Successor Representations Enable Emergent Compositional Instruction Following

Vivek Myers[*], **Bill Chunyuan Zheng**[*], **Anca Dragan**, **Kuan Fang**[†], **Sergey Levine**

University of California, Berkeley [†]Cornell University

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 tabletop robotic manipulation tasks, 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 [46]. In domains involving long-horizon decision-making like robotics, various learning approaches have been proposed to enable this property, including hierarchical learning [40], explicit subtask planning [65, 27, 1], and dynamic-programming-based "stitching" [31, 39]. In practice, these techniques are often unstable and/or data-inefficient in real-world robotics settings, making them difficult to scale [44].

By contrast, biological learners are adept at quickly composing behaviors to reach new goals [46]. Possible explanations for these capabilities have been proposed, including the ability to perform transitive inference [15], learn successor representations and causal models [19, 32], and plan with world models [70]. 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 evaluate our method, **T**emporal **R**epresentation **A**lignment (TRA), on a set of challenging multi-step manipulation tasks in the BridgeData setup [71]. These tasks specifically test 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 scenes.

[*]Equal contribution.

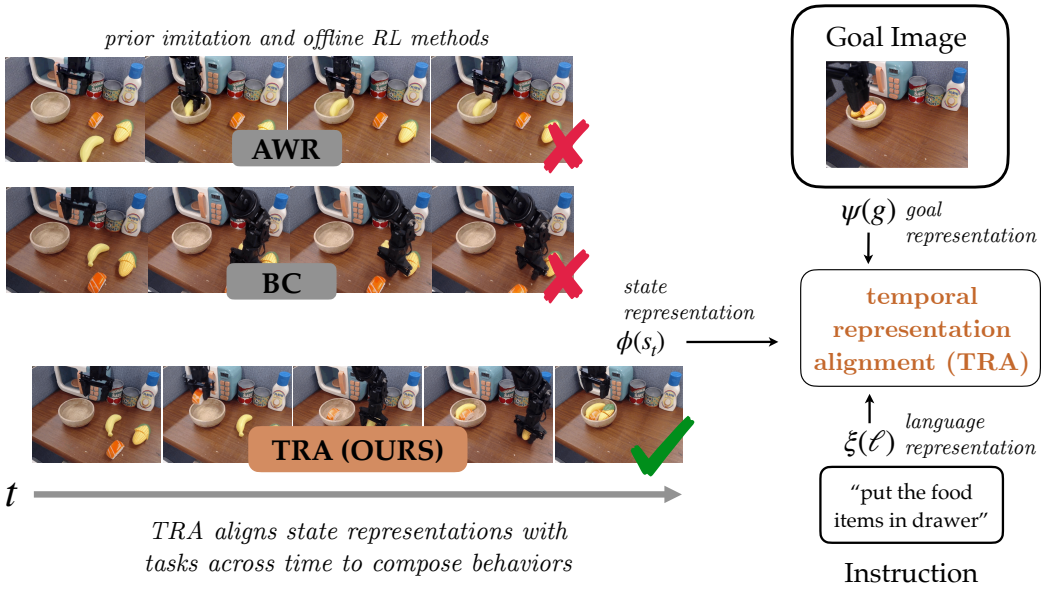


Figure 1: Example rollouts of a task with TRA and GCBC to put all food items in the bowl. While TRA can implicitly decompose the task into steps and execute them one by one, GCBC is unable to do that and fails to ground to any relevant objects. GCBC+AWR on the other hand only grounds one object, failing to display any compositionality

