DecisionNCE: Embodied Multimodal Representations via Implicit Preference Learning

Jianxiong Li [*1] **Jinliang Zheng** [*1 2] **Yinan Zheng** [*1] **Liyuan Mao** [1 3] **Xiao Hu** [1] **Sijie Cheng** [1] **Haoyi Niu** [1] **Jihao Liu** [2 4] **Yu Liu** [2] **Jingjing Liu** [1] **Ya-Qin Zhang** [1] **Xianyuan Zhan** [1 5]

Abstract

Multimodal pretraining is an effective strategy for the trinity of goals of representation learning in autonomous robots: 1) extracting both local and global task progressions; 2) enforcing temporal consistency of visual representation; 3) capturing trajectory-level language grounding. Most existing methods approach these via separate objectives, which often reach sub-optimal solutions. In this paper, we propose a universal unified objective that can simultaneously extract meaningful task progression information from image sequences and seamlessly align them with language instructions. We discover that via implicit preferences, where a visual trajectory inherently aligns better with its corresponding language instruction than mismatched pairs, the popular Bradley-Terry model can transform into representation learning through proper reward reparameterizations. The resulted framework, *DecisionNCE*, mirrors an InfoNCE-style objective but is distinctively tailored for decision-making tasks, providing an embodied representation learning framework that elegantly extracts both local and global task progression features, with temporal consistency enforced through implicit time contrastive learning, while ensuring trajectory-level instruction grounding via multimodal joint encoding. Evaluation on both simulated and real robots demonstrates that DecisionNCE effectively facilitates diverse downstream policy learning tasks, offering a versatile solution for unified representation and reward learning. Project Page: https://2toinf.github.io/DecisionNCE/

[*]Equal contribution [1]AIR, Tsinghua University [2]SenseTime Research [3]Shanghai Jiaotong University [4]CUHK MMLab [5]Shanghai AI Lab. Correspondence to: Jianxiong Li <li-jx21@mails.tsinghua.edu.cn>, Xianyuan Zhan <zhanxianyuan@air.tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Realizing general-purpose decision-making has long been an ultimate goal in AI research, which aims to build autonomous robotic agents that can comprehend environments via raw visual inputs and execute tasks described in natural language. In pursuit of such intelligent agents, the brute-force solution is end-to-end training large models on extensive in-domain data with action annotations (Zitkovich et al., 2023). However, this can be exceedingly data-hungry, often posing a stratospheric demand for training data. Pretraining scalable and generalizable vision-language representations, on the other hand, can leverage cheap out-of-domain data without action annotations, such as in-the-wild human manipulation videos (Grauman et al., 2022), to combat the scarcity of domain-specific data and facilitate effective downstream policy learning (Nair et al., 2023). Although these generic videos may not directly subscribe to the specific downstream embodiment or tasks, their broad state coverage can greatly enhance generalizability. Moreover, these videos contain context-rich dynamics about how objects interact with their surroundings, and how task progressions are semantically inscribed in varied instructions (Ma et al., 2023a). Hence, they hold immense potential for diverse downstream task executions, eliminating the dependency on intensive domain-specific data collection.

An imperative in learning effective multimodal joint representations is the *trajectory-level grounding* of language, referring to aligning language with *a sequence of images* rather than single frames, as image sequences (unlike static frames) inherently capture agent behaviors toward completing the given instructions (Nair et al., 2022). In particular, focusing on image sequences of varied lengths is critical, as shorter sequences reveal detailed local transitions, while longer ones offer global task progression semantics. Striking the right balance between local and global views can be quite complex, yet it determines the overall quality of task representation (Zhang et al., 2023). As shown in Table 1, existing methods tend to either focus on short, fixed-length sequences, thus missing global context (Karamcheti et al., 2023; Ma et al., 2023a); or consider only long horizons that overlook local transitions (Nair et al., 2023; 2022). More-

Table 1. Comparison of vision-language representation learning methods for decision making. * Note that R3M can partially extract global and local progressions but is limited to always starting from the first image in a whole video, which overlooks intermediary transitions within a video. Please see Appendix A for details.

Methods	Temporal consistency	Global progressions	Local progressions	Unified learning
LOReL (2022)	✗	✓	✗	✗
Voltron (2023)	✗	✗	✓	✗
R3M (2023)	✓	✓*	✓*	✗
LIV (2023a)	✓	✗	✓	✗
DecisionNCE	✓	✓	✓	✓

over, they often rely on separate objectives to either capture task progression, or enforce temporal consistency or language instruction alignment, delicately balanced by a hard-to-tune hyper-parameter (Ma et al., 2023a; Nair et al., 2023), which often compromises certain objectives at the expense of others. Consequently, developing a principled solution that addresses these multifaceted challenges in a unified manner becomes crucial.

To fill this gap, we introduce *DecisionNCE*, a unified multimodal representation learning framework for decision making. DecisionNCE is realized via an elegant design that transforms the popular Bradley-Terry (BT) model (Bradley & Terry, 1952) in preference-based reinforcement learning (Akrour et al., 2011) into multimodal representation learning architecture. Specifically, we introduce the concept of *implicit preferences*, where an image trajectory inherently aligns better with its corresponding language instruction than mismatched pairs (illustrated in Figure 1). Under this umbrella, BT model can transform into representation learning through proper reward reparameterizations. The resulted algorithm mirrors an InfoNCE-style (Oord et al., 2018) objective, which is distinctively tailored for decision-making tasks, by shifting the contrast focus from single images to language-aligned trajectory segments.

Unlike previous work where instruction grounding is conducted on full video, which is expensive and time-consuming, we show that by introducing well-designed *reward reparameterizations* in embedding spaces, the binary start-end transition in a selected video segment is sufficient in capturing task progression, bypassing irrelevant intermediary transitions to reduce computational cost and enable scalable learning. Specifically, we propose two DecisionNCE variants: 1) DecisionNCE-P derives from the common inductive bias that final static images are more aligned with instructions (Nair et al., 2023; Ma et al., 2023b;a); and 2) DecisionNCE-T intuits that language instruction describes dynamical transition flows better than static images. Notably, by adopting a random segment-sampling strategy, both DecisionNCE-P/T simultaneously compare short and

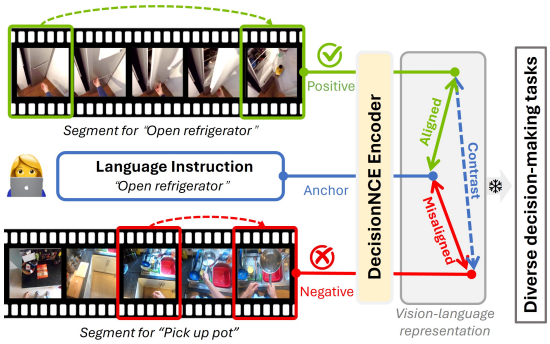


Figure 1. **Implicit Preference Learning**: Matched segments and instructions are preferred to mismatches. Thus, implicit preference learning inherently performs a trajectory-level contrastive learning that compares segments rather than single images.

long segments of various lengths, from which local/global temporal information can be extracted over diverse time spans, which artfully mirrors implicit time contrastive learning (Sermanet et al., 2018). This seamlessly fuses local/-global task progression with temporal consistency, as well as language alignment in an elegant and unified objective, eliminating ad-hoc designs in previous works (Nair et al., 2023; Ma et al., 2023b;a) and offering a surprisingly simple yet effective multimodal representation learning framework.

Pretrained on large-scale, in-the-wild human video datasets, DecisionNCE excels in capturing temporally consistent task progression and comprehensively aligning with instructions. The learned representations from DecisionNCE can then be used for downstream policy learning. Extensive evaluations on both simulated and real robots demonstrate DecisionNCE's superiority to state-of-the-art representation learning methods. Moreover, the representations can also serve as zero-shot rewards parameterized in the embedding space, which can be used for zero-shot trajectory optimization.

2. Preliminaries

Notations. We consider a language-conditioned Markov Decision Process (MDP) (Puterman, 2014) $\mathcal{M} := (\mathcal{O}, \mathcal{A}, \mathcal{L}, r, \mathcal{P}, \gamma)$. $\mathcal{O}, \mathcal{A}$ are state and action space. $\mathcal{P} : \mathcal{O} \times \mathcal{A} \to \mathcal{O}$ is dynamic, and $\gamma \in (0, 1)$ is discount factor. We assume the state space $\mathcal{O} = \mathbb{R}^{H \times W \times 3}$ is defined over RGB images. $\mathcal{L}$ is language instruction space, which specifies desired task behaviors. $r : \mathcal{O} \times \mathcal{A} \times \mathcal{L} \to \mathbb{R}$ is language-conditioned reward function. The goal is to learn a policy $\pi : \mathcal{O} \times \mathcal{L} \to \mathcal{A}$ that can solve arbitrary tasks described in language $l \in \mathcal{L}$.

We assume a video dataset $\mathcal{D} = \{v_i := (o_1^i, ..., o_{h_i}^i; l^i)\}_{i=1}^N$, where each $o \in \mathcal{O}$ is a raw RGB image, and each video v comes with a descriptive language annotation l. h is video length. The dataset $\mathcal{D}$ includes diverse out-of-domain

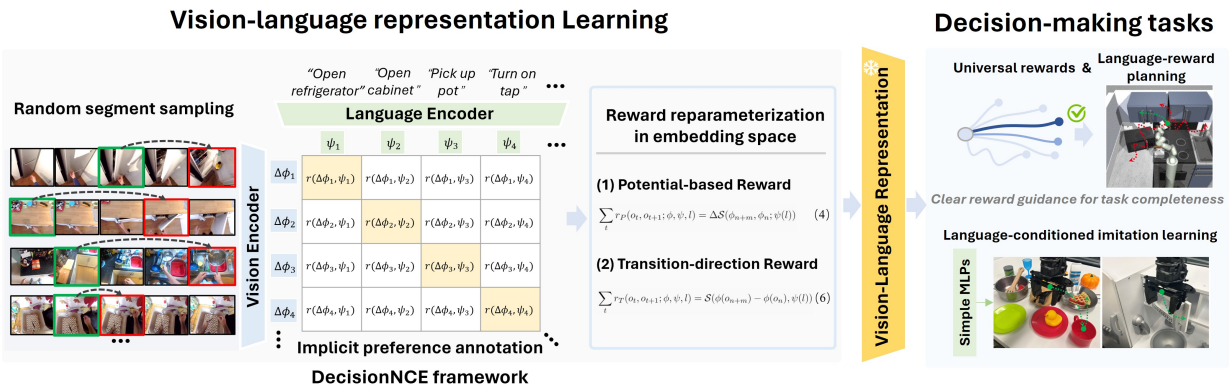


Figure 2. Overview of DecisionNCE framework. DecisionNCE focuses on jointly training vision and language encoders to achieve trajectory-level representation alignment. The learned representations can be applied to various downstream decision-making tasks.

data (Grauman et al., 2022; Damen et al., 2018), where no action labels are available. We aim to train a vision encoder $\phi : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^K$ and a language encoder $\psi : \mathcal{L} \rightarrow \mathbb{R}^K$ to map the raw image o and language l into a shared K-dimensional vision-language representation space, using the dataset $\mathcal{D}$. After this, the policy $\pi(\phi(o), \psi(l))$ can leverage these representations as inputs for policy learning via imitation learning (IL) or reinforcement learning (RL).

Bradley-Terry (BT) Model. Designing proper rewards is quite challenging for complex tasks. In contrast, the well-known BT model (Bradley & Terry, 1952) in preference learning offers a natural way for humans to convey their desired outcomes to agents, where humans are required to rank pairs of segments. Then, BT model optimizes the rewards to maximize Eq. (1) to fit human preferences (Hu et al., 2023).

$$P\left[\sigma^+ \succ \sigma^-\right] = \frac{\exp \sum_t r\left(o_t^+, o_{t+1}^+\right)}{\sum_{i\in\{+,-\}} \exp \sum_t r\left(o_t^i, o_{t+1}^i\right)}. \quad (1)$$

where $\sigma = (o_n, ..., o_{n+m})$ denotes a length-m segment. $\sigma^+ \succ \sigma^-$ indicates segment σ^+ is preferred to σ^-, and the preference for $\sigma^+ \succ \sigma^-$ grows exponentially with the total rewards of σ^+ and diminishes with those of σ^-.

3. DecisionNCE

Note the BT model in Eq. (1) is inherently a trajectory-level contrastive learning model that compares different trajectory segments and favors positive ones. This underlying equivalence inspires us to leverage BT model for representation learning. by reparameterizing the rewards with vision and language representations $\phi(o), \psi(l)$, and extending the original BT model into its language-conditioned version:

$$P\left[\sigma^+ \succ \sigma^-\right] = \frac{\exp \sum_t r\left(o_t^+, o_{t+1}^+; \phi, \psi, l^+\right)}{\sum_{i\in\{+,-\}} \exp \sum_t r\left(o_t^i, o_{t+1}^i; \phi, \psi, l^+\right)}. \quad (2)$$

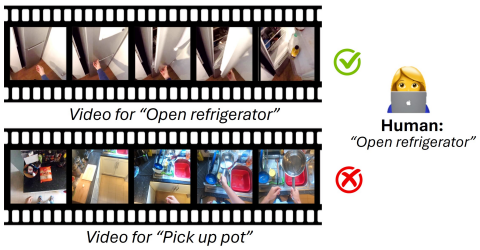


Figure 3. Implicit Preference. Segment is near-optimal for its associated language instruction, but is sub-optimal for others.

Eq. (2) essentially contrasts segments under various instructions, to extract task-relevant semantic features from image sequence segments. This offers a new possibility for trajectory-level language grounding via adapting BT model.

However, careful readers may notice that naïvely extending Eq. (2) to trajectory-level grounding faces several challenges: *1)* The preference labels are costly to collect (Lee et al., 2021), and existing video-language datasets have no explicit preference annotations; *2)* It is tricky to select appropriate segment lengths m for comparison, as overly long segments will induce extensive computations and lose local transition details, while short segments may lose global task information; *3)* It remains unclear what parameterized form of reward r is best for multimodal representation learning.

To address these challenges, we introduce three novel designs: implicit preference annotations, random segment sampling, and reward reparametrization in embedding space.

