psi(G_0)$ be the goal-feature and $\Delta a_i^k = \psi(I_i^k) - \psi(I_0^k)$ be the high-level action feature for timestep i in trajectory k (I_0^k is the initial image in trajectory k). The task for the agent is to find a pair of images in the dataset that has maximum similarity to the high-level action (i.e. transformation) in Δg. The action selection task is useful when a user need to specify an action but do not have access to images of the particular object of interest. For example, they could use images of "opening a cabinet door" to specify "opening microwave", since both share the same high-level action semantics of "opening". For this task, we identify semantically equivalent actions among trajectories and evaluate model performance with Top-N success rate.

4.3. Offline Reinforcement Learning

The task is specified by G_0 and G_f, and a robot must manipulate an object in its own environment that matches the object in G_0 and G_f. A set of sub-optimal trajectories are provided for learning consistent with the offline RL setting. *Behavioral cloning* directly learns a supervised model that maps state to action, and thus simply mimics the behavior in the dataset without any improvement. We propose to evaluate ZeST by using an offline RL algorithm to learn the policy using the ZeST similarity scores as a proxy for the reward function. If we observe an improvement over the behavior cloning baseline, then we may conclude that ZeST scores function as a reasonable reward proxy for policy learning.

5. Experimental Results and Discussion

To test the effectiveness of ZeST, we evaluate performance of various instantiations of ZeST for enabling policy execution and learning. In our experiments, we consider the following design choices for components outlined in Section 3.

1. **Embedding Foundation Models:** ImageNet-supervised ResNet50 (He et al., 2016), ImageNet-trained MoCo (He et al., 2020), and CLIP (Radford et al., 2021). We consider these models primarily due to wide adoption in computer vision, with CLIP being a natural choice for encoding both visual and language goals due to its multi-modal training.
2. **Features:** Raw embedding and delta feature variants as specified in Section 3.
3. **Feature distance:** Cosine similarity and L2 distance.

5.1. How does ZeST perform in the Goal Selection Task? How do various design choices impact performance?

To evaluate the performance of ZeST, we first consider an experience dataset in the Franka Kitchen domain (Gupta et al., 2019). The dataset consists of five different manipulation tasks: turning the top/bottom burner knob on, opening the microwave, opening the hinge door, and opening the sliding door. The experience dataset is generated by rolling out randomized expert policies that were trained using a policy gradient algorithm (Rajeswaran et al., 2017, 2018). Uniform noise of varying levels were added to the actions to create a diverse dataset that contains both successful and failed trajectories. We evaluate the performance of ZeST using different forms of goal-specification: (1) same-scene images – i.e. goal images from the Franka Kitchen scene itself; (2) off-domain images from the internet; (3) hand-sketches and drawings; (4) and instructions in natural language. The internet image and drawings dataset are part of the Visual Task Dataset (Cui, 2021), and Figure 4 provides a few representative examples from this dataset.

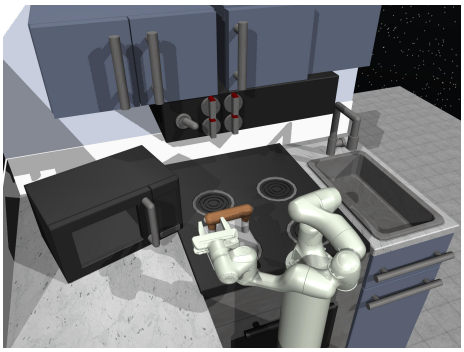

Figure 3: Visualization of the Franka Kitchen domain from Gupta et al. (2019).

We first evaluate the choice of features and feature distance function. To do so, we compute the dataset total variation (Eq. 3), averaged across different embedding models. The results are summarized in Figure 5. We find that when goals are specified through images, the use of delta features and cosine similarity leads to the best results. In case of language goals, we find that using delta features and L2 distance leads to the lowest total variation. Based on these results, for the remainder of the paper, we primarily focus on the best performing design choices.

Figure 4: Samples of different task specification modalities in the Visual Task Dataset (Cui, 2021).

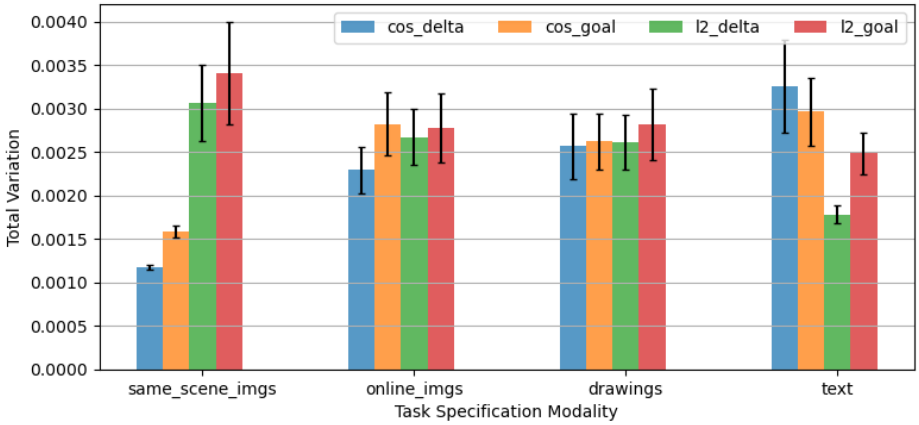

Figure 5: Avg. total variation (with std. error) across tasks in Franka kitchen domain under different task specification modality for candidate similarity computation methods across embedding models. Cosine similarity of delta features outperforms other methods.

We consider two different variations for evaluation: (1) Narrow dataset (ND), in which the experience dataset contains trajectories related to only a single manipulation task. For example, the narrow dataset would only contain (suboptimal) trajectories where the Franka robot attempts to manipulate the microwave. (2) Diverse dataset (DD), which is the entire experience dataset comprising of a diverse set of trajectories related to all the five manipulation tasks. The ND scenario is simpler and serves to evaluate if ZeST can perform goal-selection in the context of a single manipulation task. This can also be useful for policy learning similar to prior works that utilize a goal classifier as a reward function (Fu et al., 2018; Eysenbach et al., 2021). The DD scenario is harder and is also representative of goal-selection for policy execution in open-ended environments.

The experimental results are presented in Figures 6 and 7. In terms of dataset total variation, we find that CLIP has the lowest total variation among the three embedding models across all the goal-specification modalities. We hypothesize that this is due to CLIP being trained on a larger data corpus (400 million images) compared to ResNet50 and MoCo trained on ImageNet (15 million images). CLIP has also shown promising zero-shot results in downstream tasks, and our observations in the goal selection tasks are consistent with these general trends. We also observe that using same scene images leads to better results than using internet images or drawings, which is along expected lines due to smaller domain gap. The Top-25 success rates for both the goal selection and high-level action selection tasks are plotted in Figure 7, along with the performance of a random selection mechanism, which can be used to judge the level of difficulty of the task. We find that ZeST with all the embedding models vastly outperform a random selection scheme suggesting that ZeST is capable of computing useful similarity scores. We again observe that CLIP is the best embedding model, resulting in a near 14-fold improvement over a random selection baseline in the harder DD setting. Further details about this experiment can be found in the Appendix.

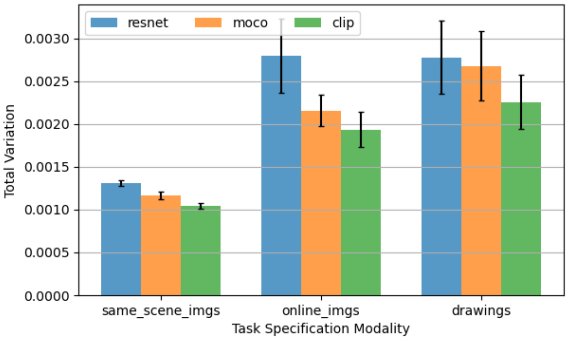

Figure 6: Avg. total variation for three embedding models in Franka Kitchen domain with **Cos+Delta**.

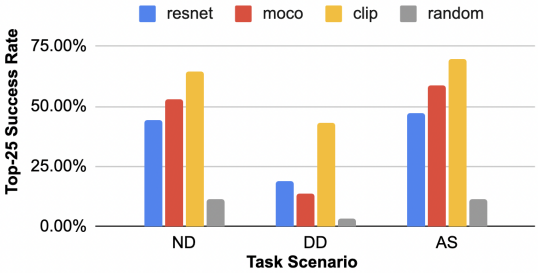

Figure 7: Top-25 success rate of embedding models with **Cos+Delta** under different task scenarios in Franka Kitchen with online images as goal specification modality.

