One-shot Imitation in a Non-Stationary Environment via Multi-Modal Skill

Sangwoo Shin [1] **Daehee Lee** [1] **Minjong Yoo** [1] **Woo Kyung Kim** [1] **Honguk Woo** [1]

Abstract

One-shot imitation is to learn a new task from a single demonstration, yet it is a challenging problem to adopt it for complex tasks with the high domain diversity inherent in a non-stationary environment. To tackle the problem, we explore the compositionality of complex tasks, and present a novel skill-based imitation learning framework enabling one-shot imitation and zero-shot adaptation; from a single demonstration for a complex unseen task, a semantic skill sequence is inferred and then each skill in the sequence is converted into an action sequence optimized for environmental hidden dynamics that can vary over time. Specifically, we leverage a vision-language model to learn a semantic skill set from offline video datasets, where each skill is represented on the vision-language embedding space, and adapt meta-learning with dynamics inference to enable zero-shot skill adaptation. We evaluate our framework with various one-shot imitation scenarios for extended multi-stage Meta-world tasks, showing its superiority in learning complex tasks, generalizing to dynamics changes, and extending to different demonstration conditions and modalities, compared to other baselines.

1. Introduction

In the context of learning control tasks for autonomous agents, one-shot imitation aims at learning such a task on a single demonstration presented (Duan et al., 2017; Finn et al., 2017; Li et al., 2022). One-shot imitation is particularly desirable, as it allows efficient use of expert demonstrations, which are often costly to collect for diverse tasks. However, it is considered challenging to achieve reliable one-shot imitation for complex tasks across various domains and environments (Pertsch et al., 2022; Jang et al., 2022; Brohan et al., 2022).

In this paper, we investigate skill-based imitation learning techniques to conduct one-shot imitation for long-horizon tasks in a non-stationary environment, where underlying dynamics can be changed over time. To do so, we leverage the compositionality in control tasks and explore the semantic representation capability of a large scale vision-language pretrained model. Specifically, we develop a novel one-shot imitation learning framework using semantic skills, OnIS to enable one-shot imitation for new long-horizon tasks across different dynamics in a non-stationary environment. The framework employs a vision-language pretrained model and contrastive learning on expert data to represent semantic skills and environment dynamics separately, and explores meta-learning to adapt semantic skills to unseen dynamics. This establishes the capability of dynamics-aware skill transfer.

In deployment, given a single video demonstration for a new task, the framework translates it to a sequence of semantic skills through a sequence encoder built on the vision-language model. It then combines the skills with inferred current dynamics to generate actions optimized for not only the demonstration but also the current dynamics.

As such, through the decomposition of skills and dynamics, each skill can be well represented in the vision-language aligned embedding space without much deformation by time-varying environment dynamics. This decomposition structure with multi-modality renders OnIS unique to explore the compositionality of control tasks with skills, different from prior skill-based reinforcement learning (RL) approaches, e.g., (Pertsch et al., 2021; Nam et al., 2022).

The main contributions of our work are as follows.

- We present the OnIS framework to enable one-shot imitation and zero-shot adaptation.
- We develop a dynamics-aware skill transfer scheme to adapt skills in a non-stationary environment.
- We create an expert dataset for long-horizon, multi-stage Meta-world tasks with diverse robotic manipulation skills, and make it publicly available for other research works.
- We test the OnIS framework with several imitation

[1]Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, Republic of Korea. Correspondence to: Honguk Woo <hwoo@skku.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

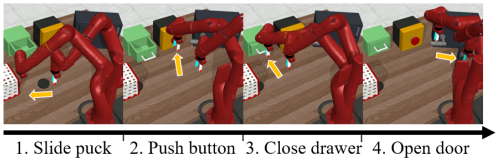

Figure 1. Multi-stage Meta-world task

learning use cases, and evaluate it in terms of one-shot imitation performance, generalization ability to different environment conditions, and extensibility to demonstrations in different modalities.

2. Problem Formulation

In vision-based tasks, one-shot imitation aims at learning a task upon a single expert demonstration presented in a video clip. Among several related problems in one-shot imitation learning approaches, our work considers a challenging yet practical problem setting, inherent in tackling complex tasks via imitation in a non-stationary environment.

Unlike recent one-shot imitation approaches, our problem setting requires the followings to be addressed in a unified framework. (1) A policy should not only be learned from a single expert demonstration, but also adapt to unseen dynamics conditions inherent in a non-stationary environment, (2) an expert demonstration specifies a long-horizon multi-stage task, and (3) an expert demonstration can be presented either in video or language. Here, we formally describe our problem, "one-shot imitation and zero-shot adaptation" in a non-stationary environment. Given a demonstration $d^{\mathcal{T}}$ for task $\mathcal{T} \sim \mathbb{T}_{\text{train}}$, we consider a model ϕ which maps $d^{\mathcal{T}}$ to a policy $\pi_{\mathcal{T}} = \phi(d^{\mathcal{T}})$. Then, the objective of one-shot imitation is to find the optimal model ϕ^* such that

$$\underset{\phi}{\text{argmax}} \left[\underset{\mathcal{T}' \sim \mathbb{T}_{\text{eval}}}{\mathbb{E}} \left[\text{E}(\mathcal{T}', \phi(d^{\mathcal{T}'})) \right] \right] \tag{1}$$

where E is an evaluation function for a policy $\phi(d^{\mathcal{T}'})$ against a new task $\mathcal{T}'$.

To deal with one-shot imitation upon various dynamics, we consider a problem setting of a Hidden Parameter Markov decision process without reward (HiP-MDP\R) with a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{H}, P_{\mathcal{H}})$. Here, $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{P}$ is a transition probability, γ is a discount factor, and $\mathcal{H}$ is a hidden parameter space, where $h \in \mathcal{H}$ determines the dynamics $s' \sim \mathcal{P}(s, a; h)$ for states $s, s' \in \mathcal{S}$ and action $a \in \mathcal{A}$. $P_{\mathcal{H}}$ is a prior distribution of $\mathcal{H}$. Then, the objective of one-shot imitation and zero-shot adaptation across various dynamics is to find the optimal model ϕ^* such that

$$\underset{\phi}{\text{argmax}} \left[\underset{h \sim P_{\mathcal{H}}}{\mathbb{E}} \left[\underset{\mathcal{T}' \sim \mathbb{T}_{\text{eval}}}{\mathbb{E}} \left[\text{E}_h(\mathcal{T}', \phi(d^{\mathcal{T}'})) \right] \right] \right] \tag{2}$$

where E_h is an evaluation function for a policy $\phi(d^{\mathcal{T}'})$ against task $\mathcal{T}'$ with fixed h in a HiP-MDP\R.

Regarding task definitions, we consider long-horizon, multi-stage tasks, in the form of K-staged tasks $\mathcal{T}_{0:K-1}$ (e.g., in Figure 1, a 4-stage task with sequential subtasks, slide puck and others). Note that $\mathcal{T}_{0:K-1}$ should be done in a correct order, e.g., slide puck, push button, close drawer, and then open door, specifying that the evaluation function E_h yields some positive value for performing subtask $\mathcal{T}_j$ only when subtasks $\mathcal{T}_{0:j-1}$ have been completed.

3. Our Approach

3.1. Overall Framework

To enable the one-shot imitation and zero-shot adaptation in a non-stationary environment, we develop the OnIS framework, leveraging the semantic representation capability of a large scale vision-language pretrained model and compositionality of complex tasks.

As illustrated in Figure 2, the framework has two phases: (a) training phase and (b) deployment phase. In the training phase, we leverage task compositionality and pretrained vision-language models to extract semantic skills from video demonstrations. Specifically, we explore skill decomposition (task compositionality) for dealing with complex tasks where each task consists of sequential skills (achievable goals). Each task is decomposed and translated into a sequence of skills that are task-agnostically learned on offline datasets. From demonstrations for various tasks and diverse dynamics, the framework employs contrastive learning with video and text retrieval tasks to disentangle dynamics-invariant task-relevant features from the demonstrations, thus representing them in a vision-language semantic space, i.e., in the CLIP embedding space (Radford et al., 2021).

At the same time, for dealing with adaptation across different dynamics, we develop a novel skill representation, dynamics-aware skills. The framework uses contrastive learning with dynamics reconstruction tasks to disentangle dynamics information from the trajectories in state and action pairs. These two individual contrastive learning procedures tend to decompose unstructured features contained in the demonstrations and trajectories into semantic skill sequences and environment dynamics, respectively. Then, to render a skill sequence adaptive to different dynamics, the decomposed skill sequence and dynamics are combined in training the skill transfer capability through a meta-learning procedure.