3.1. Implicit Preference Annotations

Despite the absence of explicit preference labels in existing datasets, it's crucial to recognize that language instructions naturally communicate human intentions to agents. We can reasonably assume a video v^i is more preferred for its

associated language instruction l^i than other mismatched instructions $l \neq l^i$, as shown in Figure 3. This mild assumption holds in many cases. For example, a video on "open microwave" is clearly sub-optimal for "close microwave" instruction. Thus, the existing video-language dataset $\mathcal{D}$ already possesses a wealth of such *implicit preference* that has yet to be fully exploited, eliminating the need for extra preference annotations.

3.2. Random Segment Sampling

Selecting a certain segment length m that best covers both local and global information is challenging, and previous works typically focus on either short or long video segments (Nair et al., 2023; 2022; Ma et al., 2023a;b; Karamcheti et al., 2023) (see Appendix A.1 for detailed discussion). A good balance between the two is preferable, enjoying the complementary advantages of both (Zhang et al., 2023). To realize this, we adopt the widely used random-sampling strategy in action recognition for segment sampling (Shi et al., 2013; Wang et al., 2018; Zhi et al., 2021; Zhang et al., 2023). Specifically, we randomly select a start image o_n from video $v = (o_1, ..., o_h)$ and then a goal image o_{n+m} from the following frames $(o_{n+1}, ..., o_h)$. Repetitively, this leads to varied segment lengths m, enabling the preference model to both capture local transitions in short segments and global task progressions in longer ones.

Despite the simplicity, we show in Section 4 that surprisingly, this simple sampling strategy induces implicit time contrastive learning for various DecisionNCE variants, guaranteeing smooth temporal consistency over time.

3.3. Implicit Preference Learning via Reward Reparameterization

This section explains how to train the vision and language encoders $\phi(o), \psi(l)$. In principle, the DecisionNCE framework is versatile to any parameterized form of reward. Demonstratively, we show that with two simple reparameterization schemes in embedding space, i.e., *potential-based* and *transition-direction reward*, we can adeptly extend the original reward learning objective (Eq. (2)) to effective representation learning , allowing the optimization of vision and language encoder $\phi(o), \psi(l)$ to implicitly optimize rewards.

(1) Potential-based Reward (DecisionNCE-P). Inspired by prior works that reformulate MDP in embedding space $\mathcal{M}_{\phi,\psi} := (\phi(\mathcal{O}), \mathcal{A}, \psi(\mathcal{L}), r, \mathcal{P}, \gamma)$ (Li et al., 2022; Ma et al., 2023b;a), we define the reward as the language-embedding distance difference in Eq. (3):

$$r_P(o_t, o_{t+1}; \phi, \psi, l) := \mathcal{S}(\phi(o_{t+1}), \psi(l)) - \mathcal{S}(\phi(o_t), \psi(l)). \tag{3}$$

Here, we choose cosine similarity for $\mathcal{S}$. This reward behaves similarly to potential-based reward shaping as

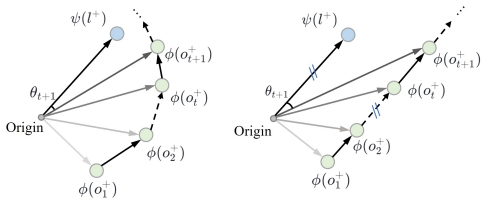


Figure 4. Illustration of DecisionNCE-P and DecisionNCE-T.

$\mathcal{S}_{t+1} - \mathcal{S}_t = (1-\gamma)\mathcal{S}_{t+1} + (\gamma\mathcal{S}_{t+1} - \mathcal{S}_t)$, which is beneficial for policy learning (Ng et al., 1999). Substitute Eq. (3) into Eq. (2), the total reward for a segment σ becomes:

$$\sum_t r_P(o_t, o_{t+1}; \phi, \psi, l) = \Delta\mathcal{S}(\phi_{n+m}, \phi_n; \psi(l)), \tag{4}$$

where $\Delta\mathcal{S}(\phi_{n+m}, \phi_n); \psi(l)) := \mathcal{S}(\phi(o_{n+m}), \psi(l)) - \mathcal{S}(\phi(o_n), \psi(l))$ denotes the embedding distance shifts. Then, the resulting DecisionNCE-P instantiation is:

Definition 3.1 (DecisionNCE-P)**.**

$$P_P\left[\sigma^+ \succ \sigma^-\right] = \frac{\exp \Delta\mathcal{S}(\phi_{n+m}^+, \phi_n^+; \psi(l^+))}{\sum_{i\in\{+,-\}} \exp \Delta\mathcal{S}(\phi_{n+m}^i, \phi_n^i; \psi(l^+))}. \tag{5}$$

Basically, maximizing Eq. (5) will attract $\phi(o_{n+m}^+)$ to $\psi(l^+)$ while repelling $\phi(o_n^+)$ from $\psi(l^+)$, leading to an inductive bias that *final images are more aligned with instructions* (Nair et al., 2023; Ma et al., 2023b).

(2) Transition-direction Reward (DecisionNCE-T). We also introduce an alternative inductive bias: *language represents a transition direction instead of final images*, as instructions most likely describe dynamical behaviors rather than static images (Karamcheti et al., 2023). To achieve this, we simply modify the total rewards in Eq. (4) to Eq. (6), leading to the DecisionNCE-T variant in Definition 3.2.

$$\sum_t r_T(o_t, o_{t+1}; \phi, \psi, l) = \mathcal{S}(\phi(o_{n+m}) - \phi(o_n), \psi(l)), \tag{6}$$

Definition 3.2 (DecisionNCE-T)**.**

$$P_T\left[\sigma^+ \succ \sigma^-\right] = \frac{\exp \mathcal{S}(\phi(o_{n+m}^+) - \phi(o_n^+), \psi(l^+))}{\sum_{i\in\{+,-\}} \exp \mathcal{S}(\phi(o_{n+m}^i) - \phi(o_n^i), \psi(l^+))}. \tag{7}$$

Here, we obtain another representation learning objective. Maximizing Eq. (7) aligns the transition direction of $\phi(o_{n+m}^+) - \phi(o_n^+)$ and the text embedding vector $\psi(l^+)$. In this case, $\phi(o_{n+m}^+)$ and $\phi(o_n^+)$ can remain distant from $\psi(l^+)$ as long as their transition direction aligns effectively.

Algorithm 1 DecisionNCE-P/T

Initialize: Out-of-domain video dataset with language annotations $\mathcal{D} := \{v_i := (o_1^i, ..., o_{h_i}^i; l^i)\}_{i=1}^N$, vision-language encoder ϕ, ψ
for each iteration **do**:
Randomly sample segments $\{\sigma_i\}_{i=1}^B = \{(o_{n_i}^i, ..., o_{n_i+m_i}^i; l^i)\}_{i=1}^B$ from dataset $\mathcal{D}$ with random segment length m_i.

$$\mathcal{L}_{\text{DecisionNCE-P}}(\phi,\psi) = \frac{1}{B}\sum_{i=1}^{B}\left[-\log\frac{\exp\Delta\mathcal{S}(\phi(o_{n_i+m_i}),\phi(o_{n_i});\psi(l^i))}{\sum_{j=1}^{B}\exp\Delta\mathcal{S}(\phi(o_{n_j+m_j}),\phi(o_{n_j});\psi(l^i))} - \log\frac{\exp\Delta\mathcal{S}(\phi(o_{n_i+m_i}),\phi(o_{n_i});\psi(l^i))}{\sum_{j=1}^{B}\exp\Delta\mathcal{S}(\phi(o_{n_j+m_j}),\phi(o_{n_j});\psi(l^j))}\right]$$

$$\mathcal{L}_{\text{DecisionNCE-T}}(\phi,\psi) = \frac{1}{B}\sum_{i=1}^{B}\left[-\log\frac{\exp\mathcal{S}(\phi(o_{n_i+m_i})-\phi(o_{n_i}),\psi(l^i))}{\sum_{j=1}^{B}\exp\mathcal{S}(\phi(o_{n_j+m_j})-\phi(o_{n_j})),\psi(l^i))} - \log\frac{\exp\mathcal{S}(\phi(o_{n_i+m_i})-\phi(o_{n_i}),\psi(l^i))}{\sum_{j=1}^{B}\exp\mathcal{S}(\phi(o_{n_i+m_i})-\phi(o_{n_i}),\psi(l^j))}\right]$$

Update ϕ and ψ using SGD: $(\phi,\psi) \leftarrow (\phi,\psi) - \alpha\nabla\mathcal{L}_{\text{DecisionNCE-P}}(\phi,\psi)$ or $(\phi,\psi) \leftarrow (\phi,\psi) - \alpha\nabla\mathcal{L}_{\text{DecisionNCE-T}}(\phi,\psi)$

Connections between DecisionNCE-P/T. Comparing the two variants, we notice that DecisionNCE-P explicitly concentrates on the absolute similarities between single images and text instructions, whereas DecisionNCE-T is less concerned with this aspect, but emphasizes more on the correctness of the relative transition direction. However, DecisionNCE-T can implicitly ensure a correct cosine similarity between $\phi(o_{t+1}^+)$, $\phi(o_t^+)$, and $\psi(l^+)$. As shown in Figure 4 (b), their angle θ_t can monotonically decrease if the transition direction is correct. Thus, Decision-T not only explicitly enforces a meaningful embedding learning direction, but also implicitly guarantees correct absolute embedding positions like DecisionNCE-P.

3.4. Practical Implementation

The pseudo-code of DecisionNCE for practical implementation is presented in Algorithm 1. Specifically, we extend the single negative segment in Eq. (5), (7) to all mismatched pairs that appeared in a mini-batch to improve training efficiency (highlighted as blue), as one gradient step can establish contrastive learning signals to more negative samples. This does not affect the analysis in Section 4, since the optimization direction remains unchanged for positive and negative segments. This final objective resembles an InfoNCE-style (Oord et al., 2018) loss function, but the contrastive terms are start-end transitions rather than single images. Thus, we term our method as *DecisionNCE*. Please see Appendix E for more implementation details.

4. Analyses and Insights

Given the two instantiations of DecisionNCE, in the following, we offer more in-depth analyses and insights of the algorithm designs and potentials of the proposed framework.

4.1. From Full Segments to Start-end Transitions

Note that by reparameterizing the reward as in Eq. (4), the intermediary transitions are conveniently canceled out. This greatly simplifies the comparison of full segments σ to just a binary start-goal transition $o_n \rightarrow o_{n+m}$ in any segment, which has proven advantageous in action recognition (Korbar et al., 2019; Zhang et al., 2023). This can effortlessly scale to any segment length, and by doing so, training efficiency is significantly improved, as comparing every transition within segments can be prohibitively expensive. Moreover, comparing all transitions often covers numerous meaningless intermediary points, especially over long segments, which can overwhelm informative transitions with noise and complicate training. In contrast, Eq. (4) concentrates on the critical transition $o_n \rightarrow o_{n+m}$, maintaining the essential task progression while bypassing most noisy data. To investigate this, in Figure 5, we train DeicisionNCE-T using the total segment rewards calculated by not only $o_n \rightarrow o_{n+m}$, but also including some intermediary points (see Appendix E.2 for experiment setups). The results show that including intermediary transitions sometimes can be harmful for representation learning, leading to decreased success rate on policy learning. Therefore, we design our *transition-direction reward* in Eq. (6) solely on the start-goal transitions.

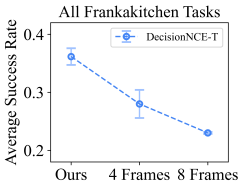


Figure 5. Ablation on different numbers of frames used for preference learning.

4.2. Mirroring Time Contrastive Learning

At first glance, some works also compare binary transitions similar to DecisionNCE-P/T, but they are limited to either long transitions $o_1 \rightarrow o_h$ across entire trajectories (Nair et al., 2022) or short transitions with a fixed k-interval $o_t \rightarrow o_{t+k}$ (Ma et al., 2023a;b; Karamcheti et al., 2023), and require additional objective for language grounding, which leads to complicated trade-offs among various elements. By contrast, we find that our segment comparisons in Eq. (5) and Eq. (7) effortlessly integrate implicit time contrastive learning for temporal consistency and meanwhile conduct local/global language grounding within a single, cohesive objective, effectively overcoming these challenges.

Specifically, a segment σ^+ is randomly selected from v^+, so any frame in v^+ can be a "goal" image or a "start" image. However, thanks to the *random segment sampling* strategy, the chance for frame o_t^+ being selected as the goal is not

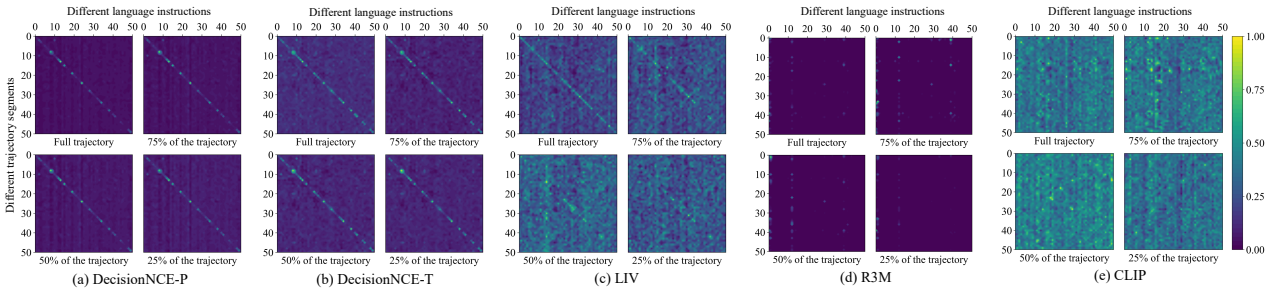


Figure 6. Heatmap of the learned rewards for various instruction/segment pairs. The diagonals are matched pairs and off-diagonals denote mismatches. DecisionNCE identifies unmatched pairs with varied segment lengths, extracting both local/global task progressions.

isotropic but gradually increases as the video progresses. Mathematically, the probability for $o_t^+, 0 < t < h$ being a goal image grows monotonically w.r.t its time stamp t (see Appendix B for proof):

$$P(o_t^+ \text{ is selected as goal}) = \frac{1}{h}\sum_{i=1}^{i<t}\frac{1}{h-i} \tag{8}$$

Therefore, for DecisionNCE-P, Eq. (8) clearly implies that later images o_j^+ have an increasing likelihood of being attracted to $\psi(l^+)$ than earlier ones $o_i^+, j > i$, with the attraction strength smoothly increasing over time step t. Thus, DecisionNCE-P implicitly promotes smooth temporal progressions in representation spaces, mirroring time contrastive learning (Sermanet et al., 2018). This characteristic also holds for DecisionNCE-T. As depicted in Figure 4(b), as the transition directions between randomly sampled image embedding pairs are driven to be similar within the same video, the cosine similarity between consecutive image embeddings $\phi(o_t), \phi(o_{t+1})$ can also remain large. Indicated by Eq. (8), the transition direction from initial to later image embeddings will be increasingly aligned with the direction of text embedding vector, as later images are more frequently chosen as goal images in video trajectories.

