Temporal Representation Alignment: Emergent Compositionality in Instruction Following with Successor Features

Vivek Myers* UC Berkeley

Bill Chunyuan Zheng* UC Berkeley

Anca Dragan UC Berkeley

Kuan Fang Cornell University

Sergey Levine UC Berkeley

February 6, 2025

Abstract

Effective task representations should facilitate compositionality, such that after learning a variety of basic tasks, an agent can perform compound tasks consisting of multiple steps simply by composing the representations of the constituent steps together. While this is conceptually simple and appealing, it is not clear how to automatically learn representations that enable this sort of compositionality. We show that learning to associate the representations of current and future states with a temporal alignment loss can improve compositional generalization, even in the absence of any explicit subtask planning or reinforcement learning. This approach is able to generalize to novel composite tasks specified as goal images or language instructions, without assuming any additional reward supervision or explicit subtask planning. We evaluate our approach across diverse robotic manipulation tasks as well as in simulation, showing substantial improvements for tasks specified with either language or goal images.

1 Introduction

Compositionality is a core aspect of intelligent behavior, describing the ability to sequence previously learned capabilities and solve new tasks (Lashley, 1951). In domains involving long-horizon decision-making like robotics, various learning approaches have been proposed to enable this property, including hierarchical learning (Kulkarni et al., 2016), explicit subtask planning (Ahn et al., 2022; Fang et al., 2022b; Schrittwieser et al., 2021), and dynamic-programming-based "stitching" (Ghugare et al., 2023; Kostrikov et al., 2022). In practice, these techniques are often unstable or data-inefficient in real-world robotics settings, making them difficult to scale (Laidlaw et al., 2024).

By contrast, humans and animals are adept at quickly composing behaviors to reach new goals (Lashley, 1951). Possible explanations for these capabilities have been proposed, including the ability to perform transitive inference (Ciranka et al., 2022), learn successor representations and causal models (Dayan, 1993b; Gopnik et al., 2017), and plan with world models (Vikbladh et al., 2019). In common among these theories is the idea of learning structured representations of the world, which inference about which actions will lead to certain goals.

How might these concepts translate to algorithms for robot learning? In this work, we study how adding an auxiliary successor representation learning objective affects compositional behavior in a real-world tabletop manipulation setting. We show that learning this representation structure improves the ability of the robot to perform long-horizon, compositionally-new tasks, specified either through goal images or natural language instructions. Perhaps surprisingly, we found that this temporal alignment does not need to be used for training the policy or test-time inference, as long as it is used as an auxiliary loss over the same representations used for the tasks. An example of this can be seen in Fig. 1.

We compare our method, **T**emporal **R**epresentation **A**lignment (TRA), against past imitation and rein-

Website: https://anonymous.4open.science/w/tra-website-B10A

*Equal contribution.

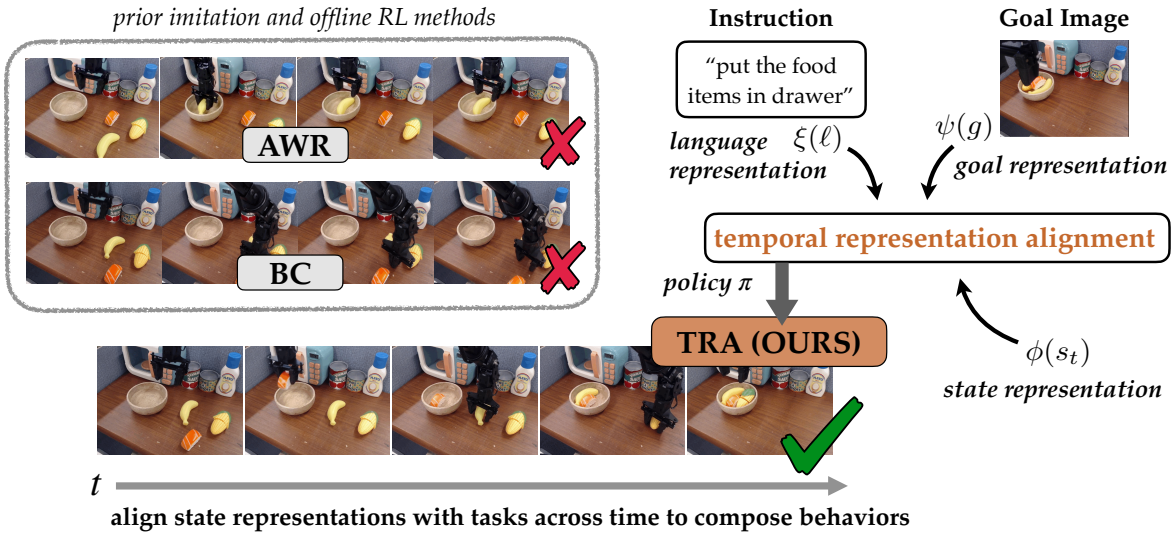


Figure 1: We show our Temporal Representation Alignment (TRA) method performing a language task, "put all food items in the bowl." TRA adds a time-contrastive loss for learning task representations to use with a goal- and language-conditioned policy. While TRA can implicitly decompose the task into steps and execute them one by one, the behavioral cloning (BC) and offline RL (AWR) methods fail at this compositional task. The structured representations learned by TRA enable this compositional behavior without explicit planning or hierarchical structure.

forcement learning baselines across a set of challenging multi-step manipulation tasks in the BridgeData setup (Walke et al., 2023) as well as the OGBench simulation benchmark (Park et al., 2025). Unlike prior work in setup, we focus on the compositional capabilities of the robot policies: as a whole, the tasks are out-of-distribution, but each distinct subtask can be described through a goal image that lies in the training distribution. Adding a simple time-contrastive alignment loss improves compositional performance on these tasks by >40% across 13 tasks in 4 evaluation scenes, and simulation results show better performance compared to behavioral cloning (i.e., no structured representation learning), and comparable performance to offline RL methods that explicitly learn and use a value function.

2 Related Work

Our approach builds upon prior work on goal- and language-conditioned control, focusing particularly on the problem of compositional generalization.

Robot manipulation with language and goals. Recent improvements in robot learning datasets have enabled the development of robot policies that can be commanded with image goals and language instructions (Ahn et al., 2022; Shridhar et al., 2021; Walke et al., 2023). These policies can be trained with goal- and language-conditioned imitation learning from human demonstrations (Brohan et al., 2023; Chowdhery et al., 2023; Jiang et al., 2023; Lynch and Sermanet, 2021; Lynch et al., 2023), reinforcement learning (Chebotar et al., 2023; Chen et al., 2021), or other forms of supervision (Bobu et al., 2023; Cui et al., 2023). When trained to reach goals, methods can additionally use hindsight relabeling (Andrychowicz et al., 2017; Kaelbling, 1993) to improve performance (Dehaene et al., 2022; Ding et al., 2019; Myers et al., 2023; Walke et al., 2023). Our work shows how the benefits of goal-conditioned and language-conditioned supervised learning can be combined with temporal representation alignment to enable compositionality that would otherwise require planning or reinforcement learning.

Compositional generalization in sequential decision making. In the context of decision making, compositional generalization refers to the ability to generalize to new behaviors that are composed of known sub-behaviors (Rubino et al., 2023; Steedman, 2004). Biological learning systems show strong compositional generalization abilities (Ciranka et al., 2022; Dehaene et al., 2022; Dickins, 2011; Lake et al., 2019), and recent work has explored how similar capabilities can be achieved in artificial systems (Akyürek et al., 2021; Ito et al., 2022; Lewis et al., 2024). In the context of policy learning, exploiting the compositionality of the behaviors can lead to generalization to unseen and temporarily extended tasks (Fang et al., 2022b, 2019; Ghugare et al., 2023; Kumar et al., 2022; Mandlekar et al., 2021; Nasiriany et al., 2019). Hierarchical and planning-based approaches also aim to enable compositional behavior by explicitly partitioning a task into its components (Fang et al., 2022a; Myers et al., 2024a; Park et al., 2023; Zhang et al., 2022). With im-

provements in vision-language models (VLMs), many recent works have explored using a pre-trained VLM to decompose a task into subtasks that are more attainable for the low-level manipulation policy (Ahn et al., 2022; Attarian et al., 2022; Belkhale et al., 2024; Kwon et al., 2023; Myers et al., 2024a; Singh et al., 2023; Zhang et al., 2023). These approaches are limited by the need for robust pre-trained models that can be fine-tuned and prompted for embodied tasks. Our contribution is to show compositional properties can be achieved *without* any explicit hierarchical structure or planning, by learning a structured representation through time-contrastive representation alignment.

Representation learning for states and tasks. State and task representations for decision making aim to improve generalization and exploit additional sources of data. Recent work in the robotics domain have explored the use of pre-trained representations across multimodal data, including images and language, for downstream tasks (Cui et al., 2022; Jang et al., 2021; Karamcheti et al., 2023; Li et al., 2022; Ma et al., 2023a; Myers et al., 2023; Nair et al., 2022; Pari et al., 2022; Shah and Kumar, 2021). In reinforcement learning problems, representations are often trained to predict future states, rewards, goals, or actions (Anand et al., 2019; Fan et al., 2022; Ma et al., 2023b; Zhang et al., 2021), and can improve generalization and sample efficiency when used as value functions (Barreto et al., 2017; Blier et al., 2021; Choi et al., 2021; Dayan, 1993a; Dosovitskiy and Koltun, 2017). Some recent works have explored the use of additional structural constraints on representations to enable planning (Eysenbach et al., 2024; Fang et al., 2022a; Hafner et al., 2019; Myers et al., 2025; Zhang et al., 2022), or enforced metric properties to improve compositional generalization (Liu et al., 2023; Myers et al., 2024b; Wang et al., 2023).