(b) In the deployment phase, given a single video demonstration for an unseen task, the framework derives such a policy not only imitating the demonstrated expert behavior to complete the unseen task but also being capable of adapting to

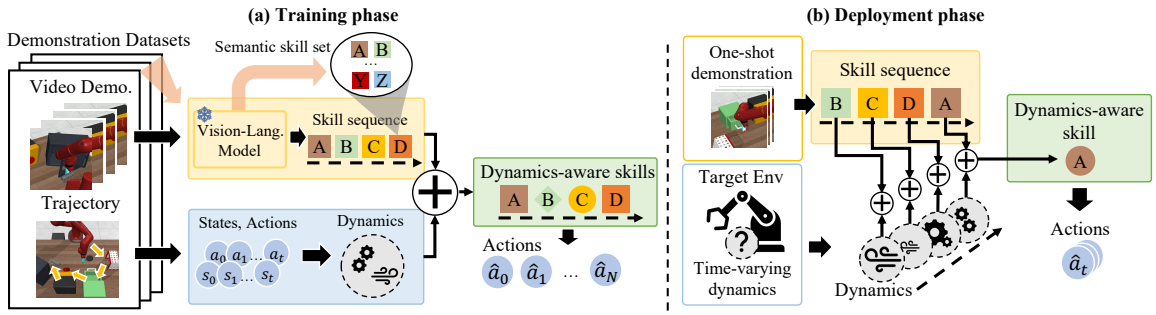

Figure 2. One-shot imitation and zero-shot adaptation in OnIS: (a) In the training phase, a semantic skill set is established from video demonstrations in offline datasets by leveraging a pretrained vision-language model. For each demonstration in the dataset, semantic skill sequence and environment dynamics are first disentangled, and then they are combined to enable dynamics-aware skill transfer to different environments. (b) In the deployment phase, given a single demonstration for an unseen task, its semantic skill sequence can be immediately inferred (one-shot imitation), where each in the sequence adapts temporally to the hidden, time-varying dynamics in the environment (zero-shot adaptation).

dynamics changes in the environment. In this phase, the framework uses only a single demonstration (one-shot imitation), and it enables rapid adaptation to dynamics changes (zero-shot adaptation) in a non-stationary environment, by making use of the knowledge constructed in the training phase.

For one-shot imitation, in the training phase, we exploit offline datasets of video demonstrations and their matched trajectories in state and action pairs, presuming the availability of such datasets from the environment. The contrastive learning procedures on the datasets and the meta-learning procedure together enable to construct the knowledge required for one-shot imitation in a non-stationary environment. The contrastive learning procedures are implemented in the framework, as a semantic skill sequence encoder and a dynamics encoder, and the meta-learning procedure is implemented as a skill transfer module. We explain these modules in Figure 3, and Sections 3.2 and 3.3.

3.2. Learning Semantic Skill Sequences

Semantic skill sequence encoder. A semantic skill sequence encoder Φ_{enc} is responsible for mapping an expert demonstration d to a sequence of semantic skills, in that d is segmented into a sequence of dynamics-invariant behavior patterns (action sequences). Each pattern corresponds to a short sequence of expert actions and it can be described in a language instruction in the environment. Such an expert behavior pattern associated with some language instruction is referred to as a semantic skill, as for an expert behavior pattern, its associated language instruction is used to represent expert behaviors on the semantic embedding space of a vision-language aligned model.

To implement the encoder Φ_{enc}, we use the CLIP vision-language pretrained model and sample-efficient prompt learning techniques (Zhou et al., 2022). Specifically, we assume that a dataset of expert trajectories $\mathcal{D} = \{\tau_1, \tau_2, \cdots, \tau_N\}$ contains language instruction data, in that each T-length trajectory $\tau = \{(s_1, v_1, l_1, a_1), \cdots, (s_T, v_T, l_T, a_T)\}$ consists of state s, visual observation v, language instruction l, and action a, where each language instruction is an element of the instruction set $\mathbf{L}$, similar to (Zheng et al., 2022).

Furthermore, we consider two distinct cases of language instructions annotated in the trajectories, one is the subtask-level instruction, (e.g.,"Push lever" is annotated on the transitions for its relevant single subtask, as in (Pertsch et al., 2022)), and the other is the episode-level instruction, (e.g., "Push lever, open door and close the box" is annotated on all the transitions in an episode, as in (Garg et al., 2022; Liu et al., 2022)).

For the subtask-level instruction case, we assume that language instructions exist only for a small subset $\mathcal{D}_0$ of $\mathcal{D}$, and consider the CLIP model with a video encoder Φ_V and a language encoder Φ_L such that

$$\Phi_V : v_{t:t+H} \mapsto z_v, \quad \Phi_L : l_t \mapsto z_l. \tag{3}$$

These Φ_V and Φ_L encoders map a video demonstration $v_{t:t+H}$ and a language instruction l_t into the same embedding space for timesteps t. To construct Φ_{enc} based on Φ_V and Φ_L using a small number of samples, we use the language prompt (Zhou et al., 2022). Given $v_{t:t+H}$ or l_t as input d, the semantic skill sequence encoder Φ_{enc} learns to translate each input into a sequence of semantic skills by

$$\Phi_{enc}(d) = \begin{cases} \underset{z \in z_{\mathbf{L}}}{\operatorname{argmax}}\{\operatorname{sim}(z, \Phi_V(v_{t:t+H}))\}, & d = d_v \\ \{\Phi_L(l_t; \theta_p)\}, & d = d_l \end{cases} \tag{4}$$

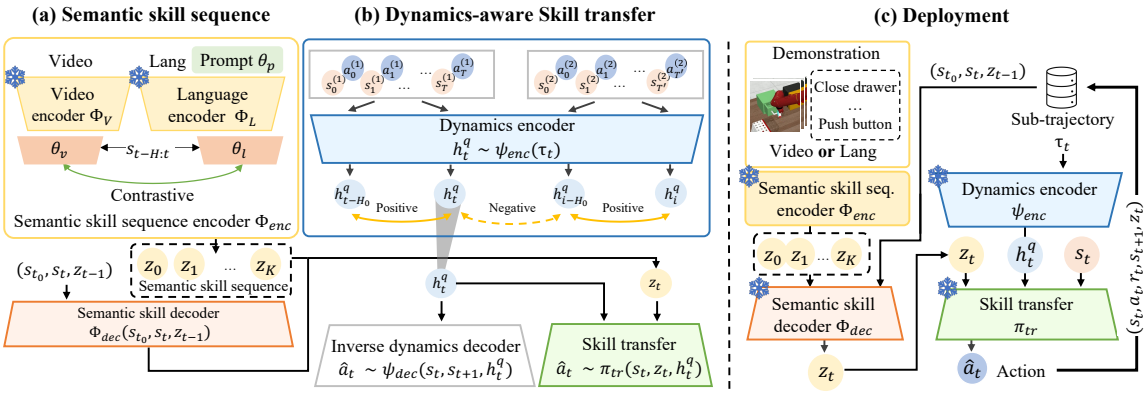

Figure 3. OnIS framework: In (a), the semantic skill sequence encoder Φ_{enc} and the semantic skill decoder Φ_{dec} are trained offline using the CLIP vision-language pretrained model, where Φ_{enc} translates video demonstrations to semantic skill sequences and is contrastively learned, and Φ_{dec} learns to infer an optimal skill (from a sequence) upon a state. The language prompt θ_p is only used for the subtask-level instruction (S-OnIS) case, and the additional encoders θ_v, θ_l are only used for episode-level instruction case (U-OnIS). In (b), the skill transfer π_{tr} and the dynamics encoder ψ_{enc} are trained offline, where π_{tr} learns to infer an action sequence optimized for the deployment setting from a given semantic skill sequence and inferred dynamics, and ψ_{enc} learns to infer dynamics from sub-trajectories. These modules establish dynamics-aware skill transfer. In (c), for a given demonstration, Φ_{enc} first infers a sequence of semantic skills, Φ_{dec} infers a current semantic skill, and ψ_{enc} infers current dynamics in the non-stationary deployment environment. Then, π_{tr} yields actions optimized through the current semantic skill and dynamics.

where $d_v = (v_{t:t+H} : 0 \leq t \leq T')$, $d_l = (l_t : 0 \leq t \leq N)$, $z_{\mathbf{L}} = \Phi_L(\mathbf{L}; \theta_p)$, and $\mathbf{L}$ denotes a set of language instructions. Specifically, a language prompt θ_p in learnable parameters are trained via contrastive learning on positive pairs $(v_{t:t+H}, l_t)$ in $\mathcal{D}_0$, where the contrastive loss is defined as $\mathcal{L}_{CON}(\theta_p) =$

$$-\log\left(\frac{\text{sim}(\Phi_V(v_{t:t+H}), \Phi_L(l_t; \theta_p))}{\sum_{l_t \neq l \in \mathbf{L}} \text{sim}(\Phi_V(v_{t:t+H}), \Phi_L(l; \theta_p))}\right). \quad (5)$$

The similarity of latent vectors z and z' is calculated by

$$\text{sim}(z, z') = \frac{1}{\alpha} \exp\left(\frac{\langle z, z' \rangle}{\|z\| \|z'\|}\right) \quad (6)$$

where α is a temperature coefficient. As such, for the subtask-level instruction case, the semantic skill sequence encoder Φ_{enc} can be established from the pretrained visual and language encoders Φ_V, Φ_L through prompt-based contrastive learning on a small set of video and text retrieval samples.