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 [1, 71, 67]. These policies can be trained with goal- and language-conditioned imitation learning from human demonstrations [14, 36, 50, 51, 10], reinforcement learning [11, 12], or other forms of supervision [9, 16]. When being trained to reach goals, methods can additionally use hindsight relabeling [4, 37] to improve performance [71, 55, 20, 22]. 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 [64, 69]. Biological learning systems show strong compositional generalization abilities [15, 20, 21, 45], and recent work has explored how similar capabilities can be achieved in artificial systems [2, 34, 47]. In the context of policy learning, exploiting the compositionality of the behaviors can lead to generalization to unseen and temporarily extended tasks [31, 42, 29, 28, 54, 59]. Hierarchical and planning-based approaches also aim to enable compositional behavior by explicitly partitioning a task into its components [26, 56, 74, 62]. With improvements 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 [1, 5, 7, 43, 56, 68, 75]. 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 [38, 48, 52, 55, 58, 61, 66, 17, 35]. In reinforcement learning problems, representations are often trained to predict future states, rewards, goals, or actions [3, 53, 73, 25], and can improve generalization and sample efficiency when used as value functions [6, 8, 18, 23, 13]. Some recent works have explored the use of additional structural constraints on representations to enable planning [26, 74, 24, 33], or enforced metric properties to improve compositional generalization [49, 57, 72].

Our work is distinct in that we show that temporal representation alignment can enable compositional generalization in a real-world manipulation setting without being used for policy extraction or defining a value function.

3 Temporal Representation Alignment

Given training on 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. The key insight is that temporal alignment

Notation. We take the setting of a goal- and language-conditioned MDP $\mathcal{M}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution ρ, dynamics $\mathrm{P}(s' \mid s, a)$, discount factor γ, and language task distribution $\mathcal{W}$. 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:

$$\begin{aligned} p^\pi_1(s_1 \mid s_0, a_0) &\triangleq p(s_1 \mid s_0, a_0), \\ p^\pi_{k+1}(s_{k+1} \mid s_0, a_0) &\triangleq \int_{\mathcal{A}} \int_{\S} p(s_{k+1} \mid s, a) \, \mathrm{d}p^\pi_k(s \mid s_0, a_0) \, \mathrm{d}\pi(a \mid s) \\ p^\pi_{k+t}(s_{k+t} \mid s_t, a_t) &\triangleq p^\pi(s_k \mid s_0, a_0). \end{aligned} \tag{1}$$

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:

$$p^\pi_\gamma(s^+ \mid s, a) \triangleq \sum_{k=0}^{\infty} \gamma^k p^\pi_k(s^+ \mid s, a). \tag{2}$$

3.1 Temporal Alignment

We propose temporal representation alignment (TRA) as an auxiliary objective to structure the representation space of goals and language instructions to better enable compositional generalization.

$$\mathcal{L}_{\texttt{NCE}}(\{x_i, y_i\}_{i=1}^K; f, h) = \sum_{i=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{j=1}^K e^{f(y_j)^T h(x_i)}}\right) + \sum_{i=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{j=1}^K e^{f(y_i)^T h(x_j)}}\right) \tag{3}$$

$$\mathcal{L}_{\texttt{BC}}\big(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi\big) = \sum_{i=1}^K \log \pi\big(a_i \mid s_i, \xi(\ell_i)\big) + \log \pi\big(a_i \mid s_i, \psi(s_i^+)\big) \tag{4}$$

$$\begin{aligned} &\mathcal{L}_{\mathrm{TRA}}\big(\{s_i, a_i, s_i^+, g_i, \ell_i\}_{i=1}^K; \pi, \phi, \psi, \xi\big) \\ &\quad = \underbrace{\mathcal{L}_{\texttt{BC}}\big(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi\big)}_{\text{behavioral cloning}} + \underbrace{\mathcal{L}_{\texttt{NCE}}\big(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi\big)}_{\text{temporal alignment}} + \underbrace{\mathcal{L}_{\texttt{NCE}}\big(\{g_i, \ell_i\}_{i=1}^K; \psi, \xi\big)}_{\text{task alignment}} \end{aligned} \tag{5}$$

Our overall objective is to minimize Eq. (5) across states, actions, future states, goals, and language tasks within the training data:

$$\min_{\pi,\phi,\psi,\xi} \mathbb{E}_{\substack{(s_{1,i},a_{1,i},\ldots,s_{H,i},a_{H,i},\ell)\sim\mathcal{D} \\ i\sim\text{Unif}(1\ldots H) \\ k\sim\text{Geom}(1-\gamma)}} \left[\mathcal{L}_{\text{TRA}}\left(\{s_{t,i}, a_{t,i}, s_{\min(t+k,H),i}, s_{H,i}, \ell\}_{i=1}^{K}; \pi, \phi, \psi, \xi\right)\right]. \tag{6}$$