Notably, both DecisionNCE-P/T seamlessly integrate language grounding and temporal consistency into a single, simplified loss function, avoiding the intricate balance required between these two objectives. Additionally, the use of randomly selected segments eliminates the need for extra tedious hyper-parameter tuning, unlike prior methods that rely on fixed or meticulously adjusted intervals (Ma et al., 2023b;a; Karamcheti et al., 2023; Nair et al., 2023; 2022).

4.3. Positioning Task-Irrelevant Image Embeddings

In DecisionNCE-P/T, frames in negative segments $\sigma^- = (o_n^-, ..., o_{n+m}^-; l^-) \subseteq v^- = (o_1^-, ..., o_h^-; l^-)$ are optimized oppositely as compared to positive segments. This can potentially attract sampled initial image embedding $\phi(o_n^-)$ to $\psi(l^+)$, and the first image o_1^- in negative video v^- will be the most attracted to $\psi(l^+)$. Counter-intuitively, this seems "incorrect" at first glance, as o_1^- is typically irrelevant to positive language instructions. However, we can demonstrate that this seemingly "incorrect" objective will actually lead to desired representation spaces, unlike other methods.

Table 2. Cosine similarities among the first image embeddings $\phi(o_1)$ and similarities to language embeding means $\psi(\tilde{l})$. See Appendix E.3 for experimental setups.

	LIV	DeicisionNCE (Ours)	
		-P	-T
$\phi(o_1)$ similarities↑	0.08 ± 0.005	0.44 ± 0.01	0.96 ± 0.004
Similarities to $\psi(\tilde{l})$ ↑	-0.06 ± 0.02	0.04 ± 0.008	0.02 ± 0.01

In fact, based on our designs in DecisionNCE, $\phi(o_1^-)$ is not only drawn to $\psi(l^+)$, but to all mismatched language embeddings with $l \neq l^-$. This essentially drives first image embeddings $\phi(o_1)$ of different videos to the central position among all language embeddings, as depicted in Figure 8. This is actually desirable, as the first images of videos hardly carry any substantial information about the goal of tasks. Determining the nature of the tasks based solely on the first images in videos is highly difficult even for humans. As an example, we notice that in video-language datasets like KITCHEN-100 (Damen et al., 2018), the videos for "open door" instruction might not even show a door in the first image. So ideally, $\phi(o_1)$ should be positioned away from any specific instruction as it lacks task-specific information, which is exactly what DecisionNCE-P/T do. In Table 2, we demonstrate that DecisionNCE-P/T successfully cluster potentially task-irrelevant first image embeddings into similar positions, yielding high similarities among them, whereas other methods cannot.

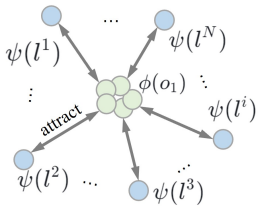


Figure 8. Illustration for task-irrelevant first image embeddings $\phi(o_1)$.

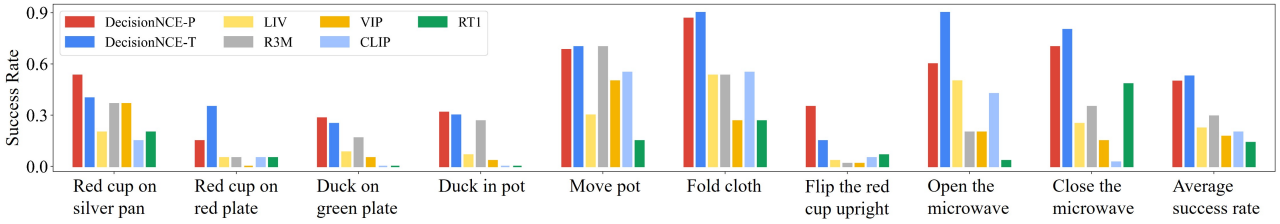


Figure 7. **Real robot LCBC experimental results**. Success rate is averaged over 10 episodes and 3 seeds.

4.4. Advanced Local/Global Trajectory-level Grounding

Beyond previously discussed properties, we also examine the ability of DecisionNCE to capture both local and global task progressions. Specifically, similarities for segments of varied lengths should be higher for matched instructions than mismatched pairs. The visualization in Figure 6 clearly demonstrates that DecisionNCE is superior in this aspect, as our training strategy explicitly guarantees this with the effective *random segment sampling* scheme under the implicit preference learning framework, capturing local and global task progressions by comparing both short and long segments. Previous methods, however, are often impaired by noises.

5. Experiments

We pretrain the DecisionNCE encoders using large-scale human video dataset EPIC-KITCHEN-100 (Damen et al., 2018) and conduct extensive experiments on both simulated and real robotic environments (Figure 9). Specifically, we investigate whether the representation learned by DecisionNCE can power effective downstream policy learning using methods like Language-Conditioned Behavior Cloning (LCBC), and whether DecisionNCE can offer universal language-conditioned rewards that describe the desired task progression directions.

(a) WidowX real robot (b) FrankaKitchen

Figure 9. Experimental environments.

Baselines. We compare DecisionNCE with following baselines: 1) *CLIP* (Radford et al., 2021b): Aligns images and language through contrastive learning. 2) *R3M* (Nair et al., 2023): Combines LOReL (Nair et al., 2022) and time contrastive learning (Sermanet et al., 2018) with a frozen language encoder. 3) *VIP* (Ma et al., 2023b): Trains representations via an RL-based objective but has no language modality. We use pretrained DistilBERT (Sanh et al., 2019) as the language encoder for VIP. 4) *LIV* (Ma et al., 2023a): Extends VIP to vision-language representation learning using CLIP to align final images with instructions. 5) *RT1* (Brohan et al., 2022): An end-to-end LCBC method. We initialize RT1 with its original pretrained vision-language encoder for a fair comparison. We do not compare with Voltron (Karamcheti et al., 2023) as it is trained on small-scale Something-Something-v2 (Goyal et al., 2017) data.

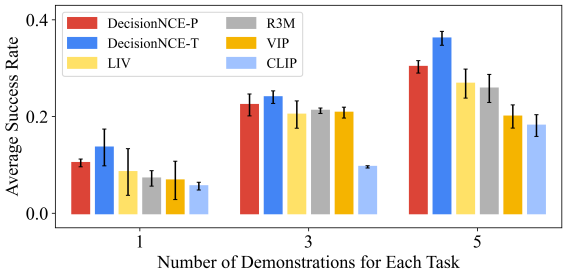


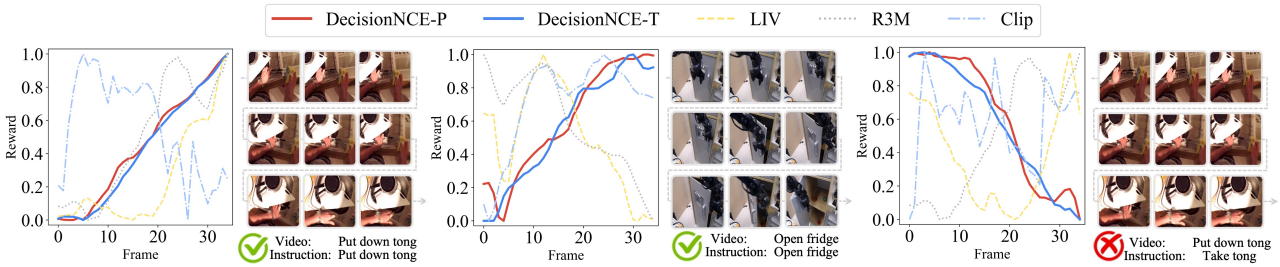
Figure 10. **Simulation LCBC results**. Max success rate averaged over 25 evaluation episodes and 3 seeds.

5.1. Language-conditioned Behavior Cloning Results

We freeze the pretrained vision-language encoders and use their output representations as input to a 256-256 MLP to train LCBC policies, trained using a few domain-specific demonstrations with action annotations. Extensive results show that DecisionNCE can extract valuable information that facilitates efficient downstream policy learning.

Results on real robots. For real robots, we evaluate 9 distinct tasks. For all tasks, the environment is highly stochastic with randomly initialized robots and objects, and contains lots of distractors with random locations, colors, and shapes. Ideally, the representations should be robust against these disturbances. Please see Appendix E for detailed setups.

Results are presented in Figure 7. DecisionNCE significantly outperforms baselines across almost all tasks. Among all baselines, LIV and R3M achieve the best results, benefiting from the alignment between images and languages as well as temporal consistency. CLIP, however, performs poorly as aligning languages with single images misses key task progression signals encoded in sequential data. More-



(a) In-distribution reward (b) Zero-shot reward (c) Contradictory video-instruction pairs

Figure 11. Visualization of the learned reward for diverse data. DecisionNCE provides more accurate rewards than baseline methods.

over, VIP cannot obtain satisfactory results as its pretrained vision and language embeddings are not aligned. RT1 also obtains subpar performances as the downstream data is too small to train a large model.

Results on simulated robots. We also evaluate on the FrankaKitchen (Gupta et al., 2019) benchmark, following LIV (Ma et al., 2023a). We train LCBC policies on 5 tasks in FrankaKitchen environment using 1/3/5 demonstrations for each task. Figure 10 shows that DecisionNCE achieves the highest success rate across diverse dataset quantities, demonstrating its effectiveness in extracting valuable information from out-of-domain data.

5.2. Universal Reward Learning

Note that the BT model is widely used in PBRL for reward learning, thus our learned representations can also be parameterized into universal reward signals (as in Section 3.3), offering direct guidance for accomplishing downstream tasks. We explore this ability of DecisionNCE by visualizing the normalized rewards calculated by image/language similarities on different videos. We also visualize rewards derived by other methods. Figure 11(a-b) demonstrates that DecisionNCE not only offers accurate reward guidance on in-distribution data, but also provides effective zero-shot rewards on out-of-distribution views and embodiments. Furthermore, Figure 11(c) shows DeicisionNCE is the only model that clearly identifies mismatched video/language pairs, revealing strong robustness against erroneous instructions compared to other methods. This robustness likely stems from DecisionNCE's training on randomly sampled segments (resists noisy transitions) and the use of contrastive learning against negative segments (enhances the ability to discern mismatched pairs).

5.3. Language-reward Planning

To further assess the quality of implicit reward, we apply it in a model-based planning method, MPPI (Williams et al., 2017), where future actions are optimized according to returns derived from the parameterized rewards, in a zero-shot manner. For baselines, we choose CLIP, LIV and R3M, and also use their customized rewards to run MPPI. We conduct experiments on FrankaKitchen, adhering to experimental protocols from Ma et al. (2023b) (See Appendix E.6 for details). Results in Table 3 clearly show that DecisionNCE-P/T both significantly outperform other methods, indicating superior capability in capturing the similarity between the images and the language goals, thus offering more accurate guidance for generic language-directed tasks. Also, note that DecisionNCE-T performs better than DecisionNCE-P in most scenarios. We believe this is because DecisionNCE-T not only ensures correct relative transition directions, but also correct absolute positions, which makes it well-suited for reward/representation learning.

Table 3. Average success rate overall FrankaKitchen tasks using MPPI planning based on zero-shot implicit rewards.

Model	CLIP	R3M	LIV	DecisionNCE (Ours)	
				-P	-T
Average Success Rate	0.016	0.036	0.054	0.101	0.122

6. Related Work

Representation learning for decision making. Pretrained representations are useful for downstream decision-making tasks (Shah & Kumar, 2021; Zhang et al., 2021; Parisi et al., 2022; Cui et al., 2022; Laskin et al., 2020b). Yet, most prior works focus on narrow and expensive in-domain data (Laskin et al., 2020a; Yarats et al., 2021; Seo et al., 2022; Kumar et al., 2022; Myers et al., 2023; Bhateja et al., 2023). Our work explores a broader setting that leverages extensive cheap out-of-domain data like human demonstration videos (Damen et al., 2018; Goyal et al., 2017; Grauman et al., 2022) to improve representation quality. While some works have studied similar settings, they mostly limit to uni-model representations that focus only on visual modality (Mendonca et al., 2023). For instance, Xiao et al. (2022) and Radosavovic et al. (2023) noticed that Masked-

Autoencoder (He et al., 2022; Liu et al., 2023) for vision representation learning can improve decision making, and VIP (Ma et al., 2023b) performed Goal-Conditioned Reinforcement Learning (GCRL) (Ma et al., 2022) with final image as goal specification to extract task-relevant features.

Language grounding for decision making. Several studies directly apply existing multimodal representations, like CLIP (Radford et al., 2021a), without tailoring them for decision making (Radford et al., 2021b; Khandelwal et al., 2022; Reid et al., 2022), potentially overlooking critical elements such as environmental dynamics and task progressions. For instance, LIV (Ma et al., 2023a) aligns language with static goal images using CLIP, missing these essential elements. By contrast, *trajectory-level grounding*, which aligns language with image sequences, is more suited for decision making. Voltron (Karamcheti et al., 2023) proposed a joint vision-language objective, yet its temporal length is quite short, failing to capture global task progressions. Conversely, R3M (Nair et al., 2023) and LOReL (Nair et al., 2022) employed a language reward head to extract global task progressions, but miss some local transition details. HULC (Mees et al., 2022) used CLIP to align language with image sequences, but is limited to short sequence length due to high computation costs. Moreover, most previous works often employ additional time contrastive learning objective (Sermanet et al., 2018) for temporal consistency, leading to a complex trade-off between two stages: grounding and ensuring temporal consistency, balanced by a hard-to-tune hyper-parameter (Nair et al., 2023; Ma et al., 2023a). In contrast, our DecisionNCE framework elegantly merges the two-stage objectives into a unified loss, bypassing all nuanced trade-offs, presenting a principled vision-language representation learning framework for decision-making. Please see Appendix A for extended discussions.