In the case of episode-level instruction, all the transitions in an episode in $\mathcal{D}$ are associated only with a single instruction without detailed subtask-level instructions. Thus, we adapt unsupervised skill learning techniques in (Garg et al., 2022) with contrastive learning on video features. Accordingly, $\Phi_{enc}(d)$ learns to translate a video demonstration $d = v_{0:T'}$ or language instruction $d = l$ to a sequence of semantic skills by

$$\begin{cases} \{\underset{z \in z_{\mathbf{L}}}{\text{argmax}}\{\text{sim}(z, \theta_v(\Phi_V(v_{0:T'}), s_{t-H:t})\}\}, & d = v_{0:T'} \\ \{\theta_l(\Phi_L(l), s_{t-H:t})\}, & d = l \end{cases} \quad (7)$$

where $z_{\mathbf{L}} = \theta_l(\Phi_L(\mathbf{L}), s_{t-H:t})$. Similar to the skill predictor used in (Garg et al., 2022), two additional encoders θ_v and θ_l such as

$$\begin{aligned} \theta_v &: (\Phi_V(v_{0:T'}), s_{t-H:t}) \mapsto z_v \\ \theta_l &: (\Phi_L(l), s_{t-H:t}) \mapsto z_l \end{aligned} \quad (8)$$

are trained. Specifically, the encoder θ_v is contrastively learned on positive pairs $(v_{0:T'}, l)$ in $\mathcal{D}$, where the contrastive loss in (5) is rewritten as $\mathcal{L}_{CON}(\theta_v) =$

$$-\log\left(\frac{\text{sim}(\theta_v(\Phi_V(v_{0:T'}), s_{t-H:t}), z_l)}{\sum_{z_l \neq z \in z_{\mathbf{L}}} \text{sim}(\theta_v(\Phi_V(v_{0:T'}), s_{t-H:t}), z)}\right) \quad (9)$$

for $z_l = \theta_l(\Phi_L(l), s_{t-H:t})$ and $z_{\mathbf{L}} = \theta_l(\Phi_L(\mathbf{L}), s_{t-H:t})$. The encoder θ_l is trained via behavior cloning that maximizes the mutual information between the actions and semantic skills, where the loss is defined as

$$\mathcal{L}_{BC}(\theta_l, f) = \mathbb{E}[\|a_t - f(s_t, z_l)\|] \quad (10)$$

for the action reconstruction model $f : (s_t, z_l) \mapsto a_t$.

Semantic skill decoder. Given a sequence of semantic skills, the semantic skill decoder Φ_{dec} is responsible for obtaining

a semantic skill for a current state. Φ_{dec} is implemented as a binary model to determine whether the current semantic skill is terminated or not, similar to the task specification interpreter in (Xu et al., 2018), i.e.,

$$\Phi_{dec} : (s_{t_0}, s_t, z_t) \mapsto \{0, 1\} \tag{11}$$

for a semantic skill z_t being currently executed, the initial state s_{t_0} for z_t, and a current state s_t. Φ_{dec} is learned on $\mathcal{D}$ with binary cross entropy (BCE),

$$\mathcal{L}_{BCE}(\Phi_{dec}) = \text{BCE}(\mathbb{1}_{z_t \neq z_{t+1}}, \Phi_{dec}(s_{t_0}, s_t, z_t)) \tag{12}$$

where $z_t = \Phi_v(v_{t:t+H})$.

As illustrated in Figure 3(a), the semantic skill sequence encoder Φ_{enc} is trained through contrastive learning to extract the inherent semantic skills in the video. The semantic skill decoder Φ_{dec}, which is basically a binary classifier, is conditioned on a skill sequence and is trained to predict the appropriate semantic skill at the current timestep.

3.3. Learning Dynamics-aware Skill Transfer

Given a semantic skill sequence encoder Φ_{enc}, the skill transfer module π_{tr} is responsible for transferring a semantic skill $z_t = \Phi_{enc}(v_{t:t+H})$ to an action sequence adapted for environment dynamics. That is,

$$\pi_{tr} : (s_t, z_t, h_t^q) \mapsto a_t \tag{13}$$

for a current state s_t, a semantic skill z_t, and the dynamics embedding $h_t^q = \psi_{enc}(\tau_t)$. For H_0-length sub-trajectories $\tau_t = (s_{t-H_0}, a_{t-H_0}, ..., s_{t-1}, a_{t-1})$, the dynamics encoder ψ_{enc} takes it as input, and maps to a quantized vector h_t^q, i.e.,

$$\psi_{enc} = \mathbf{q} \circ \psi_{enc}^c : \tau_t \mapsto h_t^q \tag{14}$$

where ψ_{enc}^c maps τ_t to a continuous latent vector. We use the vector quantization operator $\mathbf{q}$ to avoid the posterior collapse (van den Oord et al., 2017). The output of $\mathbf{q}$ is the nearest vector among the learnable parameters, called codebook, $h^q \in \{h^{q_1}, \cdots, h^{q_M}\}$. Then, the quantized dynamics embedding h_t^q is obtained by

$$h_t^q = \mathbf{q}(\psi_{enc}^c(\tau_t)) = \underset{j \in \{1, \cdots, M\}}{\text{argmin}} \{\|\psi_{enc}^c(\tau_t) - h^{q_j}\|\}. \tag{15}$$

For a semantic skill $z_t = \Phi_{enc}(v_{t:t+H})$ and dynamics embedding $h_t^q = \psi_{enc}(\tau_t)$, the skill transfer module and dynamics encoder are jointly trained by minimizing the behavior cloning (BC) loss $\mathcal{L}_{BC}(\pi_{tr}, \psi_{enc}) =$

$$\mathbb{E}_{s_T, a_T \sim \mathcal{D}} \left[\|a_t - \pi_{tr}(s_t, z_t, \psi_{enc}(\tau_t))\|^2 \right]. \tag{16}$$

Furthermore, to disentangle task-irrelevant dynamics from sub-trajectories, we also use contrastive learning on sub-trajectories in various dynamics. Specifically, suppose that a batch of H_0-length sub-trajectories $\{\tau_{t_i}\}_{1 \leq i \leq N}$ contains one positive sample $\{(\tau_{t_j}, \tau_{t_k})\}$ which comes from the same trajectory starting at different timesteps. Then, ψ_{enc} is learned on the positive and negative pair, where the contrastive loss is defined as $\mathcal{L}_{CON}(\psi_{enc}) =$

$$-\log \left(\frac{\text{sim}(\psi_{enc}(\tau_{t_j}), \psi_{enc}(\tau_{t_k}))}{\sum_{i \neq i'} \text{sim}(\psi_{enc}(\tau_{t_i}), \psi_{enc}(\tau_{t_{i'}}))} \right). \tag{17}$$

To maximize the mutual information between the inferred embedding h_t^q and underlying dynamics, we adopt reconstruction-based feature extraction by using the inverse dynamics decoder $\psi_{dec} : (s_t, s_{t+1}, h_t^q) \mapsto a_t$, where the action reconstruction loss is defined as

$$\begin{aligned} &\mathcal{L}_{REC}(\psi_{enc}, \psi_{dec}) \\ &= \mathbb{E}_{t-H_0 \leq i < t} \left[\|a_i - \psi_{dec}(s_i, s_{i+1}, \psi_{enc}(\tau_t)\|^2 \right]. \end{aligned} \tag{18}$$

Overall, the skill transfer module π_{tr}, the dynamics encoder ψ_{enc}, and the inverse dynamics decoder ψ_{dec} are jointly trained to minimize the losses $\mathcal{L}_{BC}, \mathcal{L}_{CON}$ and $\mathcal{L}_{REC}$, where these modules are presented in Figure 3(b). During the deployment phase, the skill transfer module acts as a policy network that determines actions ($\hat{a}_t$) upon the skill and dynamics embeddings (z_t and h_t^q) along with the current state (s_t), as illustrated in Figure 3(c).

Algorithm 1 lists the learning procedures for skill transfer in Section 3.3.

Algorithm 1 Learning to transfer skills

Semantic skill sequence encoder Φ_{enc}
Dynamics encoder ψ_{enc}, Inverse dynamics decoder ψ_{dec}
Skill transfer module π_{tr}, Dataset $\mathcal{D}$
repeat
  Sample a batch $\{(\tau_{t_i}, v_{t_i:t_i+H})\}_i \sim \mathcal{D}$
  $\{z_{t_i}\}_i = \Phi_{enc}(\{v_{t_i:t_i+H}\}_i)$
  / Calculate loss with $\{(\tau_{t_i}, z_{t_i})\}$ */*
  $\text{loss}_{bc} \leftarrow \mathcal{L}_{BC}(\pi_{tr}, \psi_{enc})$ using (16)
  $\text{loss}_{con} \leftarrow \mathcal{L}_{CON}(\psi_{enc})$ using (17)
  $\text{loss}_{rec} \leftarrow \mathcal{L}_{REC}(\psi_{enc}, \psi_{dec})$ using (18)
  $\psi_{enc} \leftarrow \psi_{enc} - \nabla_{\psi_{enc}}(\text{loss}_{bc} + \text{loss}_{con} + \text{loss}_{rec})$
  $\psi_{dec} \leftarrow \psi_{dec} - \nabla_{\psi_{dec}}\text{loss}_{rec}$
  $\pi_{tr} \leftarrow \pi_{tr} - \nabla_{\pi_{tr}}\text{loss}_{bc}$
until converge

4. Evaluations

In this section, we evaluate the performance of our OnIS framework under various configurations of non-stationary environments.