Algorithm 1: Temporal Representation Alignment (TRA)

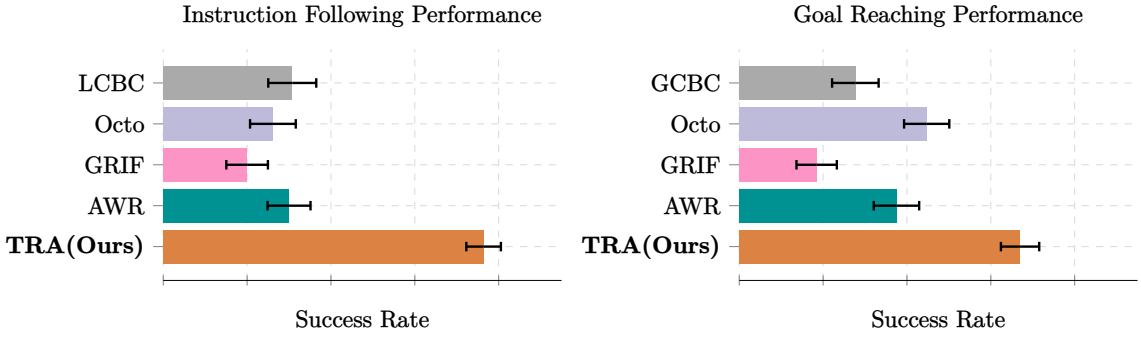
input: dataset $\mathcal{D} = (\{s_{t,i}, a_{t,i}\}_{t=1}^{H}, \ell_i)_{i=1}^{N}$
initialize networks $\Theta \triangleq (\pi, \phi, \psi, \xi)$
while training **do**
  sample a batch of transitions $\left\{(s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i)\right\}_{i=1}^{K} \sim \mathcal{D}$ for $k \sim \text{Geom}(1-\gamma)$
  $\Theta \leftarrow (\pi, \phi, \psi, \xi) - \alpha\nabla_{\Theta}\mathcal{L}_{\text{TRA}}\left(\{s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i\}_{i=1}^{K}; \Theta\right)$
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.

4 Experiments

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

1. Can TRA enable zero-shot composition of multiple sequential tasks without additional prompting or planning methods?
2. How well does TRA perform compared to conventional offline RL algorithms in terms of task generalization and composition?
3. How well does TRA capture skills that are seen at a lower percentage within the dataset, compared to the numerous entries of object manipulation?
4. Is time 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 Experimental Details

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 a towel, and then sweep the towel",

Table 1: Compositional Generalization Error of Methods

Modality	**TRA**	GRIF	LCBC	GCBC	Octo
image	**4.25 ± 0.37**	5.24 ± 0.34	—	4.84 ± 0.11	5.15 ± 0.41
language	**3.82 ± 0.25**	4.95 ± 0.32	4.84 ± 0.11	—	4.56 ± 0.32

both the tasks "remove the bell pepper from the towel" and "sweep the towel" have similar entries within BridgeData, but such combined trajectory and language description does not exist. 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 [71], which contains over 50k trajectories with 72k language annotations. We augment the dataset by rephrasing the language annotations, as described by [55], with 5 additional rephrased language instruction for each language instruction present in the dataset, and randomly sample them during training.

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:

Scene A – One-Step Drawer: this is the only scene that are not compositionally-OOD, as all the tasks are one-step tasks. This scene involves opening, putting an item in, and closing a drawer. These tasks have been seen in BridgeData, although at a lower frequency than object manipulation, but the position in which they are initialized are unseen. They will be used to compare TRA's ability to baselines when solving single-step tasks.

Scene B – Task Concatenation: this 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).

Scene C – Semantic Generalization: Unlike scene B, these tasks require manipulation with different objects of the same class. We test this using various food items seen within BridgeData and instruct the policy to put various food items within a container. An example of such task would be to have 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", a more general statement such as"put the food items in a container" will be used.

