LIV: Language-Image Representations and Rewards for Robotic Control

Yecheng Jason Ma [1] **William Liang** [1] **Vaidehi Som** [1]
Vikash Kumar [2] **Amy Zhang** [2] **Osbert Bastani** [1] **Dinesh Jayaraman** [1]

Abstract

We present **L**anguage-**I**mage **V**alue learning (LIV), a unified objective for vision-language representation and reward learning from action-free videos with text annotations. Exploiting a novel connection between dual reinforcement learning and mutual information contrastive learning, the LIV objective trains a multi-modal representation that implicitly encodes a universal value function for tasks specified as language or image goals. We use LIV to pre-train the first control-centric vision-language representation from large human video datasets such as EpicKitchen. Given only a language or image goal, the pre-trained LIV model can assign dense rewards to each frame in videos of unseen robots or humans attempting that task in unseen environments. Further, when some target domain-specific data is available, the same objective can be used to fine-tune and improve LIV and even other pre-trained representations for robotic control and reward specification in that domain. In our experiments on several simulated and real-world robot environments, LIV models consistently outperform the best prior input state representations for imitation learning, as well as reward specification methods for policy synthesis. Our results validate the advantages of joint vision-language representation and reward learning within the unified, compact LIV framework. Project website: penn-pal-lab.github.io/LIV

1. Introduction

What are the key machine learning challenges in building a general-purpose robot? Consider a home robot for non-expert end users. Such a robot must acquire common-sense knowledge applicable to generic homes, permitting it to operate from visual observations with some minimal proficiency right off the shelf. Then, it must be able to quickly and robustly adapt to the specifics of the user's home, conditioning its language understanding in the particular visual context of its new habitat. Finally, it must be able to autonomously learn arbitrary new skills specified by its user, most naturally in plain language.

Motivated by such considerations, we identify three key desiderata for control-aware vision-language representations. The first two deal with qualities of the trained representation: (1) It must **align the two modalities to permit grounding language specifications** for effective task representation, and (2) It must **capture task-directed progress grounded in language** to supply intermediate learning signals for autonomous skill acquisition. The last desideratum is concerned with how these control-aware VLMs must be trained. Language grounding is commonly context-dependent, so effective representations must be domain-aware. On the flip side, domain-specific data is typically expensive to collect and therefore scarce in robotics settings, making any domain-specific fine-tuning of large models challenging. Our third criterion for our vision-language representation then is that: (3) It must permit **both extensive domain-generic pre-training as well as domain-specific fine-tuning** on small datasets.

To achieve all three criteria, we propose **L**anguage-**I**mage **V**alue Learning (LIV), a unified objective for joint vision-language representation and reward learning. LIV can flexibly pre-train representations on arbitrary video activity datasets with text annotations, even including purely observational datasets of human activity, for which there are several large and conveniently available options (Damen et al., 2018; Grauman et al., 2022; Goyal et al., 2017). Afterwards, the very same objective can be used to fine-tune those representations on small datasets of in-domain robot data, to overcome domain gaps and ground language in context-specific ways.

At a technical level, LIV builds on our prior work Value-Implicit Pre-Training (VIP) (Ma et al., 2022b), a self-supervised approach for learning visual goal-conditioned value functions and representations from videos, generalizing it to learning multi-modal, vision-language values and representations from *language-aligned* videos, as described above. Interestingly, we show that LIV is a more

[1]University of Pennsylvania [2]Meta AI. Correspondence to: Jason Ma <jasonyma@seas.upenn.edu>.

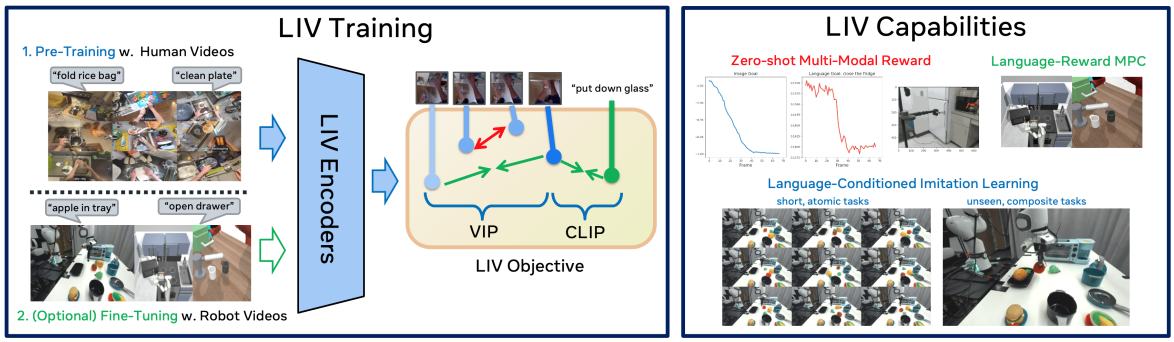


Figure 1. **Language-Image Value Learning (LIV)** for vision-language reward and representation learning. Using the same objective for pre-training and fine-tuning, LIV induces a cross-modal embedding with both temporal coherence and semantic alignment. LIV's multi-modal value representations enable diverse visuomotor control applications.

general variation of the well-known mutual-information based image-text contrastive representation learning objective, as used in CLIP (Radford et al., 2021); this observation simplifies LIV's practical implementation to a simple yet principled combination of the VIP and CLIP objectives; see Figure 1 for an overview of LIV.

We perform extensive experimental evaluations on several simulated and *real-world* household robotic manipulation settings. Our experiments evaluate LIV vision-language representations not only in their capacity as input state representations for language-conditioned behavior cloning of task policies, but also to directly ground language-based task specifications into visual state-based rewards for robot trajectory optimization, thereby stress-testing alignment across modalities. In many cases, the pre-trained LIV model, without ever seeing robots in its pre-training human video dataset, can *zero-shot* produce dense language-conditioned reward on unseen robot videos. Along another axis of evaluation, we assess both the "generic" representations pre-trained on large human video datasets as well as the specialized representations fine-tuned on in-domain robot data. Our results comparing to several representative recent works from the three distinct categories of pre-training, fine-tuning, and reward learning, confirm the advantages of the LIV objective for joint vision-language representation and reward learning for control.

2. Related Work

Pre-trained Representations for Control. Our work is related to the literature on pre-training representations for control (Shah & Kumar, 2021; Cui et al., 2022; Parisi et al., 2022; Nair et al., 2022b; Xiao et al., 2022; Ma et al., 2022b; Fan et al., 2022; Majumdar et al., 2023). These works all seek to use pre-existing data, typically out-of-domain, to pre-train effective representations for downstream unseen robotic tasks. Conceptually, VIP (Ma et al., 2022b) is closest to LIV in learning an implicit value function as joint reward and representation, but VIP focuses only on visual pre-training. Likewise, Nair et al. (2022b) uses a language alignment loss (Nair et al., 2022a) with respect to a fixed language encoder (Sanh et al., 2019) to shape the visual representation temporally, but the learned representation itself is still uni-modal. In this context, our work is the first to propose a multi-modal vision-language objective that is simultaneously suitable for pre-training, fine-tuning, and reward-learning for language-conditioned robotic control. A concurrent work (Karamcheti et al., 2023) has proposed joint pixel and language reconstruction as a pre-training objective and focuses more on high-level robotics reasoning tasks, such as grasp affordance prediction and referring expression grounding.

Fine-Tuning Pre-Trained Representations. Several recent works study how to adapt pre-trained representations for downstream tasks (Kumar et al., 2022; Wortsman et al., 2022; Ilharco et al., 2022; Lee et al., 2022; Kirichenko et al., 2022; Goyal et al., 2022; Dong et al., 2022), motivated by the emergence of large pre-trained models (Radford et al., 2021; Brown et al., 2020) capable of zero-shot transfer. Most closely related to our work are a few concurrent works that find using the CLIP objective to fine-tune CLIP is more effective than alternative fine-tuning approaches (Goyal et al., 2022; Dong et al., 2022). However, in the setting of vision-language robotic control, we find CLIP to be a sub-optimal fine-tuning choice, even for the original CLIP model itself, due to the static image-text alignment nature of the CLIP objective that ignores the temporal structure in video data and discards non-goal frames in videos when fine-tuning image-level visual representation. In contrast, LIV induces a natural synergy between VIP and CLIP that does not require any hyperparameter tuning and can fine-tuning various pre-trained vision-language representations for downstream robotic control.

Language-Conditioned Robotic Manipulation. There has been a surge of interest in language-conditioned vision-based robotic manipulation systems (Lynch & Sermanet, 2020; Stepputtis et al., 2020; Ahn et al., 2022; Jang et al.,

2022; Lynch et al., 2022; Shridhar et al., 2022a; Brohan et al., 2022; Shridhar et al., 2022b; Guhur et al., 2022; Liu et al., 2022; Mees et al., 2022b). While several works have considered policy learning on top of pre-trained vision-language representations (Shridhar et al., 2022a; Liu et al., 2022; Mees et al., 2022a), they do not consider how a better representation can be learned in the first place by leveraging large-scale out-of-domain text-annotated video data. Our work is the first to study how to pre-train new control-centric vision-language representations that surpass existing representations such as CLIP for language-conditioned visuomotor robotic control tasks.

Further, our solution, LIV, also affords flexibility in policy synthesis algorithms. Most existing works in this area focus on language-conditioned behavior cloning (LCBC) (Lynch & Sermanet, 2020; Stepputtis et al., 2020). This paradigm demands the expensive collection and text labeling of demonstration data, which can take months to complete (Jang et al., 2022; Lynch et al., 2022; Brohan et al., 2022). In contrast, not only is LIV an effective pre-trained representation for LCBC, it can also be used as a language-conditioned visual reward model that supports autonomous skill acquisition via reinforcement learning (Goyal et al., 2021; Nair et al., 2022a; Mahmoudieh et al., 2022). Our experiments show that LIV outperforms prior state-of-art language-conditioned reward models (Nair et al., 2022a;b) in model-based planning settings.

3. Preliminaries & Problem Setting

In this section, we review the VIP algorithm and describe our problem setting.

Value Implicit Pre-Training (VIP). VIP (Ma et al., 2022b) learns the optimal goal-conditioned value function via the dual goal-conditioned RL formulation (Ma et al., 2022a;c):

$$\mathcal{L}(\phi) = \mathbb{E}_{p(g)}[(1-\gamma)\mathbb{E}_{\mu_0(o;g)}\left[-\mathcal{S}(\phi(o);\phi(g))\right] + \log \mathbb{E}_{(o,o';g)\sim D}\left[\exp\left(\mathcal{S}(\phi(o);\phi(g)) + 1 - \gamma\mathcal{S}(\phi(o');\phi(g))\right)\right]], \tag{1}$$

where $\mu_0(o;g)$ is the distribution of initial frame conditioned on the goal frame g and $D(o,o';g)$ is the goal-conditioned distribution of two successive intermediate frames. In VIP, the value function is implicitly parameterized as a similarity metric (e.g., L_2 distance) in the embedding space $V(o;g) := \mathcal{S}(\phi(o);\phi(g))$, making it both a representation learning and a reward learning algorithm. Since it does not depend on actions, VIP can be pre-trained on large-scale human video datasets. The resulting implicit value function serves the dual purposes of (1) visual representation for unseen robot tasks, and (2) goal-conditioned dense reward specification. In particular, given a goal g, VIP assigns a potential-based reward at each time t:

$$R(o_t, o_{t+1}; g) := \mathcal{S}(\phi(o_{t+1});\phi(g)) - \mathcal{S}(\phi(o_t);\phi(g)) \tag{2}$$

Vision-Language Representation Pre-Training for Control. We assume access to a dataset of language-annotated videos $D = \{v_i := (o_1^i, ..., g^i; l^i)\}_{i=1}^N$, where each $o \in O := \mathbb{R}^{H\times W\times 3}$ is a raw RGB image, g^i the last frame of the video, and l^i is the textual annotation associated with v^i, describing the video outcome in g^i. As the video dataset can be out-of-domain with respect to our robot agent (e.g., human videos), we do not assume access to action labels. Datasets of this nature, such as human daily activity videos (Damen et al., 2018; Miech et al., 2019; Grauman et al., 2022), are readily available for research use. A pre-training algorithm $\mathcal{A}$ ingests this training data and outputs vision-language encoders $(\phi, \psi) := \mathcal{A}(D)$, where the vision encoder $\phi : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^K$ and the language encoder $\psi : L \to \mathbb{R}^K$, where L is the space of natural strings, each map to the same K-dimensional vision-language representation space.

A standard way to learn a vision-language representation is by learning a cross-modal joint-embedding (LeCun, 2022) that aligns the modalities via contrastive learning. Specifically, the two modalities are semantically aligned by minimizing the InfoNCE objective (Oord et al., 2018):

$$\mathcal{L}_{\text{InfoNCE}}(\phi,\psi) = \mathbb{E}_{p(o,l)}\left[-\log \frac{e^{\mathcal{S}(\phi(o);\psi(l))}}{\mathbb{E}_{D(o')}\left[e^{\mathcal{S}(\phi(o');\psi(l))}\right]}\right], \tag{3}$$

where $\mathcal{S}$ is a choice of similarity metric. Intuitively, this objective aims to attract the representations of matching image-text pairs (o, l), while repelling mismatching pairs. Many state-of-art vision-language models (Radford et al., 2021; Jia et al., 2021; Li et al., 2022) train with this InfoNCE objective at scale to deliver strong zero-shot performance on a myriad of vision-language tasks.

4. LIV: Language-Image Value Learning

4.1. Algorithm

We begin by extending the VIP framework to multi-modal goal specifications. This is straightforward given the goal-conditioned nature of Eq. (1), since we can simply replace encoded image goal $\phi(g)$ with encoded text goal $\psi(l)$ and optimize for a *multi-modal* VIP objective:

$$\begin{aligned}\mathcal{L}(\phi,\psi) = &\\ &+ \mathbb{E}_{p(g)}[(1-\gamma)\mathbb{E}_{\mu_0(o;g)}[-\mathcal{S}(\phi(o);\phi(g))]\\ &\underbrace{+ \log \mathbb{E}_{(o,o';g)\sim D}\left[\exp\left(\mathcal{S}(\phi(o);\phi(g)) + 1 - \gamma\mathcal{S}(\phi(o');\phi(g))\right)\right]]}_{\text{VIP-I}}\\ &+ \mathbb{E}_{p(l)}[(1-\gamma)\mathbb{E}_{\mu_0(o;l)}\left[-\mathcal{S}(\phi(o);\psi(l))\right]\\ &\underbrace{+ \log \mathbb{E}_{(o,o';l)\sim D}\left[\exp\left(\mathcal{S}(\phi(o);\psi(l)) + 1 - \gamma\mathcal{S}(\phi(o');\psi(l))\right)\right]]}_{\text{VIP-L}}\end{aligned} \tag{4}$$

As shown, this objective consists of two independent components; VIP-I (Image) encourages the representation to

encode an *image* goal-conditioned value function, and likewise, VIP-L (Language) for *language* goal.

At first glance, this objective does not appear to be directly optimizing for semantic alignment between goals in the two modalities, as the respective modality-specific VIP objectives are independently optimized. Without alignment, semantically equivalent goals in the respective modality may actually be distant in the representation space. This is undesirable for reward specification, which requires visual grounding of linguistic task descriptions. Intriguingly, in the next section, we show that such semantic alignment is in fact automatically achieved from optimizing Eq. (4).