7. Conclusion

In this paper, we show that by introducing several smart designs, i.e., implicit preference annotations, random segment sampling, and reward reparameterization in embedding spaces, the Bradley-Terry model in preference learning can be elegantly extended to vision-language representation learning. This adaptation naturally leads to an elegant and cohesive trajectory-level InfoNCE-style optimization objective that is specifically tailored for decision making. The derived DecisionNCE framework effectively addresses the central challenges in trajectory-level grounding problem, offering a principled approach to universal representation and reward learning. Due to limited resources, we instantiate DecisionNCE based on the CLIP encoders. Future research can further scale up DecisionNCE to train large Vision-Language Models (VLMs). More discussion on limitations and future direction can be found in Appendix F.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. One potential ethical concern is that the abroad data used for representation pretraining may contain some potential privacy issues and biases. However, in this paper, we use the open-sourced EPIC-KITCHEN-100 human video dataset (Damen et al., 2018), which has been well peer-reviewed, thus resolved this ethical concern. We also encourage researchers to filter out privacy-sensitive and biased data when they train DecisionNCE on their customized dataset.

Acknowledgement

This work is supported by National Key Research and Development Program of China under Grant (2022YFB2502904), and funding from Haomo.AI. The authors would like to thank Yeqi Feng and Wei Wang for discussions on the earlier draft of the work; The authors would also like to thank the anonymous reviewers for their feedback on the manuscripts.

References

Ajay, A., Han, S., Du, Y., Li, S., Gupta, A., Jaakkola, T. S., Tenenbaum, J. B., Kaelbling, L. P., Srivastava, A., and Agrawal, P. Compositional foundation models for hierarchical planning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 15

Akrour, R., Schoenauer, M., and Sebag, M. Preference-based policy learning. In *Proceedings of the 2011th European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part I*, pp. 12–27, 2011. 2

Bhateja, C. A., Guo, D., Ghosh, D., Singh, A., Tomar, M., Vuong, Q., Chebotar, Y., Levine, S., and Kumar, A. Robotic offline rl from internet videos via value-function pre-training. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. 8

Black, K., Nakamoto, M., Atreya, P., Walke, H., Finn, C., Kumar, A., and Levine, S. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning*, 2023. 15

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 2, 3

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. 7, 15

Collaboration, O. X.-E., Padalkar, A., Pooley, A., Jain, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Singh, A., Brohan, A., Raffin, A., Wahid, A., Burgess-Limerick, B., Kim, B., Schölkopf, B., Ichter, B., Lu, C., Xu, C., Finn, C., Xu, C., Chi, C., Huang, C., Chan, C., Pan, C., Fu, C., Devin, C., Driess, D., Pathak, D., Shah, D., Büchler, D., Kalashnikov, D., Sadigh, D., Johns, E., Ceola, F., Xia, F., Stulp, F., Zhou, G., Sukhatme, G. S., Salhotra, G., Yan, G., Schiavi, G., Su, H., Fang, H.-S., Shi, H., Amor, H. B., Christensen, H. I., Furuta, H., Walke, H., Fang, H., Mordatch, I., Radosavovic, I., Leal, I., Liang, J., Kim, J., Schneider, J., Hsu, J., Bohg, J., Bingham, J., Wu, J., Wu, J., Luo, J., Gu, J., Tan, J., Oh, J., Malik, J., Tompson, J., Yang, J., Lim, J. J., Silvério, J., Han, J., Rao, K., Pertsch, K., Hausman, K., Go, K., Gopalakrishnan, K., Goldberg, K., Byrne, K., Oslund, K., Kawaharazuka, K., Zhang, K., Majd, K., Rana, K., Srinivasan, K., Chen, L. Y., Pinto, L., Tan, L., Ott, L., Lee, L., Tomizuka, M., Du, M., Ahn, M., Zhang, M., Ding, M., Srirama, M. K., Sharma, M., Kim, M. J., Kanazawa, N., Hansen, N., Heess, N., Joshi, N. J., Suenderhauf, N., Palo, N. D., Shafiullah, N. M. M., Mees, O., Kroemer, O., Sanketi, P. R., Wohlhart, P., Xu, P., Sermanet, P., Sundaresan, P., Vuong, Q., Rafailov, R., Tian, R., Doshi, R., Martín-Martín, R., Mendonca, R., Shah, R., Hoque, R., Julian, R., Bustamante, S., Kirmani, S., Levine, S., Moore, S., Bahl, S., Dass, S., Song, S., Xu, S., Haldar, S., Adebola, S., Guist, S., Nasiriany, S., Schaal, S., Welker, S., Tian, S., Dasari, S., Belkhale, S., Osa, T., Harada, T., Matsushima, T., Xiao, T., Yu, T., Ding, T., Davchev, T., Zhao, T. Z., Armstrong, T., Darrell, T., Jain, V., Vanhoucke, V., Zhan, W., Zhou, W., Burgard, W., Chen, X., Wang, X., Zhu, X., Li, X., Lu, Y., Chebotar, Y., Zhou, Y., Zhu, Y., Xu, Y., Wang, Y., Bisk, Y., Cho, Y., Lee, Y., Cui, Y., hua Wu, Y., Tang, Y., Zhu, Y., Li, Y., Iwasawa, Y., Matsuo, Y., Xu, Z., and Cui, Z. J. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023. 28

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. 8

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. 3, 6, 7, 8, 9, 17, 24, 27

Du, Y., Yang, M., Dai, B., Dai, H., Nachum, O., Tenenbaum, J. B., Schuurmans, D., and Abbeel, P. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023a. 15

Du, Y., Yang, M., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J. B., et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023b. 15

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pp. 5842–5850, 2017. 7, 8, 14, 28

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. 1, 3, 8, 28

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. 8, 25

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. 24

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022. 9

Hu, X., Li, J., Zhan, X., Jia, Q.-S., and Zhang, Y.-Q. Query-policy misalignment in preference-based reinforcement learning. *arXiv preprint arXiv:2305.17400*, 2023. 3

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. 15

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. 1, 2, 4, 5, 6, 7, 9, 14, 26

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. 9

Korbar, B., Tran, D., and Torresani, L. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6232–6242, 2019. 5, 15, 24

Kumar, A., Singh, A., Ebert, F., Nakamoto, M., Yang, Y., Finn, C., and Levine, S. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022. 8

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895, 2020a. 8

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020b. 8

Lee, K., Smith, L. M., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, pp. 6152–6163. PMLR, 2021. 3

Li, J., Hu, X., Xu, H., Liu, J., Zhan, X., Jia, Q.-S., and Zhang, Y.-Q. Mind the gap: Offline policy optimization for imperfect rewards. *arXiv preprint arXiv:2302.01667*, 2023. 14

Li, Y., Wei, C., and Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 16

Li, Y., Gao, T., Yang, J., Xu, H., and Wu, Y. Phasic self-imitative reduction for sparse-reward goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pp. 12765–12781. PMLR, 2022. 4

Liu, H.-T. D., Williams, F., Jacobson, A., Fidler, S., and Litany, O. Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–13, 2022. 16

Liu, J., Huang, X., Zheng, J., Liu, Y., and Li, H. Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6252–6261, 2023. 9

Ma, J. Y., Yan, J., Jayaraman, D., and Bastani, O. Offline goal-conditioned reinforcement learning via f-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323, 2022. 9

Ma, Y. J., Kumar, V., Zhang, A., Bastani, O., and Jayaraman, D. LIV: Language-image representations and rewards for robotic control. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23301–23320. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/ma23b.html. 1, 2, 4, 5, 6, 7, 8, 9, 14, 16, 24, 26, 27

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. In *The Eleventh International Conference on Learning Representations*, 2023b. 2, 4, 5, 6, 7, 8, 9, 14, 26, 27

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, 2022. 9

Mendonca, R., Bahl, S., and Pathak, D. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023. 8

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 16

Myers, V., He, A. W., Fang, K., Walke, H. R., Hansen-Estruch, P., Cheng, C.-A., Jalobeanu, M., Kolobov, A., Dragan, A., and Levine, S. Goal representations for instruction following: A semi-supervised language interface to control. In *Conference on Robot Learning*, pp. 3894–3908. PMLR, 2023. 8

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, 2022. 1, 2, 4, 5, 6, 7, 9, 14, 27

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 14–18 Dec 2023. 1, 2, 4, 6, 7, 9, 14, 25, 26

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999. 4

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 5

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826, 2020. 15

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pp. 17359–17371. PMLR, 2022. 8

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 2

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, 2021a. 9, 14, 24, 26

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, 2021b. 7, 9

Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., and Darrell, T. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pp. 416–426. PMLR, 2023. 8

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 15

Reid, M., Yamada, Y., and Gu, S. S. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*, 2022. 9

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. 7, 26

Seo, Y., Lee, K., James, S. L., and Abbeel, P. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022. 8

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. 2, 6, 7, 9, 14

Shah, R., Martín-Martín, R., and Zhu, Y. Mutex: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. 15

Shah, R. M. and Kumar, V. Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning*, pp. 9465–9476. PMLR, 2021. 8

Shi, F., Petriu, E., and Laganiere, R. Sampling strategies for real-time action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2595–2602, 2013. 4, 15

Shridhar, M., Manuelli, L., and Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023. 15

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 28

Virmaux, A. and Scaman, K. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018. 16

Walke, H. R., Black, K., Zhao, T. Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A. W., Myers, V., Kim, M. J., Du, M., et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023. 17, 25

Wang, G., Cheng, S., Zhan, X., Li, X., Song, S., and Liu, Y. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*, 2023. 28

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 4, 15

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. 8, 27

Xiao, T., Radosavovic, I., Darrell, T., and Malik, J. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 8

Yarats, D., Fergus, R., and Kostrikov, I. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *9th International Conference on Learning Representations, ICLR 2021*, 2021. 8

Zakharov, A., Guo, Q., and Fountas, Z. Long-horizon video prediction using a dynamic latent hierarchy. *arXiv preprint arXiv:2212.14376*, 2022. 15

Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu. 8

Zhang, Y., Zhao, J., Chen, Z., Mi, S., Zhu, H., and Geng, X. A closer look at video sampling for sequential action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1, 4, 5, 15

Zhi, Y., Tong, Z., Wang, L., and Wu, G. Mgsampler: An explainable sampling strategy for video action recognition. In *Proceedings of the IEEE/CVF International conference on Computer Vision*, pp. 1513–1522, 2021. 4, 15

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023. 1, 15

A. Additional Discussions on Related Works

A.1. In-depth Comparisons with Closely Related Studies

This section provides detailed comparisons with several closely related works, including R3M (Nair et al., 2023), LIV (Ma et al., 2023a), and Voltron (Karamcheti et al., 2023).

R3M (Nair et al., 2023) designs two separate loss functions, one is time contrastive learning (Sermanet et al., 2018) to enforce temporal consistency and an alignment loss based on LOReL (Nair et al., 2022) to achieve vision-language alignment:

$$\begin{aligned}
\mathcal{L}_{\text{tcn}}(\phi) &= \frac{1}{B}\sum_{b=1}^{B}\left[-\log\frac{\exp(\mathcal{S}(\phi(o_i^b),\phi(o_j^b))}{\exp(\mathcal{S}(\phi(o_i^b),\phi(o_j^b)))+\exp(\mathcal{S}(\phi(o_i^b),\phi(o_k^b)))+\exp(\mathcal{S}(\phi(o_i^b),\phi(o_j^{\neq b}))}\right]\\
\mathcal{L}_{\text{alignment}}(\phi,\theta) &= \frac{1}{B}\sum_{b=1}^{B}\left[-\log\frac{\exp(\mathcal{G}_\theta(\phi(o_0^b),\phi(o_j^b);\psi(l^b)))}{\exp(\mathcal{G}_\theta(\phi(o_0^b),\phi(o_j^b);\psi(l^b)))+\exp(\mathcal{G}_\theta(\phi(o_0^b),\phi(o_i^b);\psi(l^b)))+\exp(\mathcal{G}_\theta(\phi(o_0^{\neq b}),\phi(o_j^{\neq b});\psi(l^{\neq b})))}\right]\\
\mathcal{L}_{\text{R3M}}(\phi,\theta) &= \lambda_1\mathcal{L}_{\text{tcn}}+\lambda_2\mathcal{L}_{\text{alignment}}+\lambda_3\|\phi(o_i)\|_1+\lambda_4\|\phi(o_i)\|_2,
\end{aligned} \tag{9}$$

where $k > j > i$ are the frame ids of video v^b, $\mathcal{S}$ is negative L2 distance, $\mathcal{G}_\theta$ is a reward model and $\lambda_{1,2,3,4}$ are the weight parameters. For the alignment loss, the image o_0^b is selected from the first 20% of the video. R3M employs o_0^b along with other randomly sampled images to align with language.

See from Eq. (9) that the choice of o_0^b appears somewhat ad-hoc. This makes the task progression features extracted by R3M always limit to starting from the first image, which may overlook some intermediary transitions within a video. Moreover, R3M presents a two-stage training, separately considering the temporal consistency and language grounding, which inevitably introduces complex trade-offs among these two potential contradicting objectives. In addition, see from $\mathcal{L}_{\text{R3M}}$ that R3M requires to balance a lot of hyperparameters of different loss, making it hard to tune. In contrast, DecisionNCE inherently merges the complex two-stage training processes into a simple and unified objective in Eq. (5) and Eq. (7), naturally marries language grounding and temporal consistency in a more principled and hyper-parameter-free manner, eliminating all hyper-parameters to balance these complex loss.