Scene D – Tasks with Dependency: This is the most challenging of the set of tasks: these tasks have subtasks that require previous subtasks being completed for them to succeed. An example of this would be to open a drawer, and to take out an item in the drawer, as one cannot take out an item from the drawer if the drawer is not open.

The complete list of tasks is noted in Appendix C.

4.2 Baselines

We compare against the following baselines:

GRIF [55] 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 [71] learns a goal-conditioned behavioral cloning policy that concatenates the goal image with the image observation.

LCBC [71] learns a language-conditioned policy that concatenates the language with the image observation.

OCTO [30] uses a multimodal transformer to learn a goal- and language-conditioned policy. The policy is trained on Open-X dataset [60], which incorporates BridgeData in its entirety.

Table 2: Real-world Language Conditioned Evaluation

	Task	**TRA**	GRIF	LCBC	Octo	AWR
	open the drawer	**0.80±0.1**†	0.20±0.2	**0.60±0.2**	**0.60±0.2**	0.40±0.2
	mushroom in drawer	**0.80±0.1**	**0.80±0.2**	0.40±0.2	0.00±0.0	**0.60±0.2**
	close drawer	**0.60±0.2**	**0.60±0.2**	**0.40±0.2**	**0.60±0.2**	**0.40±0.2**
(*)	put the spoons on towels	**1.00±0.0**	0.40±0.2	0.20±0.2	0.00±0.0	0.20±0.2
(*)	put the spoons on the plates	**0.80±0.2**	0.20±0.2	0.20±0.2	0.20±0.2	0.00±0.0
(*)	fold cloth into the center	**1.00±0.0**	0.20±0.2	0.40±0.2	0.40±0.2	0.40±0.2
(*)	sweep to the right	**0.80±0.1**	0.20±0.2	0.40±0.2	0.40±0.2	0.00±0.0
(*)	put the corn and sushi on plate	**0.90±0.1**	0.00±0.0	0.40±0.2	0.00±0.0	0.50±0.2
(*)	sushi and mushroom in bowl	**0.80±0.2**	0.00±0.0	**0.60±0.2**	0.20±0.2	**0.60±0.2**
(*)	corn, banana, and sushi in bowl	**0.80±0.1**	0.00±0.0	0.00±0.0	0.00±0.0	0.20±0.1
(*)	take the item out of the drawer	**0.60±0.2**	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*)	move bell pepper and sweep towel	**0.50±0.2**	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*)	corn in plate then sushi in pot	**0.70±0.1**	0.00±0.0	0.40±0.2	**0.60±0.2**	0.20±0.2

*indicates task is compositionally-OOD (has multiple steps never seen together in training)
†The best-performing method(s) up to statistical significance are **highlighted**

AWR [63] 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 B. 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 Experimental Evaluation

Does TRA enable compositionality? In Table 1, we compare the normalized mean squared error (MSE) of the TRA method with other methods on held-out compositionally-OOD image- and goal-specified tasks. These values are derived from passing the inputs through the policy network and sampling the mode of the distribution without unnormalizing the outputs based on the dataset. The validation MSE for these tasks are lower with a statistically significant margin, demonstrating that in a compositionally-OOD setting, TRA provides a trajectory closer to expert demonstrations.

Section 4.2 and Section 4.2 show 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 Scene A. We observe that while TRA performs well with drawer tasks, its performance against baseline methods are not statistically significant. However, when being evaluated on compositionally-OOD **instruction following** tasks, TRA performs considerably better than that of any baseline methods.