5.2. How does ZeST perform in real-world video datasets?

In this experiment, we examine whether our proposed approach can generalize to real world video data and how different embedding models compare. To this end, we adopt a challenging real-world video dataset, SomethingSomething-V2 (Goyal et al., 2017), that consists of short clips of everyday human actions. These videos are recorded with hand-held devices and therefore do not have a stable view of the scene. Both the motion and human hand in the view can cause occlusion. We extract a subset of these video clips that all are *opening something* as the dataset for experimental evaluation. In addition to real video frames, we further test how synthetic images from a simulated environment interact with real world data, to test cross-domain task specification. Using objects from 3D Warehouse dataset, we generate object trajectories in simulation that matches with the *opening something* tasks from the SomethingSomething-V2 dataset.

Similar to the evaluation in Franka kitchen domain, we consider narrow (ND) and diverse (DD) datasets for evaluation. In real2real modality of goal-specification, we use the frames from real trajectories to specify the goal while in the sim2real modality, we use rendered object frames to specify goal. As shown in Figure 8, averaging over all three embedding models, ZeST instantiations are able to outperform random guessing. Please see the Appendix for additional details and visualizations. Figure 9 presents the results for individual embedding models in the real2real setting. Interestingly, we observe that CLIP is only marginally better than random guessing while ResNet and MoCo are substantially better. Overall, we observe that performing goal-selection and action-selection using real-world video datasets remains challenging. An interesting direction for future work would involve the learning of foundation models using internet-scale video datasets, which we hypothesize would lead to improved results.

5.3. Can ZeST enable offline reinforcement learning?

In this experiment, we test how ZeST signals could be used to enable policy learning in offline RL (Levine et al., 2020). Offline RL is the setting where an agent is presented with a dataset of sub-optimal trajectories, and the agent must learn a competent policy using this dataset without additional environment interactions. We study if the similarity computed by ZeST can be used as a proxy for the reward function in offline RL. To do so, we use the Franka Kitchen domain outlined in Section 5.1, and use internet images for goal-specification. For our experimental evaluation, we compare: (1) behavior cloning (BC) which performs straightforward supervised learning to predic-

tion actions in the dataset conditioned on observations; and (2) decision transformer (DT) (Chen et al., 2021) – a state of the art offline RL algorithm that performs reward conditioned behavior cloning using a sequence model. This comparison involves two closely related algorithms with the major difference being reward conditioning. In the case of DT, we use our ZeST similarity as the reward function. Table 1 shows the learned policies' average performance across task. When goal images with high similarity are selected, DT achieves 90.77% of expert performance. This suggests that ZeST similarities serve as a reasonable proxy, and has the potential to enable offline RL.

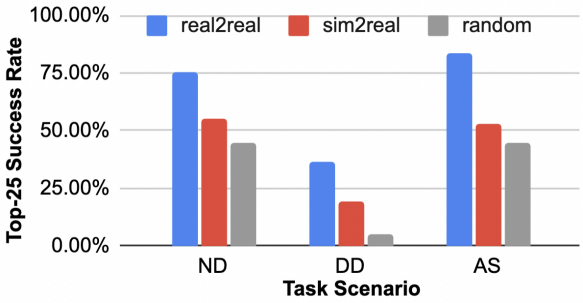

Figure 8: Top-25 success rate of **Cos+Delta** under different task scenarios with different task specification modalities in SSV2 dataset.

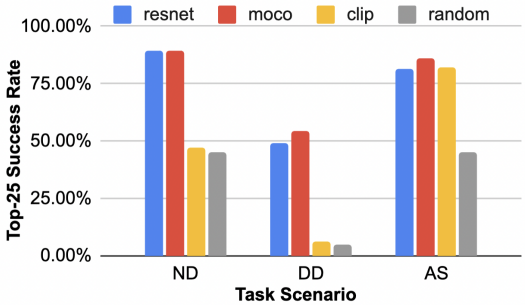

Figure 9: Top-25 success rate of different embedding models under different task scenarios in **real2real** setting in SSV2 dataset.

	ZeST+DT	**Demo.**	**BC**
normalized return	90.77%	81.19%	80.67%

Table 1: Normalized return for ZeST+DT comparing with demonstration dataset average and behavior cloning baseline (expert performance is 100%).

6. Conclusion

In this work, we introduce a *framework* for studying foundation models for zero-shot task specification (ZeST). We evaluate the effectiveness of ZeST for enabling zero-shot *policy execution* through a set of goal selection tasks; and we evaluate ZeST for enabling *policy learning* in offline reinforcement learning. Leveraging existing pre-trained models, we demonstrate the effectiveness of ZeST on goal and action selection tasks in several different domains with different task specification modalities. We find that ZeST is quite effective in zero-shot goal-selection and results in a 14-fold increase in performance over a random guessing baseline. In offline RL, we find that using ZeST scores as a proxy for the reward function enables the learning of policies that perform better than a behavior cloning baseline. Our results show the potential of foundation models and their applications in closing the domain gap for robot learning between real world and simulation.

Instantiations of ZeST in this work are limited in several aspects. The embedding model can be sensitive to occlusions, which often requires removing the robot arm from the camera view. Existing computer vision techniques for object segmentation and in-painting can be leveraged to address this problem for real applications. At the same time, our proposed approach is limited to tasks that have salient visual feature changes from unsuccessful states to successful states and such visual feature is not positional/location-based. However, explicit localization modules can be developed to address such issue and we leave this for future investigation.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

Yizhak Ben-Shabat, Xin Yu, Fatemehsadat Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. 2020.

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.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pages 783–792. PMLR, 2019.

Tom B. Brown et al. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language berts. *Transactions of the Association for Computational Linguistics*, 9:978–994, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Marcos V Conde and Kerem Turgutlu. Clip-art: Contrastive pre-training for fine-grained art classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3956–3960, 2021.

Yuchen Cui. Visual task dataset. https://github.com/yuchen93/visual_tasks_dataset, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Simon Shaolei Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient rl with rich observations via latent state decoding. *ArXiv*, abs/1901.09018, 2019.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. *ArXiv*, abs/2011.08909, 2021.

Kevin Frans, LB Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*, 2021.

Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. *Advances in Neural Information Processing Systems*, 31:8538–8547, 2018.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Advances in Neural Information Processing Systems*, 31:9094–9104, 2018.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. *Proceedings of the International Conference on Robot and Automation*, 2021.

Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer, 1993.

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. *arXiv preprint arXiv:2111.09888*, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

Youngwoon Lee, Edward S Hu, and Joseph J Lim. IKEA furniture assembly environment for long-horizon complex manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. URL https://clvrai.com/furniture.

Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.

Ajay Mandlekar, Danfei Xu, Roberto Martın-Martın, Silvio Savarese, and Li Fei-Fei. Gti: Learning to generalize across long-horizon tasks from human demonstrations. *Proceedings of Robotics: Science and Systems*, 2020.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in Neural Information Processing Systems*, 31:9191–9200, 2018.

Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*, 2019.

Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The Unsurprising Effectiveness of Pre-Trained Vision Models for Control. *ArXiv*, abs/2203.03580, 2022.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

Lerrel Pinto and Abhinav Kumar Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3406–3413, 2016.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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. *arXiv preprint arXiv:2103.00020*, 2021.

Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham Kakade. Towards Generalization and Simplicity in Continuous Control. In *NIPS*, 2017.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *Self-Supervision for Reinforcement Learning Workshop-ICLR 2021*, 2021.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *Proceedings of Robotics: Science and Systems*, 2019.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020.

Appendix

Table of Contents

A Details of Foundation Models under Test 15

B Further Analysis of ZeST and Additional Design Considerations 15

C Qualitative Evaluations of ZeST on Real Image Trajectories 17

D Details of Goal Selection Experiments in Franka Kitchen Domain 17

E Details of Goal Selection Experiments in SSV2 Dataset 27

F Goal Similarity Scores for Multi-step Tasks 27

G T-REX Training 29

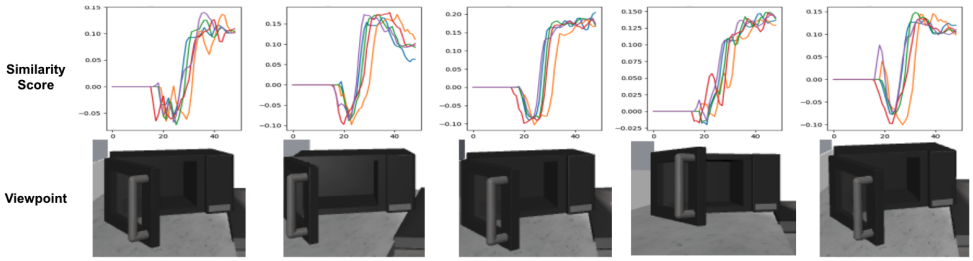

Figure 10: **Cos+Delta** signals under different viewpoints for microwave opening task.