LIV (Ma et al., 2023a) trains temporally consistent vision representations using goal conditioned reinforcement learning based on VIP (Ma et al., 2023b), and utilizes CLIP (Radford et al., 2021a) to only align the static goal images with language:

$$\begin{aligned}
\mathcal{L}_{\text{VIP}}(\phi) &= \frac{1-\gamma}{B}\sum_{b=1}^{B}\left[-\mathcal{S}(\phi(o_i^b),\phi(g^b))\right]+\log\frac{1}{B}\sum_{b=1}^{B}\exp\left[\mathcal{S}(\phi(o_k^b),\phi(g^b))+r-\gamma\mathcal{S}(\phi(o_{k+1}^b),\phi(g^b))\right]\\
\mathcal{L}_{\text{CLIP}}(\phi,\psi) &= \frac{1-\gamma}{B}\sum_{b=1}^{B}\left[-\log\frac{\exp(1-\gamma)\mathcal{S}(\phi(g^b),\psi(l^b))}{\frac{1}{B}\sum_{b'=1}^{B}\left[\exp(1-\gamma)\mathcal{S}(\phi(g^{b'}),\psi(l^{b'}))\right]}\right]\\
\mathcal{L}_{\text{LIV}}(\phi,\psi) &= \lambda_1\mathcal{L}_{\text{VIP}}+\lambda_2\mathcal{L}_{\text{CLIP}},
\end{aligned} \tag{10}$$

where g^b is the last frame of video v^b. The employment of VIP loss aids in acquiring representations that adhere to temporal consistency. Nevertheless, aligning simply static goal images with language falls short of achieving trajectory-level grounding as single images cannot fully describe a dynamical behavior. In addition, see from Eq. (10) that LIV also faces a complex balance between temporal consistency and language grounding, but DecisionNCE solves this in an elegant and unified learning objective.

Moreover, note that LIV uses reinforcement learning to train representations, and thus must require correct reward signals. In LIV, the authors defined the rewards as 0 for goal frames and -1 for all other frames. However, it is widely known that hand-engineered reward design is challenging, and thus the human defined rewards are typically imperfect and may lead to unsatisfactory results (Li et al., 2023). In contrast, DecisionNCE is extended from the popular reward learning method, the Bradley-Terry model in Preference-based Reinforcement Learning or widely known Reinforcement Learning from Human Feedback (RLHF), to train representations, leading to fewer assumptions on rewards than LIV.

Voltron (Karamcheti et al., 2023) proposed two tasks, language conditioning and language generation, to jointly train images and language encoders. However, the selected image sequences are always short with a fixed length (the sequence length is fixed to only 2), leading to a lack of trajectory-level alignment. We do not compare with Voltron, as it is pre-trained solely on small-scale Something-Something-V2 (Goyal et al., 2017) dataset.

A.2. Language-Conditioned Decision Making

To achieve generalist agent, another branch of works is end-to-end training language-conditioned vision-based agent (Jang et al., 2022; Reed et al., 2022; Brohan et al., 2022; Zitkovich et al., 2023; Shah et al., 2023; Shridhar et al., 2023). These approaches, however, is quite data-hungry, requiring tremendous expensive in-domain data. Some works reuse the broad out-of-domain data to improve decision making via a hierarchical structure, such as vidio planning (Black et al., 2023; Du et al., 2023b;a; Ajay et al., 2023), but does not show how representation affects decision making. DecisionNCE, however, effectively reuse out-of-domain data to extract decision-centric representations for efficient downstream policy learning, largely eliminating the significant demands for extensive task-specific data collection.

A.3. Other Related Works

We notice that trajectory-level grounding is closely related with action recognition and video prediction. Action recognition (Shi et al., 2013; Wang et al., 2018; Korbar et al., 2019; Zhi et al., 2021; Zhang et al., 2023) centers on detecting and categorizing movements of humans or objects in video sequences. Also, video prediction (Oprea et al., 2020; Zakharov et al., 2022) aims to forecast upcoming frames or events within a video. We can see that the core of these technologies also requires to extract local and global task progressions embedded in image sequences. Therefore, It would be an interesting topic to adopt the techniques from action recognition and video prediction to improve decision-centric vision-language representation learning.

B. Proof of Eq. (8)

In this section, we briefly present the proof for Eq. (8). Here, we first review the conclusion in Eq. (8). In specific, the probability of image o_t^+ being selected as a goal is not isotropic but progressively increases as the video progresses. Mathematically, the probability for $o_t^+, 0 < t < h$ being a goal image grows monotonically w.r.t its time stamp t:

$$P(o_t^+ \text{ is selected as goal}) = \frac{1}{h}\sum_{i=1}^{i<t}\frac{1}{h-i}$$

Proof.

$$\begin{aligned}
&P(o_t^+ \text{ is selected as goal}) \\
&\overset{1}{=} \sum_{i=1}^{h} P(o_i^+ \text{ is selected as start } \cap o_t^+ \text{ is selected as goal when } o_i^+ \text{ is start}) \\
&\overset{2}{=} \sum_{i=1}^{h} P(o_i^+ \text{ is selected as start}) \times P(\mathrm{o_t^+} \text{ is selected as goal when } o_i^+ \text{ is start}) \\
&\overset{3}{=} \frac{1}{h}\sum_{i=1}^{i<t} P(\mathrm{o_t^+} \text{ is selected as goal when } o_i^+ \text{ is start}) + \frac{1}{h}\sum_{i=1}^{t\le i\le h} P(\mathrm{o_t^+} \text{ is selected as goal when } o_i^+ \text{ is start}) \\
&\overset{4}{=} \frac{1}{h}\sum_{i=1}^{i<t} P(\mathrm{o_t^+} \text{ is selected as goal when } o_i^+ \text{ is start}) + 0 \\
&\overset{5}{=} \frac{1}{h}\sum_{i=1}^{i<t}\frac{1}{h-i},
\end{aligned} \tag{11}$$

where the 3rd equation holds since the start image o_i is uniformly sampled from the whole video, and thus $P(o_i^+ \text{ is selected as start}) = \frac{1}{h}$ where h is the video length. The 4th equation holds because it is impossible to select o_t^+ as a goal image when $i > t$, as o_t^+ is selected from the following images after o_i with $t > i$. The 5th equation holds because o_t^+ is uniformly selected from the following frames after o_i, and thus the remaining video length becomes $h - i$. □

C. Theoretical Analysis (Informal)

What does DecisionNCE update do? For a mechanistic understanding of DecisionNCE, it is useful to analyze the gradient of its loss function $\mathcal{L}_{\text{DecisionNCE}}$. In the following, we present the gradient analysis of DecisionNCE-P (Definition 3.1) and DecisionNCE-T (Definition 3.2). We begin by introducing the main assumption used in our analysis.

Assumption C.1 (Differentiable and Lipschitz continuous function approximators)**.** *The vision encoder ϕ we trained is differentiable and L-Lipschitz continuous, i.e., $\|\nabla_x \phi(x)\| \leq L, \forall x \in \mathcal{O}$.*

Note that assumption C.1 is a mild assumption which is frequently utilized in plenty of works (Li et al., 2019; Miyato et al., 2018; Virmaux & Scaman, 2018; Liu et al., 2022).

DecisionNCE-P. According to the Definition 3.1 of the DecisionNCE-P:

$$\begin{aligned} P_P\left[\sigma^+ \succ \sigma^-\right] &= \frac{\exp(\mathcal{S}(\phi(o_{n+m}^+); \psi(l^+)) - \mathcal{S}(\phi(o_n^+); \psi(l^+)))}{\exp(\mathcal{S}(\phi(o_{n+m}^+); \psi(l^+)) - \mathcal{S}(\phi(o_n^+); \psi(l^+))) + \exp(\mathcal{S}(\phi(o_{n+m}^-); \psi(l^+)) - \mathcal{S}(\phi(o_n^-); \psi(l^+)))} \\ &= \frac{1}{1 + \exp(\mathcal{S}(\phi(o_{n+m}^-); \psi(l^+)) - \mathcal{S}(\phi(o_n^-); \psi(l^+)) - \mathcal{S}(\phi(o_{n+m}^+); \psi(l^+)) + \mathcal{S}(\phi(o_n^+); \psi(l^+)))} \end{aligned} \tag{12}$$

The loss function of DecisionNCE-P is as follows:

$$\begin{aligned} \mathcal{L}_{\text{DecisionNCE-P}}(\phi, \psi) &= -\log P_P\left[\sigma^+ \succ \sigma^-\right] \\ &= \log\left[1 + \exp(\mathcal{S}(\phi(o_{n+m}^-); \psi(l^+)) - \mathcal{S}(\phi(o_n^-); \psi(l^+)) - \mathcal{S}(\phi(o_{n+m}^+); \psi(l^+)) + \mathcal{S}(\phi(o_n^+); \psi(l^+)))\right] \end{aligned} \tag{13}$$

The gradient with respect to the parameters θ of ψ cab be written as:

$$\begin{aligned} &\nabla_\theta \mathcal{L}_{\text{DecisionNCE-P}}(\phi, \psi_\theta) \\ &= \left(1 - P_P\left[\sigma^+ \succ \sigma^-\right]\right)\left[\left[\mathcal{S}'(\phi(o_{n+m}^-), \psi_\theta(l^+)) - \mathcal{S}'(\phi(o_n^-), \psi_\theta(l^+))\right]\nabla_\theta \psi_\theta(l^+) - \left[\mathcal{S}'(\phi(o_{n+m}^+), \psi_\theta(l^+)) - \mathcal{S}'(\phi(o_n^+), \psi_\theta(l^+))\right]\nabla_\theta \psi_\theta(l^+)\right] \\ &= \left(1 - P_P\left[\sigma^+ \succ \sigma^-\right]\right)\underbrace{\left[\mathcal{S}'(\phi(o_{n+m}^-), \psi_\theta(l^+)) - \mathcal{S}'(\phi(o_n^-), \psi_\theta(l^+)) - \mathcal{S}'(\phi(o_{n+m}^+), \psi_\theta(l^+)) + \mathcal{S}'(\phi(o_n^+), \psi_\theta(l^+))\right]}_{g_P(o_n^-, o_{n+m}^-, o_n^+, o_{n+m}^+, l^+)}\nabla_\theta \psi_\theta(l^+) \\ &= \left(1 - P_P\left[\sigma^+ \succ \sigma^-\right]\right) g_P(o_n^-, o_{n+m}^-, o_n^+, o_{n+m}^+, l^+)\nabla_\theta \psi_\theta(l^+) \end{aligned} \tag{14}$$

where we denote $\nabla_b \mathcal{S}(a, b)$ as $\mathcal{S}'(a, b)$. As we choose metric $\mathcal{S}$ as cosine similarity throughout the paper, $\mathcal{S}'(a, b) = \frac{a}{|a||b|} - \mathcal{S}\frac{b}{|b|^2}$. We perform a Taylor expansion on $\mathcal{S}'(\phi(x), \psi_\theta(l^+))$ at $x = o_n$:

$$\mathcal{S}'(\phi(x), \psi_\theta(l^+)) = \mathcal{S}'(\phi(o_n), \psi_\theta(l^+)) + \nabla_\phi \mathcal{S}'(\phi(x); \psi_\theta(l^+))\nabla_x \phi(x)|_{x=o_n}(x - o_n) + \mathcal{R}(x) \tag{15}$$

where $R(x)$ is higher-order infinitesimal of $(x - o_n)$. Hence we have:

$$\mathcal{S}'(\phi(o_n), \psi_\theta(l^+)) - \mathcal{S}'(\phi(o_{n+m}), \psi_\theta(l^+) = -\nabla_\phi \mathcal{S}'(\phi(x); \psi_\theta(l^+))\nabla_x \phi(x)|_{x=o_n}(o_{n+m} - o_n) - \mathcal{R}(o_{n+m}) \tag{16}$$

And

$$\begin{aligned} g_P(o_n^-, o_{n+m}^-, o_n^+, o_{n+m}^+, l^+) =& \nabla_\phi \mathcal{S}'(\phi(x); \psi_\theta(l^+))\nabla_x \phi(x)|_{x=o_n^-}(o_{n+m}^- - o_n^-) + \mathcal{R}(o_{n+m}^-) \\ &- \nabla_\phi \mathcal{S}'(\phi(x); \psi_\theta(l^+))\nabla_x \phi(x)|_{x=o_n^+}(o_{n+m}^+ - o_n^+) - \mathcal{R}(o_{n+m}^+) \end{aligned} \tag{17}$$

where $\nabla_a S'(a, b) = \nabla_a \nabla_b S(a, b) = \nabla_a(\frac{a}{|a||b|} - \mathcal{S}\frac{b}{|b|^2})$ is bounded. According to assumption C.1, $\nabla_x \phi(x)$ is bounded as well. When m is set to a large value, there may be a substantial difference between o_{n+m} and o_n, resulting in a situation where $g_P(o_n^-, o_{n+m}^-, o_n^+, o_{n+m}^+, l^+)$ contributes more to the loss gradient $\nabla_\theta \mathcal{L}_{\text{DecisionNCE-P}}(\phi, \psi_\theta)$. In our work, m is randomly set from 1 to the maximum segment length. In contrast, some other works (such as VIP-L in Equ.4 of (Ma et al., 2023a)) resemble setting $m = 1$, leading to a minor difference between o_{n+m} and o_n, resulting in a slower change in the loss. This can be less effective in training language encoder compared to our method.