4.2. Theoretical Analysis

Now, we show that Eq. (4) in fact naturally optimizes semantic alignment between multi-modal goals with a simple data augmentation applied to the training videos. Specifically, if we were to consider a *degenerate* distribution of videos, i.e., videos consisting of solely static text-aligned frames $v = ((o, o); l)$, we recover InfoNCE from VIP-L:

Proposition 1. *Let the video distribution consist of solely degenerate videos of repeated frames that align with the text annotation, $D := \{v := ((g, g; l))\}$. Then, the VIP-L objective is equivalent to the InfoNCE objective up to a constant:*

$$\mathcal{L}_{\text{VIP-L}}(\phi, \psi) = \mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[e^{(1-\gamma)\mathcal{S}(\phi(g');\psi(l))}\right]}\right] + 1, \tag{5}$$

where $p(g, l)$ is the distribution of goal frame and text pair.

The proof is in Appendix A. This result, though simple to derive, has several important implications. First, note that Eq. (5) is precisely what CLIP (Radford et al., 2021) optimizes (Eq. (3), modulo the constant factor) by contrastively learning similarity between matching image-text pairs. The fact that this objective can be obtained by optimizing VIP-L with a degenerate video distribution suggests that VIP-L is a natural generalization of the InfoNCE objective to the decision making setting, where the data is temporal. In practice, as we will show, this degenerate video distribution can be trivially obtained by augmenting any existing annotated video in the dataset by repeating the last frame.

This finding also directly suggests a method for *fine-tuning* pre-trained contrastive vision-language models (e.g., CLIP) for control: use LIV on in-domain labeled videos such as text-annotated robot demonstrations. Several concurrent works (Goyal et al., 2022; Dong et al., 2022) have suggested that fine-tuning a pre-trained model using the same objective (in particular, using CLIP objective to fine-tune CLIP model) can be more effective than fine-tuning using the downstream task objective. When working with CLIP-like pre-trained contrastive joint-embeddings, it is then natural to fine-tune them for control with the LIV objective, which is but a natural extension of CLIP that exploits sequential, goal-directed video data.

As we show in our experiments, using the LIV objective to fine-tune the pre-trained CLIP model is far more effective than using the CLIP objective. CLIP fine-tuning aligns the last frame in the video to its text description but fails to leverage earlier frames from the same video sequence.

4.3. Implementation

Based on the analysis above, we see that, despite initial appearances, Eq. (4) does in fact naturally induce semantic alignment between visual and language goals. In particular, it is implicitly optimizing for a pathway that connects the two modalities via mutual information maximization. Given this pathway that makes goals interchangeable across modalities, the final LIV objective optimizes the VIP objective in just one modality in conjunction with the vision-language InfoNCE objective in Eq. (5), which we find to be a simple yet effective objective:

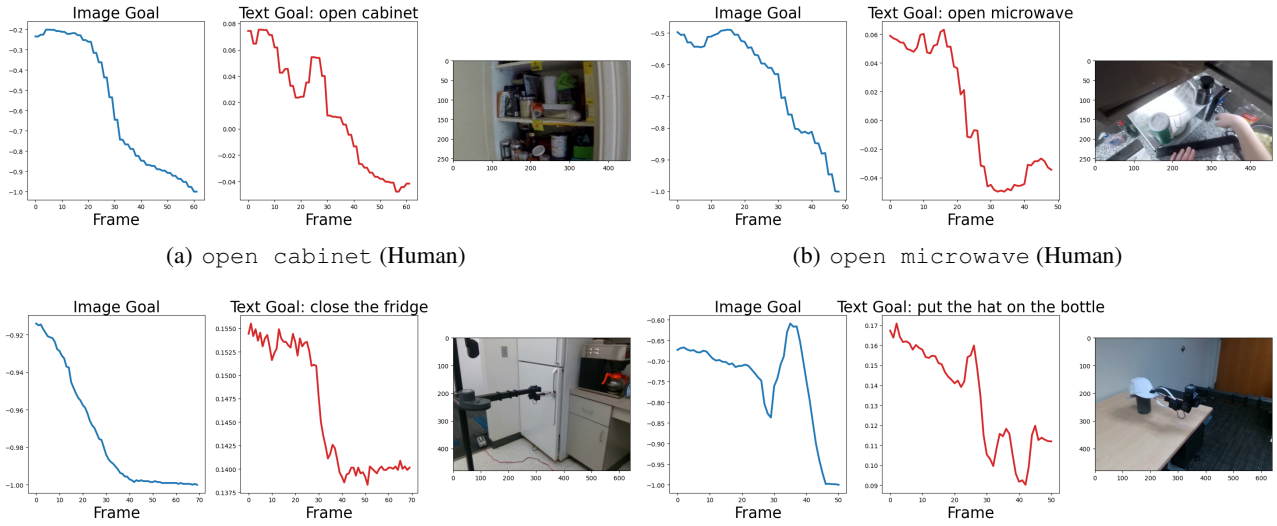
$$\begin{aligned}
&\mathcal{L}_{\text{LIV}}(\phi, \psi) = \\
&+ \mathbb{E}_{p(g)}[(1-\gamma)\mathbb{E}_{\mu_0(o;g)}[-\mathcal{S}(\phi(o);\phi(g))] \\
&\underbrace{+ \log \mathbb{E}_{(o,o';g)\sim D}\left[\exp\left(\mathcal{S}(\phi(o);\phi(g)) + 1 - \gamma\mathcal{S}(\phi(o');\phi(g))\right)\right]]}_{\text{VIP-I}} \\
&\underbrace{+\mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[e^{(1-\gamma)\mathcal{S}(\phi(g');\psi(l))}\right]}\right]}_{\text{InfoNCE}},
\end{aligned} \tag{6}$$

We use a γ-weighted cosine similarity metric for $\mathcal{S}(\phi(\cdot), \psi(\cdot)) := \frac{1}{1-\gamma} CS(\phi(\cdot), \psi(\cdot))$ so it represents a valid value function. Pseudocode is presented in Algorithm 1. In each gradient step, a minibatch of video sub-clip consisting of initial, intermediate, and final frames are sampled along with the corresponding text annotations. These samples are used to estimate the VIP-I and InfoNCE losses, which then update the vision-language architecture via back-propagation.

LIV Pre-Training. We have shown above that the LIV objective subsumes CLIP-style contrastive objectives. In implementing LIV, we stay close to CLIP architecture and design choices to allow fair comparison to pre-trained CLIP with ResNet50 (He et al., 2016) vision backbone. Initialized with CLIP weights, we pre-train LIV on EpicKitchen (Damen et al., 2018), a text-annotated ego-centric video dataset of humans completing tasks in diverse household kitchens; this dataset consists of 90k video segments, totalling 20M frames and 20k unique text annotations, and offers diverse camera views and action-centric videos, making it an ideal choice for vision-language pre-training. See Appendix B for more pre-training details.

Algorithm 1 Language-Image Value Learning (LIV)

Require: Offline text-annotated (human) videos $D = \{(o_1^i, ..., g^i; l^i)\}_{i=1}^N$, vision-language architecture (ϕ, ψ)
for number of training iterations **do**
  Sample sub-trajectories $\{o_t^i, ..., o_k^i, o_{k+1}^i, ..., g^i; l^i\}_{i=1}^B \sim D, t \in [1, h_i - 1], t \leq k < h_i, \forall i$
  $\mathcal{L}_{\text{VIP-I}}(\phi) := \frac{1-\gamma}{B} \sum_{i=1}^B \left[-\mathcal{S}(\phi(o_t^i); \phi(g^i))\right] + \log \frac{1}{B} \sum_{i=1}^B \exp\left[\mathcal{S}(\phi(o_k^i); \phi(g^i)) + 1 - \gamma \mathcal{S}(\phi(o_{k+1}^i); \phi(g^i))\right]$
  $\mathcal{L}_{\text{InfoNCE}}(\phi, \psi) := \frac{1-\gamma}{B} \sum_{i=1}^B \left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g^i); \psi(l^i))}}{\frac{1}{B}\sum_{j=1}^B \left[e^{(1-\gamma)\mathcal{S}(\phi(g^j); \psi(l^i))}\right]}\right]$
  Update (ϕ, ψ) using SGD: $(\phi, \psi) \leftarrow (\phi, \psi) - \alpha \nabla (\mathcal{L}_{\text{VIP-I}}(\phi) + \mathcal{L}_{\text{InfoNCE}}(\phi, \psi))$
end for



(a) `open cabinet` (Human) (b) `open microwave` (Human)

(c) `close the fridge` (Robot) (d) `put the hat on the bottle` (Robot)

Figure 2. **LIV Zero-Shot Multi-Modal Cost on Unseen Human and Robot Videos.** The y-axis is the real-valued negative cosine similarities computed in the LIV embedding space.

5. Experiments

Our experiments aim to answer the following questions:

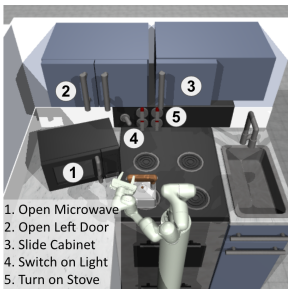
1. Can LIV produce multi-modal goal-conditioned rewards?
2. Does pre-trained LIV enable effective vision-language representations for control?
3. Can LIV successfully fine-tune pre-existing vision-language models?

To assess LIV's reward learning capability, we assess whether the pre-trained LIV can zero-shot provide multi-modal rewards for unseen text-annotated human and robot videos (Section 5.1) and use its reward function for model-based planning to solve language-specified tasks (Section 5.5). We evaluate LIV's effectiveness for pre-training (Section 5.3) and fine-tuning (Section 5.4) by using the resulting representations as the vision-language backbone in language-conditioned imitation learning (LCBC) in both simulations and a real robot platform. LIV model and training code are released: github.com/penn-pal-lab/LIV. Our qualitative results, including animated reward curves and real-robot videos, are best viewed on our project website: penn-pal-lab.github.io/LIV

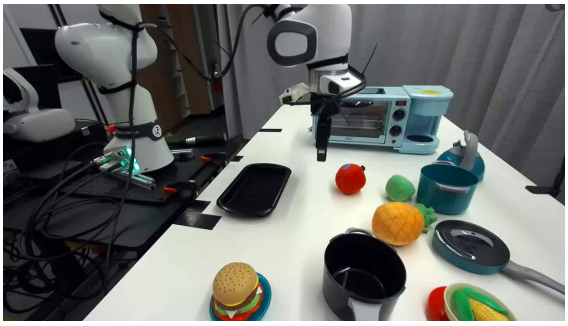
5.1. Pre-Trained LIV as Zero-Shot Reward

Recall that LIV objective encourages similarities in the embedding space to encode multi-modal goal-conditioned value functions. Intuitively, on videos depicting direct progress towards a goal, the distances (negative cosine similarities $-\mathcal{S}(\phi(o), \phi(g))$ or $-\mathcal{S}(\phi(o), \psi(l))$) between video frames o and image or language goals g or l, should steadily fall over time, reflecting progression towards those goals. Before using the embeddings for policy learning, we therefore first ask: to what extent does LIV pre-training achieve this property, and does it hold on new, unseen domains with robots? For unseen, goal-directed human videos from the test split of EpicKitchen, Figure 2 (a&b) plot frame distances from image and language goals, validating that this property indeed holds within the training distribution. However, to use LIV for training robots, we are interested in generalization to robot videos. This is challenging, since

(a) MetaWorld



(b) FrankaKitchen



(c) RealRobot

Environment	Train Tasks	Test Tasks	Horizon	Dataset Size	Dataset Type
MetaWorld	1000	6	20	1M	Random
FrankaKitchen	5	5	50	12.5K	Machine Demos
RealRobot	9	9	100	90k	Human Teleoperation

Figure 3. Multi-Task Vision-Language Environments.

these pre-trained LIV models have never encountered any robot data. Yet, as Figure 2 (c&d) show, LIV embedding distances are still informative of task progress, indicating transferability to unseen viewpoints and embodiments. Interestingly, in Figure 2 (d), we observe a bump in the middle of both cost curves; upon examination, this corresponds to a portion of the video where the robot lifts the hat unnecessarily high for the task of putting the hat on the bottle. In other words, the cost bump is indicative of sub-optimal actions.

In Appendix G.2, we include more examples, such as untrimmed robot videos where LIV rewards can effectively distinguish the opposite actions (e.g., `open the fridge` and `close the fridge`) that both appear in the video, and some failure cases. These results are promising; the fact that LIV embedding distances track task value functions means that we can use them to assign dense rewards (Eq. (2)) based purely on image or language goal specifications. Throughout these results, the distance plots are less smooth with respect to the *language* goal than the *image* goal, due to the language grounding gap; we show later (Section 5.5) that this can be effectively resolved with in-domain LIV fine-tuning. Finally, we also plot these distance progression plots with CLIP in Appendix G.1 and find that LIV's zero-shot reward capability is largely absent in CLIP. In the rest of our experiments, we present several ways in which the LIV pre-trained model and training objective can aid in language-conditioned robotic manipulation.

5.2. Policy Learning Environments

We consider three multi-task language conditioned visual manipulation environments, spanning two simulation and one real robot setup. The two simulated environments extend the MetaWorld (Yu et al., 2020) and FrankaKitchen (Gupta et al., 2019) benchmarks. The MetaWorld benchmark is taken from Nair et al. (2022a), which has also released a dataset consisting of 1M transitions collected via random actions for policy learning; the trajectories are labeled with task descriptions based on true environment state. The FrankaKitchen benchmark takes existing tasks supported in the environment but makes them specified via fixed language descriptions; we use the tasks and the dataset from Nair et al. (2022b).

The real-world environment (referred to as RealRobot) consists of a table-top toy kitchen setup, in which a Franka robot is tasked with placing various fruits, {`apple`, `pear`, `pineapple`} in various containers, {`tray`, `black pot`, `green pot`} in the scene given a sentence task description (e.g., `apple in black pot`). Unlike prior works that have considered simplified action space (Shridhar et al., 2022a) or reduced action frequency (Brohan et al., 2022; Mees et al., 2022a), we use 6-DOF end-effector displacement as the action space with 15Hz control. For each task, we collect 100 trajectories using human teleoperation with the fruits randomly initialized in the center workspace of the table for each trajectory. This environment is more challenging than the simulated ones because it requires grounding language to fine-grained spatial understanding of the scene in order to pick up the correct fruit and place it into the correct container. The environments and associated tasks are illustrated in Figure 3; for all three environments, the visual observation consists of the 3rd-person view shown in the figure. For RealRobot, we additionally include a robot wrist view. See Appendix C for more details on these environments and datasets.

5.3. Pre-Trained LIV as Representation

Pre-trained on diverse annotated human videos accomplishing daily household tasks, we posit that LIV can serve as an effective multi-modal representation for language-conditioned robotics manipulation.