The key distinction between our approach and past contrastive representation methods for robotics like VIP (Ma et al., 2023b), GRIF (Myers et al., 2023), and R3M (Nair et al., 2022) is that we focus on the real-world compositional generalization capabilities enabled by simply aligning representations across time in addition to the task modalities, without using the learned representations for policy extraction or defining a value function.

3 Temporal Representation Alignment

When training a series of short-horizon goal-reaching and instruction-following tasks, our goal is to learn a representation space such that our policy can generalize to a new (long-horizon) task that can be viewed as a sequence of known subtasks. We propose to structure this representation space by aligning the representations of states, goals, and language in a way that is more amenable to compositional generalization.

Notation. We take the setting of a goal- and language-conditioned MDP $\mathcal{M}$ with state space $\mathcal{S}$, continuous action space $\mathcal{A} \subseteq (0,1)^{d_{\mathcal{A}}}$, initial state distribution p_0, dynamics $\mathrm{P}(s' \mid s, a)$, discount factor γ, and language task distribution p_ℓ. A policy $\pi(a \mid s)$ maps states to a distribution over actions. We inductively define the k-step (action-conditioned) policy visitation distribution as:

$$\begin{aligned} p_1^\pi(s_1 \mid s_1, a_1) &\triangleq p(s_1 \mid s_1, a_1), \\ p_{k+1}^\pi(s_{k+1} \mid s_1, a_1) &\triangleq \\ &\int_{\mathcal{A}} \int_{\mathcal{S}} p(s_{k+1} \mid s, a)\, \mathrm{d}p_k^\pi(s \mid s_1, a_1)\, \mathrm{d}\pi(a \mid s) \\ p_{k+t}^\pi(s_{k+t} \mid s_t, a_t) &\triangleq p^\pi(s_k \mid s_1, a_1). \end{aligned} \tag{1}$$

Then, the discounted state visitation distribution can be defined as the distribution over s^+, the state reached after $K \sim \mathrm{Geom}(1-\gamma)$ steps:

$$p_\gamma^\pi(s^+ \mid s, a) \triangleq \sum_{k=0}^{\infty} \gamma^k p_k^\pi(s^+ \mid s, a). \tag{2}$$

We assume access to a dataset of expert demonstrations $\mathcal{D} = \{\tau_i, \ell_i\}_{i=1}^K$, where each trajectory

$$\tau_i = \{s_{t,i}, a_{t,i}\}_{t=1}^H \in \mathcal{S} \times \mathcal{A} \tag{3}$$

is gathered by an expert policy π^{E}, and is then annotated with $p_\ell(\ell_i \mid s_{1,i}, s_{H,i})$. Our aim is to learn a policy π that can select actions conditioned on a new language instruction ℓ. As in prior work (Walke et al., 2023), we handle the continuous action space by both our policy and the expert policy as an isotropic Gaussian with fixed variance; we will equivalently write $\pi(a \mid s, \varphi)$ or denote the mode as $\hat{a} = \pi(s, \varphi)$ for a task φ.

3.1 Representations for Reaching Distant Goals

We learn a goal-conditioned policy $\pi(a \mid s, g)$ that selects actions to reach a goal g from expert demonstrations with behavioral cloning. Suppose we directly selected actions to imitate the expert on two trajectories in $\mathcal{D}$:

$$\left.\begin{array}{l} s_1 \longrightarrow s_2 \longrightarrow \ldots \longrightarrow s_H \longrightarrow w \\ w \longrightarrow s'_1 \longrightarrow \ldots \longrightarrow s'_H \longrightarrow g \end{array}\right\} \ \tau_i \in \mathcal{D} \tag{4}$$

When conditioned with the composed goal g, we would be unable to imitate effectively as the composed state-goal (s, g) is jointly out of the training distribution.

What *would* work for reaching g is to first condition the policy on the intermediate waypoint w, then upon reaching w, condition on the goal g, as the state-goal

pairs (s_i, w), (w, g), and (s_i', g) are all in the training distribution. If we condition the policy on some intermediate waypoint distribution $p(w)$ (or sufficient statistics thereof) that captures all of these cases, we can stitch together the expert behaviors to reach the goal g.

Our approach is to learn a representation space that captures this ability, so that a GCBC objective used in this space can effectively imitate the expert on the composed task. We begin with the goal-conditioned behavioral cloning (Kaelbling, 1993) loss $\mathcal{L}_{\text{BC}}^{\phi,\psi,\xi}$ conditioned with waypoints w.

$$\mathcal{L}_{\text{BC}}\big(\{s_i, a_i, s_i^+, g_i\}_{i=1}^K\big) = \sum_{i=1}^K \log \pi\big(a_i \mid s_i, \psi(g_i)\big). \quad (5)$$

Enforcing the invariance needed to stitch Eq. (4) then reduces to aligning $\psi(g) \leftrightarrow \psi(w)$. The temporal alignment objective $\phi(s) \leftrightarrow \phi(s^+)$ accomplishes this indirectly by aligning both $\psi(w)$ and $\psi(g)$ to the shared waypoint representation $\phi(w)$:

$$\mathcal{L}_{\text{NCE}}\big(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi\big) = \log\left(\frac{e^{\phi(s_i^+)^T \psi(s_i)}}{\sum_{j=1}^K e^{\phi(s_i^+)^T \psi(s_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{\phi(s_i^+)^T \psi(s_i)}}{\sum_{i=1}^K e^{\phi(s_i^+)^T \psi(s_j)}}\right) \quad (6)$$

3.2 Interfacing with Language Instructions

To extend the representations from Section 3.1 to compositional instruction following with language tasks, we need some way to ground language into the ψ (future state) representation space. We use a similar approach to GRIF (Myers et al., 2023), which uses an additional CLIP-style (Radford et al., 2021) contrastive alignment loss with an additional pretrained language encoder ξ:

$$\mathcal{L}_{\text{NCE}}\big(\{g_i, \ell_i\}_{i=1}^K; \psi, \xi\big) = \sum_{i=1}^K \log\left(\frac{e^{\psi(g_i)^T \xi(\ell_i)}}{\sum_{j=1}^K e^{\psi(g_i)^T \xi(\ell_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{\psi(g_i)^T \xi(\ell_i)}}{\sum_{i=1}^K e^{\psi(g_i)^T \xi(\ell_j)}}\right) \quad (7)$$

3.3 Temporal Alignment

The Temporal Representation Alignment (TRA) approach structures the representation space of goals and language instructions to better enable compositional generalization. We learn encoders ϕ, ψ, and ξ to map states, goals, and language instructions to a shared representation space.

$$\mathcal{L}_{\text{NCE}}(\{x_i, y_i\}_{i=1}^K; f, h) = \sum_{i=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{j=1}^K e^{f(y_i)^T h(x_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{i=1}^K e^{f(y_i)^T h(x_j)}}\right) \quad (8)$$

$$\mathcal{L}_{\text{BC}}\big(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi\big) = \sum_{i=1}^K \log \pi\big(a_i \mid s_i, \xi(\ell_i)\big) + \log \pi\big(a_i \mid s_i, \psi(s_i^+)\big) \quad (9)$$

$$\mathcal{L}_{\text{TRA}}\big(\{s_i, a_i, s_i^+, g_i, \ell_i\}_{i=1}^K; \pi, \phi, \psi, \xi\big) = \underbrace{\mathcal{L}_{\text{BC}}\big(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi, \psi, \xi\big)}_{\text{behavioral cloning}} + \underbrace{\mathcal{L}_{\text{NCE}}\big(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi\big)}_{\text{temporal alignment}} + \underbrace{\mathcal{L}_{\text{NCE}}\big(\{g_i, \ell_i\}_{i=1}^K; \psi, \xi\big)}_{\text{task alignment}} \quad (10)$$

Note that the NCE alignment loss uses a CLIP-style symmetric contrastive objective (Eysenbach et al., 2024; Radford et al., 2021) — we highlight the indices in the NCE alignment loss (8) for clarity.

Our overall objective is to minimize Eq. (10) across states, actions, future states, goals, and language tasks within the training data:

$$\min_{\pi,\phi,\psi,\xi} \mathbb{E}_{\substack{(s_{1,i}, a_{1,i}, \ldots, s_{H,i}, a_{H,i}, \ell) \sim \mathcal{D} \\ i \sim \text{Unif}(1 \ldots H) \\ k \sim \text{Geom}(1-\gamma)}} \Big[\mathcal{L}_{\text{TRA}}\big(\{s_{t,i}, a_{t,i}, s_{\min(t+k,H),i}, s_{H,i}, \ell\}_{i=1}^K; \pi, \phi, \psi, \xi\big)\Big]. \quad (11)$$

Algorithm 1: Temporal Representation Alignment (TRA)

input: dataset $\mathcal{D} = (\{s_{t,i}, a_{t,i}\}_{t=1}^H, \ell_i)_{i=1}^N$
initialize networks $\Theta \triangleq (\pi, \phi, \psi, \xi)$
while training **do**
  sample a batch of transitions $\{(s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i)\}_{i=1}^K \sim \mathcal{D}$ for $k \sim \text{Geom}(1-\gamma)$
  $\Theta \leftarrow (\pi, \phi, \psi, \xi) - \alpha \nabla_\Theta \mathcal{L}_{\text{TRA}}\big(\{s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i\}_{i=1}^K; \Theta\big)$
output: language ℓ-conditioned policy $\pi(a_t | s_t, \xi(\ell))$
  goal g-conditioned policy $\pi(a_t | s_t, \psi(g))$

A summary of our approach is shown in Algorithm 1.