Table 3: Real-world Goal-Conditioned Evaluation

	Task	**TRA**	GRIF	GCBC	Octo	AWR
	open the drawer	**0.60±0.2**†	0.60±0.2	**0.40±0.2**	**0.50±0.2**	**0.80±0.2**
	mushroom in drawer	**0.90±0.1**	0.40±0.2	**0.80±0.2**	**0.90±0.1**	**0.60±0.2**
	close drawer	**1.00±0.0**	0.40±0.2	0.80±0.2	0.60±0.2	0.40±0.2
(*)	put the spoons on towels	**1.00±0.0**	0.20±0.2	0.60±0.2	0.40±0.2	0.60±0.2
(*)	put the spoons on the plates	**1.00±0.0**	0.00±0.0	0.40±0.2	0.00±0.0	0.80±0.2
(*)	fold cloth into the center	**1.00±0.0**	0.00±0.0	0.00±0.0	0.60±0.2	0.00±0.0
(*)	sweep to the right	**0.70±0.1**	0.40±0.2	0.00±0.0	**0.80±0.2**	0.00±0.0
(*)	put the corn and sushi on plate	**0.70±0.1**	0.00±0.0	0.20±0.2	0.00±0.0	0.30±0.1
(*)	sushi and mushroom in bowl	**0.60±0.2**	0.00±0.0	0.20±0.2	**0.40±0.2**	**0.60±0.2**
(*)	corn, banana, and sushi in bowl	**0.50±0.2**	0.00±0.0	0.00±0.0	**0.40±0.2**	0.50±0.2
(*)	take the item out of the drawer	**0.40±0.2**	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*)	move bell pepper and sweep towel	**0.60±0.2**	0.20±0.2	0.20±0.2	**0.40±0.2**	0.00±0.0
(*)	corn in plate then sushi in pot	**0.30±0.1**	**0.20±0.2**	0.00±0.0	0.00±0.0	0.00±0.0

*indicates task is compositionally-OOD (has multiple steps never seen together in training)
†The best-performing method(s) up to statistical significance are **highlighted**

While TRA completed 88.9% of tasks seen in Scene B, 83.3% of evaluations in Scene C, and 60% of tasks in Scene D with instruction following, the best-performing baseline for Scene B was 30% with LCBC, 43.3% for Scene C with AWR, and 33.3% on Scene D with Octo. The same improvement was also present in goal reaching tasks, although at a lower level, in which Scene C produced 60% success rate and scene D produced a 43.3% success rate, as compared to 46.7% and 20% for the best-performing 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. 3, 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.

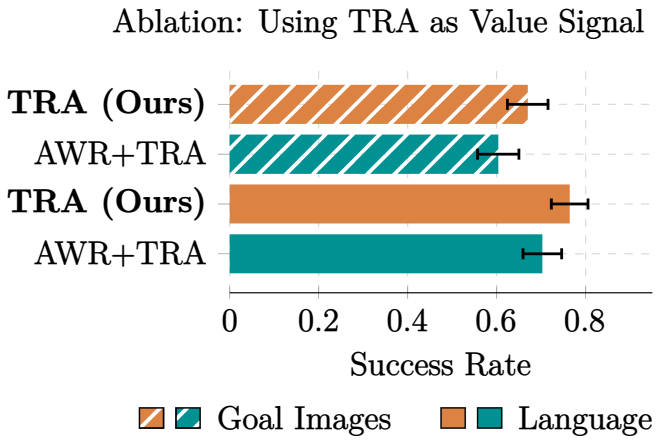


Figure 4: Aggregated success rate of using AWR as an additional policy learning metric over all 4 scenes.

How well does TRA perform against Conventional Offline RL Algorithms? While offline reinforcement learning promises good stitching behavior [41], we demonstrate that TRA still outperforms offline reinforce-