Baselines. We compare against **CLIP** (Radford et al., 2021), a state-of-art vision-language representation that has seen wide adoption in various robotics tasks (Shridhar et al., 2022a; Cui et al., 2022; Khandelwal et al., 2022; Tam et al., 2022); as LIV is trained using the CLIP architecture and

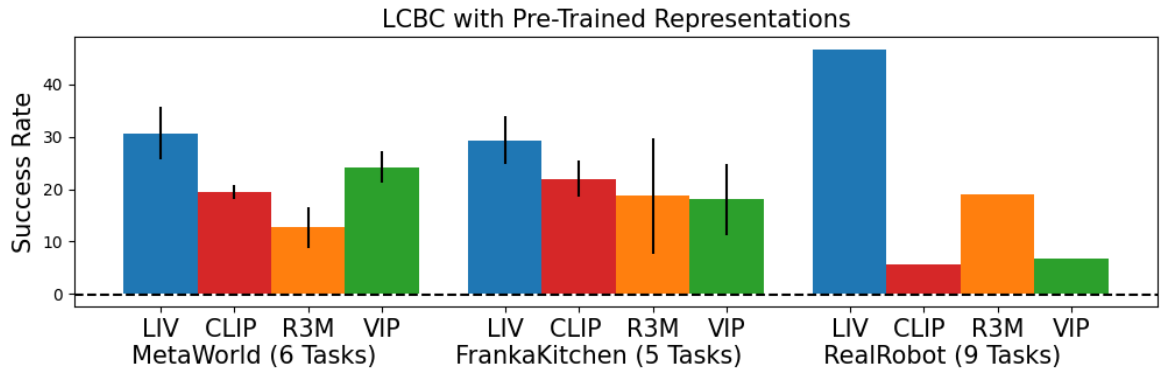


Figure 4. **Pre-Trained Representations for Language-Conditioned Behavior Cloning:** LIV achieves the highest average success rates across three distinct environments.

initialized with CLIP weights, this is the closest comparison. We also compare against **R3M** (Nair et al., 2022b) and **VIP** (Ma et al., 2022b), two state-of-art pre-trained visual representations. While unimodal, both are strong baselines; they are pre-trained on ego-centric videos similar to EpicKitchen (Grauman et al., 2022) and employ similar vision architecture (ResNet50) as LIV. We adapt them to the vision-language setting by coupling them with a pre-trained DistilBERT encoder (Sanh et al., 2019) to process language input. We note that R3M does employ this very same model for shaping its visual representation during training, making DistilBERT a natural design choice.

Policy Learning and Evaluation. Our experimental protocols closely follow prior works on evaluating pre-trained representations for robotic manipulation both in simulation (Nair et al., 2022b; Xiao et al., 2022) and on a real robot platform (Ma et al., 2022b). At a high level, we perform language-conditioned behavior cloning (LCBC), where a single multi-task policy, which takes concatenated current observation embedding and language task embedding as input, is trained for all tasks within an environment using the given environment dataset. The representations are kept frozen during policy learning, and we employ a simple MLP architecture on top of the pre-trained representations for the policy network. For the simulation environments, as in (Nair et al., 2022b), we report the success rate of the best training checkpoints on 20 evaluation rollouts per task. For the real environment, we evaluate only the final checkpoint on 10 evaluation rollouts per task due to the cost of real-world policy evaluation (Mandlekar et al., 2021). For each backbone representation, we train policies using 3 random seeds in simulation and report the mean and the standard deviation of the success rates over all evaluation episodes; on RealRobot, we train one policy per backbone representation. See Appendix D for additional details on training hyperparameters.

Results. Full results are reported in Figure 4 (full numeric results in Appendix D). As shown, our pre-trained LIV model, without any in-domain fine-tuning, performs best in all environments. In particular, LIV gains are largest on RealRobot, which is as expected since it is pre-trained on real data. R3M and VIP are also both pre-trained on human videos, yet they do not achieve the same level of performance as LIV, suggesting the importance of a proper vision-language pre-training objective for control that cannot be substituted by ad-hoc combinations of vision-only pre-trained representations and language models. This finding holds true especially on the real-world environment. There, given the lack of any "shortcut" visual cues to infer the correct action (e.g., objects of interest always appear centrally in the image), an effective vision-language representation needs to facilitate learning language-grounded hand-eye coordination for effective policy learning. Indeed, in Figure 9, Appendix D.1, we visualize the BC training loss each policy realizes on RealRobot and find that the policy with LIV backbone achieves significantly lower training loss than baselines; this result indicates that the baselines underfit as they fail to distinguish the correct action based on the conditioning text description and consequently achieve much lower success rates. In Appendix D.1, we also include an additional experiment ablating language task encoding with one-hot encoding. There, we find that LIV provides the best contextual representation (Sodhani et al., 2021) that improves policy learning even for our simulation tasks that do not heavily rely on language for task disambiguation.

Building on these real-world results, in Appendix D.2, we further find that the LIV policy can even *zero-shot* generalize to long-horizon composite tasks consisting of chained atomic tasks. This requires out-of-distribution generalization to unseen starting configurations for the atomic tasks.

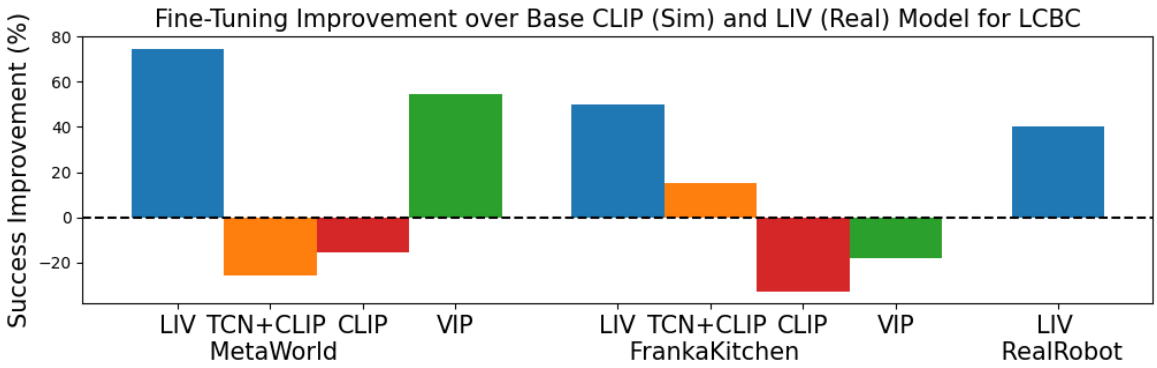


Figure 5. **LIV is an effective fine-tuner for pre-trained vision-language representations for robotic control.** Compared to the base model performance (dash line at 0), LIV fine-tuning consistently improves success rate by more than 40% in all environments.

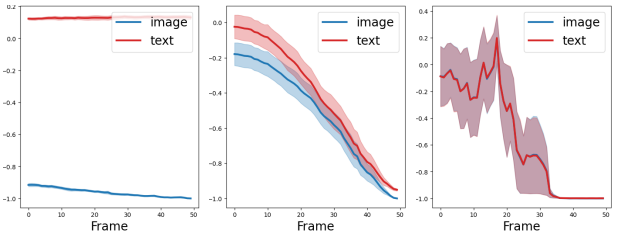
5.4. Fine-Tuning

Next, we show that the LIV objective can also be used to effectively fine-tune pre-trained vision-language models for downstream policy learning. Specifically, we first take the same in-domain task data as in Section 5.3 to fine-tune the pre-trained representations using the LIV objective (Algorithm 1) as well as several alternatives (see below). Then, as before, we freeze the fine-tuned representations and train policies on top using LCBC.

Baselines. The LIV objective can be understood as a principled combination of VIP and CLIP, so we consider a baseline that amounts to an alternative combination of a visual self-supervised learning (SSL) objective and the CLIP objective. In particular, we employ time contrastive learning (TCN) (Sermanet et al., 2018) because TCN is a SSL objective that also utilizes temporal information and has been incorporated in a prior work (Nair et al., 2022b). For this baseline, we sweep through $\alpha = \{0.1, 0.3, 0.5, 0.8\}$ and report the best value for weighing the two terms in the combined objective: $\alpha\text{TCN} + (1-\alpha)\text{CLIP}$. In addition, we consider using the **CLIP** InfoNCE objective (Eq. (5)) in isolation, as well as the **VIP-I** objective (Eq. (1)); these fine-tuning methods are ablations of LIV that focus only on semantic (CLIP) or temporal value (VIP) alignment. To isolate the impact of LIV fine-tuning independently of LIV pre-training, we use the pre-trained CLIP as the base model to be fine-tuned using LIV and the baselines; In Appendix E.2, we also present results on fine-tuning the pre-trained LIV. On RealRobot, due to the cost of real-world evaluation, we evaluate only LIV fine-tuning to see whether a strong fine-tuning algorithm vetted in simulation can remain effective in the real world. Likewise, we use the pre-trained LIV as the base model since CLIP LCBC performs poorly (Section 5.3).

Results. The relative performance improvements in percentages are shown in Figure 5. LIV fine-tuning substantially improves policy success rates on all three environments that vary significantly in terms of in-domain dataset size and quality. On RealRobot, we find that LIV fine-tuning remains effective and significantly improves the already strong base LIV model, validating its efficacy for real-world usage in both the pre-training and the fine-tuning stages. in Appendix G.3, we compare the embedding distance curves of LIV before and after LIV fine-tuning and show that LIV fine-tuning indeed acquires a smoother multi-modal representation. Qualitatively, we observe that the policy trained using fine-tuned LIV generates more coordinated and smoother grasping and placing motions that are critical for task success on RealRobot.

The weak performance of CLIP and VIP demonstrates that neither semantic or temporal-perception fine-tuning alone is sufficient for robotics manipulation; in the case of FrankaKitchen, due to the smaller dataset size, both ablations overfit and in fact decrease policy performance. TCN+CLIP similarly delivers mixed results, highlighting that LIV's fine-tuning capability cannot be easily obtained by combining another SSL objective, even one that considers temporal information, with the CLIP objective. Furthermore, given that TCN and CLIP do not bear natural connection to one another, we find their combination to be quite sensitive to the weighting parameter α; see Appendix E.2 for the numerical results for a comprehensive sweep over α for TCN+CLIP. In particular, whereas $\alpha = 0.5$ works best on FrankaKitchen, the same value leads to diverged training on MetaWorld; decreasing α to 0.1 prevents divergence on MetaWorld, but the resulting policy is even worse than that of the base CLIP model on MetaWorld and substantially worse than $\alpha = 0.5$ on FrankaKitchen. LIV does not suffer from this issue; our theoretical result in Section 4.2 suggests that the VIP and CLIP components in LIV should be weighed exactly in one-to-one, and indeed, we find that this ratio works well on all environments, requiring no hyperparameter tuning. In Appendix E.2, we further show that LIV fine-tuning is effective for different base models (e.g., LIV) and different in-domain dataset sizes, showcasing its versatility over a



(a) Pre-Trained LIV (b) LIV FT (c) CLIP FT

Figure 6. **LIV fine-tuning (FT) improves both temporal coherence and semantic alignment of the multi-modal cost curves;** in contrast, CLIP fine-tuning over-aggressively aligns the goal frame-text pair that damages representations of earlier frames.

large spectrum of domain specificities.

Qualitative Analysis. To better understand LIV fine-tuning's empirical gains, we visually compare the LIV models fine-tuned by LIV and CLIP objectives by overlaying their multi-modal cost curves (Section 5.1) over the in-domain demonstrations used for fine-tuning on FrankaKitchen. We use one demonstration from each distinct task and average over all curves to produce Figure 6; individual task reward curves as well as the plots for TCN+CLIP are included in Appendix G.4. As shown, the cost curves for the pre-trained LIV are far apart, illustrative of the real-to-sim domain gap that handicaps in-domain language grounding. LIV (Fig. 6(b)) naturally preserves a structured representation, in which the visual and text similarity curves are near-linear, monotonic, and converge to similar locations, suggesting that the representation has successfully constructed a latent value function that aligns goals in different modalities and preserve temporal coherence due to the recursive nature of value functions. In contrast, CLIP fine-tuning (Fig. 6(c)), as intended, maps the goal frame and the goal text to a near identical point in the representation space as the two curves almost perfectly overlap. However, the CLIP similarity scores of the intermediate frames exhibit uneven trends and high variance, indicating that the representation lacks temporal coherence possibly due to over-prioritizing semantic alignment. This temporal consistency is crucial for effective representation as it automatically prevents incorrect observation aliasing and preserves feature scale across time for effective policy learning (Ma et al., 2022b; Nair et al., 2022b). Yet, we have already shown that temporal consistency alone is not sufficient, evident from VIP-I's poor fine-tuning performance on FrankaKitchen due to overfitting. As such, LIV's unique effectivenss can be attributed to its principled combination of VIP and CLIP that enables the two objectives to regularize each other and work together in learning a structured, multi-modal latent value function. On our project website, we also animate these reward curves generated using successive fine-tuning checkpoints to qualitatively capture the learning dynamics of LIV fine-tuning.

Table 1. **Planning with Learned Reward:** LIV-EPIC is both the strongest zero-shot and adapted reward model.

Model	FrankaKitchen	MetaWorld
LIV (Pre-Trained)	1.3 $\pm$ 0.8	29.7 $\pm$ 4.7
LIV (LIV Fine-Tuned)	**20.0** $\pm$ 4.5	**55.2** $\pm$ 5.5
CLIP	0 $\pm$ 0.0	18.2 $\pm$ 4.4
CLIP (LIV Fine-Tuned)	15.2 $\pm$ 4.6	45.3 $\pm$ 2.5
CLIP (CLIP Fine-Tuned)	3.2 $\pm$ 0.9	30.7 $\pm$ 3.3
LOREL	9.6 $\pm$ 3.0	47.9 $\pm$ 3.2
LOREL (R3M Initialized)	16.8 $\pm$ 3.8	47.5 $\pm$ 12.7
R3M (Pre-Trained)	8.8* $\pm$ 2.7	18.3 $\pm$ 7.7
R3M (R3M Fine-Tuned)	16.1 $\pm$ 4.2	43.9 $\pm$ 3.2

5.5. LIV Reward-Based Behavior Synthesis

Finally, we demonstrate how to use LIV's dense goal-conditioned reward generation capability to directly acquire new language conditioned skills, in particular, via language-reward model predictive control.

Baselines. We compare to LOREL (Nair et al., 2022a), a state-of-art language-conditioned reward learning method that learns a classifier $f_\theta(o_0, o_t, l)$ for whether the progression from o_0 to o_t completes task description l. In addition, we compare to R3M (Nair et al., 2022b), which incorporates a similar language-progression score function trained via contrastive learning. As the original LOREL does not leverage pre-trained visual representations, we also consider a variant of LOREL initialized with R3M model weights to improve its performance. Similarly, to circumvent the out-of-domain language grounding problem for pre-trained R3M, we consider a variant where we fine-tune the pre-trained R3M using the R3M objective on the same in-domain data used for LIV fine-tuning.

Evaluations. We evaluate all reward models in a model-based planning setup, in which a trajectory optimizer synthesizes a sequence of actions to be executed in the true environment based on scores from the utilized reward function. For all LIV models (pre-trained and fine-tuned), we use the potential-based reward (Eq. 2) as in Ma et al. (2022b) using encoded language goal: On MetaWorld, we use the identical experimental setup as in Nair et al. (2022a), whereas on FrankaKitchen, we closely follow the experimental protocol of Ma et al. (2022b). See Appendix F for more details on our model-based planning experiments.

Results. The aggregated success rate over all test instances are reported in Table 1. LIV fine-tuning significantly im-

proves the success rate over the base pre-trained LIV and CLIP models, and the fine-tuned LIV achieves the best performance overall across both benchmarks; LIV fine-tuning's performance also steadily improves as the quality of the base model improves. LOREL and R3M models both perform adequately with the respective modifications we have introduced, but they still trail behind LIV; in Appendix F.2, we present additional analysis on these results. In conclusion, we have shown that LIV's implicit value learning paradigm gracefully combines both reward and representation learning in one unified objective and results in a flexible combined model that is highly effective across all evaluation settings.