3.4 Temporal Alignment and Compositionality

We will formalize the intuition from Section 3.1 that TRA enables compositional generalization by considering the error on a "compositional" version of $\mathcal{D}$,

denoted $\mathcal{D}^*$. Using the notation from Eq. (3), we can say $\mathcal{D}$ is distributed according to:

$$\mathcal{D} \triangleq \mathcal{D}^H \sim \prod_{i=1}^{K} p_0(s_{1,i}) p_\ell(\ell_i \mid s_{1,i}, s_{H,i}) \prod_{t=1}^{H} \pi^{\mathrm{E}}(a_{t,i} \mid s_{t,i}) \, \mathrm{P}(s_{t+1,i} \mid s_{t,i}, a_{t,i}), \quad (12)$$

or equivalently

$$\mathcal{D}^H \sim \prod_{i=1}^{K} p_0(s_{1,i}) p_\ell(\ell_i \mid s_{1,i}, s_{H,i}) \prod_{t=1}^{H} e^{\sigma^2 \|\pi^{\mathrm{E}}(s_{t,i}) - a_{t,i}\|^2} \, \mathrm{P}(s_{t+1,i} \mid s_{t,i}, a_{t,i}), \quad (13)$$

by the isotropic Gaussian assumption. We will define $\mathcal{D}^* \triangleq \mathcal{D}^{H'}$ to be a longer-horizon version of $\mathcal{D}$ extending the behaviors gathered under π^{E} across a horizon $\alpha H \geq H' \geq H$ that additionally satisfies a "time-isotropy" property: the marginal distribution of the states is uniform across the horizon, i.e., $p_0(s_{1,i}) = p_0(s_{t,i})$ for all $t \in \{1 \dots H'\}$.

We will relate the in-distribution imitation error $\textsc{Err}(\bullet; \mathcal{D})$ to the compositional out-of-distribution imitation error $\textsc{Err}(\bullet; \mathcal{D}^*)$. We define

$$\textsc{Err}(\hat{\pi}; \tilde{\mathcal{D}}) = \mathbb{E}_{\tilde{\mathcal{D}}}\Big[\frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{\hat{\pi}}\left[\|\tilde{a}_{t,i} - \hat{\pi}(\tilde{s}_{t,i}, \tilde{s}_{H,i})\|^2 / d_{\mathcal{A}}\right]\Big] \quad \text{for} \quad \{\tilde{s}_{t,i}, \tilde{a}_{t,i}, \tilde{\ell}_i\}_{t=1}^{H} \sim \tilde{\mathcal{D}}. \quad (14)$$

On the training dataset this is equivalent to the expected behavioral cloning loss from Eq. (9).

Assumption 1. *The policy factorizes through inferred waypoints as:*

$$\textit{goals: } \pi(a \mid s, g) = \int \pi(a \mid s, w) \, \mathrm{P}(s_t = w \mid s_{t+k} = g) \, \mathrm{d}w \quad (15)$$

$$\textit{language: } \pi(a \mid s, \ell) = \int \pi(a \mid s, w) \, \mathrm{P}(s_t = w \mid s_{t+k} = g) \, \mathrm{P}(s_{t+k} = g \mid \ell) \, \mathrm{d}w \, \mathrm{d}g, \quad (16)$$

where denote by $\pi(s, g)$ the MLE estimate of the action a.

Theorem 1. *Suppose $\mathcal{D}$ is distributed according to Eq. (12) and $\mathcal{D}^*$ is distributed according to Eq. (12). When $\gamma > 1 - 1/H$ and $\alpha > 1$, for optimal features ϕ and ψ under Eq. (11), we have*

$$\textsc{Err}(\pi; \mathcal{D}^*) \leq \textsc{Err}(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbb{1}\{\alpha > 2\}. \quad (17)$$

We can also define a notion of the language-conditioned compositional generalization error:

$$\textsc{Err}^{\ell}(\pi; \mathcal{D}^*) \triangleq \mathbb{E}_{\mathcal{D}^*}\Big[\frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{\pi}\left[\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{\ell}_i)\|^2\right]\Big].$$

Corollary 1.1. *Under the same conditions as Theorem 1,*

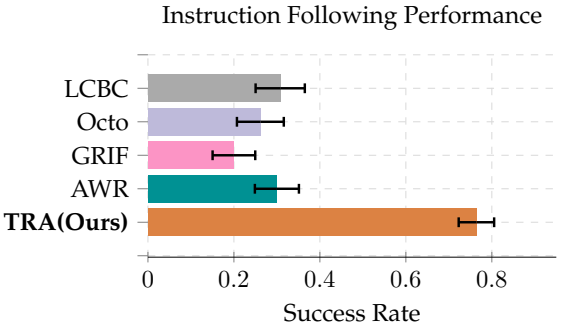
$$\textsc{Err}^{\ell}(\pi; \mathcal{D}^*) \leq \textsc{Err}^{\ell}(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbb{1}\{\alpha > 2\}.$$

The proofs as well as a visualization of the bound are in Appendix F. Policy implementation details can be found in Appendix B

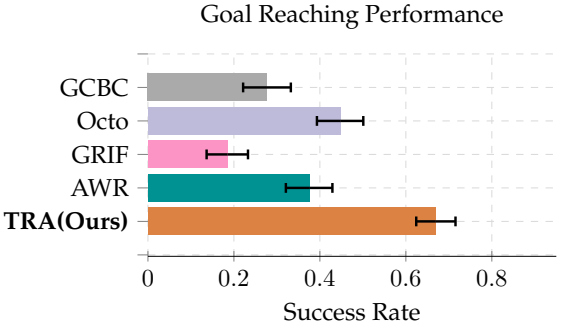
4 Experiments

Our experimental evaluation aims to answer the following research questions for TRA:

1. Can TRA enable zero-shot composition of sequential tasks without additional rewards or planning methods?
2. Does TRA improve compositional generalization over past methods?
3. How well does TRA capture skills that are less common within the dataset?
4. Is temporal alignment by itself sufficient for effective compositional generalization?



(a) Language instruction tasks



(b) Goal-image conditioned tasks

Figure 2: Aggregated performance on compositional generalization tasks, consisting of instruction-following and goal-reaching tasks.

4.1 Real-World Experimental Setup

We evaluate TRA on a collection of held-out *compositionally-OOD* tasks — tasks for which the individual substeps are represented in the dataset, but the combination of those steps is unseen. For example, in a task such as "removing a bell pepper from

Table 1: Real-world Evaluation

		Language-conditioned					Goal-conditioned				
	Task	**TRA**	GRIF	LCBC	Octo	AWR	**TRA**	GRIF	GCBC	Octo	AWR
(A)	open the drawer	**0.80**$^{(\pm 0.1)}$ †	0.20$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$ †	0.60$^{(\pm 0.2)}$	**0.40**$^{(\pm 0.2)}$	**0.50**$^{(\pm 0.2)}$	**0.80**$^{(\pm 0.2)}$
(A)	mushroom in drawer	**0.80**$^{(\pm 0.1)}$	**0.80**$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.60**$^{(\pm 0.2)}$	**0.90**$^{(\pm 0.1)}$	0.40$^{(\pm 0.2)}$	**0.80**$^{(\pm 0.2)}$	**0.90**$^{(\pm 0.1)}$	**0.60**$^{(\pm 0.2)}$
(A)	close drawer	**0.60**$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	**0.40**$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	**0.40**$^{(\pm 0.2)}$	**1.00**$^{(\pm 0.0)}$	0.40$^{(\pm 0.2)}$	0.80$^{(\pm 0.2)}$	0.60$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$
(D)	take the item out of the drawer	**0.60**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.40**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$
(B)	put the spoons on towels	**1.00**$^{(\pm 0.0)}$	0.40$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	**1.00**$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.60$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.60$^{(\pm 0.2)}$
(B)	put the spoons on the plates	**0.80**$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**1.00**$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.40$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.80$^{(\pm 0.2)}$
(C)	put the corn and sushi on plate	**0.90**$^{(\pm 0.1)}$	0.00$^{(\pm 0.0)}$	0.40$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.50$^{(\pm 0.2)}$	**0.70**$^{(\pm 0.1)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.30$^{(\pm 0.1)}$
(C)	sushi and mushroom in bowl	**0.80**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.60**$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	**0.40**$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$
(C)	corn, banana, and sushi in bowl	**0.80**$^{(\pm 0.1)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.1)}$	**0.50**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	**0.40**$^{(\pm 0.2)}$	0.50$^{(\pm 0.2)}$
(D)	corn on plate then sushi in pot	**0.70**$^{(\pm 0.1)}$	0.00$^{(\pm 0.0)}$	0.40$^{(\pm 0.2)}$	**0.60**$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	**0.30**$^{(\pm 0.1)}$	**0.20**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$
(A)	sweep to the right	**0.80**$^{(\pm 0.1)}$	0.20$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.70**$^{(\pm 0.1)}$	0.40$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.80**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$
(B)	fold cloth into the center	**1.00**$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	0.40$^{(\pm 0.2)}$	**1.00**$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.60$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$
(B)	move bell pepper and sweep towel	**0.50**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	0.00$^{(\pm 0.0)}$	0.20$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$	**0.60**$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	0.20$^{(\pm 0.2)}$	**0.40**$^{(\pm 0.2)}$	0.00$^{(\pm 0.0)}$

(A) One step tasks
(B) Task concatenation
(C) Semantic generalization
(D) Tasks with dependency

†The best-performing method(s) up to statistical significance are **highlighted**

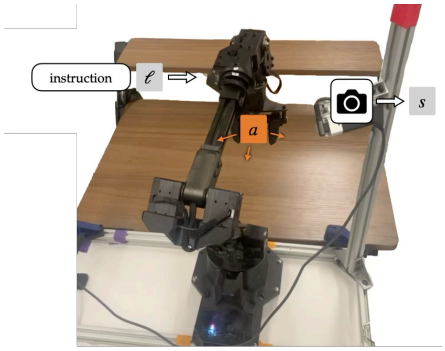


Figure 3: The tabletop manipulation setup used for the real-world evaluation (see Walke et al., 2023).

a towel, and then sweep the towel", both the tasks "remove the bell pepper from the towel" and "sweep the towel" have similar entries within BridgeData, but such a combination of behaviors is unseen. We utilize a real-world robot manipulation interface with a 7 DoF WidowX250 manipulator arm with 5Hz execution frequency. We train on an augmented version of the BridgeDataV2 dataset (Walke et al., 2023), which contains over 50k trajectories with 72k language annotations. More details can be seen Appendix B.