Appendix A. Details of Foundation Models under Test

Vision-based robot learning enables training end-to-end robot policies that directly work with visual inputs. However, real-world robot manipulation data are expensive and often cannot generate data at the scale required to train vision models. Therefore, existing vision-based robot learning methods often rely on pre-trained vision models and fine-tune with limited data. Deep residual network (ResNet), proposed by He et al. (2016), greatly increased the depth of neural networks through using residual connections. ResNet has been used as the backbone for many state-of-the-art vision models. Pre-trained ResNet on image classification task on ImageNet (Deng et al., 2009) has been a popular and very effective embedding model for extracting visual features for down-stream robotic manipulation tasks (Gupta et al., 2018; Mandlekar et al., 2020; Shah and Kumar, 2021). MoCo (He et al., 2020) is a recent self-supervised pre-training technique, i.e. pre-train without using the labels, and has shown better performance for downstream vision tasks than supervised pre-training with image classification. Moco employs a contrastive loss, for some query q and a set of encoded samples $\{k_0, k_1, ...k_K\}$ (where k_+ is the desired key for query q):

$$\mathcal{L}_q = -\log \frac{\exp\left(q \cdot k_+/\tau\right)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)} \tag{4}$$

CLIP (Radford et al., 2021) is a multi-modal pre-training technique that leverages both images and text descriptions to learn embeddings of both modality in the same space. The training objective of CLIP is to maximize the cosine similarity between matching pairs of encoded image i_k and text t_k while minimizing the similarities between unmatched pairs:

$$\mathcal{L}_k = -\log \frac{\exp\left(i_k \cdot t_k/\tau\right)}{\sum_{n=0}^{N} \exp(i_k \cdot t_n/\tau)} - \log \frac{\exp\left(i_k \cdot t_k/\tau\right)}{\sum_{n=0}^{N} \exp(i_n \cdot t_k/\tau)} \tag{5}$$

CLIP is trained with over 400 million pairs of internet image and text and has shown state-of-the-art zero-shot performance on image classification tasks. Self-supervision models such as Moco and CLIP have found wide applicability in computer vision as backbone representations. However, they have not been widely used in RL, and our work explores this possibility.

Appendix B. Further Analysis of ZeST and Additional Design Considerations

Viewpoints To mitigate potential viewpoint differences between the observations and goal specification images, it is desired to use multi-viewpoint observations and take the average similarity

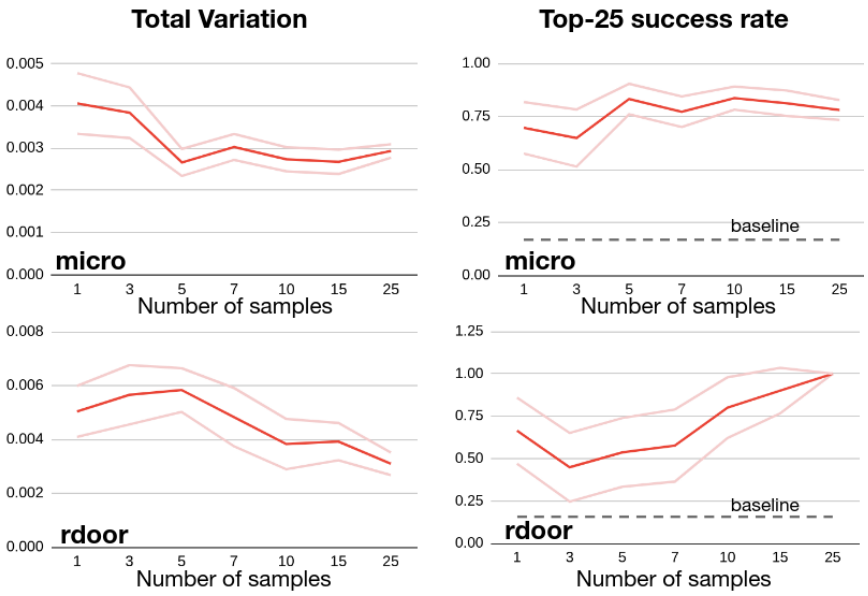

Figure 11: Example performance profiles based on number of goal samples used.

score across different viewpoints. As shown in Figure 10, we observe that the cosine similarity for same trajectories of opening a microwave differ from different viewpoints. However, this may not be possible in certain real world applications and can be mitigated by filtering or reweighing goal specifications using similarities with embeddings of the robot's observations.

Ensemble of task specifications To reduce potential noise in the computed signals, it is desired to use an **ensemble** of goal specification images. When multiple goal and initial images are provided, for each goal image, we will find an initial image that is the most similar to the goal image's embedding using cosine similarity measure and then compute the delta feature between the two. With this ensemble of images, the average of all cosine similarity scores is used as the final prediction.

We find the use of an ensemble of goal specifications help to reduce noise and improve the smoothness of the predicted signal. We further investigate how the performance of one instantiation of ZeSTS change based on the number of goal samples used in the dataset. We evaluated with goal selection task in the Franka Kitchen domain using internet images as the goal specification modality. Each task has a total of 50 goal samples (for both the initial and goal state). We sample a desired number of goal samples from the dataset and repeat for 5 times. The average performance with standard errors is plotted in Figure 11. In general, more samples would improve the performance of the model, but the initial number of samples necessary to achieve a certain performance varies from task to task.

Pixel differences and Cos+Delta scores A natural question to ask when considering whether cosine similarities between delta features are meaningful is that "Does pixel differences between image frames correlate with high cosine similarities?". Figure 12 shows that large pixel differences do not correlate with high similarity scores.

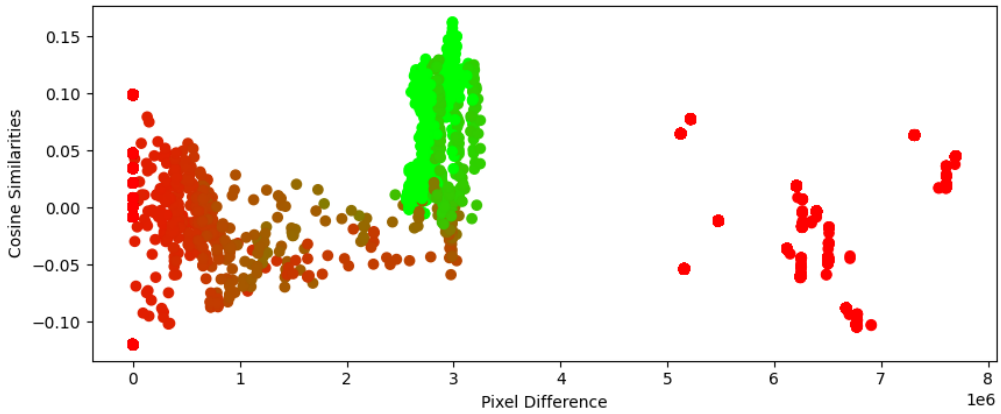

Figure 12: Relationship between cosine similarities between delta features and pixel differences between frames for microwave opening task in Franka Kitchen with CLIP embeddings (the greenness of a dot is indicative of its distance to true goal states).

2*Modality	2*model	Tasks						2*Avg.	2*Std. Err
		knob3	rdoor	sdoor	knob2	ldoor	micro		
Random (goal frame ratio)		10.90%	5.3%	33.4%	4.9%	5.7%	8.0%	11.4%	-
3*same_scene_imgs	resnet	3.0%	100.0%	76.0%	97.0%	100.0%	90.0%	77.7%	2.33%
	moco	31.0%	100.0%	98.0%	99.0%	93.0%	100.0%	86.8%	3.17%
	clip	98.0%	100.0%	62.0%	100.0%	91.0%	94.0%	**90.8%**	4.73%
3*online_imgs	resnet	0.0%	28.3%	36.0%	34.3%	21.0%	59.7%	29.9%	14.44%
	moco	26.3%	79.3%	1.0%	67.7%	8.3%	77.0%	43.3%	11.35%
	clip	17.7%	83.7%	80.3%	13.3%	77.0%	92.0%	**60.7%**	8.95%
3*drawings	resnet	100.0%	0.0%	0.0%	33.0%	0.0%	16.0%	24.8%	2.95%
	moco	27.0%	98.0%	12.0%	12.5%	0.0%	26.0%	29.3%	4.92%
	clip	24.5%	0.0%	49.0%	23.0%	98.0%	57.0%	**41.9%**	10.33%
text	clip	0.0%	46.0%	4.0%	0.0%	8.0%	1.0%	9.8%	2.30%

Table 2: Top-25 goal selection success rate in Franka kitchen domain (SGSD).