6. Conclusion

We have presented the Language-Image Value Learning (LIV) algorithm. LIV is at once the first pre-training objective for control-oriented vision-language representations, a fine-tuning objective for domain-specific language grounding, and a language-conditioned task reward function. Trained on large generic human video datasets and fine-tuned on small robotics datasets, LIV outperforms state-of-the-art approaches in each of three distinct evaluation settings and successfully operates on real-world robotic tasks.

Acknowledgements

This work was supported in part by ONR grant number N00014-22-1-2677 and gift funding from NEC Laboratories.

Author Contributions

YJM conceived the project idea, implemented models and experiments, and wrote the paper. WL and VS assisted on the real-world experiments. VK, AZ, OB, and DJ provided feedback on the project and edited the writing.

References

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

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

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

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

Cui, Y., Niekum, S., Gupta, A., Kumar, V., and Rajeswaran, A. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for Dynamics and Control Conference*, pp. 893–905. PMLR, 2022.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–736, 2018.

Dong, X., Bao, J., Zhang, T., Chen, D., Gu, S., Zhang, W., Yuan, L., Chen, D., Wen, F., and Yu, N. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022.

Goyal, P., Niekum, S., and Mooney, R. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. In *Conference on Robot Learning*, pp. 485–497. PMLR, 2021.

Goyal, R., Kahou, S. E., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense, 2017.

Goyal, S., Kumar, A., Garg, S., Kolter, Z., and Raghunathan, A. Finetune like you pretrain: Improved finetuning of zero-shot vision models. *arXiv preprint arXiv:2212.00638*, 2022.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Guhur, P.-L., Chen, S., Garcia, R., Tapaswi, M., Laptev, I., and Schmid, C. Instruction-driven history-aware policies for robotic manipulations. *arXiv preprint arXiv:2209.04899*, 2022.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. *arXiv preprint arXiv:2208.05592*, 2022.

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Karamcheti, S., Nair, S., Chen, A. S., Kollar, T., Finn, C., Sadigh, D., and Liang, P. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

LeCun, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62, 2022.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

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

Liu, H., Lee, L., Lee, K., and Abbeel, P. Instruction-following agents with jointly pre-trained vision-language models. *arXiv preprint arXiv:2210.13431*, 2022.

Lynch, C. and Sermanet, P. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

Lynch, C., Wahid, A., Tompson, J., Ding, T., Betker, J., Baruch, R., Armstrong, T., and Florence, P. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

Ma, Y. J., Shen, A., Jayaraman, D., and Bastani, O. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv preprint arXiv:2202.02433*, 2022a.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022b.

Ma, Y. J., Yan, J., Jayaraman, D., and Bastani, O. How far i'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022c.

Mahmoudieh, P., Pathak, D., and Darrell, T. Zero-shot reward specification via grounded natural language. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14743–14752. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mahmoudieh22a.html.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Abbeel, P., Malik, J., et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Mees, O., Hermann, L., and Burgard, W. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022a.

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

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2630–2640, 2019.

Nair, S., Mitchell, E., Chen, K., Savarese, S., Finn, C., et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pp. 1303–1315. PMLR, 2022a.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022b.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

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

Rubinstein, R. Y. and Kroese, D. P. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, volume 133. Springer, 2004.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Shah, R. and Kumar, V. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022a.

Shridhar, M., Manuelli, L., and Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022b.

Sodhani, S., Zhang, A., and Pineau, J. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pp. 9767–9779. PMLR, 2021.

Stepputtis, S., Campbell, J., Phielipp, M., Lee, S., Baral, C., and Ben Amor, H. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

Tam, A. C., Rabinowitz, N. C., Lampinen, A. K., Roy, N. A., Chan, S. C., Strouse, D., Wang, J. X., Banino, A., and Hill, F. Semantic exploration from language abstractions and pretrained representations. *arXiv preprint arXiv:2204.05080*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Williams, G., Aldrich, A., and Theodorou, E. A. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Xiao, T., Radosavovic, I., Darrell, T., and Malik, J. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

A. Proof of Proposition 1

In this section, we provide a full proof of Proposition 1 in the main text. For ease of reading, we begin by reproducing the proposition.

Proposition. *Let the video distribution consist of solely degenerate videos of repeated frames that align with the text annotation, $D := \{v := ((g, g; l))\}$. Then, the VIP-L objective is equivalent to the InfoNCE objective up to a constant:*

$$\mathcal{L}_{\text{VIP-L}}(\phi, \psi) = \mathbb{E}_{p(g,l)} \left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')} \left[e^{(1-\gamma)\mathcal{S}(\phi(g');\psi(l))} \right]} \right] + 1, \tag{7}$$

where $p(g, l)$ is the distribution of goal frame and text pair.

Proof. We begin with the VIP-L objective:

$$\mathbb{E}_{p(l)}[(1-\gamma)\mathbb{E}_{\mu_0(o;l)}\left[-\mathcal{S}(\phi(o);\psi(l))\right] + \log \mathbb{E}_{(o,o';l)\sim D}\left[\exp\left(\mathcal{S}(\phi(o);\psi(l)) + 1 - \gamma\mathcal{S}(\phi(o');\psi(l))\right)\right]] \tag{8}$$

We can massage this expression as follow:

$$\mathbb{E}_{p(l)}[\mathbb{E}_{\mu_0(o;l)}\left[-(1-\gamma)\mathcal{S}(\phi(o);\psi(l))\right] + \log \mathbb{E}_{(o,o';l)\sim D}\left[\exp\left(1 + (1-\gamma)\mathcal{S}(\phi(o);\psi(l))\right)\right]], \tag{9}$$

assuming $o = o'$ in the log-sum-exp term.

Now, the joint distribution of language and initial-frame $p(l)\mu_0(o; l)$ reduces to the marginal distribution of goal-frame and text distribution $p(g, l)$ when the videos are just concatenations of the goal frames. Similarly, The language-conditioned distribution of successive intermediate frames $D(o, o'; l)$ reduces to the marginal distribution of goal frames $D(g')$ in the dataset. Plugging these substitution back into Equation (9) gives

$$\begin{aligned}
&\mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[\exp\left(1 + (1-\gamma)\mathcal{S}(\phi(g');\psi(l))\right)\right]}\right] \\
=&\mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[e \cdot \exp\left((1-\gamma)\mathcal{S}(\phi(g');\psi(l))\right)\right]}\right] \\
=&\mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[\exp\left((1-\gamma)\mathcal{S}(\phi(g');\psi(l))\right)\right]}\right] + 1 \\
=&\mathbb{E}_{p(g,l)}\left[-\log \frac{e^{(1-\gamma)\mathcal{S}(\phi(g);\psi(l))}}{\mathbb{E}_{D(g')}\left[e^{(1-\gamma)\mathcal{S}(\phi(g');\psi(l))}\right]}\right] + 1
\end{aligned} \tag{10}$$

□

B. LIV Model Details

We implement LIV using the open-sourced CLIP architecture[1] without modifications; we use the modified ResNet50 (He et al., 2016) from CLIP for the vision encoder, and the CLIP Transformer (Vaswani et al., 2017; Radford et al., 2019) architecture for the language encoder. The training hyperparameters used during the pre-training and fine-tuning stages are listed in Table 2. During pre-training, we also incorporate the VIP-L objective, which we find to produce better pre-trained LIV models in our preliminary experiments; we hypothesize that adding the explicit language-based VIP loss is instrumental in shaping the representation with semantic structure early on. During the fine-tuning stage, the same set of fine-tuning hyperparameters is used for fine-tuning CLIP as well as the ablation fine-tuning methods presented in Section 5.4.

Since LIV uses -1 as the constant fixed reward for all observations, the range of valid state value is $[\frac{-1}{1-\gamma}, 0]$; however, cosine similarity, as used in CLIP, has range of $[-1, 1]$. Thus, to be able to represent all possible values, we set $\mathcal{S}(\phi(\cdot), \psi(\cdot)) :=$

[1] https://github.com/openai/CLIP

(a) 3rd-Person View (b) Wrist View

Figure 7. RealRobot Visual Inputs.

$\frac{1}{1-\gamma}\texttt{CosineSimilarity}(\phi(\cdot), \psi(\cdot))$. Coincidentally, with this choice of $\mathcal{S}$, the InfoNCE objective in LIV reduces to precisely the InfoNCE objective used in CLIP.

We pre-train LIV on EpicKitchen (Damen et al., 2018). We use the `EPIC-KITCHENS-100` version of the data and only utilize the RGB frames and text annotations from the dataset; the default frame rate in the raw dataset is used. The pre-training takes place on a node of 8 NVIDIA V100 GPUs.

Table 2. VIP Architecture & Pre-Training Hyperparameters.

	Pre-Training	Fine-Tuning
Model Initialization	CLIP	{LIV-EPIC, CLIP, Random}
Optimizer	Adam (Kingma & Ba, 2014)	Adam
Gradient Steps	200000	10000
Batch Size	512	64
Learning Rate	0.00001	0.00001
Weight Recay	0.001	0.001
Discount Factor γ	0.98	{0.98, 0.96}
VIP-L objective	Yes	No

C. Environment Details

MetaWorld. The MetaWorld environment consists of a tabletop scene with a Sawyer robot that can interact with 4 objects, including a drawer, faucet, and two mugs distinguished by color. The dataset is collected by running a random policy for 50000 episodes with episode length 20; each episode is labeled with procedurally generated language descriptions that it achieves via computing pre-defined success criterion for each language-specified task. A single episode can solve many distinct tasks. In that case, the labeled description will be a concatenation of all atomic instructions that the episode has solved. The whole dataset contains 2311 unique descriptions, and the evaluation tests on 6 atomic instructions: `close drawer`, `open drawer`, `turn faucet right`, `turn faucet left`, `move black mug right`, `move the white mug left`.

FrankaKitchen The FrankaKitchen environment consists of a kitchen scene with a Franka robot that can interact with a variety of common household kitchen objects. We use the same 5-task split that was evaluated in Nair et al. (2022b) for visual imitation learning; the tasks as well as their language commands are listed in Table 3. For each task, we include 50 demonstrations, so the total size of the dataset is 250 episodes, where each episode is 50 environment steps long.

RealRobot. Our RealRobot environment consists of a toy kitchen tabletop scene with a Franka robot that can interact with 3 soft fruit toys (`apple, pineapple, pear`) and is tasked with placing the fruit objects into the correct container

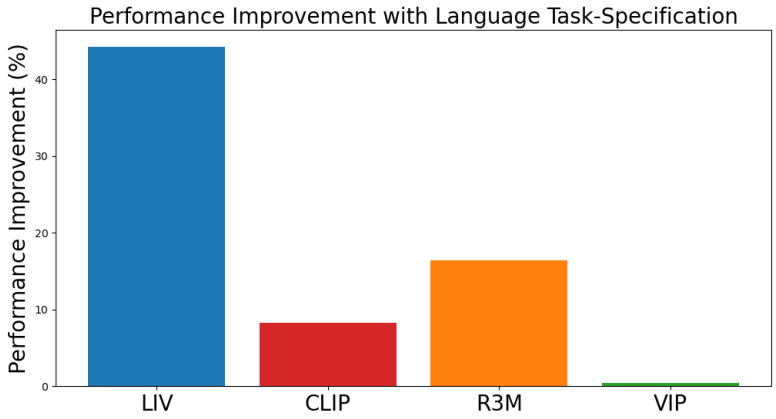


Figure 8. **Comparison Between Language vs. One-Hot Task Encoding:** LIV benefits the most from using language task-specification, resulting in near 45% gain in absolute success rates.

(`tray, black pot, green pot`) based on language input, creating 9 tasks in total. We use two cameras for visual inputs, one mounted on the robot gripper and one mounted on the side of the workspace table; see Figure 7 The dataset is collected via human teleoperation with 100 demonstrations per task.

Table 3. FrankaKitchen Task Mapping

Environment ID	Language Task
`kitchen_micro_open-v3`	`open microwave`
`kitchen_sdoor_open-v3`	`slide cabinet`
`kitchen_ldoor_open-v3`	`open left door`
`kitchen_knob1_on-v3`	`turn on stove`
`kitchen_light_on-v3`	`switch on light`

D. Language-Conditioned Imitation Learning with Pre-Trained Representations

We present the LCBC imitation learning hyperparameters in Table 4. Because the dataset size in MetaWorld is significantly larger, we use a larger MLP architecture with bigger batch size. For each distinct evaluation task, we rollout for 50 episodes and record the success rate.

Table 4. LCBC Hyperparameters.

	MetaWorld	FrankaKitchen	RealRobot
MLP Architecture	[1024, 1024, 1024]	[256, 256]	[256,256]
Non-Linear Activation	ReLU	ReLU	ReLU
Optimizer	Adam	Adam	Adam
Gradient Steps	200000	200000	1M
Batch Size	4096	32	64
Learning Rate	0.001	0.001	0.001
Proprioception	No	Yes	No
Augmentation	No	No	Yes

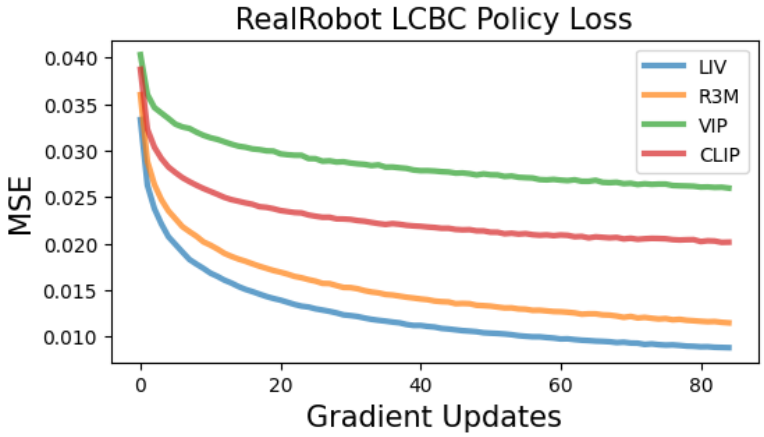


Figure 9. LIV LCBC achieves lower training loss and consequently achieves higher success rate on our RealRobot multi-task suite.