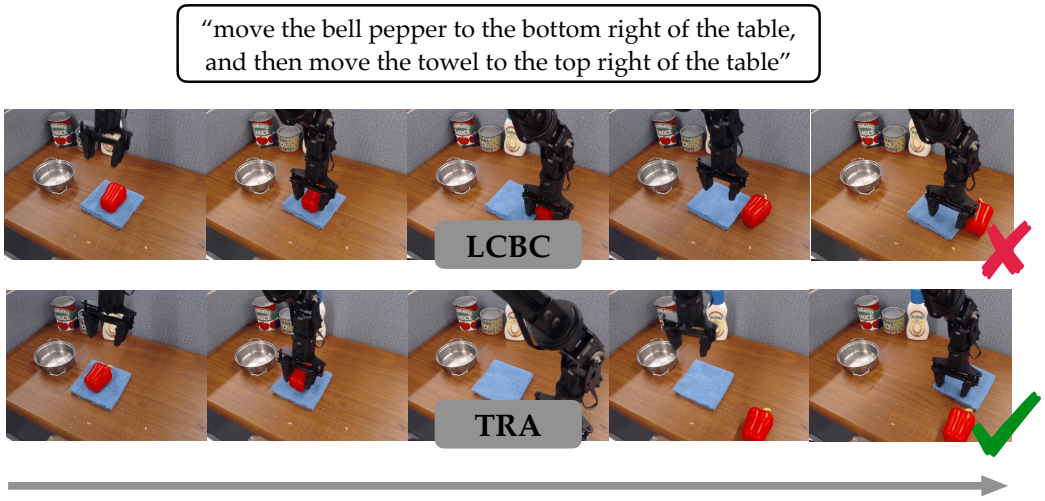


Figure 3: Example rollouts of a task with TRA and LCBC. While TRA is able to successfully compose the steps to complete the task, LCBC fails to ground the instruction correctly.

ment 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, it is often seen that a policy trained with AWR would stop after one subtask, even though the goal instruction or image demanded all of the subtasks be completed. We can see this behavior in Fig. 1, in which we have the same goal image being fed in to 3 different policies in which 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 due to the fact that AWR on an offline dataset has a goal-reaching reward function, in which it does not attempt to align the representations of all trajectories across time unlike TRA.

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 Scene D as well as folding the towel, all rarely seen skills within BridgeData, as it mainly focused on object manipulation. When compared by task within language conditioned set, we discover AWR suffered a significant drop off in effectiveness, with its average success rate plummeting from 43.3% in Scene C compared to 6.67% in Scene D, while TRA had a smaller drop off, from 83.3% to 60%, displaying that TRA generates better understanding of tasks that are rarely seen in the dataset. Other agents do not nearly achieve the same performance even as AWR in Scene 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? Finally, 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, using the same principle described Section 4.2, in which we calculate the difference between contrastive alignment losses between the current observation with goal and next observation with goal.

Fig. 4 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, as compared to showing marginal improvement over vanilla GCBC methods and a similar performance with vanilla LCBC methods. While TRA qualitatively improve the smoothness

of the execution trajectories, the same cannot be said about using AWR, in which after executing every subtask, the robot chose to return near the starting joint angles before executing the next subtask.

4.4 Failure Cases

While TRA provides an effective mechanism for compositional generalization, it is not immune to failures. 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, other cases of early grasping and incorrect reaching are also observed at a similar rate. While TRA did provide marginal improvements as seen in Scene A, it does not provide full coverage of such scenarios. More analysis of failure cases can be seen in Appendix E.

5 Conclusions and Limitations

In this paper, we studied the effects of adding a temporal representation alignment objective in behavior cloning, and we have discovered that by adding this metric, it allows a robot policy to perform robust compositional generalization even when the composition of such tasks are OOD.

Although TRA demonstrates strong performance, there are few limitations remain. First, due to restrictions placed by dataloaders, TRA cannot handle extremely long sequence of language, even though the difficulty of subtasks contained within the instructions still remain easy. It also needs to be shown that such method will be helpful for executing long-horizon tasks with bimanual manipulators or enable cross-embodiment generalization. An interesting future development for this method would look into these directions and also create such compositional generalization across multiple embodiments.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*. 2022.

[2] Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. Learning to Recombine and Resample Data for Compositional Generalization. In *International Conference on Learning Representations*. 2021.

[3] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised State Representation Learning in Atari. In *Conference on Neural Information Processing Systems*. 2019.

[4] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*, volume 30. 2017.

[5] Maria Attarian, Advaya Gupta, Ziyi Zhou, Wei Yu, Igor Gilitschenski, and Animesh Garg. See, Plan, Predict: Language-guided Cognitive Planning with Video Prediction. 2022. arXiv:2210.03825.

[6] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 30. 2017.

[7] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: Action Hierarchies Using Language. 2024. arXiv:2403.01823.