Appendix C. Qualitative Evaluations of ZeST on Real Image Trajectories

In this section we present qualitative evaluation of **Cos+Delta** with all three embedding models for real world object manipulation trajectories. Sample task specification images used, robot trajectory frames, and the corresponding normalized cosine similarity score are plotted. Figure 15,16,13,14,17, and 18 shows the evaluation for tasks of opening fridge top door, opening fridge bottom door, opening a microwave, opening a cabinet, opening an oven, and opening a drawer respectively. As shown in these figures, ZeST signals are sensitive to occlusions caused by the robot manipulator as well as different croppings of the trajectory images.

Appendix D. Details of Goal Selection Experiments in Franka Kitchen Domain

1. **SGSD**: As shown in Table 2, CLIP outperform other embedding models with image-based task specification modalities using **Cos+Delta**.

2. **DGSD**: As shown in Table 3, CLIP does not outperform the other two embedding models.

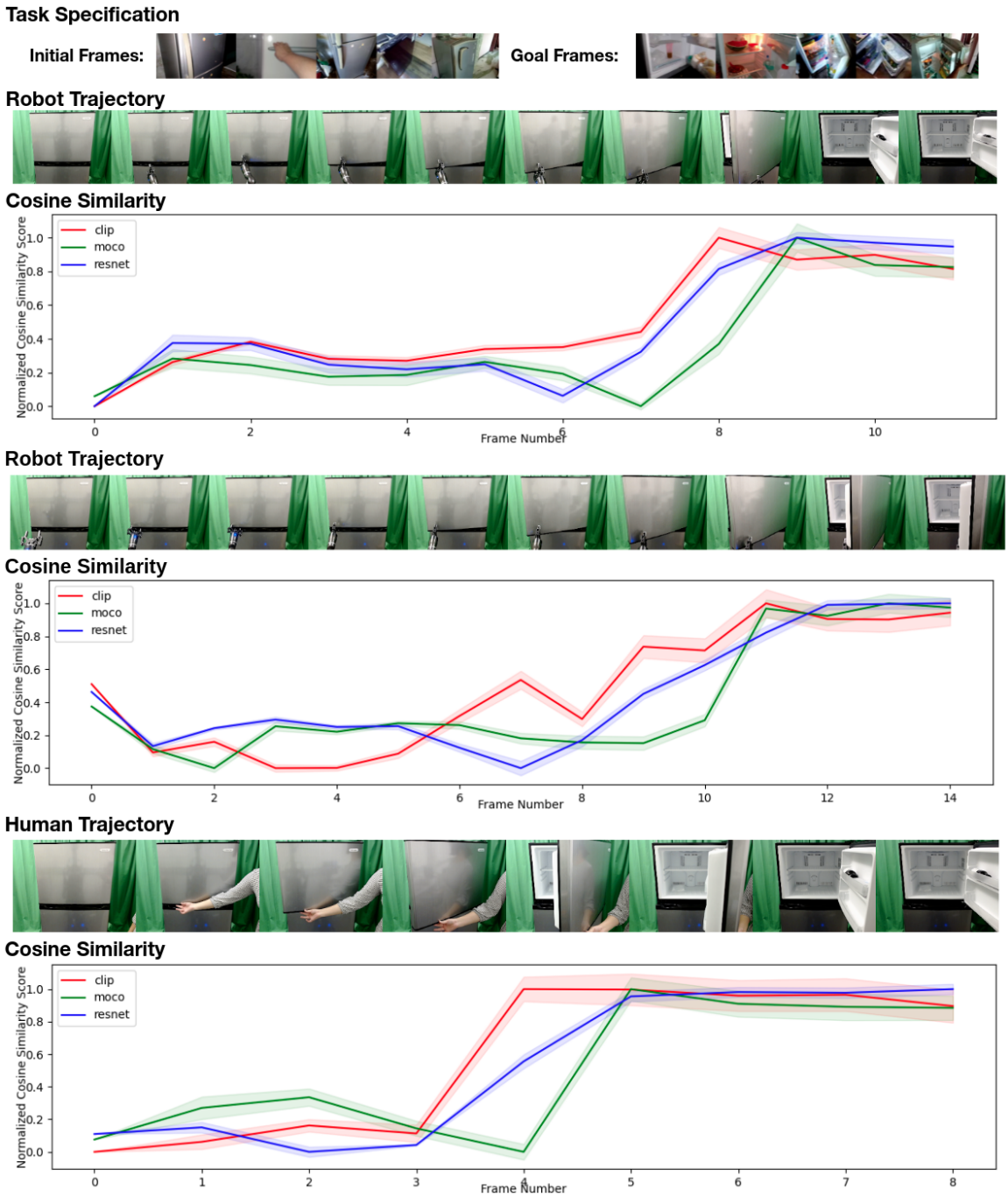

Figure 13: Opening Refrigerator Task (top door)

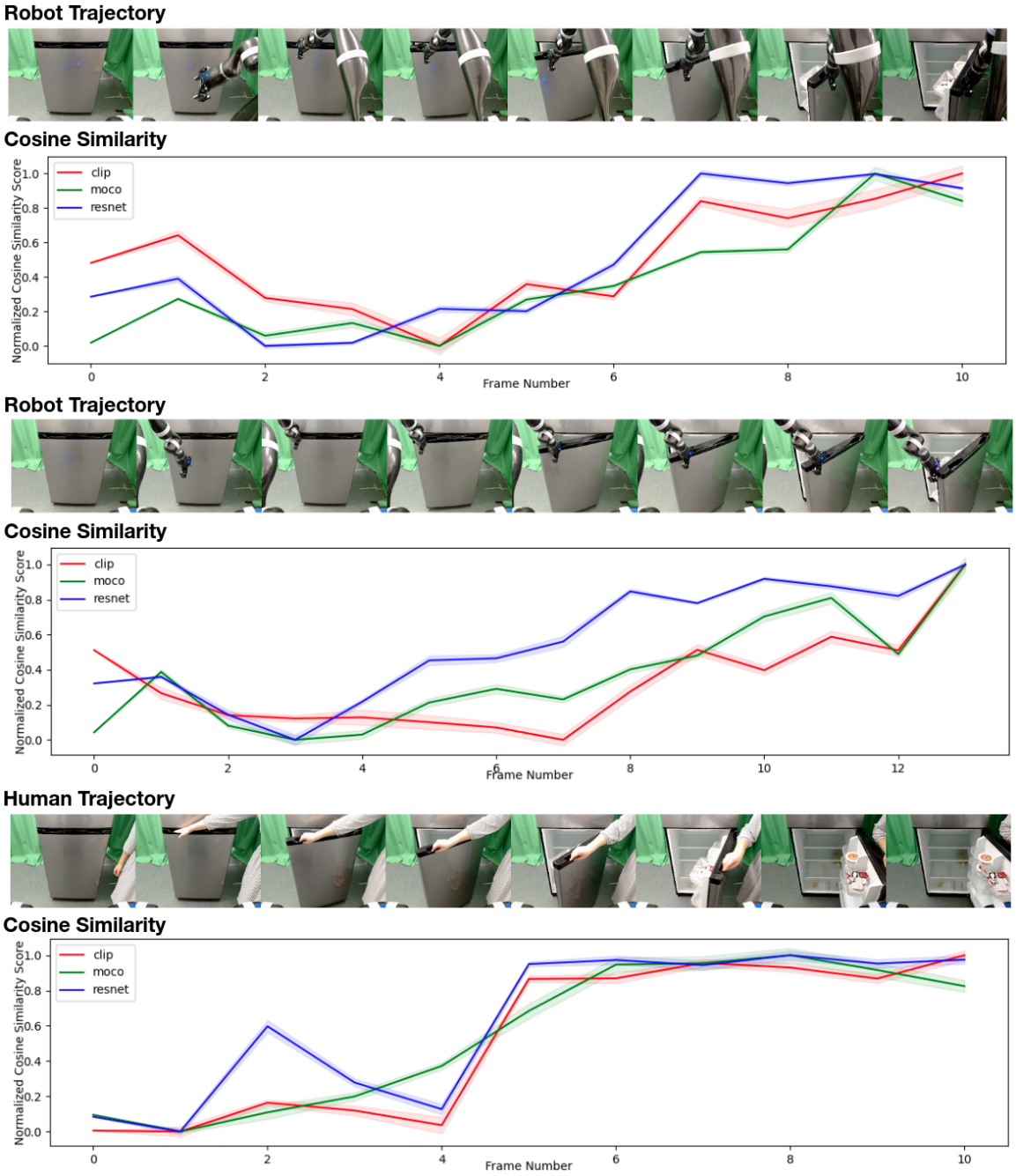

Figure 14: Opening Refrigerator Task (bottom door)