Table 5. **Pre-Trained Representations for Language-Conditioned Imitation Learning:** LIV-EPIC makes most effective use of its language embedding and realizes greater gains when using language-based contextual representation.

model	FrankaKitchen	MetaWorld
LIV-EPIC	**29.3** ± 4.6	**30.6** ± 5.0
LIV-EPIC (One-Hot)	17.6 ± 5.0	26.1 ± 5.5
CLIP	22 ± 3.5	19.4 ± 1.3
CLIP (One-Hot)	14.8 ± 0.7	28.6 ± 1.3
VIP (BERT)	18.0 ± 6.9	24.2 ± 3.0
VIP (One-Hot)	15.6 ± 6.2	28.3 ± 0.8
R3M (BERT)	18.7 ± 11.0	12.7 ± 3.9
R3M (One-Hot)	11.5 ± 1.9	18.1 ± 5.5

D.1. Additional Results

Given that our simulation environments require less sophisticated scene understanding from language, is language still helpful for policy learning? To probe this, we ablate each choice of pre-trained representations by replacing their language embeddings with one-hot task encoding to assess whether language provides contextual knowledge (Sodhani et al., 2021) that facilitates policy learning and generalization. The full numeric results comparing the policy performance with and without language task encoding are presented in Table 5; the relative improvement in percentage from using language task encoding is also displayed in Figure 8. as shown, LIV-EPIC consistently benefits from its jointly trained language representation. In particular, on both benchmarks, while LIV-EPIC and the strongest baselines (CLIP and VIP) all perform similarly with one-hot encoding, LIV-EPIC realizes much greater gain when language task-specification is used. In fact, using language task-specification *hurts* all baselines on MetaWorld. We hypothesize that this is due to the fact that the MetaWorld dataset contains many episodes whose annotations are long descriptions that consist of concatenation of shorter atomic instructions; for example, `"close drawer turn faucet right push black mug right"` is a valid annotation that contains 3 atonomic instructions. Therefore, the language embeddings from pure language model (e.g, DistilBERT) or language model trained from image-text only datasets (e.g., CLIP) may fail to disambiguate these instructions, leading to incorrect task aliasing that hampers policy learning. In contrast, one-hot encoding treats every description as distinct and does not have this aliasing problem. Together, these results highlight the challenges of adapting pure image-text representations and uni-modal visual representations to language-conditioned robotic control, thereby affirming LIV's unique effectiveness in vision-language pre-training for language-conditioned visuomotor control.

D.2. Zero-Shot Long-Horizon Task Generalization

On RealRobot, we test whether the LIV policy can solve long-horizon, composite tasks that require solving the atomic tasks in some specified sequences. We allow the policy to spend a fixed number of 100 control steps to solve the current

Table 6. **Partial Success Counts on Unseen Long-Horizon Composite Tasks.**

Task Sequence	LIV	R3M
{Pineapple in Tray, Apple in Tray, Pear in Tray}	5/10	0/10
{Pineapple in Black Pot, Apple in Tray, Pear in Green Pot}	3/10	1/10
{Pineapple in Tray, Apple in Green Pot, Pear in Black Pot}	5/10	1/10

task; then, the robot will be reset to its initial position, and the current task will transition to the next one in the specified sequence, which again allocates the policy 100 steps before transitioning. We note that the RealRobot dataset contains only demonstrations for the short-horizon, atomic tasks, and the demonstrations are never collected in configurations where some fruits have already been placed into some containers. As such, solving more than one task strictly requires the policy to generalize to unseen tabletop configuration, as success for an earlier task will change the scene into a novel configuration for the later tasks.

Given this, we report the number of trials out of 10 in which the policy is able achieve *partial success*, namely, solving at least 2 tasks out of the 3 tasks in the specified sequence. The comparison between LIV and R3M is shown in Table 6. We see that LIV policy is capable of generalizing to composite tasks that require generalization in both visual and semantic levels. In some cases, LIV policy can solve all 3 tasks. On our project website, we provide videos of the policy rollouts.

E. Fine-Tuning

E.1. Fine-Tuning on RealRobot

In our real world environment, since we use a 3rd-person and a wrist camera for policy learning, the LIV fine-tuning procedure differs slightly from our simulation experiments. In particular, since the wrist view provides a local view of the table that is significantly out-of-distribution with respect to the pre-training dataset, we elect to fine-tune using just the 3rd-person camera view. Then, we use the pre-trained LIV to process visual observations from the wrist camera view and the fine-tuned LIV to process visual observation from the 3rd-person camera view. The 3rd-person view embedding vectors are further passed through a learnable shallow MLP adapter as the feature scale of pre-trained and fine-tuned LIV varies. Then, the embeddings are conatenated with the language embedding from the fine-tuned LIV, and the final concatenated embedding vector is the input to the MLP policy network. For our real-world experiment, we also employ trajectory level frame random-cropping as data augmentation during policy learning, which we find to improve all methods.

E.2. Additional Results

In this section, we present additional experiments probing LIV's fine-tuning capability in simulation.

Can LIV effectively fine-tune base models of varying quality? We first study how LIV and its ablations (CLIP and VIP-I) fare with base models of varying quality. To this end, in addition to the base CLIP model we consider in the main text, we also include pre-trained LIV as well as a Random model (i.e., randomly initialized weights on the same CLIP architecture) as base models to be fine-tuned. The full results are presented in Table 7. We see that LIV fine-tuning is effective for all three model initializations, whereas the baseline ablations deliver mixed results. In particular, CLIP fine-tuning degrades performance in all cases except on the Random model in MetaWorld. This difference in dataset sizes also explains why VIP-I fine-tuning is reasonable on MetaWorld but very poor on FrankaKitchen, consistent with the findings in Ma et al. (2022b). As such, we have demonstrated that both terms in the LIV are indispensable for effective fine-tuning, and LIV is uniquely effective at fine-tuning vision-language models under varying pre-training objectives, pre-trained model qualities, and fine-tuning dataset sizes. The final LIV fine-tuned models perform better when they started from better pre-trained models, so that the best combined system simply uses the LIV objective for both phases, pre-training as well as fine-tuning.

How does in-domain dataset size affect LIV fine-tuning performance? Now, we probe whether LIV fine-tuning can work with even smaller in-domain robot datasets, representative of the few-shot settings that several recent works have studied (Nair et al., 2022b; Ma et al., 2022b). On the FrankaKitchen environment, instead of 50 trajectories per task, we repeat the same fine-tuning+LCBC experiment with just 10 and 25 trajectories per task. For these experiments, we include TCN+CLIP as a baseline, as it is the only baseline that actually improve policy performance with 50 trajectories per task for fine-tuning. The results are presented in Table 8. As shown, across different dataset sizes, LIV fine-tuning consistently

Table 7. **Fine-Tuning Vision-Language Representations:** LIV consistently improves the performance of pre-trained vision-language models regardless of their initial capabilities, the sizes and the qualities of the in-domain fine-tuning datasets.

MetaWorld				
Model/Method	Pre-Trained	**LIV**	CLIP	VIP-I
Random	20.6 ± 1.0	27.8 ± 4.1	30.8 ± 2.2	30.6 ± 3.5
CLIP	19.4 ± 1.3	33.9 ± 7.5	16.4 ± 4.3	30.0 ± 2.2
LIV-EPIC	30.6 ± 5.0	**35.8** ± 1.4	21.4 ± 5.7	20.3 ± 3.4
FrankaKitchen				
Model/Method	Pre-Trained	**LIV**	CLIP	VIP-I
Random	17.7 ± 3.9	19.2 ± 3.8	17.1 ± 2.2	3.2 ± 0.7
CLIP	22 ± 3.5	26.8 ± 4.9	14.0 ± 6.8	14.8 ± 1.3
LIV-EPIC	29.3 ± 4.6	**32.3** ± 5.8	15.1 ± 4.3	17.3 ± 6.6

Table 8. **Few-Shot Fine-Tuning Vision-Language Representations:** LIV fine-tuning can consistently improve over base model even with handful of demonstrations per task.

Number of Demos per Task	CLIP	CLIP (LIV fine-tuned)	CLIP (TCN+CLIP fine-tuned)
10	7.3 ± 1.2	**22.0** ± 6.0	13.3 ± 2.3
20	12.7 ± 1.5	**30.7** ± 1.7	23.3 ± 1.2
50	22.0 ± 3.5	**33.0** ± 1.4	25.3 ± 4.1

provides large gains, whereas TCN-CLIP is able to realize much smaller gains. Notably, in the most challenging setting of few-shot, 10 demos per task, LIV fine-tuning is able to match CLIP (no fine-tuning) with 50 demonstrations per task and improves the base model performance by more than 200%. These results demonstrate that LIV is fully capable of effective fine-tuning even in a very low data regime, showcasing its versatility and sample-efficiency.

The sensitivity of TCN+CLIP to objective weighting α. We present the full hyperparameter search over α for TCN+CLIP on both FrankaKitchen and MetaWorld and report the results in Table 9. As shown, while higher α values work better on FrankaKitchen, only the lowest α value of 0.1 results in a TCN-CLIP fine-tuned model that did not diverge during MetaWorld training; however, this converged TCN-CLIP-$\alpha 0.1$ model yields significantly lower downstream policy performance than LIV fine-tuning. Note that in practice, hyperparameter tuning for offline visual imitation learning/RL is fairly difficult because of the high computational footprint that limits the amount of hyperparameter tuning and, more fundamentally, the offline setting does not permit online rollouts for evaluation. As such, LIV's lack of dependency on tuning the balance between its SSL and CLIP objective is a significant advantage over these baselines in addition to its already superior performance.

F. Reward Learning

We describe our model-based planning experimental details. On MetaWorld, we use a cross-entropy Method (CEM) (Rubinstein & Kroese, 2004) planner to propose action sequences and employ the open-sourced SV2P (Babaeizadeh et al., 2017) visual dynamics model trained on the demonstration data to rollout the action sequences for optimization. On FrankaKitchen, as in Ma et al. (2022b), we use the ground-truth environment dynamics to for action rollouts and employ a model-path

Table 9. TCN+CLIP is sensitive to the relative weighting between the two components in the combined objective.

	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.8$	LIV fine-tuning
FrankaKitchen	11.3 ± 4.2	24.0 ± 1.0	25.3 ± 4.1	24.7 ± 4.2	33.0 ± 1.4
MetaWorld	14.4 ± 2.7	Diverged	Diverged	Diverged	35.8 ± 1.4

predictive integral (MPPI) (Williams et al., 2017) planner. On FrankaKitchen, due to the exploration challenge, we also warmstart the action search with a fixed open-loop sequence that brings the robot end-effector to the vicinity of the task object but does not perform the full commanded task.

F.1. Hyperparameters

On MetaWorld, we use the open-sourced implementation of Cross-Entropy Method (CEM) on this environment released by (Nair et al., 2022a). On FrankaKitchen, we follow the practice of Ma et al. (2022b) and use a publicly available implementation of MPPI[2] with the default hyperparameters.

Table 10. Model-Based Planning Hyperparameters.

	MetaWorld	FrankaKitchen
Planner	CEM	MPPI (Williams et al., 2017)
Planning Horizon	20	50
# Proposed Action Sequences	200	128
Optimization Iteration	1	1
Dynamics Model	SV2P trained on in-domain dataset	Ground truth simulation

F.2. Additional Results & Analysis

Additional analysis of Table 1. These results illustrate the orthogonal, if not competing, nature of reward and representation capability of a vision-language model. While CLIP (CLIP Fine-Tuned) exhibits improved reward learning performance over pre-trained CLIP, in Section 5.4, we have shown that CLIP fine-tuning leads to far inferior representation backbone for policy learning. We believe this is because CLIP fine-tuning aligns the last frames with text goals, and the model-based planners we use evaluate action sequences based on only the reward of the last observation. In contrast, in imitation learning, the representation needs to be well-behaved for all intermediate observations, which CLIP fine-tuning impairs, as shown in Figure 6. LOREL is a reward learning algorithm, yet it is prone to overfitting when trained from scratch on small in-domain data (i.e., FrankaKitchen) and is most performant when initialized with a pre-trained representation. Finally, though R3M training involves learning a language-reward predictor, this predictor is trained only in service of the core visual representation training. We find that this predictor is inferior to even purely in-domain trained LOREL on MetaWorld.

How does increasing planning budget affect model performance? To further assess the capability of the various learned reward models, we repeat the model-based planning experiment on MetaWorld by increasing the CEM optimization iteration from 1 to 3. The results are shown in Table 11. We see that almost all models that are trained or fine-tuned on the in-domain data see performance increase with the fine-tuned LIV-EPIC standing as the best model. However, the pre-trained models (LIV-EPIC, CLIP, R3M), with the exception of LIV-EPIC, see performance degradation, suggesting that their reward models are in fact exploited by the stronger optimizer. Finally, we observe that LIV with 1 CEM iteration already performs as well as LOREL with 3 CEM iterations, suggesting that LOREL is more prone to "false nagatives", i.e. assigning low scores to good trajectories. These results highlight both LIV's ability for zero-shot and fine-tuning reward model.

Why does R3M work well zero-shot on FrankaKitchen? Interestingly, we find R3M to perform well zero-shot on FrankaKitchen (Table 1, achieving $\approx 9\%$ success rate without any in-domain fine-tuning. Upon investigating this outcome however, we find that this result is an artifact of the specific way in which R3M was trained. In particular, R3M's pre-trained reward predictor has a bias for actions that induce visual change in the environment because it was pre-trained to output higher scores for frames that are farther apart in time, which typically correlate with larger visual changes in the scene. To confirm this, we repeat the same experiment on FrankaKitchen but this time with *random* language goals. The results are shown in Table 12. We see that R3M's performance remains surprisingly high, indicating that it does not depend at all on the language-based task specification. In contrast, other models' performance catastrophically decline. This indicates that R3M's language grounding is limited and often confuses completion of specific tasks with any indiscriminate visual changes in the environment. This finding is further supported by R3M's poor performance on the MetaWorld environment, in which random actions are enough to move the objects and induce large visual changes, and task completion requires more directed action, driven by more sophisticated language understanding. LIV-EPIC significantly outperforms R3M on MetaWorld and

[2] https://github.com/aravindr93/trajopt/blob/master/trajopt/algos/mppi.py

Table 11. LIV models consistently improve with increased planning budget; in contrast, baselines report mixed results.

Model	MetaWorld (CEM Iterations=1)	MetaWorld (CEM Iterations=3)
LIV-EPIC	29.7	34
LIV-EPIC (LIV Fine-Tuned)	**55.2**	**57.8**
CLIP	18.2	14.7
CLIP (LIV Fine-Tuned)	45.3	44.4
CLIP (CLIP Fine-Tuned)	30.7	34.4
LOREL	47.9	55.4
LOREL (R3M Initialized)	47.5	50.6
R3M	18.3	18.1
R3M (R3M Fine-Tuned)	43.9	50.8

Table 12. Performance Comparison Between Correct and Random Language Goals.

Model	Correct Goal	Random Goal
LIV-EPIC	1.3	1.0
LIV-EPIC (LIV Fine-Tuned)	**20.0**	0.0
LOREL	9.6	0.0
LOREL (R3M Initialized)	16.8	0.0
R3M	8.8	12.1
R3M (R3M Fine-Tuned)	16.1	0.0

is the best zero-shot reward model overall on this benchmark.

G. Representation Qualitative Results

In this section, we provide additional qualitative results on our pre-trained and fine-tuned models. For animated version of the qualitative reward curves, please visit our project website: penn-pal-lab.github.io/LIV

G.1. EpicKitchen (Real)

We first visualize pre-trained LIV-EPIC on representative seen and unseen EpicKitchen videos by plotting the embedding curves with respect to the image (final frame of the video) and the text goal. In both seen and unseen splits, the three videos have annotations `open cabinet`, `open door`, and `open microwave`, respectively. The results are in Figure 10 and 11. For comparison purpose, we include the results for the CLIP model in Figure 12 and 13.

G.2. HelloRobot (Real)

In Figure 14, we present additional examples where pre-trained LIV is able to capture language-conditioned progress in unseen robot videos; in Figure 15, we present the reward curves for CLIP on the same set of videos; as shown, CLIP's zero-shot language-reward is much more noisy. In Figure 16, we conduct the same reward curve analysis on untrimmed videos in which the robot completes a sequence of opposite actions in the same video (e.g., `open the fridge` and `close the fridge`; note that these results are best viewed on our project website. Since the image goal and the language goal semantically refer to opposite actions, where image goal specifies the action accomplished in the last frame and the language goal specified the action accomplished in the middle of the video, we see that LIV's reward curves exhibit inverted trend across two modalities. This demonstrates that LIV has the ability to detect fine-grained, action-induced object state changes in videos.