In order to specifically test the ability of TRA to perform compositional generalization, we organize our evaluation tasks into 4 scenes that are unseen in BridgeData, each with increasing difficulty:

Set (A) – One-Step: These are the only tasks that are not compositionally-OOD, as all the tasks are one-step tasks. These tasks involve opening, putting an item in, and closing a drawer, and have been seen in BridgeData, although at a lower frequency than object manipulation, and with new positions. We use these tasks to compare TRA's performance on single-step tasks relative to baselines.

Set (B) – Task Concatenation: These tasks scene involves concatenating multiple tasks of the same nature in sequence, where a robot must be able to perform all tasks within the same trajectory. During evaluation, we instruct the policy with instructions such as sweeping multiple objects in the scene that require composition (though are not sensitive to the *order* of the composition). or are there other tasks?

Set (C) – Semantic Generalization: Unlike set (B), these tasks require manipulation with different objects of the same type. We test this using various food items within BridgeData, instructing the policy within a container. An example of such task would be a table containing a banana, a sushi, a bowl, and various distractor objects, and instead of using specific language commands such as "put the banana and the sushi in the bowl," the instruction is "put the food items in a container".

Set (D) – Tasks with Dependency: This is the most challenging of the set of tasks: these tasks have subtasks that require previous subtasks to be completed for them to succeed. For instance, taking an object out of a drawer has this structure, as the drawer must be opened before the object can be taken out.

The complete list of tasks is noted in Appendix D.

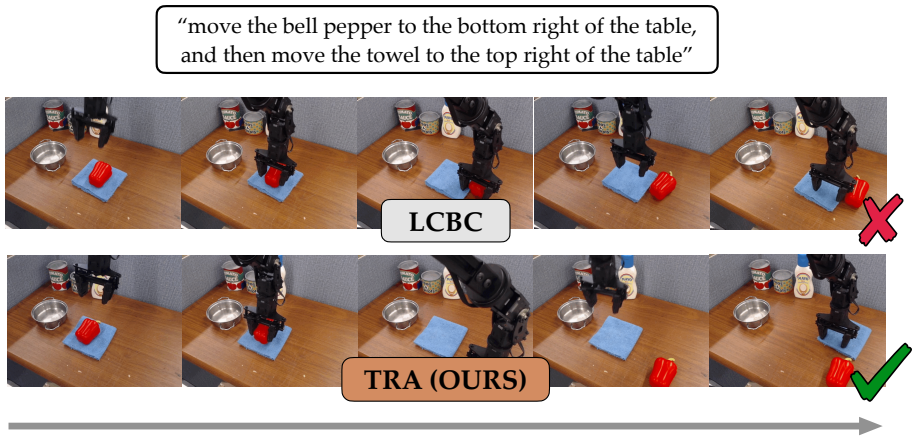


Figure 4: Example rollouts of a task with TRA and LCBC. While TRA is able to successfully compose the steps to complete the task, LCBC fails to ground the instruction correctly.

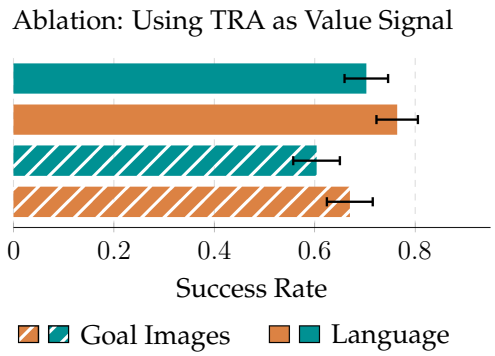


Figure 5: Aggregated success rate of using AWR as an additional policy learning metric over all 4 scenes.

4.2 Baselines

We compare against the following baselines in our real-world evaluation:

GRIF (Myers et al., 2023) learns a goal- and language- conditioned policy using aligned goal image and language representations. In our experiments, this becomes equivalent to TRA when the temporal alignment objective is removed.

GCBC (Walke et al., 2023) learns a goal-conditioned behavioral cloning policy that concatenates the goal image with the image observation.

LCBC (Walke et al., 2023) learns a language-conditioned policy that concatenates the language with the image observation.

OCTO (Ghosh et al., 2024) uses a multimodal transformer to learn a goal- and language-conditioned policy. The policy is trained on Open-X dataset (O'Neill et al., 2024), which incorporates BridgeData in its entirety.

AWR (Peng et al., 2019) uses advantages produced by a value function to effectively extract a policy from an offline dataset. In this experiment, we use the difference between the contrastive loss between the current observation and the goal representation and the contrastive loss between the next observation and the goal representation as a surrogate for value function.

We train GRIF, GCBC, LCBC, and AWR using the same augmented Bridge Dataset as TRA, and we use an Octo-Base 1.5 model for our evaluation. A more detail approach is detailed in Appendix C. During evaluation, we give all policies the same goal state and language instruction regardless of the architecture, as they are trained on the same language instruction with the exception of Octo, which doesn't benefit from paraphrased language data, but does benefit from a more diverse language annotation set across a larger dataset of varying length and complexity.

4.3 Real-world Evaluation

Our real-world evaluation aims to answer the following questions.

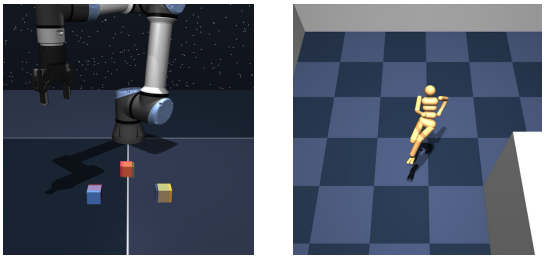


Figure 6: Two environments from the OGBench suite (Park et al., 2025). *Left:* a cube stacking environment. *Right:* a humanoid maze navigation environment.

Does TRA enable compositionality? Section 4 shows the success rates of the TRA method compared

to other methods on real-world robot evaluation tasks. We marked all policies within the task orange if they achieve the best statistically significant performance. We first compare the performance against methods in (A). Although TRA performs well with drawer tasks, its performance against baseline methods is not statistically significant. However, TRA performs considerably better than that of any baseline methods on compositionally-OOD **instruction following** tasks.

While TRA completed 88.9% of tasks seen in (B), 83.3% of evaluations in (C), and 60% of tasks in (D) with instruction following, the best-performing baseline for (B) was 30% with LCBC, 43.3% for (C) with AWR, and 33.3% on (D) with Octo. The same improvement was also present in goal reaching tasks, although at a lower level, in which (C) produced 60% success rate and scene D produced a 43.3% success rate, as compared to 46.7% and 20% for the best baselines.

Qualitatively, we see that policies trained under TRA provides a much smoother trajectory between different subtasks while following instructions, while other cannot replicate the same performance. Take removing the bell pepper + sweep task for example, with its visualization shown Fig. 4, while TRA was able to remove the bell pepper by grasping it and putting it to the bottom right corner of the table, LCBC cannot replicate the same performance, choosing to nudge the bell pepper instead and failed to execute the task.

How well does TRA perform against Conventional Offline RL Algorithms? While offline reinforcement learning promises good stitching behavior (Kumar et al., 2021), we demonstrate that TRA still outperforms offline reinforcement learning on robotic manipulation. Overall, TRA performs better than AWR for both language and image tasks, outperforming AWR by 45% on instruction following tasks, and by 25% on goal reaching tasks, showing considerable improvement over an offline RL method that promises compositional generalization via stitching.

Qualitatively, a policy trained with AWR often stops after one subtask, even though the goal instruction or image demanded all of the subtasks be completed. We see this in, e.g., Fig. 1, where 3 different policies use the same goal image for a task where all 3 food items must be put in the bowl. While TRA successfully completes all 3 subtasks, AWR chose to only complete one subtask and terminates right after putting the banana in the bowl. This is because AWR on an offline dataset has a goal-reaching reward function that disregards aligning representations across time in different trajectories.

Does TRA help capturing rarely-seen skills within the dataset? We also compare the performance of TRA against AWR across all scenes and compare the performance of the policies with all 3 tasks in (D) as well as folding the towel, all rarely seen skills within BridgeData. When conditioning on language, AWR struggles to to effectively generalize to compositionally harder tasks, with average success rate decreasing from 43.3% in to 6.67% from (C) to (D), compared to a decrease of only 83.3% to 60% for TRA. Other agents do not perform as well as AWR in (D), as the lack of such compositional generalization prevented the policies from achieving all of the tasks at a reliable rate.