Environment settings. For evaluation, we devise multi-stage robotic manipulation tasks, namely multi-stage Meta-world, using the Meta-world simulated benchmark (Yu et al.,

2020), where each multi-stage task is composed of a sequence of existing Meta-world tasks (subtasks). An agent manipulates a robot arm using actions such as slide puck, close drawer, etc., to achieve certain objectives. For emulating non-stationary environments, we exploit kinematic parameters affecting the dynamics of robotic manipulation, which are not explicitly revealed through states or videos. This non-stationarity is consistent with the prior works in robotics (Kumar et al., 2021; Qi et al., 2022).

Offline datasets. To generate the dataset that covers diverse dynamics, we implement rule-based expert policies to collect trajectories from 13 different stationary environment conditions for each subtask, where each condition corresponds to a specific kinematic configuration. We collect 3120 episodes (240 for each environment condition), and we annotate 24 episodes (0.76% of our dataset) with subtask-level language instructions. This annotated data is only used to train the semantic skill sequence encoder, especially for S-OnIS (in Section 4.1).

4.1. OnIS Implementation

We implement our OnIS framework using the opensource project Jax (Bradbury et al., 2018). OnIS is composed of 4 modules: semantic skill sequence encoder Φ_{enc}, skill decoder Φ_{dec}, dynamics encoder ψ_{enc}, and skill transfer module π_{tr}. We use the Transformer-based CLIP pretrained model (Radford et al., 2021) to implement the sequence encoder. Specifically, we implement two versions of OnIS: one with supervised semantic skills (S-OnIS) and the other with unsupervised semantic skills (U-OnIS). The former S-OnIS corresponds to subtask-level instructions where contrastive learning for semantic skills is driven by the supervision of subtask-level instructions (with sub-trajectory and skill pairs). The latter U-OnIS corresponds to episode-level instructions where contrastive learning for semantic skills is driven by the supervision of episode-level instructions (without sub-trajectory and skill pairs).

4.2. Baselines

- BC-Z (Jang et al., 2022) is a state-of-the-art imitation learning framework, which utilizes both video demonstrations and language instructions. To extract semantic skill information from multi-modal data, BC-Z is trained to maximize the similarity between a visual and language embedding encoded by the pretrained language model.
- Decision Transformer (DT) (Chen et al., 2021) is a Transformer-based model tailored for imitation learning. Considering that DT is not originally meant for multi-modal data, we modify DT to have state vectors in different modalities, along with either a video demonstration or language instruction. We use DT to compare OnIS with conventional imitation learning approaches which do not consider skill semantics.
- SPiRL (Pertsch et al., 2021) is a state-of-the-art unsupervised skill-based RL algorithm, which embeds sub-trajectories into the latent skill space. Similar to DT, we modify SPiRL to handle either a video demonstration or language instruction in a single framework.

4.3. One-shot Imitation Performance

Tables 1 and 2 compare the one-shot imitation performance in task success rates achieved by our framework (U-OnIS, S-OnIS) and other baselines (DT, SPiRL, BC-Z), given two distinct scenarios in terms of modality with a single input in either a video demonstration or language instruction. As shown in Figure 3(c), all the learned modules in the framework freeze when being deployed in a target environment, and they are immediately evaluated against various demonstrations and time-varying dynamics conditions.

Specifically, we measure the one-shot imitation performance in task success rates for multi-stage tasks with K sequential objectives ($K = 1, 2, 4$) upon several dynamics change levels (stationary, low, medium, and high). Regarding baselines, DT and SPiRL variants for leveraging video demonstrations are denoted as V-DT and V-SPiRL, and those for leveraging language instructions are denoted as L-DT and L-SPiRL. Due to its low performance, V-DT is not included in the comparison.

Table 1. One-shot imitation performance for video demonstration: for multi-stage Meta-world tasks, the performance in task success rates by our U-OnIS, S-OnIS and other baselines is measured against various test conditions on K sequential objectives in a task ($K = 1, 2, 4$) and dynamics change levels (stationary, low, medium, and high).

K	Non-st.	V-SPiRL	BC-Z	U-OnIS	S-OnIS
1	Stationary	61.48%	75.00%	95.80%	**100.0%**
	Low	48.23%	48.33%	86.28%	**94.55%**
	Medium	45.22%	44.29%	84.09%	**94.34%**
	High	39.09%	37.63%	81.10%	**90.49%**
2	Stationary	61.75%	71.08%	67.50%	**100.0%**
	Low	41.66%	42.85%	75.75%	**85.66%**
	Medium	43.39%	47.61%	73.07%	**84.29%**
	High	30.33%	22.09%	66.00%	**81.62%**
4	Stationary	47.44%	21.01%	64.38%	**91.67%**
	Low	27.31%	14.38%	50.03%	**74.95%**
	Medium	20.82%	12.79%	49.22%	**74.89%**
	High	15.54%	11.13%	49.82%	**70.19%**

Overall performance. As shown in Table 1, our U-OnIS and S-OnIS yield higher performance consistently than the baselines for all video demonstration cases. Specifically, S-OnIS outperforms the most competitive baseline SPiRL by a significant margin $38.25\% \sim 54.64\%$ on average. The performance gap between S-OnIS and the baselines increases for larger $K = 4$ where a task consists of 4 subtasks, compared to $K = 1, 2$. This specifies that our approach is competitive in learning on a relatively long-horizon demonstra-

Table 2. One-shot imitation performance for language instruction

K	Non-st.	L-DT	L-SPiRL	BC-Z	U-OnIS	S-OnIS
1	Stationary	92.66%	100.0%	100.0%	87.70%	**100.0%**
	Low	62.49%	54.29%	**96.61%**	90.40%	95.00%
	Medium	53.84%	50.71%	75.32%	81.29%	**89.04%**
	High	20.19%	47.75%	68.91%	82.19%	**88.50%**
2	Stationary	58.33%	60.08%	76.25%	74.50%	**95.00%**
	Low	47.12%	29.35%	45.27%	60.44%	**83.17%**
	Medium	33.17%	34.34%	35.00%	69.24%	**78.50%**
	High	30.28%	20.51%	25.52%	66.67%	**77.27%**
4	Stationary	37.73%	43.16%	40.30%	67.71%	**87.50%**
	Low	22.11%	22.83%	20.23%	59.64%	**76.60%**
	Medium	20.43%	15.50%	20.60%	56.68%	**71.39%**
	High	8.90%	16.39%	11.13%	54.82%	**66.92%**

tion that performs multi-stage tasks.

Consistency in multi-modality. Similar to the video demonstration scenario, in Table 2, U-OnIS and S-OnIS show competitiveness for the language instruction scenario. When Tables 1 and 2 are compared, we observe that U-OnIS and S-OnIS maintain high performance consistently for the two distinct scenarios, showing 5.69% and 3.17% performance differences (between the language and video scenarios) on average between them. Unlike OnIS, although BC-Z explores multi-modality, it achieves inconsistent performance between the two scenarios, showing a 16.01% performance drop for the video demonstration scenario from the language instruction scenario. This inconsistent performance pattern between different modalities can be also found in (Jang et al., 2022). While our approach exploits task compositionality with shareable subtasks and semantic skills, BC-Z tends to transform a whole video input into an individual task; thus, BC-Z can degrade significantly for new video demonstrations.

Generalization to non-stationarity. As shown in Tables 1 and 2, our U-OnIS and S-OnIS show more robust performance than the baselines for all non-stationary settings (including low, medium, and high). Specifically, in Table 2, S-OnIS achieves a relatively small drop of $9.49\% \sim 18.13\%$ on average in non-stationary environments compared to the stationary. The most competitive baseline, BC-Z shows a large drop of $19.72\% \sim 57.02\%$ on average in the same condition. In OnIS, the separate embedding spaces for semantic skills and dynamics are used to address time-varying dynamics. Without that decoupled embedding strategy, BC-Z can experience low-performance skills, which are entangled with certain dynamics in training environments.

U-OnIS and S-OnIS. As presented above, while both U-OnIS and S-OnIS outperform the baselines, S-OnIS shows better performance than U-OnIS with an average margin of 16.63%. In our framework with efficient prompt learning, the ability of semantic skill learning can be improved by a small annotated dataset (i.e., 0.76% of total trajectories of 3120 episodes) with subtask-level instructions.

Semantic embedding correspondence. Given a video demonstration, in Figure 4, we visualize its embeddings (as black-colored circles, Demo.) and the embeddings by its respective episodic run by U-OnIS (as blue-colored circles, OnIS) in a multi-modal semantic space. As observed, two streams of the embeddings construct a similar path, indicating their temporal correspondence in semantic skills in an episode. We also visualize semantic skill embeddings (as rhombuses of different colors) of sub-trajectories in our dataset, which are matched with the 4 subtasks in the demonstration. The path (OnIS) turns out to follow these semantic skill embeddings. This correspondence specifies that a policy achieved by one-shot imitation can perform skills, which are required to complete the subtasks in the demonstration, in the correct sequence.

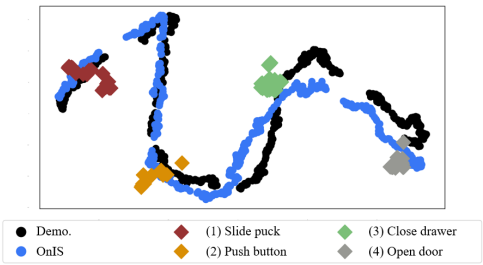

Figure 4. Semantic embedding correspondence for a demonstration and one-shot imitation run

4.4. Ablation Study

Temporal contrast. Table 3 evaluates our positive pair sampling strategy in the dynamics encoder, where Fixed$\pm N$ and Random$\pm N$ denote sampling on N-fixed interval and and random sampling within N interval, respectively, in a trajectory. Specifically, our framework employs Random$\pm$T, where T is the trajectory length. With Random$\pm$T, the framework shows $3.75\% \sim 18.25\%$ higher performance than others. This is because pairs sampled in a whole episode contain a wider range of state variations with respect to dynamics.