[8] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint. 2021. arXiv:2101.07123.

[9] Andreea Bobu, Yi Liu, Rohin Shah, Daniel S. Brown, and Anca D. Dragan. SIRL: Similarity-based Implicit Representation Learning. In *2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 565–574. 2023.

[10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning*. 2023.

[11] Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, et al. Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions. In *7th Annual Conference on Robot Learning*. 2023.

[12] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement Learning via Sequence Modeling. 2021.

[13] Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational Empowerment as Representation Learning for Goal-Conditioned Reinforcement Learning. In *International Conference on Machine Learning*, pp. 1953–1963. 2021.

[14] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, et al. PaLM: Scaling Language Modeling with Pathways. In *J. Mach. Learn. Res.* 2023.

[15] Simon Ciranka, Juan Linde-Domingo, Ivan Padezhki, Clara Wicharz, Charley M. Wu, and Bernhard Spitzer. Asymmetric Reinforcement Learning Facilitates Human Inference of Transitive Relations. *Nature Human Behaviour*, 6(4):555–564, 2022.

[16] Yuchen Cui, Siddharth Karamcheti, Raj Palleti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. No, to the Right: Online Language Corrections for Robotic Manipulation via Shared Autonomy. In *2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 93–101. 2023.

[17] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can Foundation Models Perform Zero-Shot Task Specification For Robot Manipulation? In *L4DC*. 2022.

[18] Peter Dayan. Improving Generalisation for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 1993.

[19] Peter Dayan. Improving Generalization for Temporal Difference Learning: The Successor Representation. In *Neural Computation*, volume 5, pp. 613–624. 1993.

[20] Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and Mental Programs: A Hypothesis about Human Singularity. *Trends in Cognitive Sciences*, 26(9):751–766, 2022.

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

[22] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-Conditioned Imitation Learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Alexey Dosovitskiy and Vladlen Koltun. Learning to Act by Predicting the Future. In *International Conference on Learning Representations*. 2017.

[24] Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference. 2024. arXiv:2403.04082.

[25] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *Conference on Neural Information Processing Systems*. 2022.

[26] Kuan Fang, Patrick Yin, Ashvin Nair, and Sergey Levine. Planning to Practice: Efficient Online Fine-Tuning by Composing Goals in Latent Space. In *International Conference on Intelligent Robots and Systems*. 2022.

[27] Kuan Fang, Patrick Yin, Ashvin Nair, Homer Walke, Gengchen Yan, and Sergey Levine. Generalization with Lossy Affordances: Leveraging Broad Offline Data for Learning Visuomotor Tasks. In *Conference on Robot Learning*. 2022.

[28] Kuan Fang, Patrick Yin, Ashvin Nair Homer Walke, Gengchen Yan, and Sergey Levine. Generalization with Lossy Affordances: Leveraging Broad Offline Data for Learning Visuomotor Tasks. *Conference on Robot Learning (CoRL)*, 2022.

[29] Kuan Fang, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Dynamics Learning with Cascaded Variational Inference for Multi-Step Manipulation. *Conference on Robot Learning (CoRL)*, 2019.

[30] Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, et al. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems*. 2024.

[31] Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the Gap between TD Learning and Supervised Learning - A Generalisation Point of View. In *The Twelfth International Conference on Learning Representations*. 2023.

[32] Alison Gopnik, Shaun O'Grady, Christopher G. Lucas, Thomas L. Griffiths, Adrienne Wente, Sophie Bridgers, Rosie Aboody, Hoki Fung, and Ronald E. Dahl. 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.

[33] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. 2019. arXiv:1811.04551.

[34] Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional Generalization through Abstract Representations in Human and Artificial Neural Networks. *Advances in Neural Information Processing Systems*, 35:32225–32239, 2022.

[35] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning. *Conference on Robot Learning*, p. 12, 2021.

[36] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General Robot Manipulation with Multimodal Prompts. 2023. arXiv:2210.03094.

[37] Leslie Pack Kaelbling. Learning to Achieve Goals. In *International Joint Conference On Artificial Intelligence*. 1993.