DecisionNCE-T. According to the definition 3.2 of the DecisionNCE-T:

$$\begin{aligned} P_T\left[\sigma^+ \succ \sigma^-\right] &= \frac{\exp\left(\mathcal{S}(\phi(o_{n+m}^+) - \phi(o_n^+); \psi(l^+))\right)}{\exp\left(\mathcal{S}(\phi(o_{n+m}^+) - \phi(o_n^+); \psi(l^+))\right) + \exp\left(\mathcal{S}(\phi(o_{n+m}^-) - \phi(o_n^-); \psi(l^+))\right)} \\ &= \frac{1}{1 + \exp\left(\mathcal{S}(\phi(o_{n+m}^-) - \phi(o_n^-); \psi(l^+)) - \mathcal{S}(\phi(o_{n+m}^+) - \phi(o_n^+); \psi(l^+))\right)} \end{aligned} \tag{18}$$

The loss function of DecisionNCE-T is as follow:

$$\begin{aligned}\mathcal{L}_{\text{DecisionNCE-T}}(\phi,\psi) &= -\log P_T\left[\sigma^+ \succ \sigma^-\right] \\ &= \log\left[1+\exp\left(\mathcal{S}(\phi(o_{n+m}^-)-\phi(o_n^-);\psi(l^+)) - \mathcal{S}(\phi(o_{n+m}^+)-\phi(o_n^+);\psi(l^+))\right)\right]\end{aligned} \tag{19}$$

The gradient with respect to the parameters θ of ψ cab be written as:

$$\begin{aligned}\nabla_\theta\mathcal{L}_{\text{DecisionNCE-T}}(\phi,\psi_\theta) &= \left(1-P_T\left[\sigma^+ \succ \sigma^-\right]\right)\underbrace{\left[\mathcal{S}'(\phi(o_{n+m}^-)-\phi(o_n^-);\psi(l^+)) - \mathcal{S}'(\phi(o_{n+m}^+)-\phi(o_n^+);\psi(l^+))\right]}_{g_T(o_n^-,o_{n+m}^-,o_n^+,o_{n+m}^+,l^+)}\nabla_\theta\psi_\theta(l^+) \\ &= \left(1-P_T\left[\sigma^+ \succ \sigma^-\right]\right) g_T(o_n^-,o_{n+m}^-,o_n^+,o_{n+m}^+,l^+)\nabla_\theta\psi_\theta(l^+)\end{aligned} \tag{20}$$

We perform Taylor expansion on $\mathcal{S}'(\phi(x)-\phi(o_n),\psi_\theta(l^+))$ at $x=o_n$:

$$\mathcal{S}'(\phi(x)-\phi(o_n),\psi_\theta(l^+)) = \mathcal{S}'(0,\psi_\theta(l^+)) + \nabla_\phi\mathcal{S}'(\phi(x)-\phi(o_n);\psi_\theta(l^+))\nabla_x\phi(x)|_{x=o_n}(x-o_n) + \mathcal{R}(x) \tag{21}$$

where $\mathcal{R}(x)$ is higher-order infinitesimal of $(x-o_n)$. Hence g_T can be written as:

$$\begin{aligned}g_T(o_n^-,o_{n+m}^-,o_n^+,o_{n+m}^+,l^+) &= \mathcal{S}'(0,\psi_\theta(l^+)) + \nabla_\phi\mathcal{S}'(\phi(x)-\phi(o_n^-);\psi_\theta(l^+))\nabla_x\phi(x)|_{x=o_n^-}(o_{n+m}^- - o_n^-) + \mathcal{R}(o_{n+m}^-) \\ &\quad - \mathcal{S}'(0,\psi_\theta(l^+)) - \nabla_\phi\mathcal{S}'(\phi(x)-\phi(o_n^+);\psi_\theta(l^+))\nabla_x\phi(x)|_{x=o_n^+}(o_{n+m}^+ - o_n^+) - \mathcal{R}(o_{n+m}^+) \\ &= \nabla_\phi\mathcal{S}'(\phi(x)-\phi(o_n^-);\psi_\theta(l^+))\nabla_x\phi(x)|_{x=o_n^-}(o_{n+m}^- - o_n^-) + \mathcal{R}(o_{n+m}^-) \\ &\quad - \nabla_\phi\mathcal{S}'(\phi(x)-\phi(o_n^+);\psi_\theta(l^+))\nabla_x\phi(x)|_{x=o_n^+}(o_{n+m}^+ - o_n^+) - \mathcal{R}(o_{n+m}^+)\end{aligned} \tag{22}$$

Similarly, in our method, m is randomly set from 1 to the maximum segment length, implying a faster descent in loss compared to setting $m=1$ alone.

D. Additional Results

In this section, we provide more cases to demonstrate that our DecisionNCE-P/T can learn accurate and generalizable reward, and conduct a deeper analysis.

D.1. In-distribution reward

We randomly selected segments from EPIC-KITCHEN-100 dataset (Damen et al., 2018) with their corresponding text annotations to visualize the rewards curves learned through DecisionNCE-P/T. As shown in Figure 12, DecisionNCE accurately assigns rewards to different video frames across various action types, scenes, and target objects, resulting in a smooth and trend-correct reward curve. Next, we will delve deeper into discussing some intriguing and insightful characteristics observed in specific reward curves.

Slope of curves: The slope of the reward curves provides an insight into the task's progression. For example, curves (c) and (g) demonstrate varying growth rates. This variation is due to the differing amplitudes of actions in the videos over time, as can be seen in the accompanying images.

Robustness in Complex Tasks: In the context of kitchen tasks, which may involve complex actions with subgoals such as `drying hands`, `rinsing a cloth`, or `cutting something`, the robustness of DecisionNCE is notable. As illustrated by curve (h), the DecisionNCE curve fluctuates, reflecting the presence of subgoals. This variation is a reasonable response to the complexity of the tasks.

D.2. Zero-shot reward

To assess the generalizability of the reward learned through DecisionNCE, we also visualize the reward curves on Bridgedata-V2 (Walke et al., 2023), a open-source robotics manipulation dataset. The results are presented in Figure 13. DecisionNCE successfully captures the correct trend, demonstrating promising generalizability on out-of-domain data. It is important to note the significant domain gap between EPIC-KITCHEN-100 and Bridgedata-V2 which encompasses differences in embodiments, scenes, and view points.

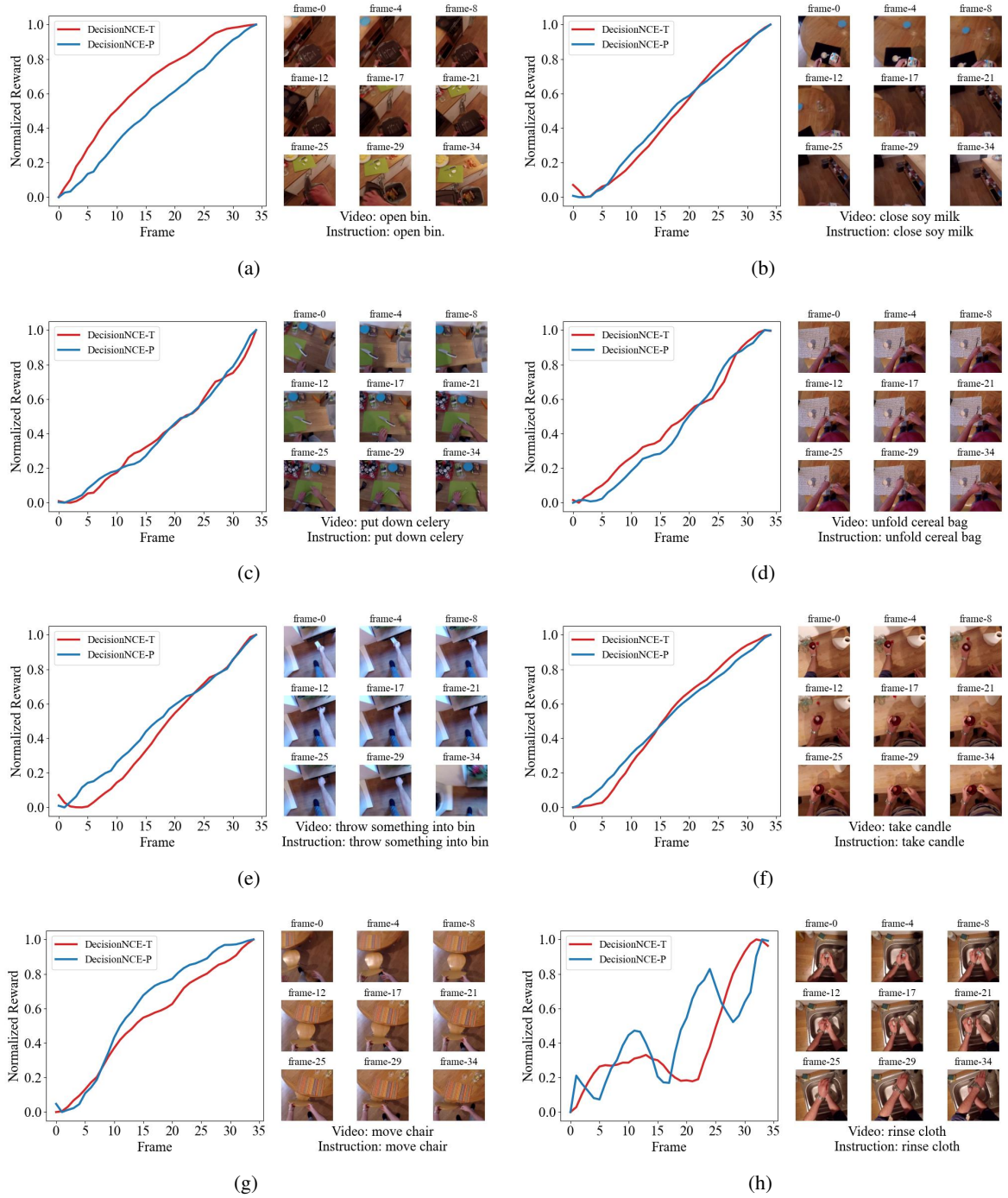


Figure 12. In-distribution Reward

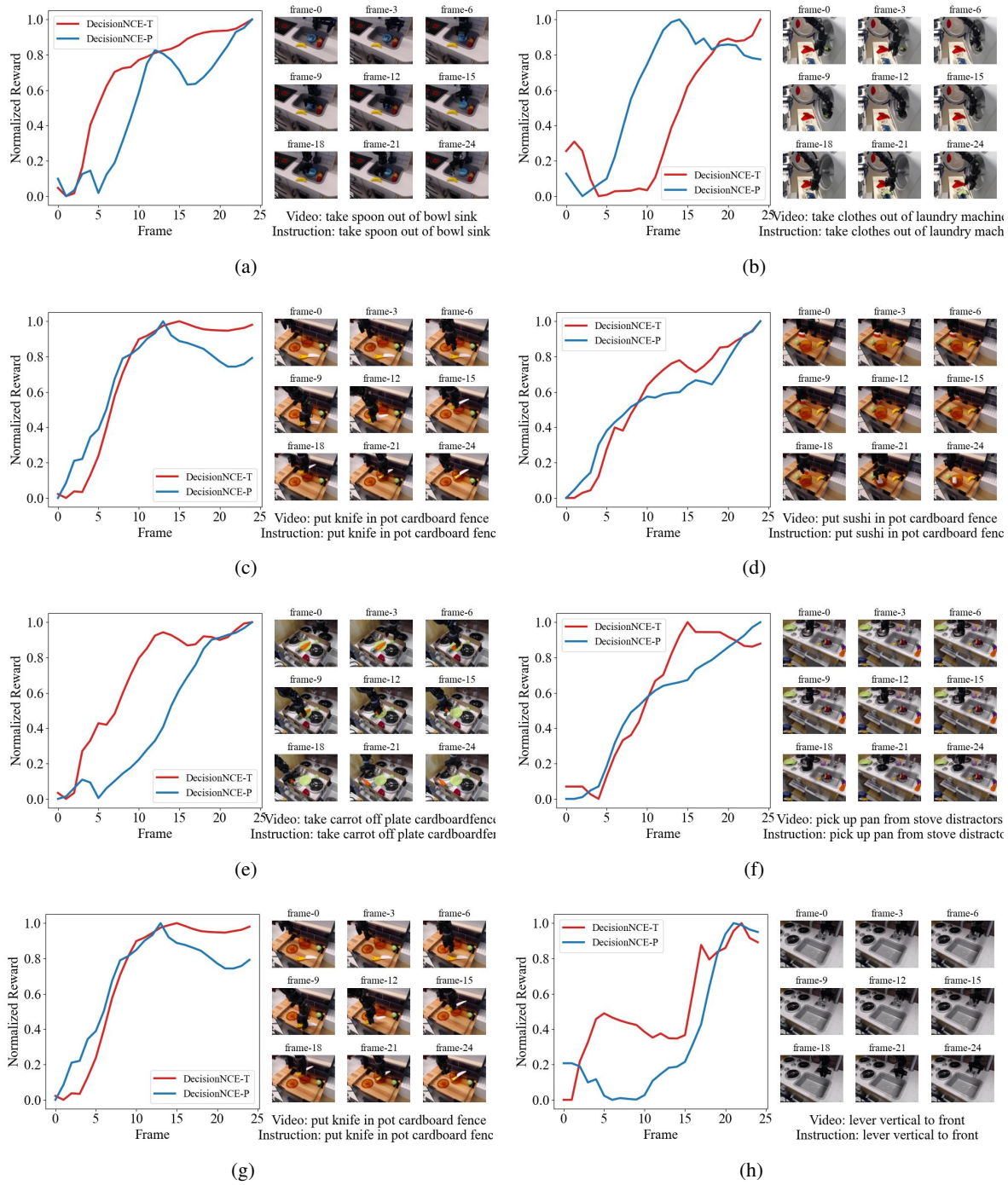


Figure 13. Zero-shot Reward

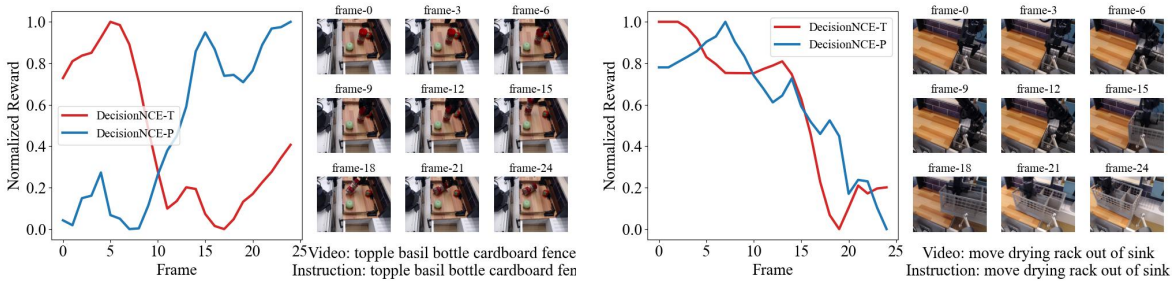
D.2.1. FAILURE CASES ANALYSIS

Indeed, while we observe numerous successful instances, there are also several results that deviate from our expectations. Upon a thorough examination of these unsuccessful cases, we discern two main factors. To provide a clearer understanding of these issues, we select two representative cases as shown in Figure 14 for in-depth analyses.

Out-of-domain skills. Understanding actions at the trajectory level is notably more challenging to transfer than image perception skills. As a result, DecisionNCE demonstrates impressive performance with in-domain skills like `pick-up`, `put-down`, `open`, and `take-out`, maintaining its effectiveness even when there are changes in the target object or action sequence. However, it struggles when confronted with instructions describing a novel skill not previously encountered in the Epic-Kitchen dataset. For instance, in the case of `Topple` as shown in (a), DecisionNCE is unable to accurately track the correct task progression.