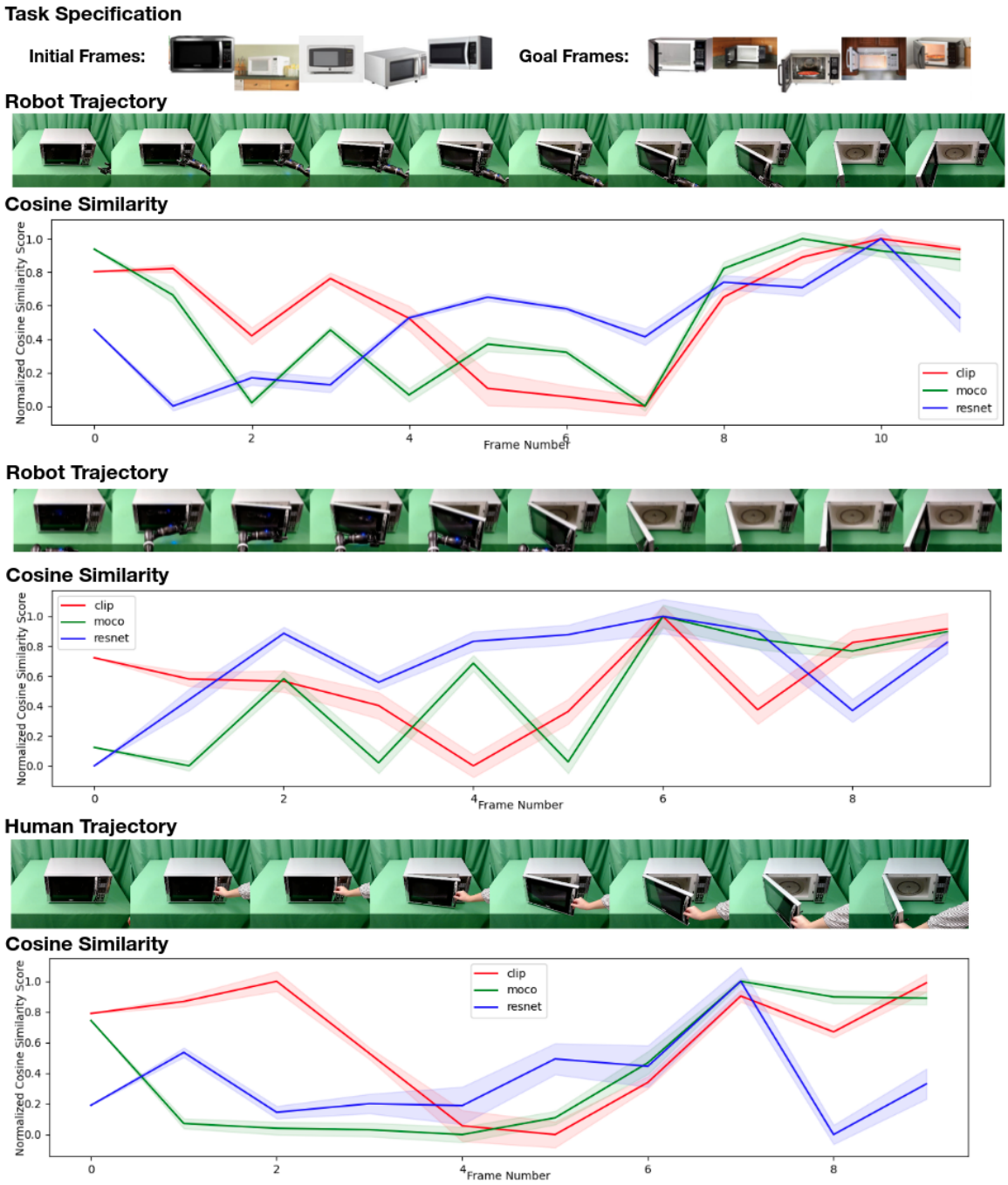

Figure 15: Opening Microwave Task

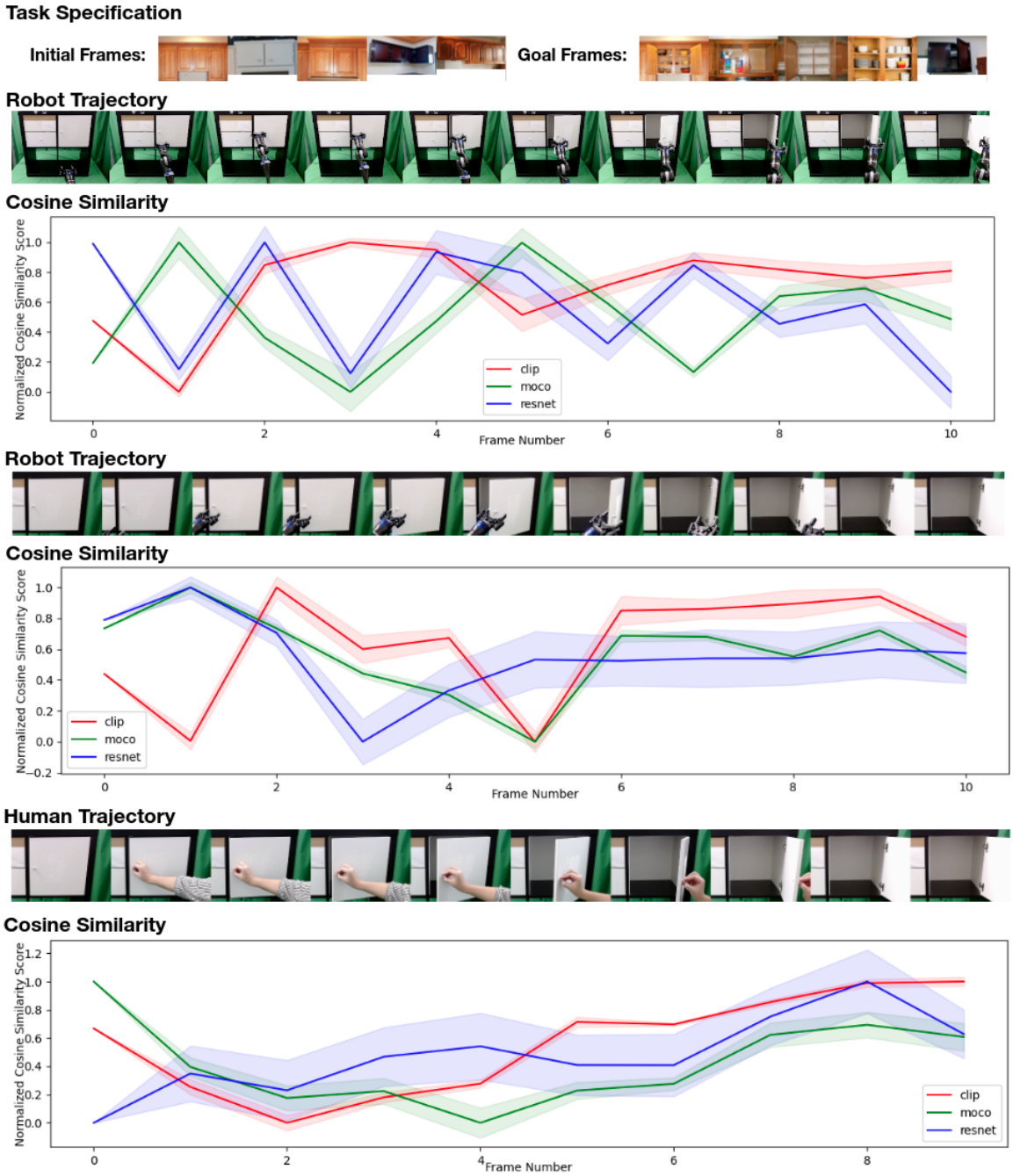

Figure 16: Opening Cabinet Task

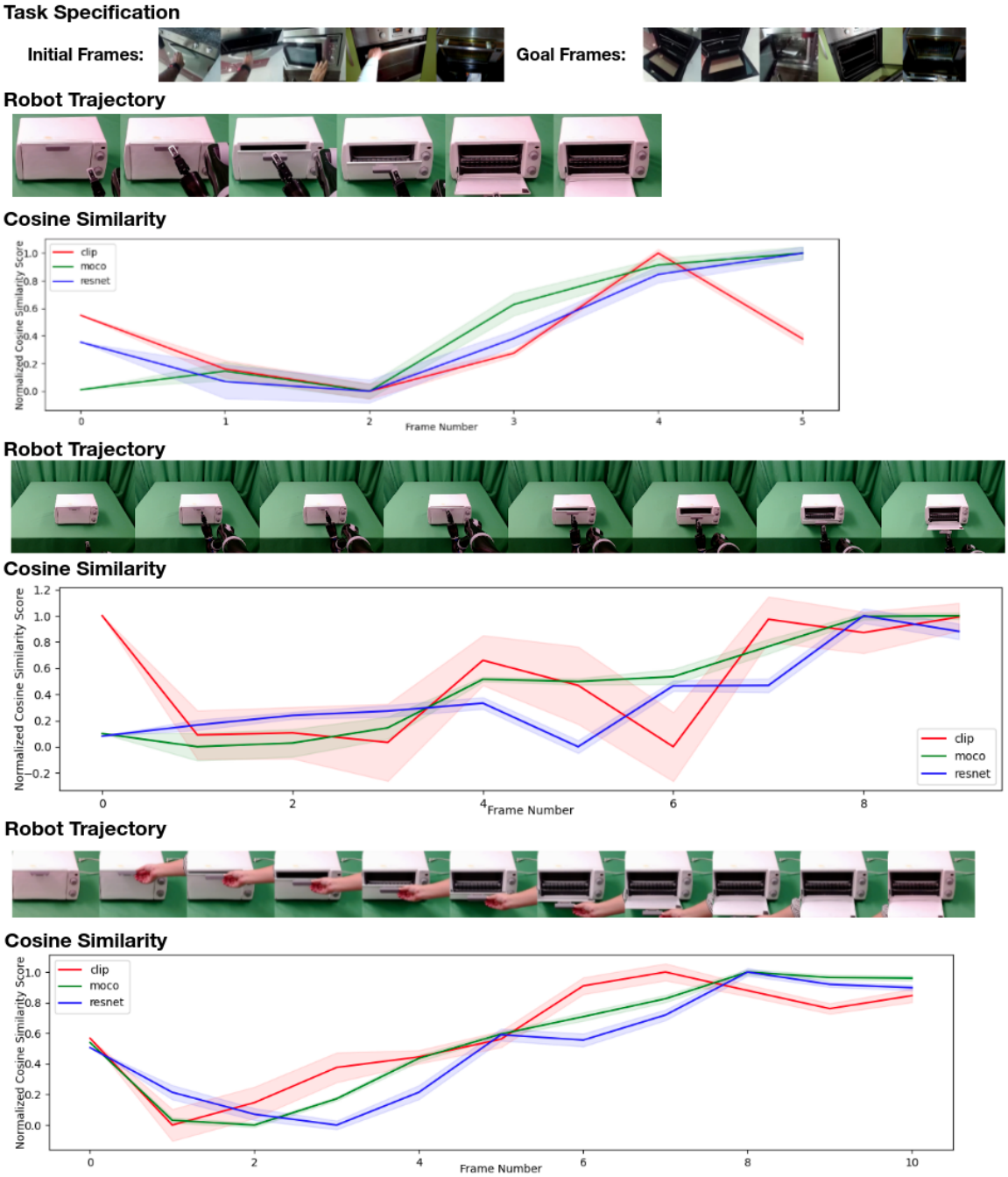

Figure 17: Opening Oven Task

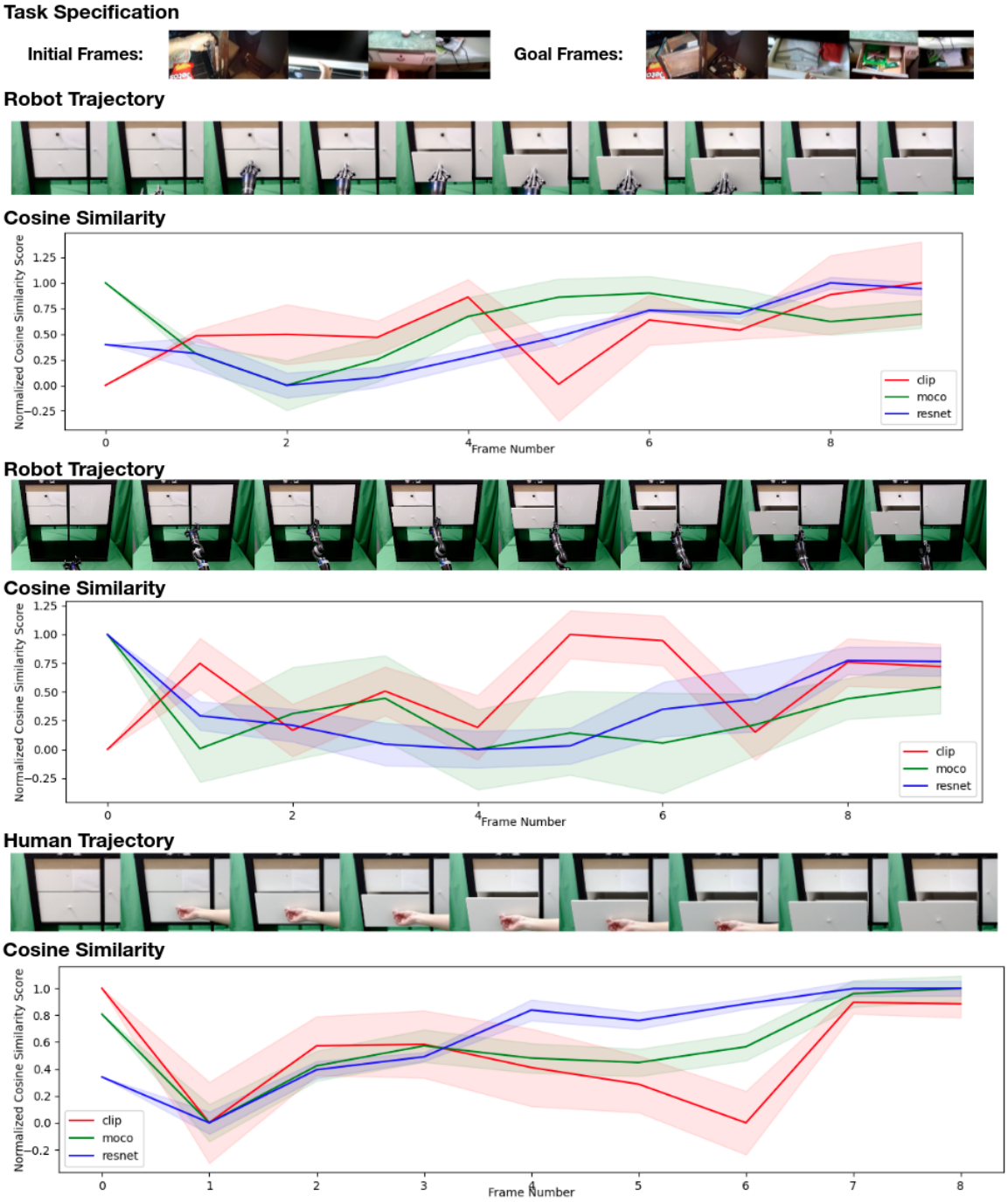

Figure 18: Opening Drawer Task

Task Specification modality	2***model**	**Task ID**						2***avg**
		knob3	**rdoor**	**sdoor**	**knob2**	**ldoor**	**micro**	
3*same_scene_imgs	ResNet	26.40%	23.20%	53.60%	40.80%	20.00%	30.67%	**32.44%**
	Moco	27.20%	19.20%	3.20%	39.20%	35.33%	40.00%	27.36%
	CLIP	18.67%	6.40%	7.20%	28.00%	28.80%	16.00%	17.51%
3*online_imgs	ResNet	23.47%	5.07%	20.00%	3.73%	11.20%	26.44%	14.99%
	Moco	26.40%	20.00%	3.20%	41.33%	31.73%	39.78%	**27.07%**
	CLIP	25.33%	40.27%	8.00%	27.73%	35.47%	14.67%	25.24%
3*drawings	ResNet	20.00%	4.00%	53.60%	43.33%	6.40%	12.53%	**23.31%**
	Moco	50.67%	20.00%	3.20%	4.80%	7.20%	52.00%	22.98%
	CLIP	9.33%	0.00%	36.00%	0.00%	20.00%	22.40%	14.62%
text	CLIP	20.00%	2.40%	0.00%	10.40%	16.67%	15.20%	10.78%

Table 3: Top-25 success rate for experiment in Franka Kitchen domain (DGSD).