Finally, in Figure 17, we also present several failure examples, where LIV language rewards fail to capture language-based progression. There are several reasons why these failures may occur, such as network capacity, and the distribution shift presented in these videos with respect to camera viewpoint, embodiment, and language commands. Given LIV's

self-supervised nature, we are hopeful that LIV's zero-shot capability will only improve with more expressive network architecture and diverse datasets.

G.3. RealRobot (Real)

In Figure 18, 19, we present the reward curves for LIV (pre-trained) and LIV (LIV fine-tuned). As shown, LIV (pre-trained) produces reasonable visual reward progress but suffers from domain gap that renders its language reward progress ineffective. LIV (LIV fine-tuned) remedies this issue and smoothens the representation in both modalities.

G.4. FrankaKitchen (Sim)

In Figure 20, 21, 22, 23. we present the reward curves for LIV-EPIC, LIV-EPIC (LIV fine-tuned), LIV-EPIC (CLIP fine-tuned), and LIV-EPIC (TCN+CLIP fine-tuned) on the FrankaKitchen tasks, respectively. For each model, the first plot is the averaged reward curve for all 5 tasks, whereas the succeeding 5 plots are the task-specific reward curves. As shown, LIV-EPIC, without any in-domain fine-tuning, is able to competently capture visual progress but lacks language grounding to capture language goal progress. LIV fine-tuning captures fine-grained language-conditioned progression while simultaneously improving visual temporal alignment. CLIP fine-tuning over-aggressively aligns the representations of the last frame and the text goal and collapses intermediate representations. TCN+CLIP lacks temporal smoothness in the learned representation that is crucial for both vision-language representation for control (Section 5.3 and language-conditioned reward-specification (Section 5.5.

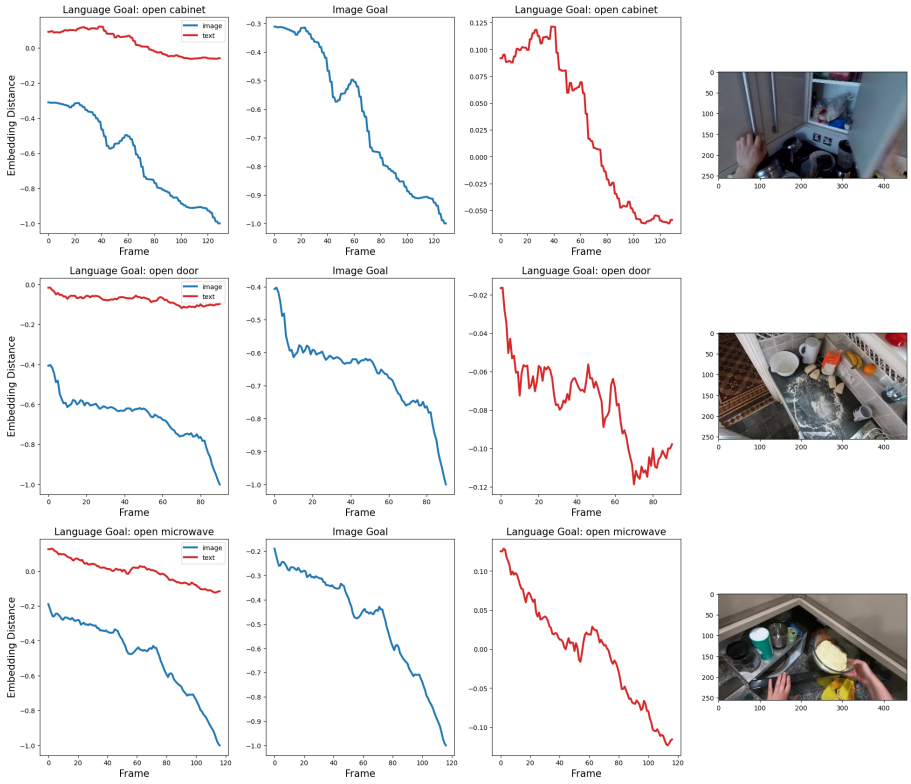


Figure 10. Pre-trained LIV-EPIC image and language goal reward curves on (seen) EpicKitchen videos.

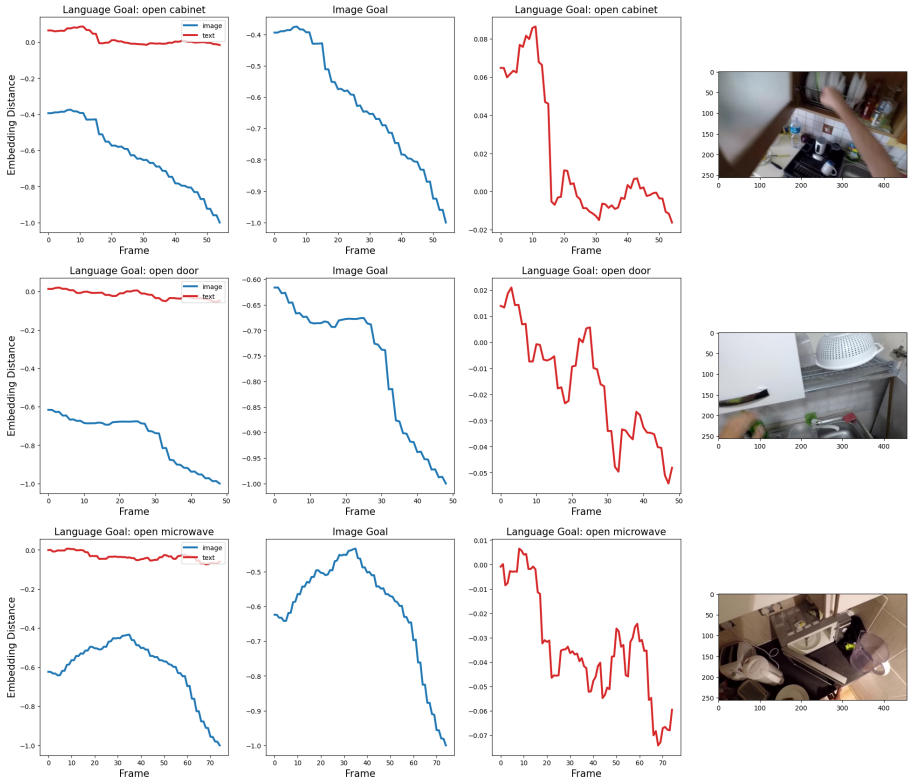


Figure 11. Pre-trained LIV-EPIC image and language goal reward curves on (unseen) EpicKitchen videos.

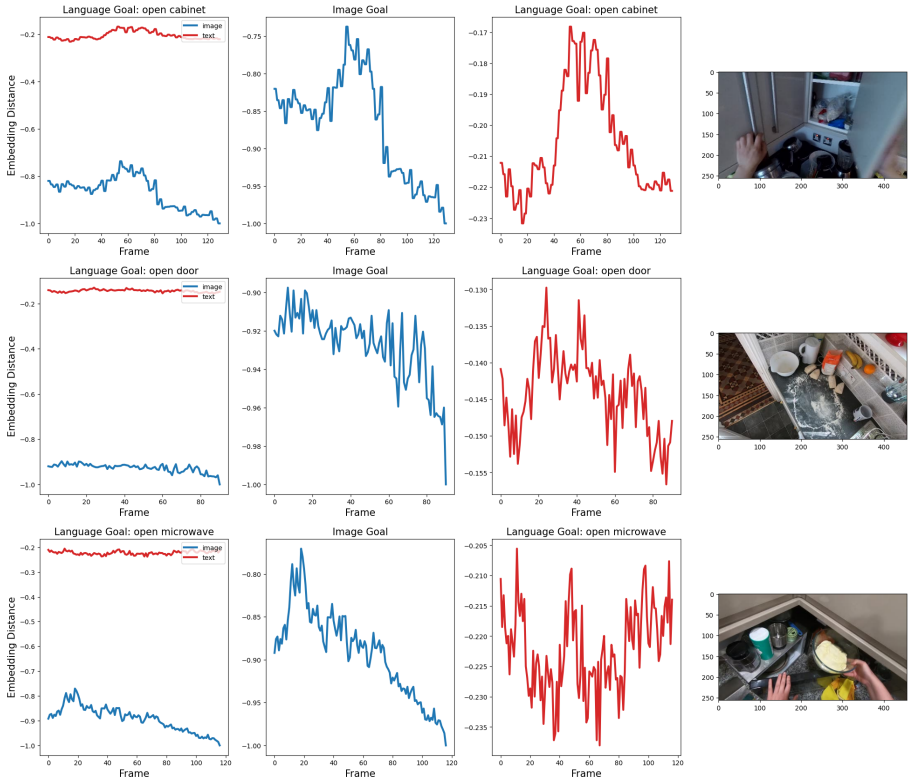


Figure 12. CLIP image and language goal reward curves on (seen) EpicKitchen (videos).

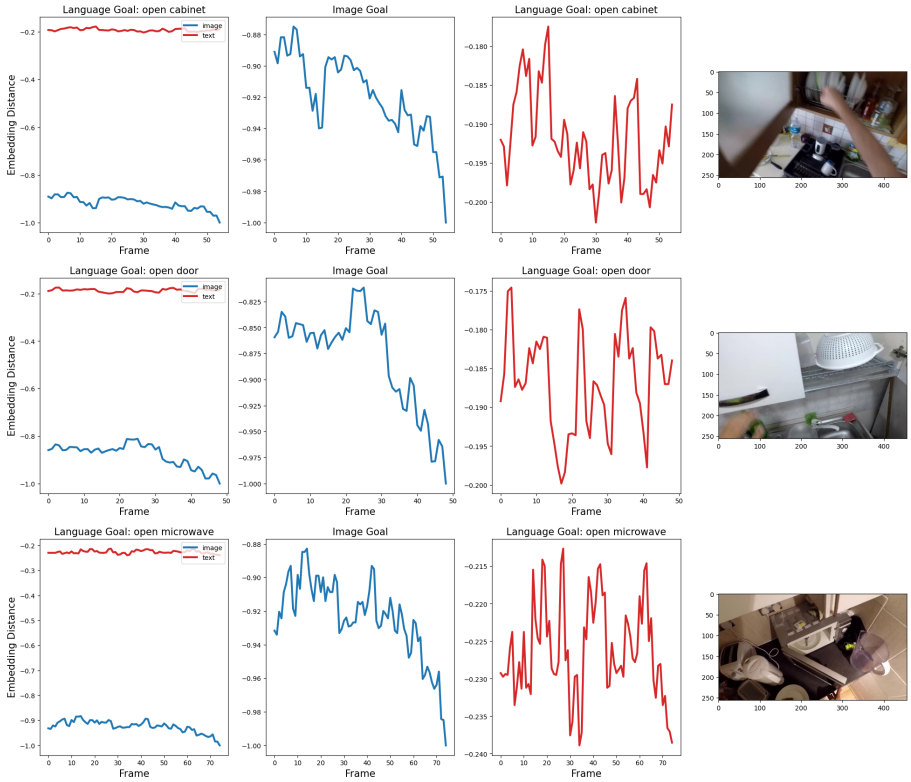


Figure 13. CLIP image and language goal reward curves on (unseen) EpicKitchen videos.

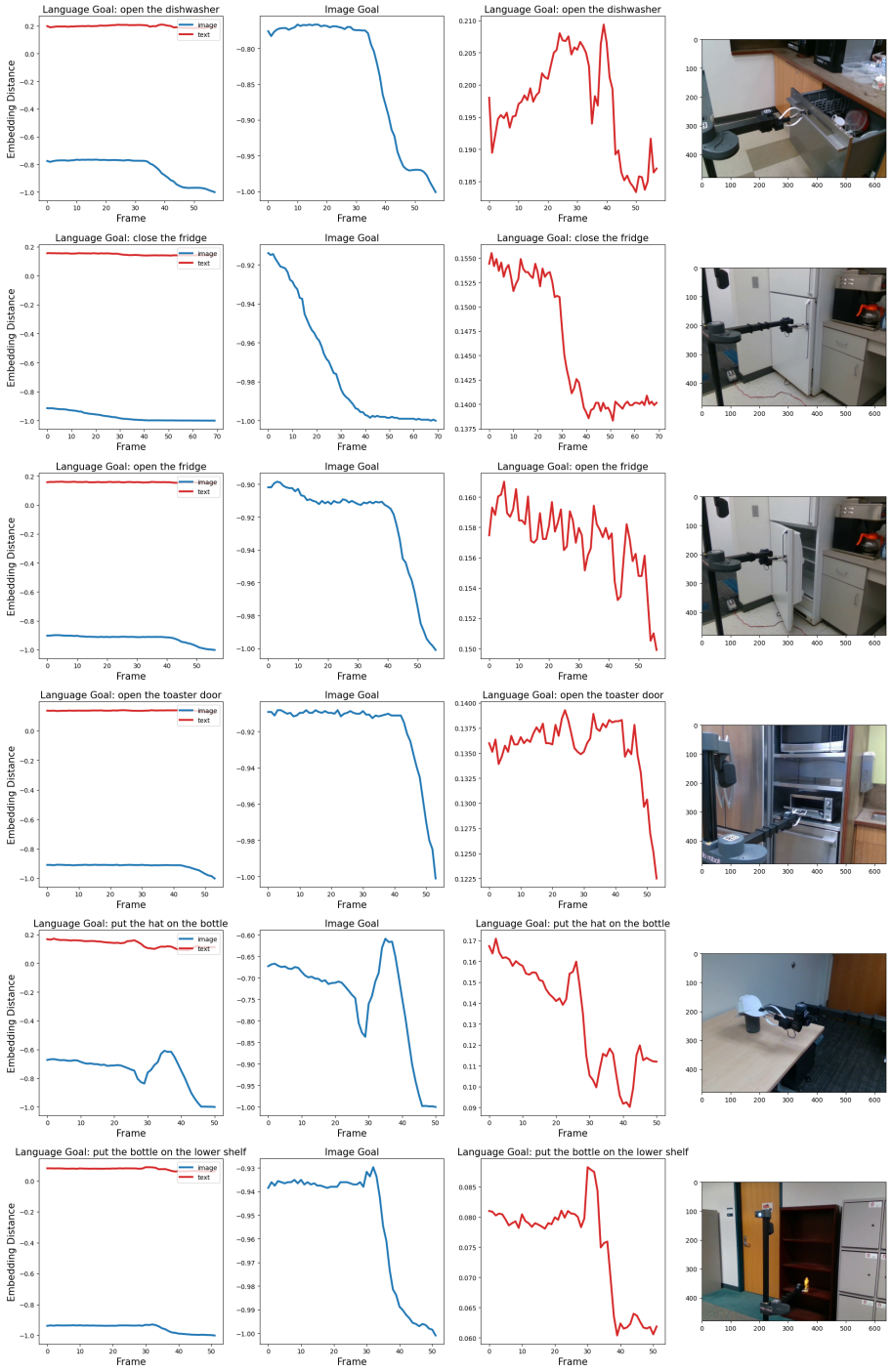


Figure 14. Success cases of LIV image and language goal reward curves on (unseen) Robot videos.