Table 3. Effect by temporal contrast

Dynamics	Fixed$\pm$1	Fixed$\pm$10	Random$\pm$10	Random$\pm$T
Unseen Stationary	49.50%	54.25%	55.00 %	63.25%
Non-Stationary	39.75%	44.50%	63.25%	62.50%

Annotation sample size. Figure 5 evaluates S-OnIS with respect to different sample sizes (4, 8, 16, 24) used for training the prompt (in (5)). As expected, the extremely low data regime of 4 samples rarely solves even a single task ($K = 1$), but the default S-OnIS implementation with 24 samples achieves robust performance for all $K = 1, 2, 4$. This shows the efficiency of prompt-based contrastive learning used for the semantic skill sequence encoder.

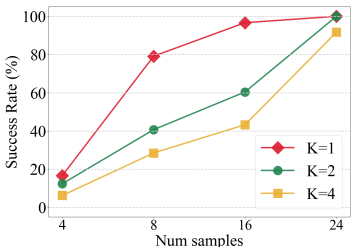

Figure 5. Effect by annotated sample size: the x-axis denotes the number of annotated samples used for training the semantic skill sequence encoder in S-OnIS, and the y-axis denotes the one-shot imitation performance by S-OnIS.

4.5. Use Cases

Noisy video demonstration. To evaluate the robustness of our framework on noisy video demonstrations, we construct different scenarios: (1) Color jitter, (2) Gaussian noise, (3) Cutout. Table 4 shows the performance of 4-stage tasks for the scenarios. The baselines are highly vulnerable to noisy video demonstrations, but OnIS is robust to them. In the training phase, OnIS employs video and text retrieval tasks with a vision-language model, tending to extract semantic information from visual observations. This renders OnIS robust to noisy demonstrations, especially when noise does not alter underlying semantics.

Table 4. Noisy video demonstrations

Noise type	V-SPiRL	BC-Z	U-OnIS	S-OnIS
w/o noise	47.44%	21.01%	64.38%	91.67%
Color jitter	0.00%	2.78%	35.19%	68.05%
Gaussian noise	0.00%	0.00%	25.42%	20.83%
Cutout	0.00%	0.00%	27.78%	8.33%

Real-world video demonstration. Here, we test our framework with real-world video demonstrations, where we use a few real-world video clips, e.g., slide puck and push button at the top of Figure 6. The graph associated with the real-world video clips specifies the inference for pretrained semantics skills over timesteps, yielded by our semantic skill sequence encoder. The inference results state that the encoder is able to extract correct semantic skills from the input clips to perform respective robotic manipulation tasks, exploring the domain-invariant semantic knowledge tuned on the pretrained CLIP visual encoder. At the bottom of the figure, we present the semantic skill execution over timesteps in the target environment, where each semantic skill for a certain timestep is determined by the semantic skill decoder and is then translated into an optimized action sequence through our skill transfer module.

Instruction variation. We evaluate our framework with language instruction variants, which are not used for training the semantic skill sequence encoder. For this, we use several

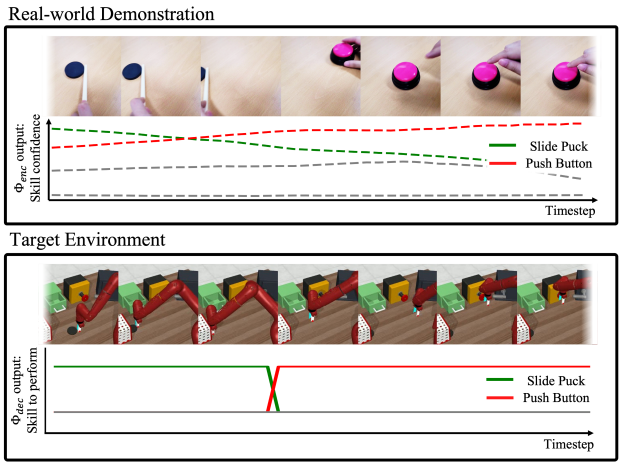

Figure 6. Real-world demonstrations

instruction variations with verb replacement, modifier extension, and verbose instructions. As shown in Table 5, both U-OnIS and S-OnIS achieve robust performance against unseen language instructions with variations. This is because the sequence encoder does not directly map an instruction in the embedding space through the CLIP language encoder, but translates it to its respective semantic skill by using its similarity to learned semantic skills.

Table 5. Language instruction variations

Variations	U-OnIS	S-OnIS
Seen instruction	70.01%	93.50%
Verb replacement	70.01%	93.50%
Modifier extension	70.01%	93.50%
Verbose instruction	40.74%	54.25%

5. Related Works

One-shot imitation learning. To adapt to a new task with a single demonstration, one-shot imitation learning has been investigated by extracting task information from state action sequences (Duan et al., 2017; Yu et al., 2018a) or from video demonstrations (Yu et al., 2018b; Mandi et al., 2022; Xu et al., 2021). For example, (Yu et al., 2018b) exploited human video demonstrations to learn robotic manipulation tasks, focusing on the generalization ability to unseen tasks in video demonstrations, and (Mandi et al., 2022) adopted contrastive learning schemes for cross-domain one-shot imitation. We share the same direction to one-shot imitation with these prior works, but we address specific one-shot imitation issues for a non-stationary environment, aiming to achieve one-shot imitation for unseen long-horizon tasks and zero-shot adaptation across various dynamics.

Skill-based RL. To tackle long-horizon multi-stage tasks, skill-based RL techniques have been explored along with hierarchical learning frameworks. (Eysenbach et al., 2019)

explored unsupervised skill learning in a hierarchical framework without requiring explicit reward signals from the environment. Recently, in (Pertsch et al., 2021; Nam et al., 2022), with the availability of diverse expert trajectory datasets in multi-task environments, skill embedding techniques in offline settings were further investigated to accelerate online policy learning for long-horizon tasks. We also use skill embedding techniques to tackle multi-stage long-horizon tasks, but adapt them with a vision-language pretrained model to exploit semantics in skills and demonstrations. Our OnIS is the first framework using semantic skill embeddings in the CLIP space to enable one-shot imitation and zero-shot adaptation.

Multi-modality for imitation learning. Recently, several imitation learning approaches using large scale pretrained models have been introduced. (Guhur et al., 2022) used a pretrained language model to tackle instruction-following tasks in a vision-based environment, and (Liu et al., 2022; Shridhar et al., 2022) used a pretrained multi-modal model to encode both instruction and visual observations for better scaling and generalization ability. (Jang et al., 2022) presented a vision-language joint learning method for one-shot imitation learning which can utilize both video demonstrations and language instructions. In the same vein as (Jang et al., 2022), we also use a multi-modal model for one-shot imitation, but we focus on zero-shot adaptation for different dynamics in a non-statinary environment.

6. Conclusion

We presented a skill-based imitation learning framework, OnIS that enables not only one-shot imitation learning from a single demonstration but also zero-shot adaptation of a learned policy to different dynamics in a non-stationary environment. To this end, we leverage the compositionality of long-horizon tasks, by which a task can be decomposed into a sequence of skills represented in the vision-language aligned semantic space, as well as explore meta-learning techniques to enable skill transfer upon various dynamics conditions. Through experiments, we demonstrated the superiority of OnIS with several one-shot imitation use cases in terms of one-shot imitation performance, generalization ability to unseen tasks and dynamics, and extensibility to different modalities in demonstrations. Our future direction includes addressing the issue of learning complex tasks via imitation in the environment with limited data, where sufficient expert data does not exist to learn diverse skills.

7. Acknowledgement

We would like to thank anonymous reviewers for their valuable comments and suggestions. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-01045, 2022-0-00043, 2019-0-00421, 2020-0-01821) and by the National Research Foundation of Korea (NRF) grant funded by the MSIT (No. RS-2023-00213118).

References

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*. PMLR, 2022.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. Albumentations: fast and flexible image augmentations. *Information*, 11(2): 125, 2020.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 15084–15097, 2021.

Duan, Y., Andrychowicz, M., Stadie, B., Jonathan Ho, O., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, volume 78, pp. 357–368. PMLR, 2017.

Garg, D., Vaidyanath, S., Kim, K., Song, J., and Ermon, S. LISA: Learning interpretable skill abstractions from language. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Guhur, P.-L., Chen, S., Pinel, R. G., Tapaswi, M., Laptev, I., and Schmid, C. Instruction-driven history-aware policies for robotic manipulations. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*. PMLR, 2022.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*. PMLR, 2019.

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 *Proceedings of the 5th Conference on Robot Learning (CoRL)*, pp. 991–1002. PMLR, 2022.

Kumar, A., Fu, Z., Pathak, D., and Malik, J. RMA: rapid motor adaptation for legged robots. In *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021.

Lee, K.-H., Nachum, O., Yang, M., Lee, L., Freeman, D., Xu, W., Guadarrama, S., Fischer, I., Jang, E., Michalewski, H., and Mordatch, I. Multi-game decision transformers, 2022.