Is TRA sufficient in achieving compositional generalization? We demonstrate in our real-world experiment that only using temporal alignment is sufficient for achieving good compositional generalization. We evaluate this by comparing a policy trained on only temporal alignment loss (our method), and another policy trained on such loss and have these losses weighed by AWR.

Figure 5 shows that across all evaluation tasks, there exists no statistically significant difference between using and not using AWR in addition to temporal alignment. In fact, using AWR marginally decreases the efficacy of TRA, unlike when used with GCBC and LCBC.

4.4 Testing Compositionality in Simulation

Table 2: OGBench Evaluation

Task	Methods					
	TRA	**GCBC**	CRL	GCIQL	GCIVL	QRL
antmaze medium stitch	**60.7**$^{(\pm 3.0)*}$	45.5$^{(\pm 3.9)}$	52.7$^{(\pm 2.2)}$	29.3$^{(\pm 2.2)}$	44.1$^{(\pm 2.0)}$	59.1$^{(\pm 2.4)}$
antmaze large stitch	12.8$^{(\pm 2.0)}$	3.4$^{(\pm 1.0)}$	10.8$^{(\pm 0.6)}$	7.5$^{(\pm 0.7)}$	**18.5**$^{(\pm 0.8)\dagger}$	**18.4**$^{(\pm 0.7)}$
antsoccer arena stitch	17.0$^{(\pm 1.2)}$	**24.5**$^{(\pm 2.8)}$	0.7$^{(\pm 0.1)}$	2.1$^{(\pm 0.1)}$	21.4$^{(\pm 1.1)}$	0.8$^{(\pm 0.2)}$
humanoidmaze medium stitch	**46.1**$^{(\pm 1.9)}$	29.0$^{(\pm 1.7)}$	36.2$^{(\pm 0.9)}$	12.1$^{(\pm 1.1)}$	12.3$^{(\pm 0.6)}$	18.0$^{(\pm 0.7)}$
humanoidmaze large stitch	**8.6**$^{(\pm 1.4)}$	5.6$^{(\pm 1.0)}$	4.0$^{(\pm 0.2)}$	0.5$^{(\pm 0.1)}$	1.2$^{(\pm 0.2)}$	3.5$^{(\pm 0.5)}$
antmaze large navigate	35.4$^{(\pm 1.8)}$	24.0$^{(\pm 0.6)}$	**82.8**$^{(\pm 1.4)}$	34.2$^{(\pm 1.3)}$	15.7$^{(\pm 1.9)}$	74.6$^{(\pm 2.3)}$
cube single noisy	9.2$^{(\pm 0.9)}$	8.4$^{(\pm 1.0)}$	38.3$^{(\pm 0.6)}$	**99.3**$^{(\pm 0.2)}$	70.6$^{(\pm 3.3)}$	25.5$^{(\pm 2.1)}$

RL methods with a separate value network to update the actor are in gray.
*The best non-RL methods up to significance are highlighted.
†We **bold** the best performance across all methods.

We also validated the compositional behavior of TRA in simulation using the recent offline RL benchmark OGBench (Park et al., 2025). This environment features environments for locomotion and manipulation, each with multiple offline datasets that can be used for training, including one that explicitly tests compositional generalization (the "stitch" datasets) by creating

multiple short datasets that comprise a single, larger task. We modify our approach to TRA to account for the lack of language instructions, and more implementation detail can be seen at Appendix G.

We evaluate the performance of TRA on 7 different environments in OGBench. 5 of those environments use the stitch dataset, and 2 other environment use a more general goal-reaching dataset ("navigate" and "noisy"). Table 2 shows the performance of TRA compared to other non-hierarchical methods on these environments from OGBench. Consistent with our real-world results Section 4 and Fig. 5, TRA outperforms other imitation and offline RL methods on certain environments that require compositional generalizations, including CRL (Eysenbach et al., 2022) that also has a separate value and critic network. In non-stitching environments, while traditional offline RL methods outperform TRA, we observe that TRA still improved compared to GCBC, although less than on the stitch environments.

4.5 Failure Cases

Qualitatively, we observe that despite showing better compositional generalization, the policy still fails at a similar rate compared to other multivariate Gaussian policies when multimodal behavior is observed, and other cases of early grasping and incorrect reaching are also observed at a similar rate. While TRA did seem to provide small improvements on the in-distribution tasks of (A), the primary benefits derived from TRA were seen on compositionally-OOD tasks. We further discuss failure cases in Appendix E.1.

5 Conclusions and Limitations

In this paper, we studied a temporal alignment objective for the representations used in (goal- and language-conditioned) behavior cloning. This additional structure provides robust compositional generalization capabilities in both real-world robotics tasks and simulated RL benchmarks. Perhaps surprisingly, these results suggest that generalization properties usually attributed to reinforcement learning methods may be attainable with supervised learning with well-structured, temporally-consistent representations.

Limitations and Future Work While TRA consistently outperformed behavior cloning in real world and simulation evaluations, the degree of improvement degrades when behavior cloning cannot solve the task at all. Future work could examine how to improve compositional generalization in such cases through additional structural constraints on the representation space. To scale to more complex settings, similar approaches with more complex architectures such as transformers and diffusion policies may be needed for policy and/or representation learning. TRA could also be combined with hierarchical task decomposition using VLMs, or with other forms of planning.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., et al. Do as I Can, Not as I Say: Grounding Language in Robotic Affordances. *Conference on Robot Learning*, 2022.

Akyürek, E., Akyürek, A.F., and Andreas, J. Learning to Recombine and Resample Data for Compositional Generalization. *International Conference on Learning Representations*, 2021.

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.A., and Hjelm, R.D. Unsupervised State Representation Learning in Atari. *Neural Information Processing Systems*, 2019.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight Experience Replay. *Neural Information Processing Systems*, volume 30, 2017.

Attarian, M., Gupta, A., Zhou, Z., Yu, W., Gilitschenski, I., and Garg, A. See, Plan, Predict: Language-Guided Cognitive Planning With Video Prediction. arXiv:2210.03825, 2022.

Barreto, A., Dabney, W., Munos, R., Hunt, J.J., Schaul, T., van Hasselt, H.P., and Silver, D. Successor Features for Transfer in Reinforcement Learning. *Neural Information Processing Systems*, volume 30, 2017.

Belkhale, S., Ding, T., Xiao, T., Sermanet, P., Vuong, Q., Tompson, J., Chebotar, Y., Dwibedi, D., and Sadigh, D. RT-H: Action Hierarchies Using Language. arXiv:2403.01823, 2024.

Blier, L., Tallec, C., and Ollivier, Y. Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint. arXiv:2101.07123, 2021.

Bobu, A., Liu, Y., Shah, R., Brown, D.S., and Dragan, A.D. SIRL: Similarity-Based Implicit Representation Learning. *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 565–574, 2023.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *Conference on Robot Learning*, 2023.

Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., et al. Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions. *Conference on Robot Learning*, 2023.

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. 2021.

Choi, J., Sharma, A., Lee, H., Levine, S., and Gu, S.S. Variational Empowerment as Representation Learning for Goal-Conditioned Reinforcement Learning. *International Conference on Machine Learning*, pp. 1953–1963, 2021.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., et al. PaLM: Scaling Language Modeling With Pathways. *J. Mach. Learn. Res*, 2023.

Ciranka, S., Linde-Domingo, J., Padezhki, I., Wicharz, C., Wu, C.M., and Spitzer, B. Asymmetric Reinforcement Learning Facilitates Human Inference of Transitive Relations. *Nature Human Behaviour*, 6(4):555–564, 2022.

Cui, Y., Karamcheti, S., Palleti, R., Shivakumar, N., Liang, P., and Sadigh, D. No, to the Right: Online Language Corrections for Robotic Manipulation via Shared Autonomy. *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 93–101, 2023.

Cui, Y., Niekum, S., Gupta, A., Kumar, V., and Rajeswaran, A. Can Foundation Models Perform Zero-Shot Task Specification for Robot Manipulation? *L4DC*, 2022.

Dayan, P. Improving Generalisation for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 1993a.

Dayan, P. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, volume 5, pp. 613–624, 1993b.

Dehaene, S., Al Roumi, F., Lakretz, Y., Planton, S., and Sablé-Meyer, M. Symbols and Mental Programs: A Hypothesis About Human Singularity. *Trends in Cognitive Sciences*, 26(9):751–766, 2022.

Dickins, D.W. Transitive Inference in Stimulus Equivalence and Serial Learning. *European Journal of Behavior Analysis*, 12(2):523–555, 2011.

Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. Goal-Conditioned Imitation Learning. *Neural Information Processing Systems*, 32, 2019.

Dosovitskiy, A. and Koltun, V. Learning to Act by Predicting the Future. *International Conference on Learning Representations*, 2017.

Eysenbach, B., Myers, V., Salakhutdinov, R., and Levine, S. Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference. arXiv:2403.04082, 2024.

Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R.R. Contrastive Learning as Goal-Conditioned Reinforcement Learning. *Neural Information Processing Systems*, 35:35603–35620, 2022.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.A., Zhu, Y., and Anandkumar, A. MineDojo: Building Open-Ended Embodied Agents With Internet-Scale Knowledge. *Neural Information Processing Systems*, 2022.