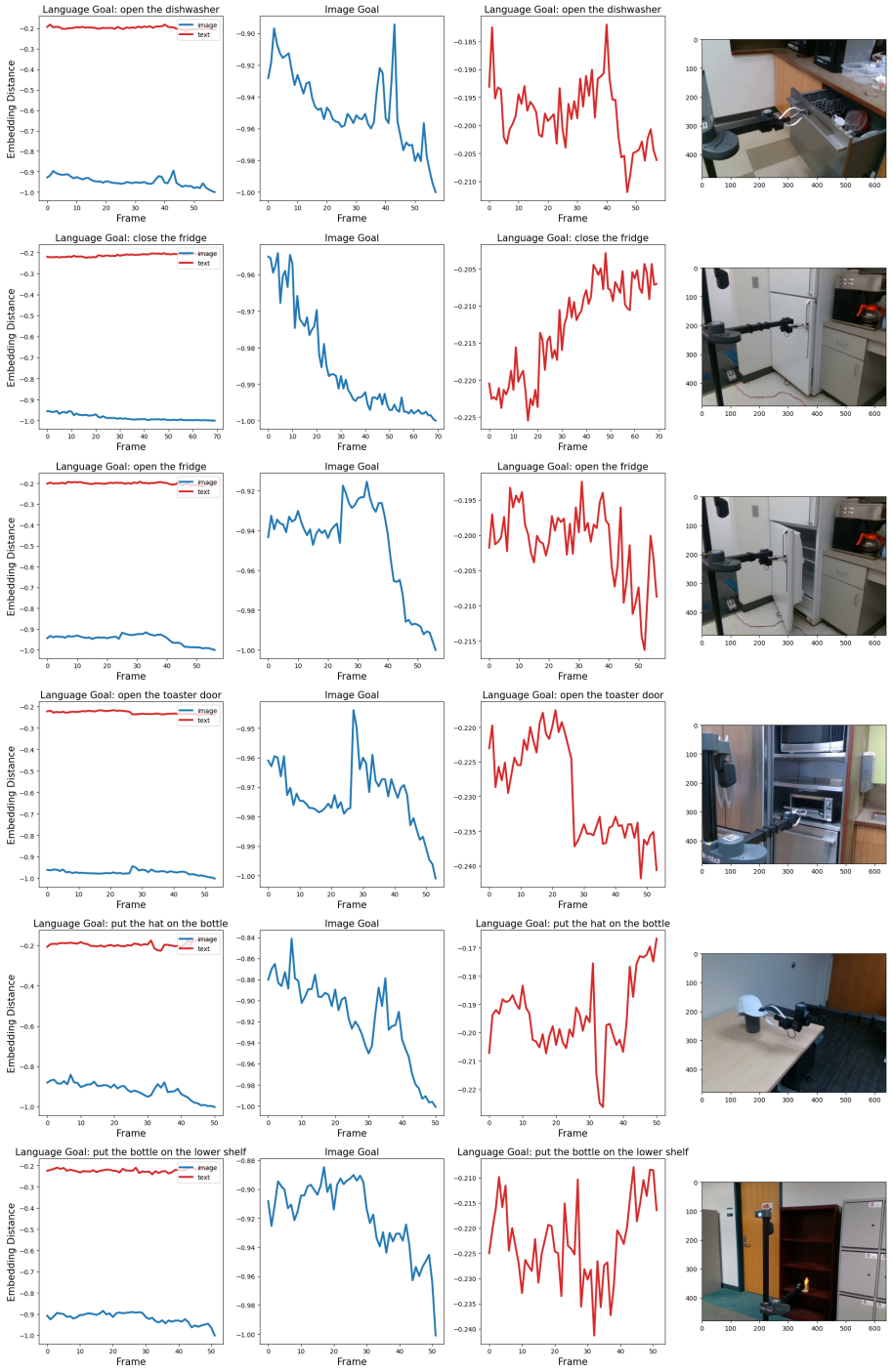


Figure 15. CLIP image and language goal reward curves on the same set of unseen Robot videos.

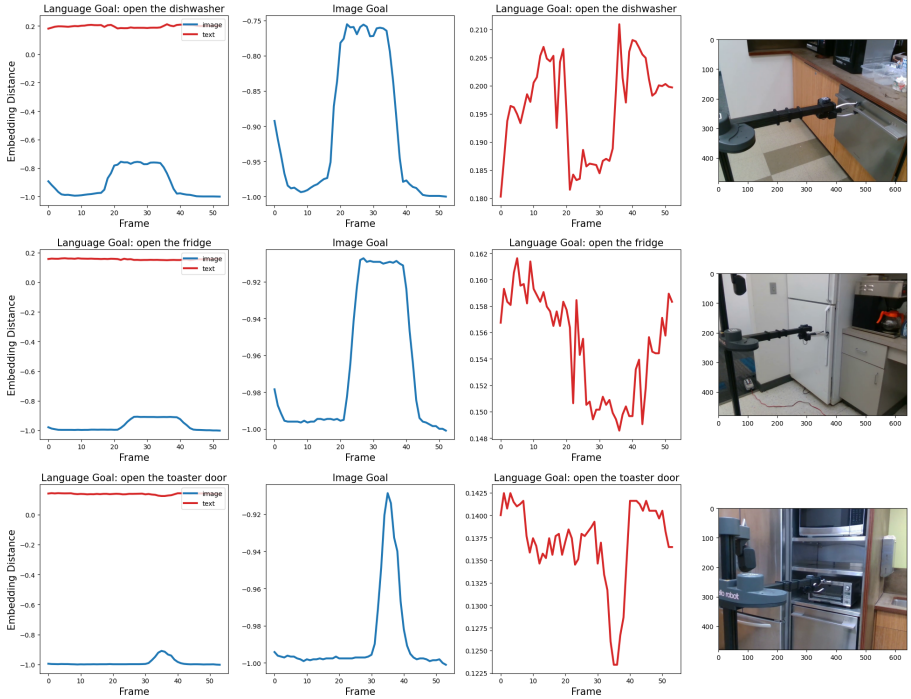


Figure 16. When the video contains opposite actions, LIV's image and language reward curves exhibit inverted trend because the image goal depicts the action completed in the last frame, which is opposite from the action described in the language goal that has already occurred in the middle of the video.

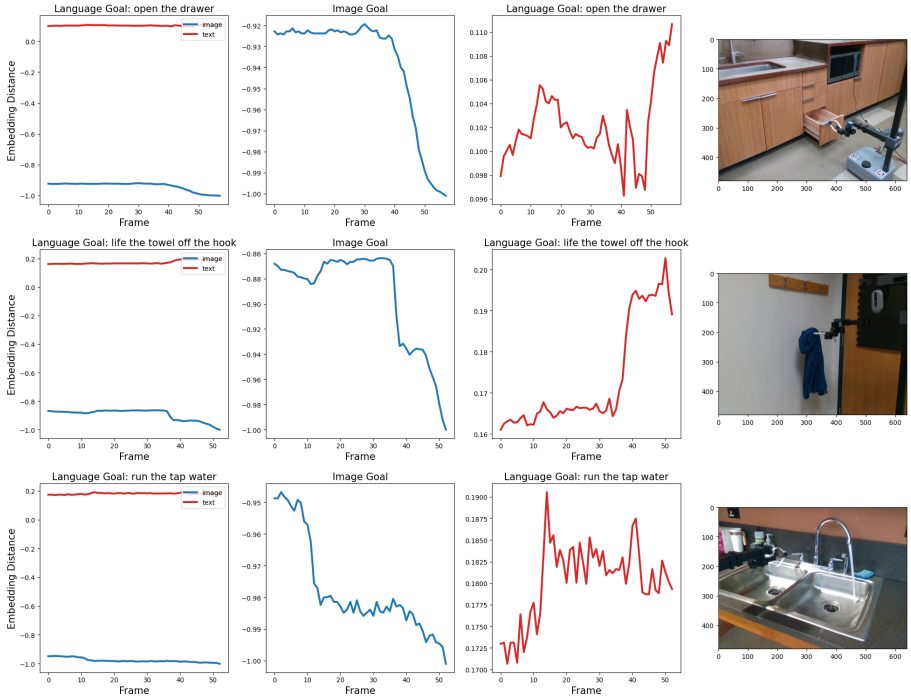


Figure 17. Failure cases of LIV image and language goal reward curves on (unseen) Robot videos.

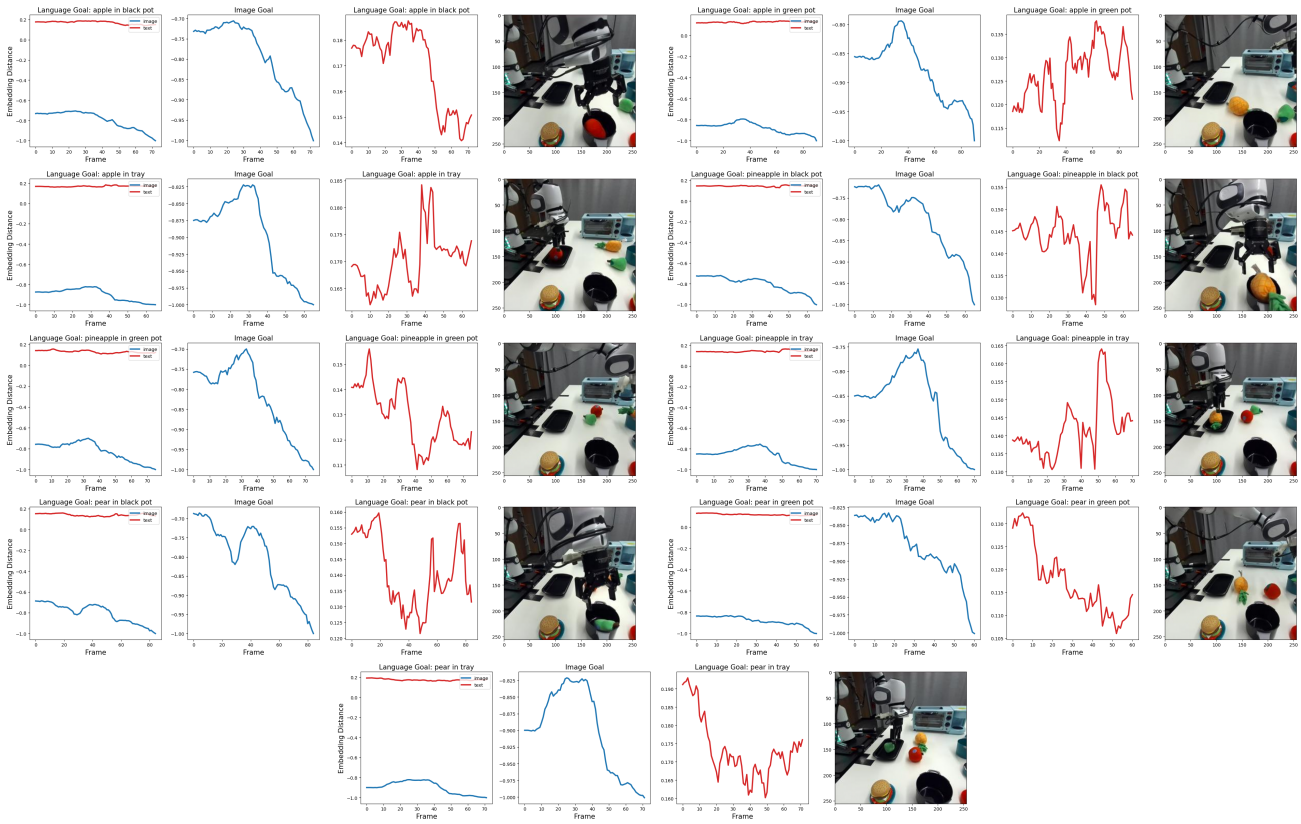


Figure 18. Pre-trained LIV image and language goal reward curves on RealRobot tasks.

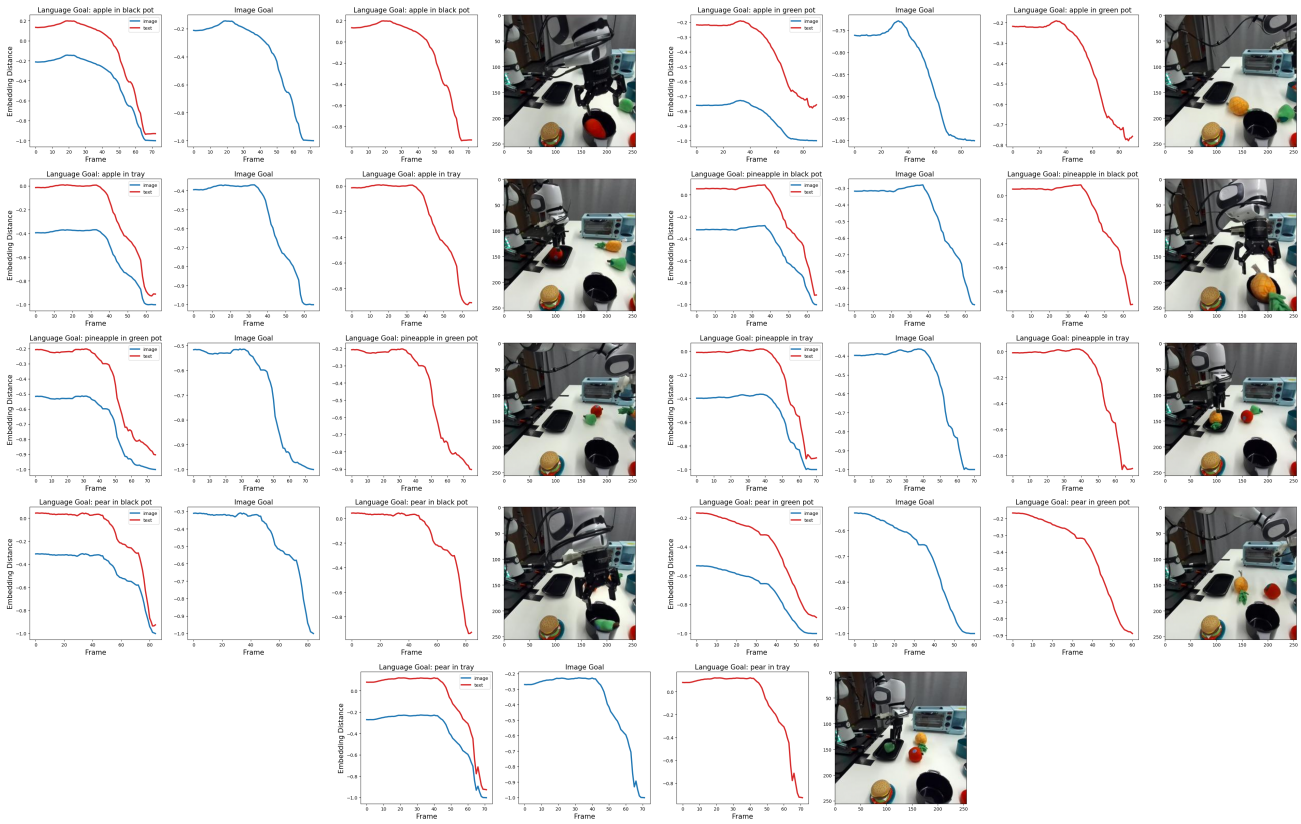


Figure 19. LIV (LIV fine-tuned) image and language goal reward curves on RealRobot tasks.