Li, J., Lu, T., Cao, X., Cai, Y., and Wang, S. Meta-imitation learning by watching video demonstrations. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

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

Mandi, Z., Liu, F., Lee, K., and Abbeel, P. Towards more generalizable one-shot visual imitation learning. In *Proceedings of the 39th International Conference on Robotics and Automation (ICRA)*, pp. 2434–2444. IEEE, 2022.

Nam, T., Sun, S.-H., Pertsch, K., Hwang, S. J., and Lim, J. J. Skill-based meta-reinforcement learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

Pertsch, K., Lee, Y., and Lim, J. Accelerating reinforcement learning with learned skill priors. In *Proceedings of the 4th Conference on robot learning (CoRL)*, pp. 188–204. PMLR, 2021.

Pertsch, K., Desai, R., Kumar, V., Meier, F., Lim, J. J., Batra, D., and Rai, A. Cross-domain transfer via semantic skill imitation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*. PMLR, 2022.

Qi, H., Kumar, A., Calandra, R., Ma, Y., and Malik, J. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, pp. 1722–1732, 2022.

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 *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR, 2021.

Shridhar, M., Manuelli, L., and Fox, D. CLIPort: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, pp. 894–906. PMLR, 2022.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. Neural discrete representation learning. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Woo, H., Yoo, G., and Yoo, M. Structure learning-based task decomposition for reinforcement learning in non-stationary environments. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI ,February 22 - March 1, 2022*, pp. 8657–8665, 2022.

Xie, A., Harrison, J., and Finn, C. Deep reinforcement learning amidst continual structured non-stationarity. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pp. 11393–11403, 2021.

Xu, C., Amato, C., and Wong, L. Hierarchical robot navigation in novel environments using rough 2-D maps. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, volume 155, pp. 1971–1991. PMLR, 2021.

Xu, D., Nair, S., Zhu, Y., Gao, J., Garg, A., Fei-Fei, L., and Savarese, S. Neural task programming: Learning to generalize across hierarchical tasks. In *Proceedings of the 35th International Conference on Robotics and Automation (ICRA)*, pp. 3795–3802. IEEE, 2018.

Yu, T., Abbeel, P., Levine, S., and Finn, C. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018a.

Yu, T., Finn, C., Dasari, S., Xie, A., Zhang, T., Abbeel, P., and Levine, S. One-shot imitation from observing humans via domain-adaptive meta-learning. In *Proceedings of the Robotics: Science and Systems*, 2018b.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the 3rd Conference on robot learning (CoRL)*, pp. 1094–1100. PMLR, 2020.

Zheng, K., Chen, X., Jenkins, O., and Wang, X. E. VLMbench: A compositional benchmark for vision-and-language manipulation. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

Zintgraf, L. M., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep RL via meta-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

A. Environments and Dataset

A.1. Multi-Stage Meta-world

Multi-stage Meta-world is a robotic arm manipulation environment specifically designed for the execution of multi-stage tasks. For each task, multiple objects are positioned on a desk, where each object corresponds to existing tasks in Meta-world (Yu et al., 2020). The successful completion of a task is determined by the execution of N subtasks in a predetermined sequence, similar to the Franka Kitchen environment (Gupta et al., 2019).

The state space is 140-dimensional, consisting of 4-dimensional position information of the robot arm, and 136-dimensional position (e.g., the location of the drawer) and status (e.g., whether the drawer is opened or closed) of objects. The action space is 4-dimensional, consisting of a 3-dimensional directional vector applied to the end-effector of the robot arm, and an 1-dimensional torque vector applied to the gripper. An example including the robot arm and several objects is illustrated in Figure 7. The reward function in the multi-stage Meta-world environment follows the same success metrics of Meta-world in a subtask level.

Figure 7. Multi-stage Meta-world

Non-stationarity. To emulate a non-stationary environment, we adopt a time-varying kinematic parameter of the robot arm. Specifically, the environment involves such a kinematic parameter w_t that varies for each timestep t, i.e.,

$$\begin{aligned} w_t &= m + 0.25b \times sin(\rho_t) \\ \rho_t &= w_{t-1} + 0.75\pi \times z \end{aligned} \tag{19}$$

where $w_0 = l$, and z is a standard normal distribution $\mathcal{N}(0, 1)$. Note that b, l and m are hyperparameters to represent various configurations of environment dynamics. Accordingly, given action a_t, the environment yields the next state which is sampled from the transition probability $s'_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t + w_t)$.

A.2. Expert Data Collection

For expert data collection, we devise rule-based policies and execute them in a stationary environment. Regarding environment hidden dynamics, we use 13 different stationary environment configurations with dynamics parameters $b = 0$ and $m \in \{-0.3, -0.25, -0.2, -0.15, -0.1, -0.05, 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$ in (19). Based on the initial location of objects and subtask execution orders, we have 24 different tasks. We use 16 tasks for training and use 8 tasks for evaluation, where for each task, 15 episodes are generated. Accordingly, we build a dataset of 15 episodes for each of the 16 tasks in 13 different environment settings with distinct dynamics parameters, containing 3,120 trajectories (by $16 \times 15 \times 13$).

B. Experiment Details

B.1. OnIS Implementation

Table 6. Hyperparameters for S-OnIS

Hyperparameter	Value
Batch size	1024
Prompt θ_p	16 vector in 512-dim
Frames H of video clip	30
Contrastive α (for (5))	1
Dynamics encoder ψ_{enc}	3 FC with LSTM
Activation function for ψ_{enc}	Leaky ReLU
Inverse dynamics decoder ψ_{dec}	5 FC with 128 units
Activation function for ψ_{dec}	Leaky ReLU
The number of codebooks M for ψ_{enc}	20
Dimension of codebook	10
Learning rate for ψ_{enc}	5.55e-4
Length of sub-trajectory H_0	3
Semantic skill decoder Φ_{dec}	4 FC with 128 units
Activation function for Φ_{dec}	Leaky ReLU
Learning rate for Φ_{dec}	1e-4
Learning rate for ψ_{dec}	1e-4
Skill transfer module π_{tr}	5 FC with 128 units
Activation function for π_{tr}	Leaky ReLU
Learning rate for π_{tr}	1e-4

S-OnIS. S-OnIS involves 4-phases of training: (1) semantic skill sequence encoder Φ_{enc}, (2) semantic skill decoder Φ_{dec}, (3) dynamics encoder ψ_{enc}, and (4) skill transfer module π_{tr}.

During the training process of the semantic skill sequence encoder, the video demonstration being passed through the encoder Φ_{enc} creates a physical time bottleneck in the end-to-end training procedure of S-OnIS. To address this issue, we assign the inferred skill $z_t = \Phi_{enc}(v_{t:t+H})$ to each tuple (s_t, v_t, l_t, a_t) in the dataset after training Φ_{enc}.

During the training process of the semantic skill decoder Φ_{dec}, we have observed that the semantic skills within our dataset, consisting of long-horizon episodes, change infrequently (i.e., a maximum of 4 times). As a result, the semantic skill decoder should predominantly predict 0 for most timesteps, leading to significantly lower performance in binary classification. To address this issue, we assign the label *skill done* to each timestep in the dataset. This label

indicates the end of a skill for a specific timestep as well as the preceding and subsequent 5-step intervals of the skill. Then, we have

$$\textit{skill done}_t = \begin{cases} 1 & \text{if } \Phi_{enc}(v_{i:i+H}) \neq \Phi_{enc}(v_{i+1:i+1+H}) \\ 0 & o.w \end{cases} \tag{20}$$

where $i \in [t-5, t+5]$. The semantic skill decoder Φ_{dec} is trained to minimize binary cross entropy

$$\mathcal{L}_{tp}(\tau) = \text{BCE}(\textit{skill done}_t, \Phi_{dec}(s_{t_0}, s_t, z_t)) \tag{21}$$

in (12).

In the training procedure for the dynamics encoder ψ_{enc}, to enable backpropagation through non-differentiable quantization operator $\mathbf{q}$, we use the straight-through gradient estimator as explained in (van den Oord et al., 2017).

The skill transfer module π_{tr} is implemented in a simple MLP which is trained via behavior cloning. The hyperparameter settings for S-OnIS are summarized in Table 6.

Table 7. Hyperparameters for U-OnIS

Hyperparameter	Value
Batch size	1024
Encoder θ_v	Causal transformer
Encoder θ_l	Causal transformer
The number of layers for θ_v, θ_l	1
The number of heads for θ_v, θ_l	4
The number of codebooks K for θ_l	20
Learning rate for θ_v, θ_l	5e-5
Dimension of codebook for θ_l	10
Length of sub-trajectory H	20
Contrastive α (for (9))	1
Dynamics encoder ψ_{enc}	3 FC with LSTM
Activation function for ψ_{enc}	Leaky ReLU
Inverse dynamics decoder ψ_{dec}	5 FC with 128 units
Activation function for ψ_{dec}	Leaky ReLU
The number of codebook for ψ_{enc}	20
Dimension of codebook for ψ_{enc}	10
Learning rate for ψ_{enc}	5.55e-4
Learning rate for ψ_{dec}	1e-4
Length of sub-trajectory H_0	3
Semantic skill decoder Φ_{dec}	4 FC with 128 units
Activation function for Φ_{dec}	Leaky ReLU
Learning rate for Φ_{dec}	1e-4
Skill transfer module π_{tr}	5 FC with 128 units
Activation function for π_{tr}	Leaky ReLU
Learning rate for π_{tr}	1e-4

U-OnIS. U-OnIS involves 4-phases of training: (1) semantic skill sequence encoder Φ_{enc}, (2) semantic skill decoder Φ_{dec}, (3) dynamics encoder ψ_{enc}, and (4) skill transfer module π_{tr}.