Dataset-specific task bias. The lack of precision in text annotations within the EPIC-KITCHEN-100 dataset results in a dataset-specific task bias, as learned by DecisionNCE. A prime example of this can be seen in (b), where the task `move drying rack out of sink` exhibits an incorrect reward trend according to DecisionNCE's learning. This issue arises crucially due to a distinctive pattern in EPIC-KITCHEN-100. In EPIC-KITCHEN-100 dataset, most `move` actions result in the target object moving away from the center of the field of view. However, in this Bridgedata-V2 example, the scenario is reversed, which contributes to this erroneous reward curve.

In conclusion, the observed failures are primarily attributed to the limitations in the quality and scope of the dataset. For future advancements, it is imperative to enhance these aspects by incorporating a broader, more diverse, and higher-quality range of pre-training datasets. Please see Appendix F for more discussions on limitations and solutions.



(a) (b)

Figure 14. Zero-shot failure cases

D.3. Reward for mismatched video-instruction pair

As metioned in Section 5, DecisionNCE effectively identifies mismatched video-language pairs. In this section, we delve deeper into the rewards assigned by DecisionNCE to these mismatched pairs, offering a detailed discussion of our findings.

D.3.1. COMPLETELY IRRELEVANT VIDEO-INSTRUCTIONS PAIRS

For most cases, altering either the target object or the action in the instruction renders it incongruent with the task progression shown in the video. Under these **Completely Irrelevant Video-Instructions Pairs**, the resulting reward curve tends to lack a clear trend and appears chaotic, as illustrated in the Figure 15. Significantly, DecisionNCE does not erroneously interpret these unmatched instructions as correct, even when there is similarity in actions or objects. This demonstrates the model's strong robustness in differentiating between matched and mismatched pairs.

Certainly, mismatched video-instruction pairs are not always entirely irrelevant. Next, we will explore two intriguing cases to further illustrate the generalizability of DecisionNCE.

D.3.2. CONTRADICTORY VIDEO-INSTRUCTIONS PAIRS

Given that tasks in EPIC-KITCHEN-100 are predominantly single-step, many have corresponding mirror tasks, such as `Open` and `Close`, `Pick-up` and `Put-down`, `Pull` and `Return`. With these mirrored instructions, the task progression

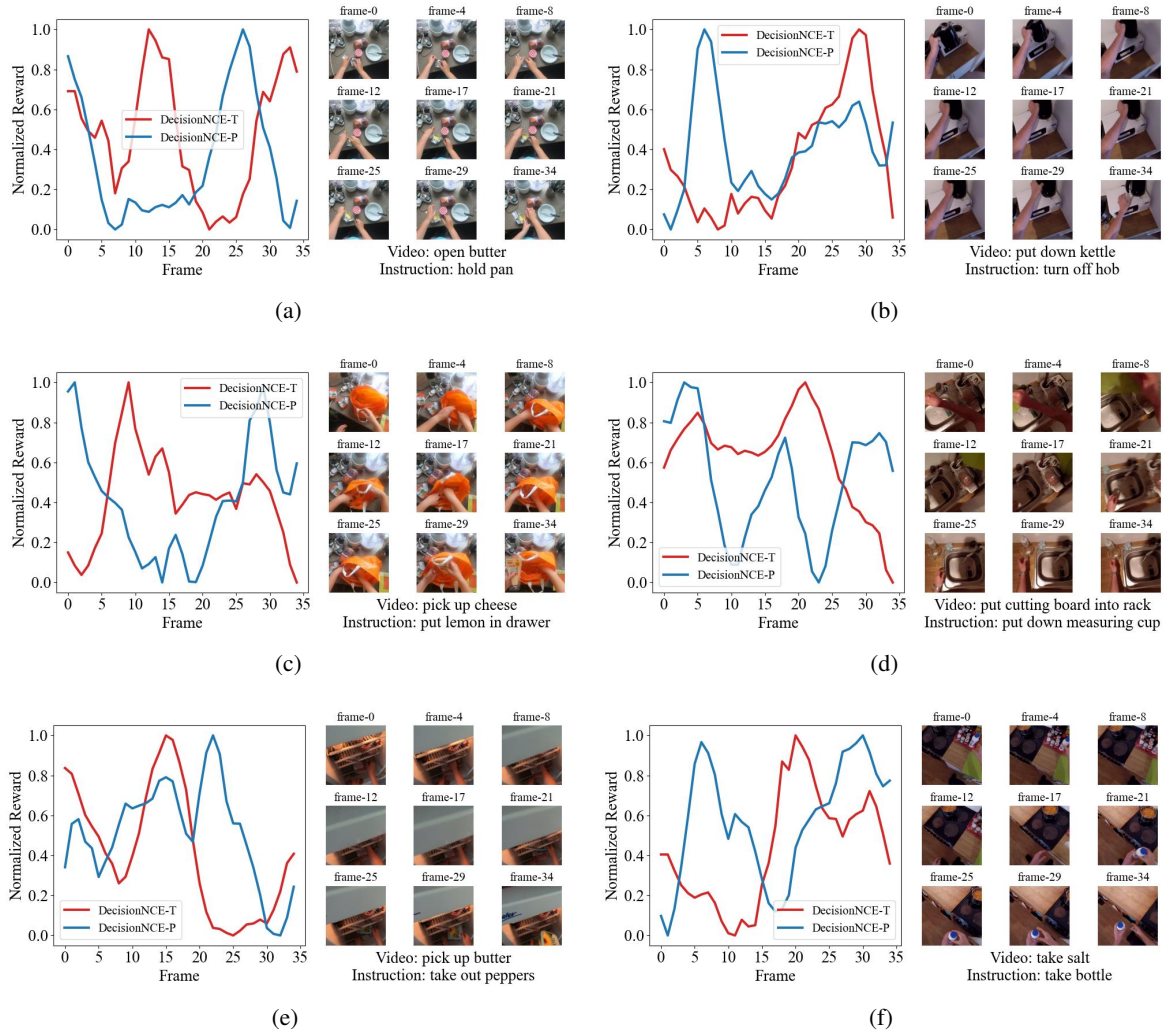


Figure 15. Irrelevant video-instruction pair reward

in the video is effectively the opposite. We have visualized the reward curve for these **Contradictory Video-Instruction Pairs** in a similar manner as before. As depicted in Figure 16, the reward curve aligns with our expectations: as the task progresses, the reward value tends to decrease.

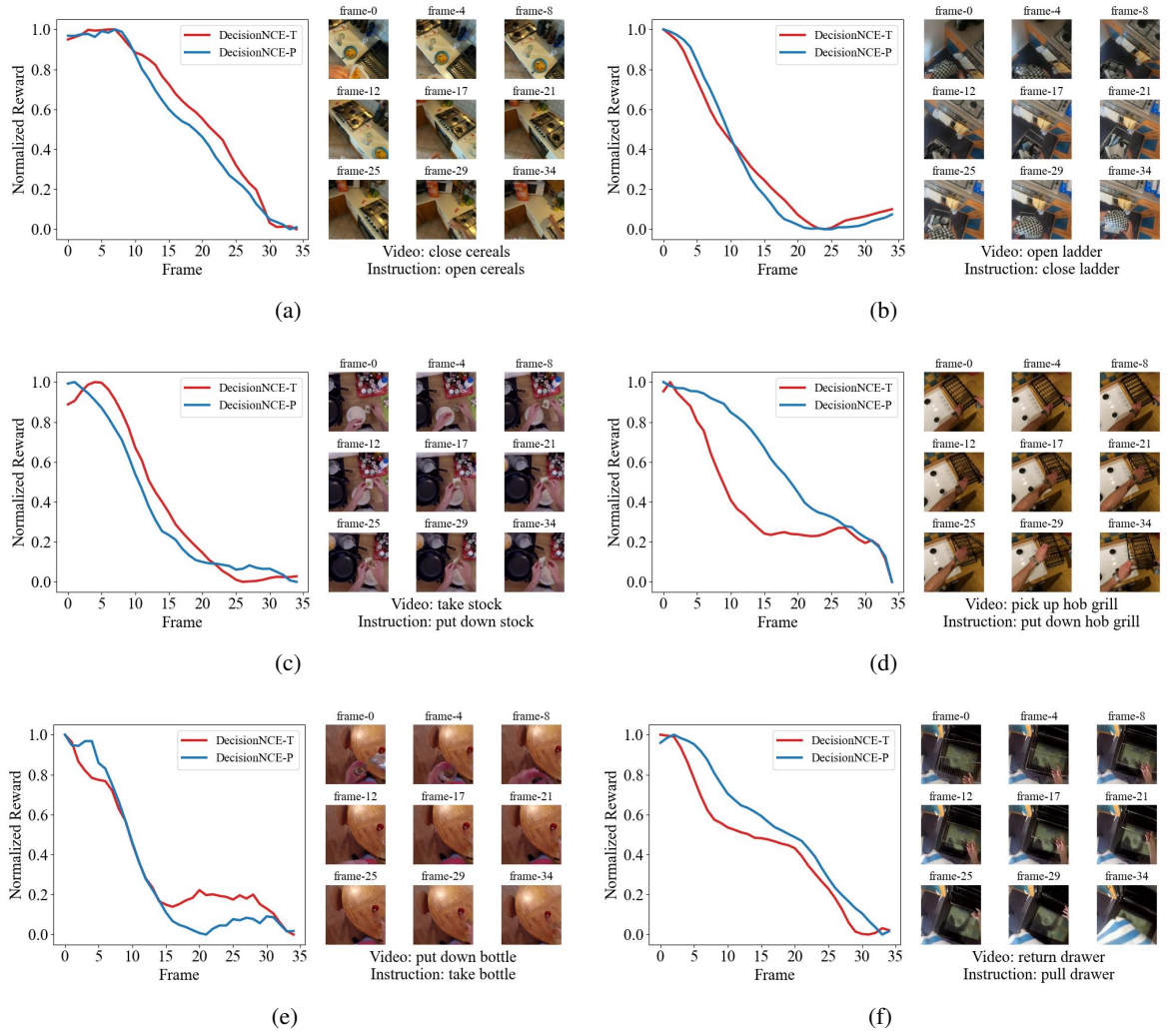
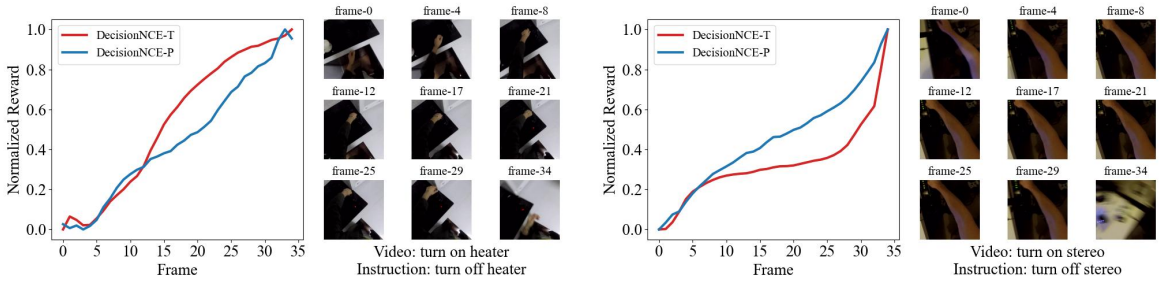


Figure 16. Contradictory video-instruction pair reward

Failure cases: We also discovered that certain mirror tasks pose significant challenges in differentiation from their original counterparts. As shown in Figure 17, tasks such as `Turn on` and `Turn off` an appliance, like a heater (a) or stereo (b), often involve almost identical movement trajectories. To accurately distinguish between these tasks, the model needs to discern the initial and final states of the target appliance, a task that becomes exceedingly difficult without supplemental annotations. In this instance, DecisionNCE fails to differentiate between these two tasks.

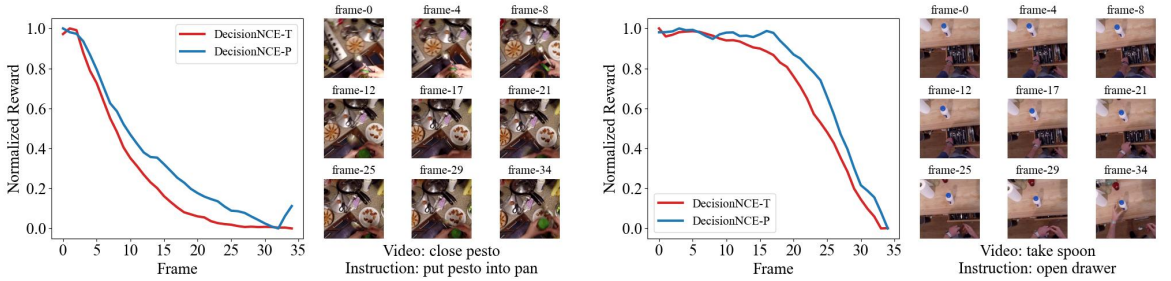
D.3.3. Indirectly relevant video-instructions pairs

Usually, the completion of a task typically involves accomplishing several sub-tasks or certain task subgoals. The Epic-Kitchen dataset, however, does not provide annotations with such granularity. For these sub-task instructions, they are indirectly related to the video. We have illustrated the reward curve for these **Indirectly Relevant Video-Instruction Pairs** in Figure 18. The results show that DecisionNCE effectively recognizes these sub-tasks, even in the absence of explicit human annotations. For instance, in the case of video (b), the annotation is `take spoon`, yet the task `close drawer` is concurrently completed. So when `open drawer` is used as the instruction, DecisionNCE appropriately assigns a reward curve with an opposing trend.

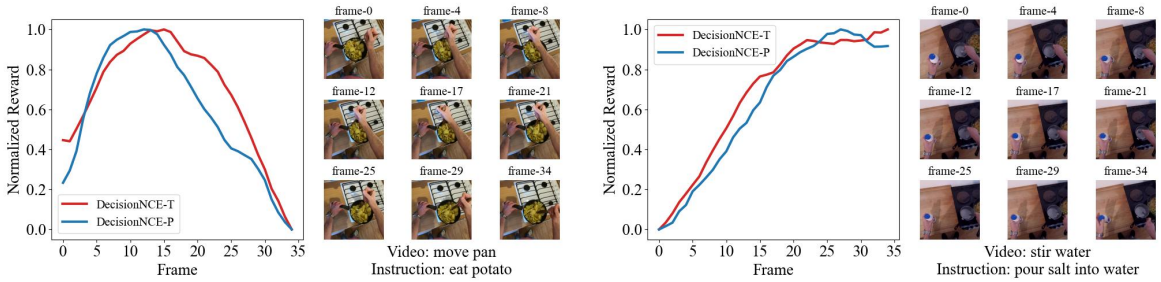


(a) (b)

Figure 17. Failure cases of contradictory pairs



(a) (b)



(c) (d)

Figure 18. Indirectly relevant video-instructions pairs reward

E. Experimental setups

In this section, we provide detailed experimental setups.