Fang, K., Yin, P., Nair, A., and Levine, S. Planning to Practice: Efficient Online Fine-Tuning by Composing Goals in Latent Space. *International Conference on Intelligent Robots and Systems*, 2022a.

Fang, K., Yin, P., Nair, A., Walke, H., Yan, G., and Levine, S. Generalization With Lossy Affordances: Leveraging Broad Offline Data for Learning Visuomotor Tasks. *Conference on Robot Learning*, 2022b.

Fang, K., Zhu, Y., Garg, A., Savarese, S., and Fei-Fei, L. Dynamics Learning With Cascaded Variational Inference for Multi-Step Manipulation. *Conference on Robot Learning*, 2019.

Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., et al. Octo: An Open-Source Generalist Robot Policy. *Robotics: Science and Systems*, 2024.

Ghugare, R., Geist, M., Berseth, G., and Eysenbach, B. Closing the Gap Between TD Learning and Supervised Learning - a Generalisation Point of View. *International Conference on Learning Representations*, 2023.

Gopnik, A., O'Grady, S., Lucas, C.G., Griffiths, T.L., Wente, A., Bridgers, S., Aboody, R., Fung, H., and Dahl, R.E. Changes in Cognitive Flexibility and Hypothesis Search Across Human Life History From Childhood to Adolescence to Adulthood. *National Academy of Sciences*, 114(30):7892–7899, 2017.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning Latent Dynamics for Planning From Pixels. arXiv:1811.04551, 2019.

Ito, T., Klinger, T., Schultz, D., Murray, J., Cole, M., and Rigotti, M. Compositional Generalization Through Abstract Representations in Human and Artificial Neural Networks. *Neural Information Processing Systems*, 35:32225–32239, 2022.

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. BC-Z: Zero-Shot Task Generalization With Robotic Imitation Learning. *Conference on Robot Learning*, p. 12, 2021.

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. VIMA: General Robot Manipulation With Multimodal Prompts. *International Conference on Machine Learning*, 2023.

Kaelbling, L.P. Learning to Achieve Goals. *International Joint Conference on Artificial Intelligence*, 1993.

Karamcheti, S., Nair, S., Chen, A.S., Kollar, T., Finn, C., Sadigh, D., and Liang, P. Language-Driven Representation Learning for Robotics. *Robotics - Science and Systems*, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline Reinforcement Learning With Implicit Q-Learning. *International Conference on Learning Representations*, 2022.

Kulkarni, T.D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *Neural Information Processing Systems*, volume 29, 2016.

Kumar, A., Hong, J., Singh, A., and Levine, S. Should I Run Offline Reinforcement Learning or Behavioral Cloning? *International Conference on Learning Representations*, 2021.

Kumar, A., Singh, A., Ebert, F., Yang, Y., Finn, C., and Levine, S. Pre-Training for Robots: Offline RL Enables Learning New Tasks From a Handful of Trials. arXiv:2210.05178, 2022.

Kwon, M., Hu, H., Myers, V., Karamcheti, S., Dragan, A., and Sadigh, D. Toward Grounded Commonsense Reasoning. *International Conference on Robotics and Automation*, 2023.

Laidlaw, C., Zhu, B., Russell, S., and Dragan, A. The Effective Horizon Explains Deep RL Performance in Stochastic Environments. *International Conference on Learning Representations*, 2024.

Lake, B.M., Linzen, T., and Baroni, M. Human Few-Shot Learning of Compositional Instructions. *CogSci*, 2019.

Lashley, K.S. The Problem of Serial Order in Behavior. *Cerebral Mechanisms in Behavior*, pp. 112–136. 1951.

Lewis, M., Nayak, N.V., Yu, P., Yu, Q., Merullo, J., Bach, S.H., and Pavlick, E. Does CLIP Bind Concepts? Probing Compositionality in Large Image Models. *Conference of the European Chapter of the Association for Computational Linguistics*, 2024.

Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., and Gao, J. Grounded Language-Image Pre-Training. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10955–10965, 2022.

Liu, B., Feng, Y., Liu, Q., and Stone, P. Metric Residual Network for Sample Efficient Goal-Conditioned Reinforcement Learning. *AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806, 2023.

Lynch, C. and Sermanet, P. Language Conditioned Imitation Learning Over Unstructured Data. *Robotics: Science and Systems XVII*, 2021.

Lynch, C., Wahid, A., Tompson, J., Ding, T., Betker, J., Baruch, R., Armstrong, T., and Florence, P. Interactive Language: Talking to Robots in Real Time. *IEEE Robotics and Automation Letters*, (arXiv:2210.06407):1–8, 2023.

Ma, Y.J., Liang, W., Som, V., Kumar, V., Zhang, A., Bastani, O., and Jayaraman, D. LIV: Language-Image Representations and Rewards for Robotic Control. *International Conference on Machine Learning*, 2023a.

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. *International Conference on Learning Representations*, 2023b.

Mandlekar, A., Xu, D., Martín-Martín, R., Savarese, S., and Fei-Fei, L. Learning to Generalize Across Long-Horizon Tasks From Human Demonstrations. arXiv:2003.06085, 2021.

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. *Conference on Robot Learning*, pp. 3894–3908, 2023.

Myers, V., Ji, C., and Eysenbach, B. Horizon Generalization in Reinforcement Learning. *International Conference on Learning Representations*, 2025.

Myers, V., Zheng, B.C., Mees, O., Levine, S., and Fang, K. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. *Conference on Robot Learning*, 2024a.

Myers, V., Zheng, C., Dragan, A., Levine, S., and Eysenbach, B. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. *International Conference on Machine Learning*, arXiv:2406.17098, 2024b.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3M: A Universal Visual Representation for Robot Manipulation. *Conference on Robot Learning*, pp. 892–909, 2022.

Nasiriany, S., Pong, V.H., Lin, S., and Levine, S. Planning With Goal-Conditioned Policies. arXiv:1911.08453, 2019.

Neumann, G. and Peters, J. Fitted Q-Iteration by Advantage Weighted Regression. *Neural Information Processing Systems*, volume 21, 2008.

O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *International Conference on Robotics and Automation*, 2024.

Pari, J., Shafiullah, N.M.M., Arunachalam, S.P., and Pinto, L. The Surprising Effectiveness of Representation Learning for Visual Imitation. *Robotics: Science and Systems XVIII*, 2022.

Park, S., Frans, K., Eysenbach, B., and Levine, S. OGBench: Benchmarking Offline Goal-Conditioned RL. *International Conference on Learning Representations*, 2025.

Park, S., Ghosh, D., Eysenbach, B., and Levine, S. HIQL: Offline Goal-Conditioned RL With Latent States as Actions. *Neural Information Processing Systems*, 2023.

Peng, X.B., Kumar, A., Zhang, G., and Levine, S. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. arXiv:1910.00177, 2019.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. *International Conference on Machine Learning*, arXiv:2103.00020, 2021.

Rubino, V., Hamidi, M., Dayan, P., and Wu, C.M. Compositionality Under Time Pressure. *Cognitive Science Society*, volume 45, 2023.

Schrittwieser, J., Hubert, T., Mandhane, A., Barekatain, M., Antonoglou, I., and Silver, D. Online and Offline Reinforcement Learning by Planning With a Learned Model. *Neural Information Processing Systems*, volume 34, pp. 27580–27591, 2021.

Shah, R. and Kumar, V. RRL: Resnet as Representation for Reinforcement Learning. *International Conference on Machine Learning*, 2021.

Shridhar, M., Manuelli, L., and Fox, D. CLIPort: What and Where Pathways for Robotic Manipulation. *Conference on Robot Learning*, 2021.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. ProgPrompt: Generating Situated Robot Task Plans Using Large Language Models. *International Conference on Robotics and Automation*, 2023.

Steedman, M. Where Does Compositionality Come From? *AAAI Technical Report*, 2004.

Vikbladh, O.M., Meager, M.R., King, J., Blackmon, K., Devinsky, O., Shohamy, D., Burgess, N., and Daw, N.D. Hippocampal Contributions to Model-Based Planning and Spatial Memory. *Neuron*, 102(3):683–693, 2019.

Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., et al. BridgeData V2: A Dataset for Conference on Robot Learning at Scale. *Conference on Robot Learning*, pp. 1723–1736, 2023.

Wang, T., Torralba, A., Isola, P., and Zhang, A. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. *International Conference on Machine Learning*, pp. 36411–36430, 2023.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning Invariant Representations for Reinforcement Learning Without Reconstruction. *International Conference on Learning Representations*, 2021.

Zhang, T., Eysenbach, B., Salakhutdinov, R., Levine, S., and Gonzalez, J.E. C-Planning: An Automatic Curriculum for Learning Goal-Reaching Tasks. *International Conference on Learning Representations*, 2022.

Zhang, Z., Li, Y., Bastani, O., Gupta, A., Jayaraman, D., Ma, Y.J., and Weihs, L. Universal Visual Decomposer: Long-Horizon Manipulation Made Easy. arXiv:2310.08581, 2023.

A Code and Website

An implementation of TRA is available at https://anonymous.4open.science/r/ogcrl-43A4/. A website with additional visualizations and videos is available at https://sites.google.com/view/tra-website-submission.

B TRA Implementation

In this section, we provide details on the implementation of temporal representation alignment (TRA) and its training process.

B.1 Dataset Curation

We use an augmented version of BridgeData. We augment the dataset by rephrasing the language annotations, as described by (Myers et al., 2023), with 5 additional rephrased language instruction for each language instruction present in the dataset, and randomly sample them during training.