In the training procedure for the semantic skill sequence encoder Φ_{enc}, we use two additional encoders θ_v and θ_l which are similar to the skill predictor used in (Garg et al., 2022), as explained in Section 3.2. Encoder θ_l takes $(\Phi_l(l), s_{t-H:t})$ as input, and maps to a quantized semantic skill z_t as

$$\theta_l = \mathbf{q} \circ \theta_l^c : (\Phi_l(l), s_{t-H:t}) \mapsto z_t \tag{22}$$

where θ_l^c maps the input $(\Phi_l(l), s_{t-H:t})$ to a continuous latent vector. As explained in (15), the quantization operator $\mathbf{q}$ maps the output of θ_l^c to one of the vectors in the skill codebook $\{c_1, \cdots, c_K\}$. That is, quantized semantic skill embedding z_t at each timestep t is defined as

$$z_t = \underset{j \in \{1, \cdots, K\}}{\text{argmin}} \{\|\theta_l^c(\Phi_l(l), s_{t-H:t}) - c_j\|\}. \tag{23}$$

Note that the contrastive loss presented in (9) is only responsible for updating the encoder θ_v, and there is no gradient from (9) to update the encoder θ_l. The whole procedure of U-OnIS is presented in Algorithm 2 and the the hyperparameter settings for U-OnIS are summarized in Table 7.

Algorithm 2 One-shot imitation of U-OnIS

Semantic skill sequence encoder Φ_{enc}
Semantic skill decoder Φ_{dec}
Skill transfer module π_{tr}, Dynamics encoder ψ_{enc}
Environment env, Demonstration d
done $\leftarrow$ False, $t \leftarrow 0$, $\tau_t = queue(size = H_0)$
$s_0 \leftarrow env$.reset()
while not done **do**
  $z_t \leftarrow \Phi_{enc}(d, s_{t-H:t})$ using (7)
  $h_t \leftarrow \psi_{enc}(\tau_{t-1})$ using (14)
  $a_t \leftarrow \pi_{tr}(s_t, h_t, z_t)$ using (13)
  s_{t+1}, done $\leftarrow env$.step(a_t)
  if $\Phi_{dec}(s_{start}, s_t, z_t) = 1$ **then**
    $s_{start} \leftarrow s_t$
  end if
  $\tau_t.enqueue((s_t, a_t))$, $t \leftarrow t + 1$
end while

B.2. Baseline Implementation

BC-Z. BC-Z is an imitation learning framework which exploits multi-modal demonstrations. In BC-Z, a language regression network is contrastively trained on an expert demonstration by maximizing the similarity between the embeddings of video demonstration and the embeddings of language instruction. Instead of training a video encoder, we use the embedding vectors extracted from the vision-language pertained model CLIP. For a fair comparison, a history of state-actions is combined with either the vision or the language embedding as input to the policy. The policy network is trained via behavior cloning, conditioned on video or language embeddings. The hyperparameter settings for BC-Z are summarized in Table 8.

Table 8. Hyperparameters for BC-Z

Hyperparameter	Value
Batch size	1024
Language regression network	5 FC with 512 units
Length of expert sub-trajectory	10
Learning rate	1e-4

Decision Transformer (DT). DT is a Transformer-based behavior cloning model. Specifically, each token in language instruction is embedded into the latent vector using the pertained CLIP, and then it is concatenated with the state vector for the input of DT. The hyperparameter settings for DT are summarized in Table 9.

Table 9. Hyperparameters for DT

Hyperparameter	Value
Batch size	64
The number of layers	3
Embedding dimension	128
The number of heads	4
Dropout	0.1
Learning rate	1e-4

SPiRL. SPiRL is an unsupervised skill learning framework, which embeds an expert sub-trajectory into a latent vector skill and decodes it into a sequence of actions via a primitive policy. Specifically, the CLIP embedding vector of video is concatenated with a state and used as input of V-SPiRL, and the CLIP embedding of language instruction is concatenated with a state and used as input to L-SPiRL. The hyperparameter settings for SPiRL are summarized in Table 10.

Table 10. Hyperparameters for SPiRL

Hyperparameter	Value
Batch size	1024
Skill encoder	1 FC with LSTM
Primitive policy	7 FC with 128 units
Length of expert sub-trajectory	10
Skill dimension	20
Learning rate	1e-4

B.3. Evaluation Details

The distribution of dynamics at test time assumes to be completely out of the distribution of training time dynamics. Specifically, the parameter m is sparsely sampled (from 13 distinct values) according to uniform distribution of $[-0.3, 0.3]$ and ω_t is fixed in each trajectory at training time with $b = 0$ in (19). At test time, different hyperparameter values for b, m are sampled at each episode from $[1, 2]$ and $[-0.45, 0.45]$, respectively. Accordingly, ω_t is changed by (19) at each timestep.

B.4. Use Case Experiment Details

Noisy video demonstrations. We add noise to video demonstrations through three different methods (Color Jitter, Gaussian Noise, and Cutout) as described in (Buslaev et al., 2020).

- **Color jitter.** In each video frame, adjustments are made to the brightness, contrast, saturation, and hue, in a way that random noise vectors are sampled from uniform distributions of $[-0.1, 0.1]$, $[-0.1, 0.1]$, $[-0.1, 0.1]$, and $[-0.025, 0.025]$ and the noise vectors are then added to video demonstrations.
- **Gaussian noise.** In each video frame, we sample a noise vector from a Gaussian distribution with a mean of 1 and a variance ranging from 1 to 10. The sampled noise is then added to video demonstrations.
- **Cutout.** 16 square regions of size 4×4 are randomly cropped in each video frame.

Real-world video demonstration. We record various real-world video demonstrations in 700×700 resolution, which involve interactions with objects that represent different stages in multi-stage Meta-world. Then, we evaluate our framework with different noisy levels. We also conduct the performance evaluation after fine-tuning the semantic skill sequence encoder (in the row of w/ Fine-tune in Table 11), presuming the cases where fine-tuning is feasible if a real-world video demonstration is given with subtask-level instructions.

Table 11. Performance on real-world demonstrations

	wo/ noise	noisy	very noisy
Success Rate	91.67%	73.75%	54.38%
Skills Matching Ratio	100.0%	72.08%	52.92%
Success Rate (w/ Fine-tune)	91.67%	83.13%	67.50%
Skills Matching Ratio (w/ Fine-tune)	100.0%	89.16%	76.46%

As observed in Table 11, the skills matching ratio is positively correlated with the success rate, specifying that if correct semantic skills are predicted from the semantic skill sequence encoder, no performance degradation is expected for noisy real-world settings. When a single real-world demonstration is given with a subtask instruction, it is feasible to improve the success rate by tuning the language prompt of the semantic skill sequence encoder. As we test several real-world demonstrations with robotic manipulation in multi-stage Meta-world scenarios, we observe that the performance in the success rate can be rather dependent

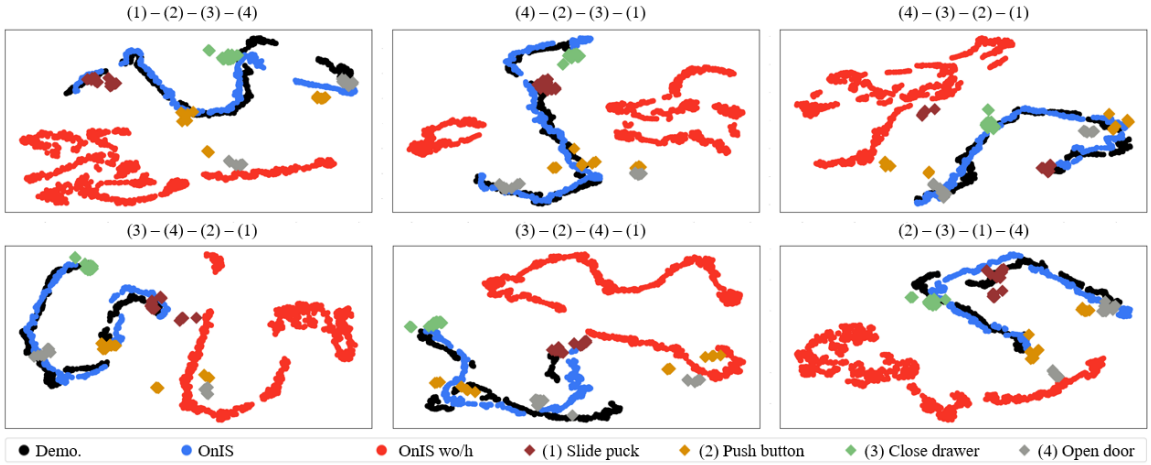

Figure 8. T-SNE embedding maps: (i) a video demonstration performing 4 subtasks (Demo.), (ii) a video generated by OnIS, and (iii) a video generated by OnIS without dynamics embedding (OnIS wo/h). Expert videos for each subtask (e.g., Slide puck, Push button, Close drawer, Open door) are represented in the same space.

on the capability of underlying pretrained vision-language models.