E.1. Vision-Language representation training

Following LIV (Ma et al., 2023a), we use a modified ResNet-50 (He et al., 2016) from CLIP (Radford et al., 2021a) for the vision encoder and a CLIP transformer for the language encoder. We initialize our DecisionNCE-P/T with CLIP model and train them on EPIC-KITCHEN-100 (Damen et al., 2018) under the same training setting. The training hyperparameters used during the pre-training are listed in Table 4. It is worth noting that our training objective in Algorithm 1 compares all mismatched pairs in mini-batch samples, which enjoys much higher training efficiency and thus only 20K training steps can obtain well-trained representations. Therefore, our training only take about 9 hours on four A100 GPUs, showing higher training efficiency compared to previous work.

Table 4. Hyper-parameters for pretraining .

config	value
training iteration	20K
optimizer	Adam
learning rate	1×10^{-5}
batch size	1024
weight decay	0.001
optimizer momentum	β_1, β_2=0.9,0.999
data augmentation	*RandomCropResize*

E.2. Experimental setup for Figure 5

In Eq. (6), we directly define the segment total rewards as the cosine similarity between the transition direction and the text embedding as follows:

$$\text{(Ours)} \ : \sum_t r_T(o_t, o_{t+1}; \phi, \psi, l) = \mathcal{S}(\phi(o_{n+m}) - \phi(o_n), \psi(l)), \tag{23}$$

Like the potential-based reward for DeicisionNCE-P, this transition-direction reward bypasses all intermediary transitions and only focuses on the binary start-goal transition directions. This has been proven advantageous in action recognition (Korbar et al., 2019). However, we would like to provide a more detailed experimental evaluation to further support this.

In specific, we also define transition-direction rewards considering some intermediary transitions as follows:

$$\text{(4 frames)} \ : \sum_t r_T(o_t, o_{t+1}; \phi, \psi, l) = \sum_{i=1}^{4} \mathcal{S}(\phi(o_{n+\lfloor \frac{m \times i}{4} \rfloor}) - \phi(o_{n+\lfloor \frac{m \times (i-1)}{4} \rfloor}), \psi(l)), \tag{24}$$

$$\text{(8 frames)} \ : \sum_t r_T(o_t, o_{t+1}; \phi, \psi, l) = \sum_{i=1}^{8} \mathcal{S}(\phi(o_{n+\lfloor \frac{m \times i}{8} \rfloor}) - \phi(o_{n+\lfloor \frac{m \times (i-1)}{8} \rfloor}), \psi(l)), \tag{25}$$

This treatment will include more intermediary transitions when calculating the preference model. Then, we use these objectives to train representations and use the obtained representations to serve as input for downstream Frankakitchen LCBC tasks. The results are presented in Figure 5. Results clearly demonstrate that including more intermediary points may complicate the training process and hamper downstream policy learning. Therefore, we focus solely on the teleporting start-goal transitions to train representations.

E.3. Experimental setup for Table 2

To validate the effectiveness of DecisionNCE-P/T in clustering the first, potentially task-irrelevant image embeddings into similar positions, we randomly select 100 trajectories from the EPIC-KITCHEN-100 dataset. Subsequently, we calculate and

compare the cosine similarities among the first image embeddings, denoted as $\phi(o_1)$, from these trajectories. Additionally, we assess the cosine similarities between $\phi(o_1)$ and $\psi(\bar{l})$, where $\psi(\bar{l})$ is derived by averaging all language embeddings within the EPIC-KITCHEN-100 annotations. The average result from both LIV and our DecisionNCE, acquired after five instances of random sampling, is presented in Table 2.

We chose LIV as the main baseline since it also builds on top of CLIP, which has the same representation embedding dimensions as our DecisionNCE. We also calculate the $\phi(o_1)$ similarities for CLIP, and obtain a value of **0.99**. However, we must mention that this comparison is not meaningful in our setting since CLIP does not aim to align text and image at the task level. Moreover, the EPIC-KITCHEN-100 dataset is significantly smaller in scope compared to the dataset used for pre-training CLIP, making a direct comparison of values unfair. Additionally, we do not compare against R3M in this experiment as the vision embedding and language embedding for R3M have different dimensions and thus cannot calculate the distances or similarities between these two embeddings. In addition, R3M chose $\mathcal{S}$ as negative L2 distance, whereas DecisionNCE chose $\mathcal{S}$ as cosine similarity. Therefore, these two methods are not comparable in this setting due to the differences in their evaluation metrics.

E.4. Environments

Frankakichen. Frankakitchen environment consists of a Franka robot interacting within a kitchen scene that contains diverse household kitchen objects such as microwaves and cabinets (Gupta et al., 2019). In our work, we follow the same 5 task definitions in LIV (Nair et al., 2023) and R3m (Nair et al., 2023), reported in Table 5. For each task, we have 1/3/5 demonstrations from (Nair et al., 2023), thus the total dataset includes only 5/15/25 episodes, and each episode has 50 environmental steps. These small datasets are not enough to train a vision-language agent from scratch but becomes possible by pretraining the vision-language representations on abroad out-of-domain data in advance.

Table 5. FrankaKitchen Tasks

Environment ID	Language Instruction
`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`

WidowX RealRobot. WidowX RealRobot environment contains a WidowX 6DoF robot arm executing diverse manipulation tasks. In specific, we evaluate on 5 distinct skills including {`pick & place, move, fold, flip, open & close`} comprising 9 sub-tasks as shown in Table 6. For each task, we collect around 100 demonstrations using the demonstration collection system in BridgedataV2 (Walke et al., 2023). For each demonstration, the environmental steps are around 50 steps. We collect more data for WidowX RealRobot than Frankakitchen as the real robot environment is far more stochastic, where the object and robot locations are randomly initialized, and the scene also has lots of randomly located distractors with varied shape and color.

Table 6. WidowX RealRobot Tasks

Environment ID	Language Instruction
`Red cup on silver pan`	`Pick up the red cup and place it on the silver pan`
`Red cup on red plate`	`Pick up the red cup and place it on the red plate`
`Duck on green plate`	`Pick up the duck and place it on the green plate`
`Duck in pot`	`Pick up the duck and place it in the pot`
`Move pot`	`Move the pot from right to left`
`Fold cloth`	`Fold the cloth from right to left`
`Flip the red cup upright`	`Flip the red cup upright`
`Open the microwave`	`Open the microwave`
`Close the microwave`	`Close the microwave`

E.5. LCBC Experiment Details

Here, we train a Language-Conditioned Behavior Cloning (LCBC) policy take the frozen vision-language representations as inputs. In details, the raw images o and language instructions l will be fed into the learned vision-language encoder and obtain the representations $\phi(o)$ and $\psi(l)$. Then, the policy $\pi(\mathrm{StopGrad}(\phi(o)), \mathrm{StopGrad}(\psi(l)))$ take these representations as input to fit the actions in demonstrations. Note that we freeze the pretrained representations and only optimize a small MLP to avoid overfit, following (Ma et al., 2023a;b; Nair et al., 2023; Karamcheti et al., 2023), as the downstream domain-specific data is quite small. The policy input also includes the proprioception information, which is concatenated with the frozen vision-language representation and fed into the downstream MLP.

(a) FrankaKitchen Left Camera View (b) WidowX RealRobot Side Camera View (c) WidowX RealRobot Wrist Camera View

Figure 19. Visual input view for LCBC policy.

FrankaKitchen. For FrankaKitchen experiments, we take single left camera view as the vision input, as shown in Figure 19 (a). We choose a small learning rate 1e-4, 16 batch size and 2e4 gradient steps, as the simulation dataset is quite small. Detailed hyper-parameters can be found in Table 7. We evaluate the policy for 25 episodes per 2e3 gradient steps and report the max success rate over the training following previous works (Ma et al., 2023a; Nair et al., 2023; Ma et al., 2023b).

WidowX RealRobot. For real robot experiments, we take both a side camera view and a wrist camera view as the vision inputs, as shown in Figure 19 (b)-(c). We choose a relatively large learning rate 1e-3, 64 batch size, as the real robot dataset contains more data. We also use data augmentation on the brightness to enhance the generalizability. See Table 7 for more details. We evaluate the success rate only for the last checkpoint after the training, as the real robot evaluation is extremely time-costly. We rollout 10 episodes for each checkpoint, and evaluate 3 policy checkpoints trained with different seeds for each pretrained representation.

Table 7. LCBC experiment details

	FrankaKitchen	WidowX RealRobot
MLP Architecture	[256, 256]	[256, 256]
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam
Learning Rate	1e-4	1e-3
Batch Size	16	64
Gradient Steps	2e4	200 epoch
Proprioception	Yes	Yes
Augmentation	No	Yes

Baseline Setups. We use the official released checkpoints of the vision/vision-language encoder from previous works. For VIP (Ma et al., 2023b) baseline, we use the pretrained DistilBERT (Sanh et al., 2019) as the language encoder, which is the same language encoder used in R3M (Nair et al., 2023), as VIP studies only the visual modality. For baselines including R3M (Nair et al., 2023), LIV (Ma et al., 2023a), VIP (Ma et al., 2023b) and CLIP (Radford et al., 2021a), we adhere the same hyper-parameters in Table 7 to train the downstream LCBC policy to ensure fair comparisons.

E.6. Frankakichen Planning Experiments

In this section, we'll describe more experimental details of language-reward planning. As we can directly roll out trajectories in FrankaKitchen environment, we combine ground-truth environment dynamics with reward induced by learned representation as the model in MPPI. This also aligns with the setting in (Ma et al., 2023b) and (Ma et al., 2023a). For CLIP, LIV, DecisionNCE-P and DecisionNCE-T, we all use $\mathcal{S}(\phi(o_{n+1}), \psi(l)) - \mathcal{S}(\phi(o_n), \psi(l))$ as reward. For R3M, the model contains a reward model $R(o_0, o_n, l)$ whose inputs are initial observation, present observation and language goal and output is a score estimating the task completion degree. So we take $R(o_0, o_{n+1}, l) - R(o_0, o_n, l)$ as reward for R3M. For all reward model we take discount factor $\gamma = 1.0$ to calculate return.

As is pointed out in (Ma et al., 2023a), tasks in FrankaKitchen demand a high level of exploration. So we follow the same setting and warmstart the action search with a fixed open-loop sequence came from expert demonstration. The warmstart stage won't directly solve the task, but will bring the robot end-effector to the vicinity of the task objective. This reduces the number of planning horizon and proposed action sequences and speed up the planning stage.

We list all hyperparameters of MPPI in Table 8. It's worth mention that because $\mathcal{S}(\phi(o_n), \psi(l))$ could be the cosine similarity between image and language embedding, $\mathcal{S}(\phi(o_{n+1}), \psi(l)) - \mathcal{S}(\phi(o_n), \psi(l))$ lies in $[-2, 2]$ and is usually small in this case. We'll then normalize the return among all proposed action sequences through $R_{normed} = \frac{R - R_{mean}}{R_{std}}$.

Table 8. MPPI Hyperparameters (parameters not listed are set to default values in (Ma et al., 2023a))

	FrankaKitchen
Planner	MPPI (Williams et al., 2017)
Planning Horizon	50
Proposed Action Sequences	64
Optimization Iteration	1
Temperature	10.0

F. Limitations & Discussions & Future Work

Here, we discuss our limitations, potential solutions to our limitations and interesting future works.

1. **Implicit Preference**. Note some few counterexample also exist that may violate the *implicit preference* assumption, which will bring noisy preference labels. For instance, consider two videos, one for "open microwave" and the other for "open the microwave". In principle, these two videos should not be treated as different, as they complete the same task. However, in our paper, these videos are treated as suboptimal for the mismatched language instructions as the instructions are a little bit different.

   Solution and future work: This problem can be solved by proper data filtering and reorganization but might demand tremendous human costs. In this paper, we directly utilize the original dataset $\mathcal{D}$ for convenience, but observe good robustness to these noisy samples, which have also been observed in previous studies (Nair et al., 2022). We believe this is an interesting observation and leave this direction as a future work.

2. **Reward definition**. In this current version, we only evaluate two types of reward definitions in the BT model. Thus, we cannot guarantee our design choices must be the optimal solution.

   Solution and future work: We have shown that by simply designing the rewards as potential-based or transition-direction rewards in embedding space, we can inherently recover an implicit time contrastive learning, naturally marries language grounding and temporal consistency in an elegant and unified objective. Therefore, we believe there must be lots of efficient reward forms still under-explored and encourage the researchers to investigate this.

3. **Out-of-domain Dataset**. In this current version, we only use the EPIC-KITCHEN-100 dataset (Damen et al., 2018) to pretrain the representations. Although this is the common choice of previous works (Ma et al., 2023a), we observe the language diversity in this dataset is far more limited, including lots of simple instructions such as `put & take & open & close & move something`. The reason is straightforward, as language is highly abstract, which can describe many videos with the same language instruction. Therefore, the language diversity is quite limited, which may be not appropriate to train a generalizable language encoder.

Solution and future work: To address this, one straightforward solution is to include more out-of-domain data, including but not limited to Ego4D (Grauman et al., 2022), Something-Something (Goyal et al., 2017), and Open-X-embodiment dataset (Collaboration et al., 2023). However, we find the language diversity in these datasets is still limited compared to image diversity. To further tackle this challenge, one promising direction is to utilize the powerful Large Language Models (LLMs) (Touvron et al., 2023; Wang et al., 2023) to augment the language instruction reasonably.

Overall, although some design choices seem quite simple and some potential limitations exist, they smartly tackle all challenges faced by previous studies. Moreover, in this work, we have demonstrated the superior effectiveness and versatility of DecisionNCE to extract value features from out-of-domain data to facilitate efficient downstream policy learning and provide several solutions to tackle these limitations. Therefore, we leave these further detailed improvements for future work. In summary, DecisionNCE provides an elegant and principled solution for decision-centric vision-language representation and reward learning.