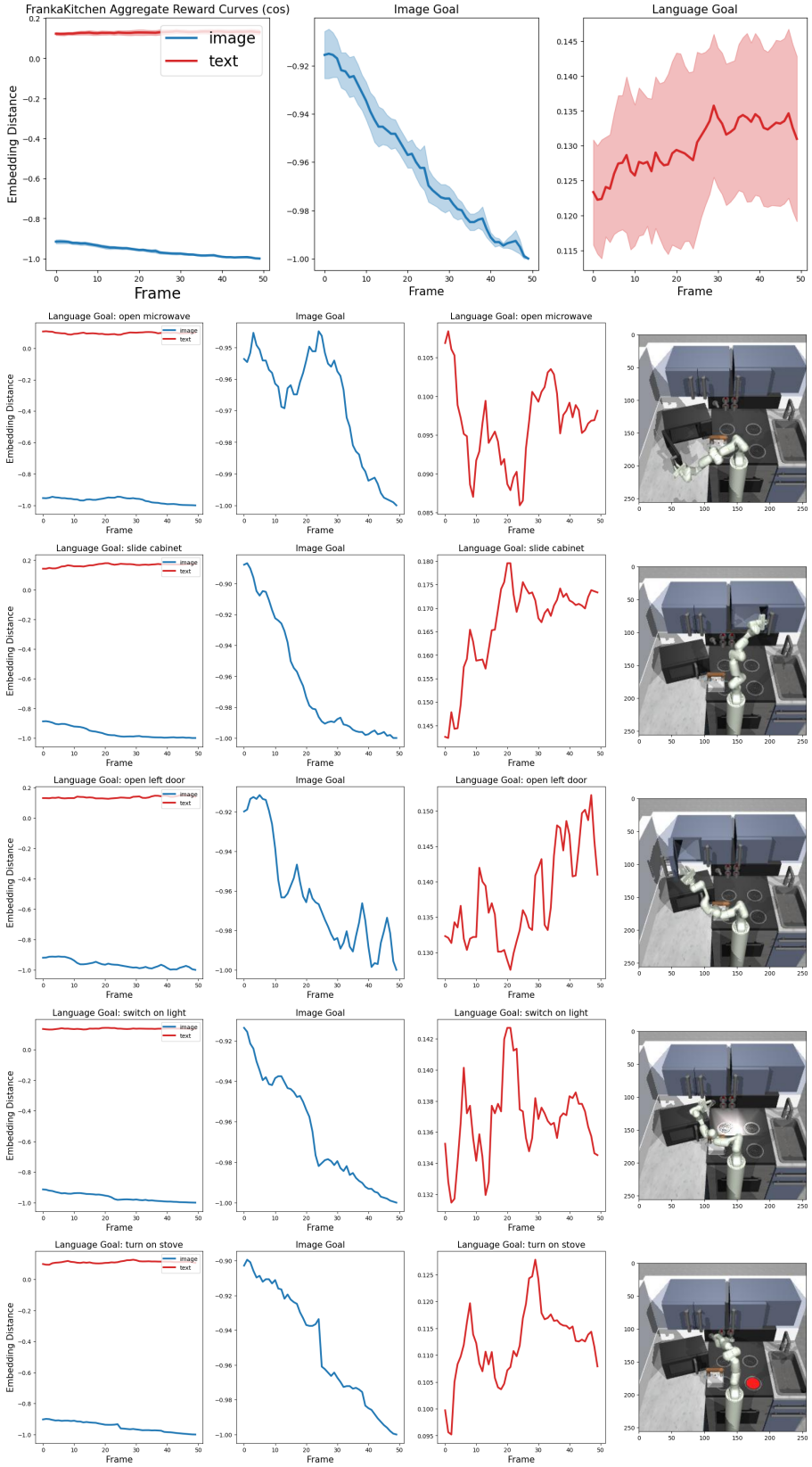


Figure 20. Pre-trained LIV-EPIC image and language goal reward curves on simulated FrankaKitchen tasks.

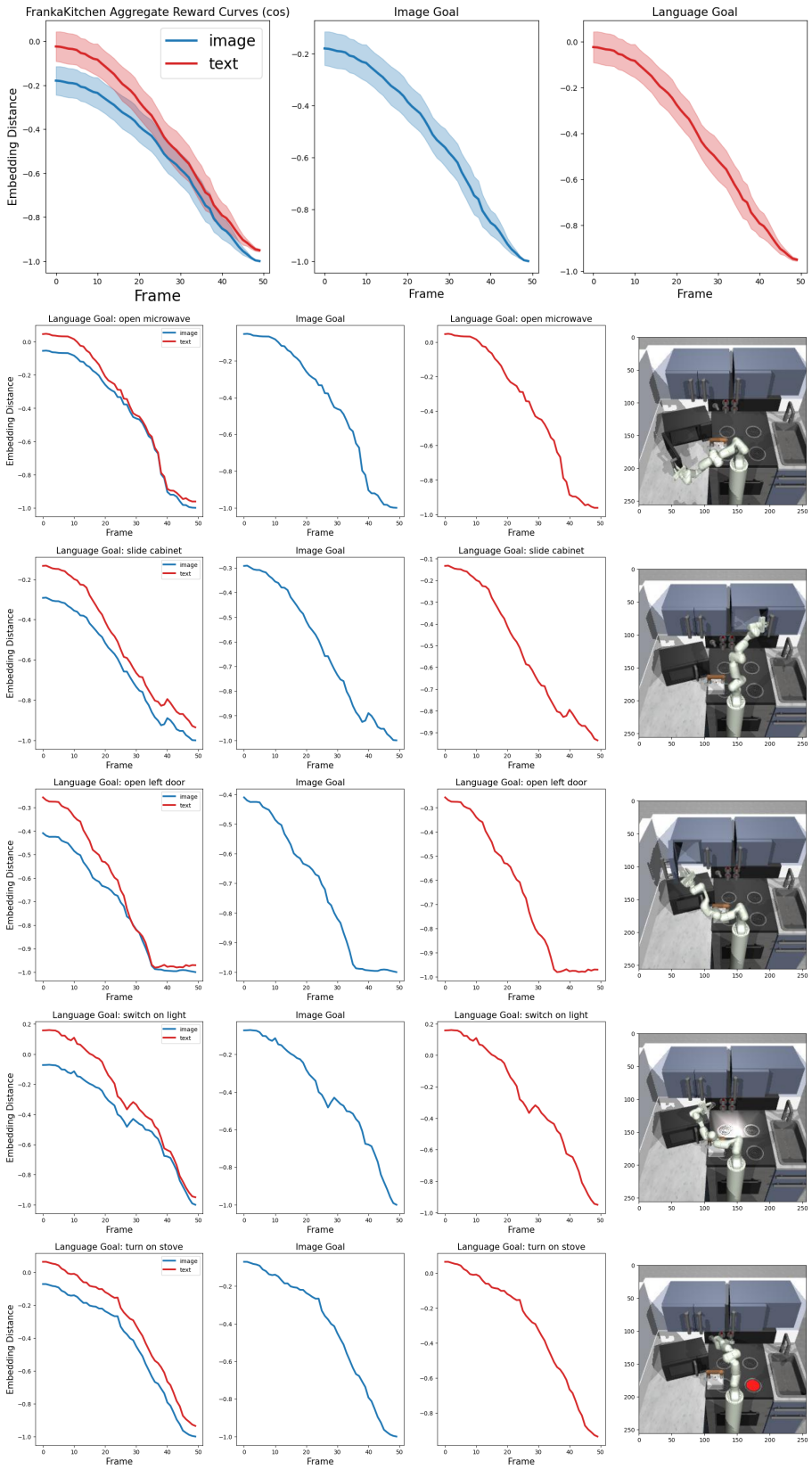


Figure 21. LIV-EPIC (LIV fine-tuned) image and language goal reward curves on simulated FrankaKitchen tasks.

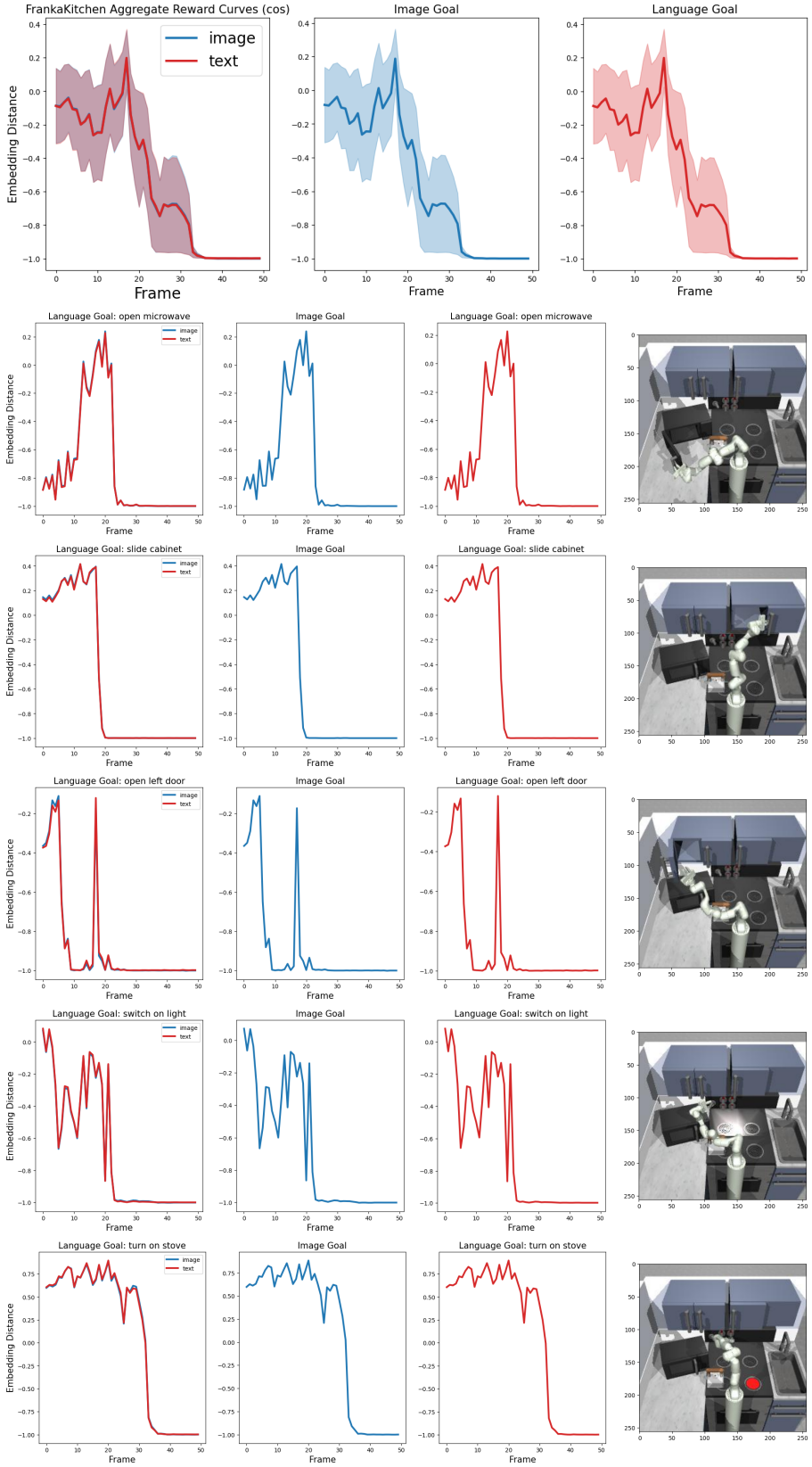


Figure 22. LIV-EPIC (CLIP fine-tuned) image and language goal reward curves on simulated FrankaKitchen.

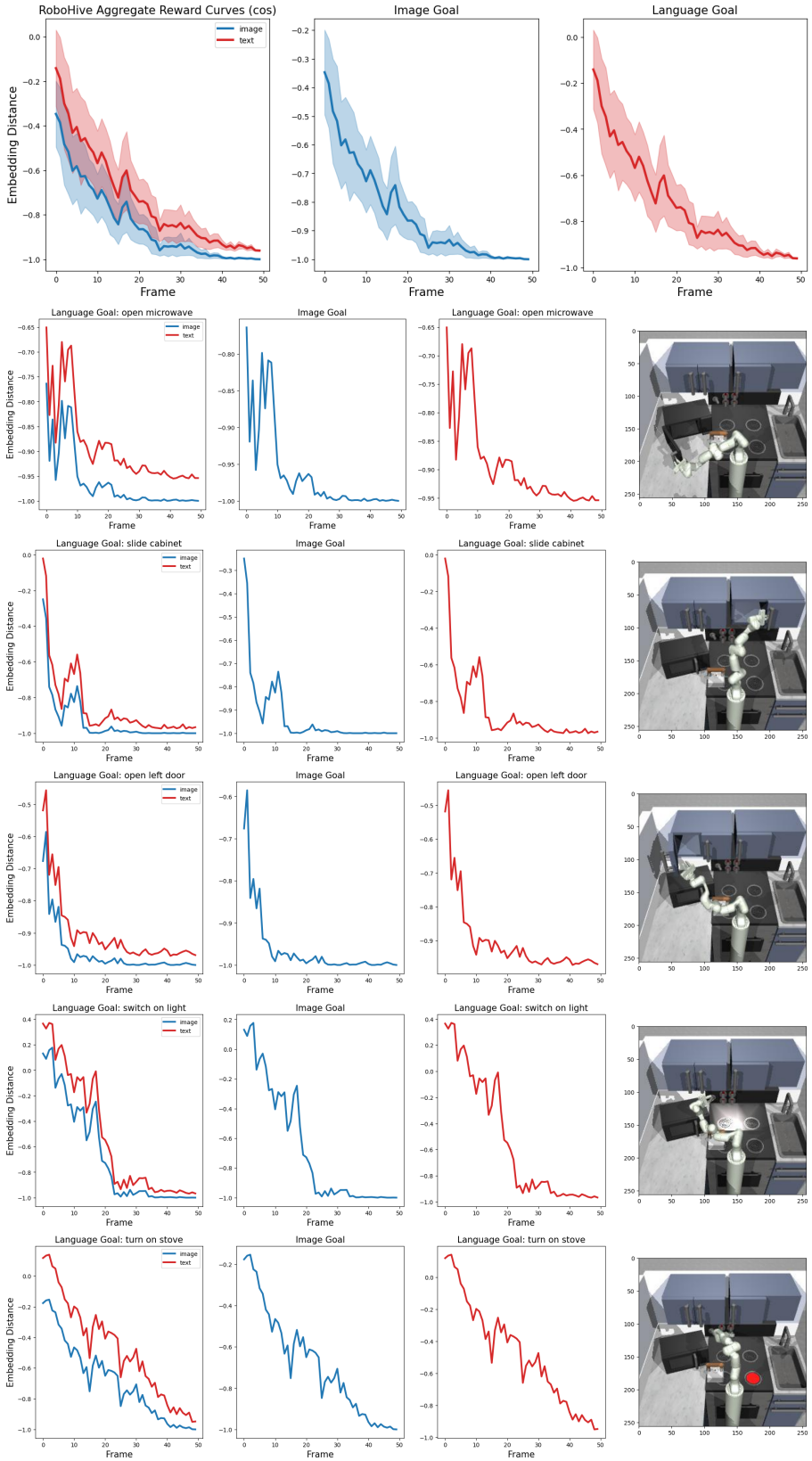


Figure 23. LIV-EPIC (TCN+CLIP fine-tuned) image and language goal reward curves on simulated FrankaKitchen.