During data loading process, for each observation that is sampled with timestep k, we also sample $k^+ \triangleq \min(k + x, H), x \sim \text{Geom}(1 - \gamma)$, and load s_k along with s_{k^+}. We employ random cropping, resizing, and hue changes during training process image robustness. We set $\gamma = 0.95$ for policy training on BridgeData.

B.2 Policy Training

We use a ResNet-34 architecture for the policy network. We train our policy with one Google V4-8 TPU VM instance for 150,000 steps, which takes a total of 20 hours. We use a learning rate of 3×10^{-4}, 2000 linear warm-up steps, and a MLP head of 3 layers of 256 dimensions after encoding the observation representations as well as goal representations.

C Baseline Implementations

We summarize the implementation details of the baselines discussed in Section 4.2.

C.1 Octo

We use the Octo-base 1.5 model publicly available on HuggingFace for evaluating Octo baselines. We use inference code that is readily available for both image- and language- conditioned tasks. During inference, we use an action chunking window of 4 and an execution horizon window of 4.

C.2 Behavior Cloning

We use the same architecture for LCBC and GCBC as in Myers et al. (2023); Walke et al. (2023). During the training process we use the same hyperparameters as TRA.

C.3 Advantage Weighted Regression

In order to train an AWR agent without separately implementing a reward critic, we follow Eysenbach et al. (2022) and use a surrogate for advantage:

$$\mathcal{A}(s_t) = \mathcal{L}_{\text{NCE}}\big(f(s_t), f(g)\big) - \mathcal{L}_{\text{NCE}}\big(f(s_{t+1}), f(g)\big). \tag{18}$$

Here, f can be any of the encoders ϕ, ξ, ψ. $\mathcal{L}$ is the same InfoNCE loss defined Section 3, and g is defined as either the goal observation or the goal language instruction, depending on the modality.

And we extract the policy using advantage weighted regression (AWR) (Neumann and Peters, 2008):

$$\pi \leftarrow \arg\max_{\pi} \mathbb{E}_{s,a\sim\mathcal{D}}\Big[\log \pi(a|s,z) \exp\big(A(s,a)/\beta\big)\Big]. \tag{19}$$

During training, we set β to 1, and we use a batch size of 128, the same value as policy training for our method.

D Experiment Details

In this section, we go through our experiment details and how they are set up. During evaluation, we randomly reset the positions of each item within the table, and perform 5 to 10 trials on each task, depending on whether this task is important within each scene. We examine tasks that are seen in BridgeData, which include conventionally less challenging tasks such as object manipulation, and challenging tasks to learn within the dataset such as cloth folding and drawer opening.

D.1 List of Tasks

Table 3 describes each task within each scene, and the language annotation used when the policy is used for inference. Every task that is outside of the drawer scene are multiple step, and require compositional generalization.

Table 3: Task Instructions

Scene	Count	Task Description	Instruction
Drawer	10	open the drawer	"open the drawer"
	10	put the mushroom in the drawer	"put the mushroom in the drawer"
	10	close the drawer	"close the drawer"
Task Generalization	5	put the spoons on the plates	"move the spoons onto the plates."
	5	put the spoons on the towels	"move the spoons on the towels"
	6	fold the cloth into the center from all corners	"fold the cloth into center"
	10	sweep the towels to the right	"sweep the towels to the right of the table"
Semantic Generalization	10	put the sushi and the corn on the plate	"put the food items on the plate"
	5	put the sushi and the mushroom in the bowl	"put the food items in the bowl"
	10	put the sushi, corn, and the banana in the bowl	"put everything in the bowl"
Tasks With Dependency	10	take mushroom out of drawer	"open the drawer and then take the mushroom out of the drawer"
	10	move bell pepper and sweep towel	"move the bell pepper to the bottom right corner of the table, and then sweep the towel to the top right corner of the table"
	10	put the corn on the plate, *and then* put the sushi in the pot	"put the corn on the plate and then put the sushi in the pot"

D.2 Inference Details

During inference, we use a maximum of 200 timesteps to account for long-horizon behaviors, which remains the same for all policies. We determine a task as successful when the robot completes the task it was instructed to within the timeframe. For evaluating baselines, we use 5 trials for each of the tasks.

E Additional Visualizations

In this section, we show additional visualizations of TRA's execution on compositionally-OOD tasks. We use *folding, taking mushroom out of the drawer*, and *corn on plate, then sushi in the pot* as examples, as these tasks require a strong degree of dependency to complete at Appendix E.

E.1 Failure Cases

We break down failure cases in this section. While TRA performs well in compositional generalization, it cannot counteract against previous failures seen with behavior cloning with a Gaussian Policy.

F Analysis of Compositionality

We prove the results from Section 3.4.

F.1 Goal Conditioned Analysis

Theorem 1. *Suppose $\mathcal{D}$ is distributed according to Eq. (12) and $\mathcal{D}^*$ is distributed according to Eq. (12). When $\gamma > 1 - 1/H$ and $\alpha > 1$, for optimal features ϕ and ψ under Eq. (11), we have*

$$\textsc{Err}(\pi; \mathcal{D}^*) \leq \textsc{Err}(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbb{1}\{\alpha > 2\}. \tag{17}$$

Proof. We have from Eq. (14) for $K \sim \text{Geom}(1 - \gamma)$:

$$\textsc{Err}(\pi; \mathcal{D}^*) \triangleq \mathbb{E}_{\mathcal{D}^*}\left[\frac{1}{H'} \sum_{t=1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\right]$$

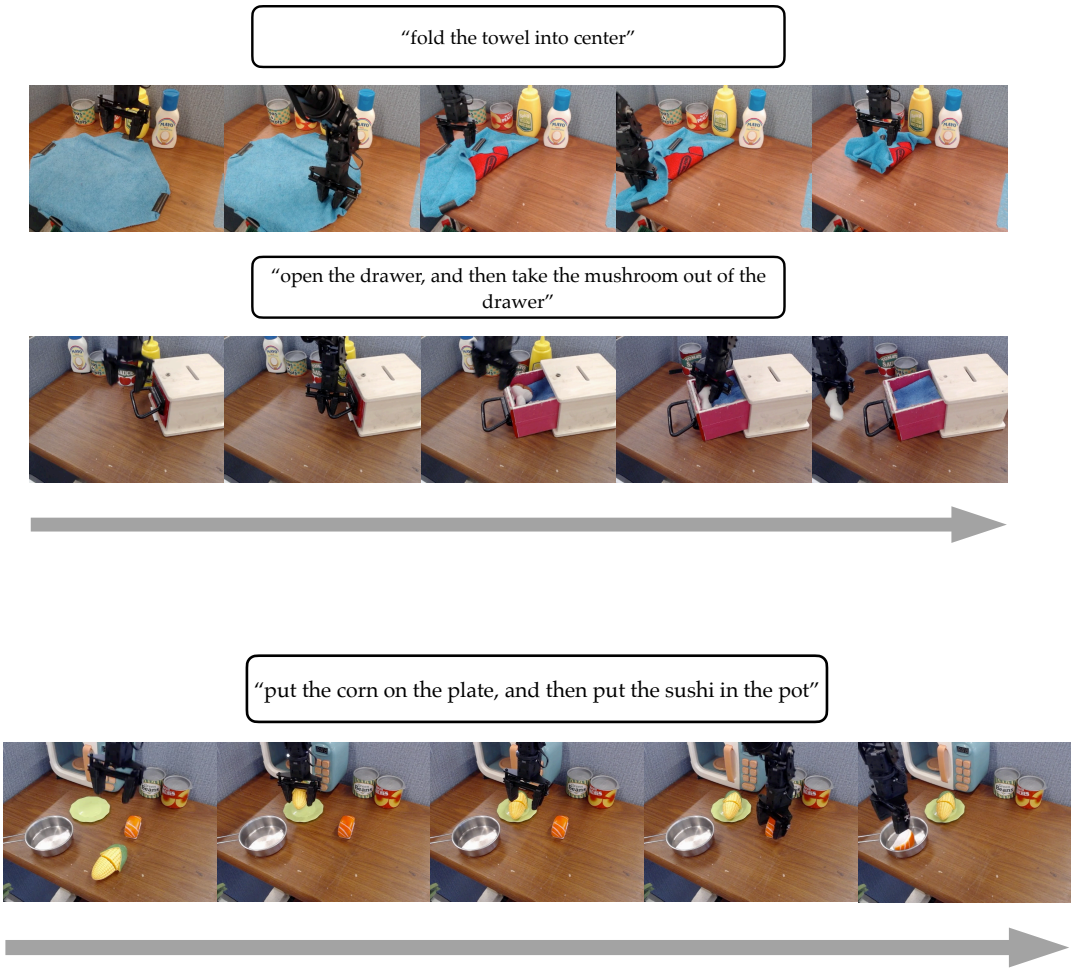


Figure 7: In these figures, we see that TRA is able to perform good compositional generatlization over a variety of tasks seen within BridgeData

$$
\begin{aligned}
&= \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=1}^{H'-2H}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\Big] + \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{H'-2H+1}^{H'-H}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\qquad + \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-H+1}^{H'}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\leq \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-H+1}^{H'}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\Big] + \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-2H+1}^{H'-H}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{g}_i)\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\qquad + \Big(\frac{\alpha-2}{2\alpha}\Big)\mathbb{1}\{\alpha>2\} \\
&\leq \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-H+1}^{H'}\frac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\qquad + \frac{1}{H'}\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-2H+1}^{H'-H}\mathbb{E}_K\Big[\frac{\|\tilde{a}_{t,i}-p^{\pi}(\tilde{s}_{t,i}|\tilde{s}_{H'-K,i})\|^2}{n_{d_{\mathcal{A}}}}\Big]\Big] + \Big(\frac{\alpha-2}{2\alpha}\Big)\mathbb{1}\{\alpha>2\}
\end{aligned}
$$