Instruction variation. Table 12 describes the examples of the instruction variations in Table 5.

Table 12. Examples of instruction variations

Verb replacement
push button → press the button
open door → pull the door
close drawer → shut the drawer
slide puck → move the puck
Modifier extension
push button → push the red button
open door → open the black door
close drawer → close the green drawer
slide puck → slide the black puck
Verbose instruction
push button → press the red button on yellow box
open door → open the closed black door on the desk
close drawer → close the green drawer on the desk
slide puck → move the black plate to the white goal on the left side

C. Additional Experiments

C.1. Additional Experiments for Unseen Environments

Table 13 shows the performance of OnIS when dynamics parameter m in the seen task increases from -0.6 to 0. As shown, S-OnIS yields the best performance for all K (the number of subtasks), implying that S-OnIS is robust to unseen dynamics. This result corresponds to Section 4.3. Furthermore, Figure 9 shows that OnIS achieves the most robust performance as the dynamics level $|m|$ in (19) increases (e.g., to unseen dynamics) over previously trained dynamics. This specifies that the dynamics embedding procedure of OnIS is able to disentangle dynamics information from expert trajectories.

Table 13. Performance in the success rate for multi-stage Meta-world with language instructions

K	L-DT	L-SPiRL	BC-Z	U-OnIS	S-OnIS
1	57.81%	29.68%	78.13%	70.32%	**94.79%**
2	33.59%	37.50%	26.57%	66.57%	**77.04%**
4	21.09%	17.96%	18.60%	47.44%	**60.58%**

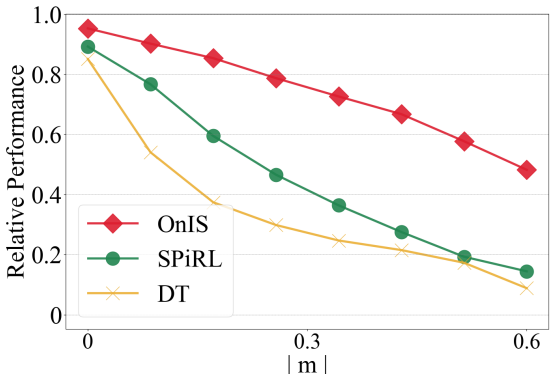

Figure 9. Relative performance of unseen dynamics to previously trained dynamics

C.2. Long-Horizon Multi-Stage Tasks

To investigate whether our approach can tackle long-horizon multi-stage tasks with a large number of skills, we evaluate OnIS with manipulation tasks that involve up to 10 objects.

Table 14. Experiments for Long-Horizon Tasks

	Number of objects (K)									
	1	2	3	4	5	6	7	8	9	10
OnIS(Language Instruction)	99.33%	76.0%	74.66%	69.33%	65.33%	58.0%	50.66%	44.0%	36.0%	26.0%
OnIS(Video Demonstration)	99.33%	72.0%	64.0%	60.33%	42.0%	29.0%	10.03%	4.0%	1.33%	0.0%

As shown in Table 14, OnIS is able to solve the tasks with up to 10 objects ($K = 1, \ldots, 10$). This clarifies that the semantic skill sequence encoder scales well to predict semantic skills even when a task involves multiple subtasks of up to 10.

C.3. Visualization of Semantic Reconstruction

Figure 8 represents the embedding maps of the expert demonstration (Demo.), U-OnIS (OnIS), and OnIS trained without dynamics embedding (OnIS wo/h) for 6 different tasks. Similar to the experiment in Section 4.3 in the main manuscript, the embeddings of episodic run by U-OnIS and demonstrations show highly similar paths, meaning that their semantic skills are aligned. On the other hand, the OnIS without dynamics embedding has low semantic skill correspondence with demonstrations.

C.4. Additional Baseline: Multigame-DT

Table 15. One-shot imitation performance for language instruction

K	Non-st.	Multi-DT	L-DT	BC-Z	U-OnIS	S-OnIS
1	Stationary	100.0%	92.66%	100.0%	87.70%	100.0%
	Low	85.58%	62.49%	**96.61%**	90.40%	95.00%
	Medium	73.07%	53.84%	75.32%	81.29%	**89.04%**
	High	66.34%	20.19%	68.91%	82.19%	**88.50%**
2	Stationary	68.75%	58.33%	76.25%	74.50%	**95.00%**
	Low	53.84%	47.12%	45.27%	60.44%	**83.17%**
	Medium	52.40%	33.17%	35.00%	69.24%	**78.50%**
	High	42.79%	30.28%	25.52%	66.67%	**77.27%**
4	Stationary	31.25%	37.73%	40.30%	67.71%	**87.50%**
	Low	20.55%	22.11%	20.23%	59.64%	**76.60%**
	Medium	18.51%	20.43%	20.60%	56.68%	**71.39%**
	High	14.90%	8.90%	11.13%	54.82%	**66.92%**

We perform the experiment with the latest version of DT, a multigame-DT (Lee et al., 2022). Table 15 shows the performance in multi-stage Meta-world achieved by OnIS and baselines including Multi-DT. We observe that Multi-DT achieves better than L-DT with an average gain of 11.73%, but shows worse performance than S-OnIS with an average of loss of 31.74% For fair comparisons, we use video and language embeddings from the pretrained CLIP model as an additional input to Multi-DT, as well as other baselines. Since Multi-DT uses rewards in offline datasets (unlike OnIS without any rewards), we consider that Multi-DT has some advantage over the original DT in terms of policy learning. Otherwise, Multi-DT becomes similar to the original DT. In our problem setting of one-shot imitation for multi-stage tasks, Multi-DT is yet limited in exploring task compositionality.

C.5. Ablation Studies

Contrastive learning on dynamics encoder. To investigate the effect of contrastive learning-based dynamics embedding in our proposed OnIS framework, we evaluate and compare several different implementations for dynamics embedding as part of ablation. Table 16 illustrates those including the cases of whether the contrastive loss is added to the typical dynamics embedding loss or not (w/ Contra and wo/ Contra, respectively, in the column names), the cases of whether the dynamics embedding vector $\psi_{enc}(\tau)$ is quantized or not (w/ VQ and wo/ VQ, respectively, in the column names), and the case of reconstruction loss (18) is not added (wo/ Recon, in the column name) to the dynamics embedding loss.

Here, the typical dynamics embedding loss is based on reconstruction, as in the formula in (18), similar to several related works such as (Zintgraf et al., 2020; Woo et al., 2022; Xie et al., 2021):

$$\mathcal{L}_{DE} = \mathbb{E}_{t-H_0<i\leq t}\left[\|a_i - \psi_{dec}(s_i, s_{i+1}, \psi_{enc}(\tau_t)\|^2\right] \tag{24}$$

where ψ_{enc} and ψ_{dec} denote the dynamics encoder and decoder, respectively.

As observed in Table 16, the contrastively trained dynamics encoder (w/ Contra) improves the performance by 19.8% in average success rates for multi-stage Meta-world tasks, compared to the dynamics encoder trained without contrastive learning (wo/ Contra).

Moreover, we also observe that quantizing the output of the dynamics encoder (w/ VQ) improves the performance by 2.2% in average success rates, compared to the dynamics embedding without quantization (wo/ VQ). Additionally, the reconstruction loss in the training of dynamics encoder (w/ Contra, w/VQ) improves the performance by 6.21%, compared to the dynamics encoder trained without reconstruction loss (wo/ Recon).

These ablation results with such performance gains specify that contrastive learning enables the dynamics encoder to disentangle task-invariant dynamics-relevant features from expert demonstrations. The task information - what to do, such as open door and push button - is extracted by the semantic skill sequence encoder in prior, and then it is feasible to enforce different sub-trajectories sharing the same dynamics have a close proximity in the dynamics embedding space, regardless of which skills the sub-trajectories involve. As two sub-trajectories in a trajectory are used as a positive

Table 16. Ablation: Contrastive Learning on Dynamics Encoder

K	Non-st.	w/ Contra, w/ VQ	w/ Contra, wo/ VQ	wo/ Contra, w/ VQ	wo/ Contra, wo/ VQ	wo/ Recon
1	Stationary	100.0%	100.0%	100.0%	100.0%	100.0%
	Low	**95.01%**	93.84%	92.30%	89.42%	89.10%
	Medium	**94.55%**	92.11%	83.65%	77.24%	84.40%
	High	88.50%	**89.42%**	72.62%	61.21%	81.76%
2	Stationary	100.0%	100.0%	68.75%	79.16%	95.10%
	Low	**85.82%**	83.75%	66.98%	76.60%	75.04%
	Medium	**84.42%**	82.37%	61.37%	58.52%	74.45%
	High	**81.14%**	69.07%	49.34%	48.96%	75.04%
4	Stationary	87.50%	**88.09%**	46.88%	58.34%	85.50%
	Low	**80.11%**	74.45%	45.97%	53.50%	71.96%
	Medium	75.24%	**75.58%**	43.50%	44.31%	67.04%
	High	**66.92%**	63.19%	34.39%	34.13%	65.32%

pair in our sampling strategy, the contrastive learning loss in (17) renders the aforementioned desired property of the dynamics embedding space.