Task Specification modality	2***model**	**Task ID**						2***avg**
		knob3	**rdoor**	**sdoor**	**knob2**	**ldoor**	**micro**	
3*same_scene_imgs	ResNet	12.00%	100.00%	0.00%	0.00%	100.00%	0.00%	35.33%
	Moco	0.00%	100.00%	100.00%	0.00%	0.00%	0.00%	33.33%
	CLIP	0.00%	100.00%	64.00%	100.00%	100.00%	100.00%	**77.33%**
3*online_imgs	ResNet	0.00%	0.00%	0.00%	0.00%	6.00%	0.00%	1.00%
	Moco	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	CLIP	0.00%	0.00%	54.00%	4.00%	100.00%	64.00%	**37.00%**
3*drawings	ResNet	100.00%	0.00%	0.00%	0.00%	0.00%	18.00%	**19.67%**
	Moco	0.00%	0.00%	64.00%	0.00%	0.00%	0.00%	10.67%
	CLIP	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	16.67%
text	CLIP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 4: Top-25 success rate for goal selection task in Franka kitchen domain (SGDD).

Task Specification modality	2***model**	**Task ID**						2***avg**
		knob3	**rdoor**	**sdoor**	**knob2**	**ldoor**	**micro**	
3*same_scene_imgs	ResNet	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Moco	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	CLIP	0.00%	24.00%	0.00%	33.33%	33.33%	0.00%	15.11%
3*online_imgs	ResNet	6.00%	0.00%	11.56%	0.00%	0.00%	0.00%	2.93%
	Moco	0.00%	32.89%	0.00%	0.00%	14.00%	0.00%	7.81%
	CLIP	12.00%	33.33%	0.00%	0.00%	0.00%	0.00%	7.56%
3*drawings	ResNet	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Moco	0.00%	33.33%	0.00%	0.00%	0.00%	0.00%	5.56%
	CLIP	18.22%	0.00%	30.67%	0.00%	32.89%	0.00%	13.63%
text	CLIP	0.00%	0.00%	2.67%	33.33%	0.00%	0.00%	6.00%

Table 5: Top-25 success rate for goal selection task in Franka kitchen domain (DGDD).

Task Specification modality	2***model**	**Task ID**						2***avg**
		knob3	**rdoor**	**sdoor**	**knob2**	**ldoor**	**micro**	
3*same_scene_imgs	ResNet	16.00%	100.00%	96.00%	100.00%	100.00%	96.00%	84.67%
	Moco	56.00%	100.00%	100.00%	100.00%	100.00%	100.00%	92.67%
	CLIP	100.00%	100.00%	64.00%	100.00%	100.00%	100.00%	**94.00%**
3*online_imgs	ResNet	32.80%	0.00%	100.00%	0.00%	54.40%	32.80%	36.67%
	Moco	60.00%	100.00%	21.60%	61.60%	71.20%	52.00%	61.07%
	CLIP	18.40%	98.40%	68.00%	8.80%	100.00%	89.60%	**63.87%**
3*drawings	ResNet	100.00%	0.00%	0.00%	8.00%	0.00%	14.40%	20.40%
	Moco	0.00%	100.00%	33.60%	0.00%	0.00%	4.00%	22.93%
	CLIP	38.40%	0.00%	96.00%	0.00%	100.00%	76.00%	**51.73%**
text	CLIP	0.00%	0.00%	8.00%	100.00%	0.00%	0.00%	18.00%

Table 6: Action Selection task for Franka Kitchen Domain

3***Exp**	3***Model**	**Task ID**							3*All tasks
		2*door	2*drawer	2*box	2*fridge	2*laptop	washing machine	2*oven	
3*real2real	CLIP	38.00%	46.00%	61.00%	38.00%	23.00%	55.00%	69.00%	47.14%
	ResNet	96.00%	99.00%	68.00%	100.00%	97.00%	68.00%	100.00%	**89.71%**
	Moco	88.00%	94.00%	66.00%	100.00%	95.00%	84.00%	100.00%	89.57%
3*sim2real	CLIP	46.00%	46.00%	53.00%	42.00%	34.00%	40.00%	81.00%	48.86%
	ResNet	61.00%	51.00%	32.00%	36.00%	98.00%	9.00%	77.00%	52.00%
	Moco	35.00%	90.00%	36.00%	68.00%	94.00%	54.00%	70.00%	**63.86%**

Table 7: Top-25 success rate for experiment with SSV2 dataset: SGSD

3. **SGDD**: As shown in Table 4, CLIP outperforms the other two models under 2 out of the 3 image-based task specification modalities in the goal selection task in the Franka kitchen domain (using **Cos+Delta**).

4. **DGDD**: As shown in Table 5, CLIP outperforms the other two models in this hardest setting as well. However, the overall performance is not very good.

5. **ASDD**(Action Selection): As shown in Table 6, CLIP outperforms the other two embedding models in this scenario.

3***Exp**	3***Model**	**Task ID**							3*All tasks
		2*door	2*drawer	2*box	2*fridge	2*laptop	washing machine	2*oven	
3*real2real	CLIP	46.00%	40.00%	42.00%	42.00%	44.00%	48.00%	44.00%	43.71%
	ResNet	50.00%	59.00%	64.00%	55.00%	31.00%	49.00%	56.00%	**52.00%**
	Moco	60.00%	56.00%	51.00%	52.00%	34.00%	55.00%	49.00%	51.00%
3*sim2real	CLIP	42.00%	40.00%	52.00%	41.00%	47.00%	46.00%	43.00%	44.43%
	ResNet	35.00%	58.00%	42.00%	58.00%	37.00%	58.00%	41.00%	47.00%
	Moco	64.00%	61.00%	50.00%	68.00%	23.00%	52.00%	67.00%	**55.00%**

Table 8: Top-25 success rate for experiment with SSV2 dataset: DGSD

3*Exp	3*Model	Task ID							3*All tasks
		2*door	2*drawer	2*box	2*fridge	2*laptop	washing machine	2*oven	
3*real2real	CLIP	12.00%	8.00%	1.00%	4.00%	1.00%	11.00%	5.00%	6.00%
	ResNet	21.00%	39.00%	0.00%	100.00%	92.00%	25.00%	67.00%	49.14%
	Moco	0.00%	57.00%	1.00%	100.00%	100.00%	25.00%	100.00%	**54.71%**
3*sim2real	CLIP	1.00%	1.00%	11.00%	5.00%	1.00%	12.00%	7.00%	5.43%
	ResNet	8.00%	29.00%	20.00%	0.00%	97.00%	0.00%	11.00%	23.57%
	Moco	33.00%	52.00%	23.00%	0.00%	92.00%	0.00%	0.00%	**28.57%**

Table 9: Top-25 success rate for experiment with SSV2 dataset: SGDD

3*Exp	3*Model	Task ID							3*All tasks
		2*door	2*drawer	2*box	2*fridge	2*laptop	washing machine	2*oven	
3*real2real	CLIP	6.00%	4.00%	6.00%	6.00%	3.00%	6.00%	7.00%	5.43%
	ResNet	9.00%	2.00%	8.00%	0.00%	0.00%	1.00%	4.00%	3.43%
	Moco	15.00%	13.00%	16.00%	0.00%	0.00%	4.00%	0.00%	**6.86%**
3*sim2real	CLIP	8.00%	6.00%	6.00%	11.00%	3.00%	8.00%	5.00%	6.71%
	ResNet	12.00%	7.00%	9.00%	9.00%	0.00%	12.00%	0.00%	7.00%
	Moco	7.00%	15.00%	7.00%	5.00%	0.00%	17.00%	4.00%	**7.86%**

Table 10: Top-25 success rate for experiment with SSV2 dataset: DGDD

3*Exp	3*Model	Task ID							3*All tasks
		2*door	2*drawer	2*box	2*fridge	2*laptop	washing machine	2*oven	
2*R2R	CLIP	0.65	0.64	0.28	1.0	1.0	0.64	1.0	0.74
	ResNet	0.63	0.75	0.62	0.94	1.0	0.78	0.8	0.79
2*S2S	CLIP	0.75	0.23	0.14	0.66	0.37	0.72	0.93	0.54
	ResNet	0.46	0.42	0.98	0.97	0.81	0.7	1.0	0.76
2*R2S	CLIP	0.2	0.32	0.2	0.0	0.13	0.13	0.24	0.17
	ResNet	0.0	0.1	0.21	0.03	0.93	0.11	0.29	0.24
2*S2R	CLIP	0.18	0.35	0.1	0.54	0.78	0.06	0.3	0.33
	ResNet	1.0	0.97	0.0	0.4	0.8	0.28	0.62	0.58