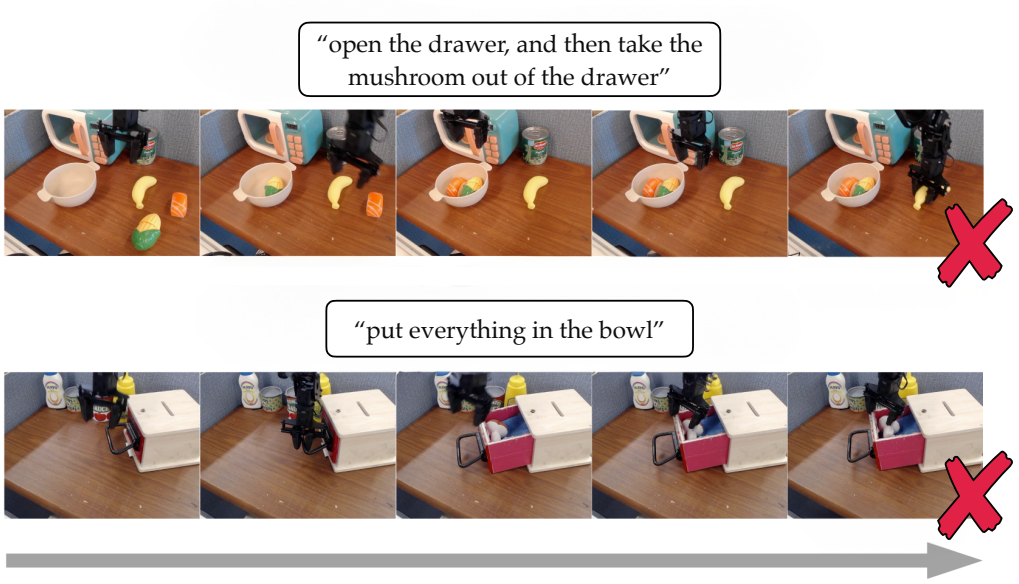


Figure 8: Most of the failure cases came from the fact that a policy cannot learn depth reasoning, causing early grasping or late release, and it has trouble reconciling with multimodal behavior

$$
\begin{aligned}
&\leq \frac{1}{H'}\,\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-H+1}^{H'} \tfrac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\qquad + \frac{1}{H'}\,\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-2H+1}^{H'-H} \mathbb{E}_K\big[\tfrac{\|\tilde{a}_{t,i}-p^{\pi}(\tilde{s}_{t,i}|\tilde{s}_{H'-K,i})\|^2}{n_{d_{\mathcal{A}}}}\big]\Big] + \Big(\frac{\alpha-2}{2\alpha}\Big)\,\mathbb{1}\{\alpha>2\} \\
&\leq \frac{1}{H'}\,\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-H+1}^{H'} \tfrac{\|\tilde{a}_{t,i}-\pi(\tilde{s}_{t,i},\tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}}\Big] \\
&\qquad + \frac{1}{H'}\,\mathbb{E}_{\mathcal{D}^*}\Big[\sum_{t=H'-2H+1}^{H'-H} \mathbb{E}_K\big[\tfrac{\|\tilde{a}_{t,i}-p^{\pi}(\tilde{s}_{t,i}|\psi(\tilde{s}_{H'-K,i}))\|^2}{n_{d_{\mathcal{A}}}}\big]\Big] + \Big(\frac{\alpha-2}{2\alpha}\Big)\,\mathbb{1}\{\alpha>2\} \\
&\leq \textsc{Err}(\pi;\mathcal{D}^*) + \frac{1}{H'}\,\mathbb{E}_{\mathcal{D}^*}\Big[\frac{1-\gamma^H}{1-\gamma}\Big] + \Big(\frac{\alpha-2}{2\alpha}\Big)\,\mathbb{1}\{\alpha>2\} \\
&\leq \textsc{Err}(\pi;\mathcal{D}^*) + \frac{\alpha-1}{2\alpha} + \Big(\frac{\alpha-2}{2\alpha}\Big)\,\mathbb{1}\{\alpha>2\}. \qquad (20)
\end{aligned}
$$

□

F.2 Language Conditioned Analysis

Corollary 1.1. *Under the same conditions as Theorem 1,*

$$\textsc{Err}^{\ell}(\pi;\mathcal{D}^*) \leq \textsc{Err}^{\ell}(\pi;\mathcal{D}) + \frac{\alpha-1}{2\alpha} + \Big(\frac{\alpha-2}{2\alpha}\Big)\,\mathbb{1}\{\alpha>2\}.$$

The proof is similar to Appendix F.1, but over the predictions of ξ instead of ψ.

F.3 Visualizing the Bound

We compare the bound from Theorem 1 with the "worst-case" bound of $\textsc{Err}(\pi;\mathcal{D}^*) - \textsc{Err}(\pi;\mathcal{D})$ in Appendix F.3. The bound from Theorem 1 is tighter than the worst-case bound, and it shows that the compositional generalization error decreases as α increases.

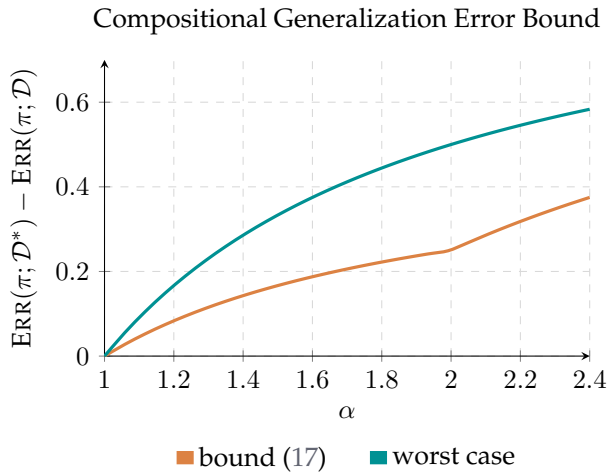


Figure 9: Visualizing the bound (Eq. 17 from Theorem 1) on the compositional generalization error.

Table 4: **Success Rate for Different GCBC Architectures in OGBench.**

Environment	**GCBC**	**GCBC-ϕ**
`antmaze medium stitch`	$\mathbf{45.5}^{\pm(\mathbf{3.9})}$	$\mathbf{48.7}^{\pm(\mathbf{2.7})}$
`antmaze large stitch`	$3.4^{\pm(1.0)}$	$\mathbf{6.8}^{\pm(\mathbf{1.3})}$
`antsoccer arena stitch`	$\mathbf{24.5}^{\pm(\mathbf{2.8})}$	$1.4^{\pm(0.3)}$
`humanoidmaze medium stitch`	$29.0^{\pm(1.7)}$	$\mathbf{34.4}^{\pm(\mathbf{1.7})}$
`humanoidmaze large stitch`	$\mathbf{5.6}^{\pm(\mathbf{1.0})}$	$3.5^{\pm(1.1)}$
`antmaze large navigate`	$\mathbf{24}^{\pm(\mathbf{0.6})}$	$16.1^{\pm(0.8)}$
`cube single noisy`	$\mathbf{8.4}^{\pm(\mathbf{1.0})}$	$\mathbf{8.7}^{\pm(\mathbf{0.9})}$

G OGBench Implementation Details

In order to implement TRA in OGBench, which does not have a corresponding language label for all goal-reaching tasks, we make the following revision to TRA to accommodate the lack of a language task. We train a policy $\pi(a|\phi(s), \psi(g))$, in which we propagate the behavior cloning loss throughout the entire network. Both the state and goal encoders are MLPs with identical architecture. We detail the configuration in 5. This is to simulate the ResNet architecture and CLIP embeddings we use from real-world policy training. We define separate state and goal encoder $\phi(s)$ and $\psi(g)$, and we modify $\mathcal{L}_{\text{TRA}}$ as:

$$\mathcal{L}_{\text{TRA}} = \mathcal{L}_{\text{BC}}(\{s_i, a_i, s_i^+\}_{i=1}^K; \pi, \phi, \psi) + \alpha \mathcal{L}_{\text{NCE}}(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi) \tag{21}$$

The rest of the implementation are carried over from OGBench. We evaluate each method with 10 seeds, and we take the final 3 evaluation epoch per seed to calculate the average success rate, the same way OGBench calculates success rate for its baselines. While we used $\alpha = 1$ in real world experiments, consistent with implementation from Myers et al. (2023), we adjust our α value in OGBench, as it is a hyperparameter. We report our optimal α configuration in Table 5.

Note that $\alpha = 0$ turns the formulation into a version of GCBC with different architecture; we denote this GCBC-ϕ. We compare the performance of GCBC and GCBC-ϕ here across the 7 environments using table 4. Although the second formulation is parameterized than the original GCBC configuration, they have similar performances across the environments that we have evaluated on — the performance of TRA does not rely on extra parameterization, but learning a structured temporal representation.

We report the value of hyperparameters in table 5. The rest of the relevant hyperparameters are implemented from OGBench unless specified in the table.

Table 5: **TRA hyperparameters.**

Hyperparameter	**Value**
State and goal encoder dimensions	(64, 64, 64)
State and goal encoder latent dimension	64
Discount factor γ	0.995 (large locomotion environments), 0.99 (other)
Alignment coefficient α	60 (medium locomotion), 100 (large locomotion), 20 (non-stitch)