Table 11: Top-25 success rate for experiment with SSV2 dataset: ASDD

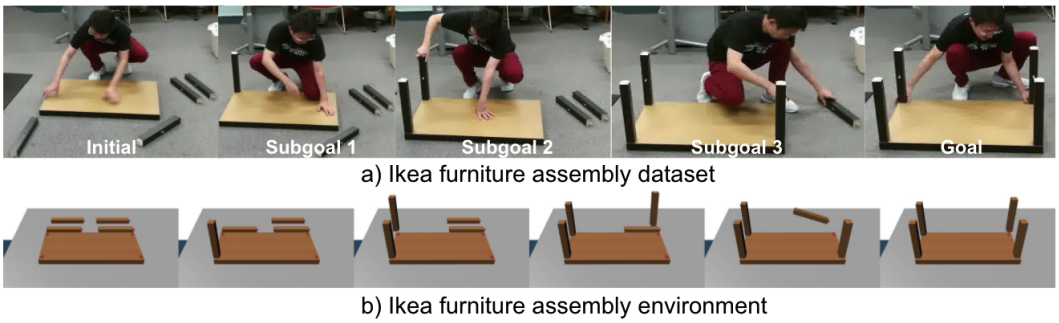

Figure 19: Sample image frames from IKEA furniture assembly dataset and simulation environment.

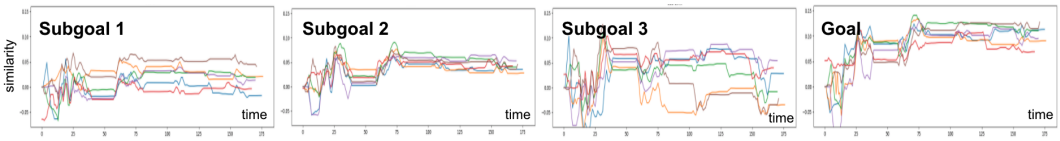

Figure 20: Cosine similarities (from the CLIP-based model) to each subgoal for simulated furniture assembly trajectories.

Appendix E. Details of Goal Selection Experiments in SSV2 Dataset

For this experiment, we test **Cos+Delta** with three different embedding models. We conducted each experiment with 10 different repetitions, each time sampling a different trajectory for specifying the task. The results for the five different task scenarios SGSD,DGSD,SGDD,DGDD, and ASDD are shown in Table 7,8,9,10 and 11 respectively. As shown, the performance of CLIP is much lower than that of ResNet and Moco with SSV2 dataset.

Appendix F. Goal Similarity Scores for Multi-step Tasks

Many realworld tasks are multi-step and consist of different numbers of sub-goals. In this experiment, we evaluate how our proposed method work for multi-step tasks and investigate whether specifying sub-goals improves its performance. More specifically, we study the task of table assembly with real world video data from the IKEA furniture assembly dataset (Ben-Shabat et al., 2020) and synthetic visual trajectories in the IKEA assembly environment (Lee et al., 2021). Sample trajectories from the dataset and the simulator are shown in Figure 19. This setting corresponds to when an agent is learning from human demonstrations in the form of video recordings and it has access to past experience data in simulation. Figure 20 shows the consine similarities to each subgoal for different simulated trajectories. The average similarity across time increases from the first subgoal to the final goal since the final goal contains the subgoals. However, the similarity score pattern is not clear enough for telling apart at which point a subgoal is achieved. A more complex method is needed to find subgoals in these trajectories.

	ResNet	Moco	CLIP
knob2	71.71%	69.51%	72.20%
knob3	76.10%	69.51%	80.49%
ldoor	76.34%	78.05%	76.34%
micro	84.88%	85.12%	86.34%
rdoor	71.71%	71.71%	71.71%
sdoor	86.34%	84.88%	80.49%

Table 12: Max. T-REX test accuracy under each model.

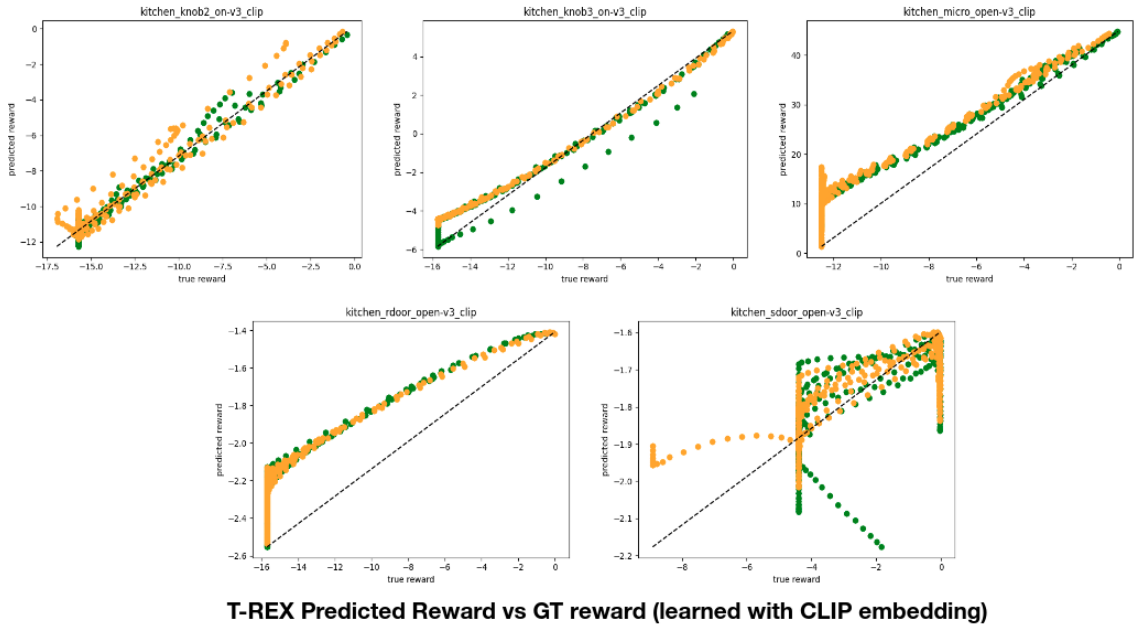

Figure 21: Return predicted by T-REX models vs GT returns. Green dots are seen trajectories that were used in training and yellow dots are unseen trajectories.

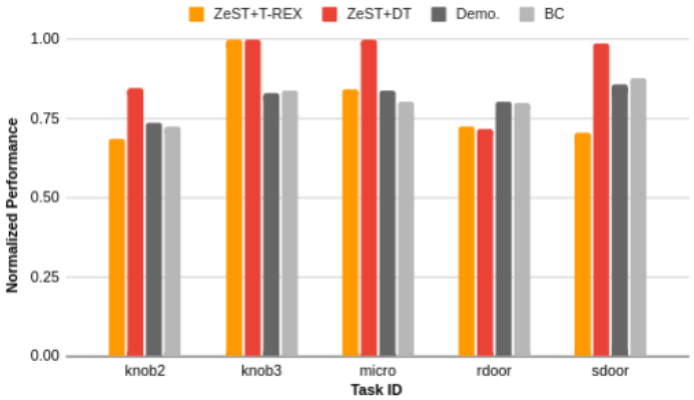

Figure 22: Performance T-REX policies comparing with other methods

Appendix G. T-REX Training

We also employ a ranking-based reward learning method, T-REX (Brown et al., 2019), to learn a reward function. We train T-REX with trajectories ranked by aggregated cosine similarities. We then test on classifying sub-trajectories (relative ranking) with ground-truth dense reward rankings. Table 12 shows the maximum test accuracy under each model for different tasks. Figure 21 shows the relationship between predicted returns and ground truth returns on both seen and unseen trajectories. The return predicted by T-REX models correlate well with ground truth return for four out of the five tasks. We observe that the predicted rewards correlate well with ground-truth rewards. We then use the learned rewards to train policy gradient agents. Figure 22 shows the learned policies' performance with different methods under each task. The return is normalized to the expert's performance as 1.0. T-REX training is able to generate policies that outperform behavioral cloning in 2 out of the 5 tasks, while T-REX does not need the action labels from the demonstrations.