[38] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-Driven Representation Learning for Robotics. In *Robotics - Science and Systems*. 2023.

[39] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*. 2022.

[40] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In *Advances in Neural Information Processing Systems*, volume 29. 2016.

[41] Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should i Run Offline Reinforcement Learning or Behavioral Cloning? In *International Conference on Learning Representations*. 2021.

[42] Aviral Kumar, Anikait Singh, Frederik Ebert, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-Training for Robots: Offline RL Enables Learning New Tasks from a Handful of Trials. 2022. arXiv:2210.05178.

[43] Minae Kwon, Hengyuan Hu, Vivek Myers, Siddharth Karamcheti, A. Dragan, and Dorsa Sadigh. Toward Grounded Commonsense Reasoning. In *International Conference on Robotics and Automation (ICRA)*. 2023.

[44] Cassidy Laidlaw, Banghua Zhu, Stuart Russell, and Anca Dragan. The Effective Horizon Explains Deep RL Performance in Stochastic Environments. In *International Conference on Learning Representations*. 2024.

[45] Brenden M. Lake, Tal Linzen, and Marco Baroni. Human Few-Shot Learning of Compositional Instructions. In *CogSci*. 2019.

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

[47] Martha Lewis, Nihal V. Nayak, Peilin Yu, Qinan Yu, Jack Merullo, Stephen H. Bach, and Ellie Pavlick. Does CLIP Bind Concepts? Probing Compositionality in Large Image Models. In *Conference of the European Chapter of the Association for Computational Linguistics*. 2024.

[48] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. 2022.

[49] Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric Residual Network for Sample Efficient Goal-Conditioned Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806. 2023.

[50] Corey Lynch* and Pierre Sermanet*. Language Conditioned Imitation Learning over Unstructured Data. In *Robotics: Science and Systems XVII*. 2021.

[51] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive Language: Talking to Robots in Real Time. *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.

[52] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-Image Representations and Rewards for Robotic Control. In *International Conference on Machine Learning*. 2023.

[53] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *International Conference on Learning Representations*. 2023.

[54] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to Generalize Across Long-Horizon Tasks from Human Demonstrations. *ArXiv*, abs/2003.06085, 2020.

[55] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control. In *Conference on Robot Learning*, pp. 3894–3908. 2023.

[56] Vivek Myers, Bill Chunyuan Zheng, Oier Mees, Sergey Levine, and Kuan Fang. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. In *Conference on Robot Learning*. 2024.

[57] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. In *International Conference on Machine Learning*, arXiv:2406.17098. 2024.

[58] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, pp. 892–909. 2022.

[59] Soroush Nasiriany, Vitchyr H. Pong, Steven Lin, and Sergey Levine. Planning with Goal-Conditioned Policies. 2019. arXiv:1911.08453.

[60] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *International Conference on Robotics and Automation*. 2024.

[61] Jyothish Pari, Nur Muhammad (Mahi) Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems XVIII*. 2022.

[62] Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline Goal-Conditioned RL with Latent States as Actions. In *Conference on Neural Information Processing Systems*. 2023.

[63] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. 2019. arXiv:1910.00177.

[64] Valerio Rubino, Mani Hamidi, Peter Dayan, and Charley M. Wu. Compositionality under Time Pressure. In *Cognitive Science Society*, volume 45. 2023.

[65] Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and Offline Reinforcement Learning by Planning with a Learned Model. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27580–27591. 2021.

[66] Rutav Shah and Vikash Kumar. RRL: Resnet as Representation for Reinforcement Learning. In *International Conference on Machine Learning*. 2021.

[67] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and Where Pathways for Robotic Manipulation. In *Conference on Robot Learning*. 2021.

[68] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating Situated Robot Task Plans Using Large Language Models. In *International Conference on Robotics and Automation*. 2023.

[69] Mark Steedman. Where Does Compositionality Come From? In *AAAI Technical Report*. 2004.

[70] Oliver M. Vikbladh, Michael R. Meager, John King, Karen Blackmon, Orrin Devinsky, Daphna Shohamy, Neil Burgess, and Nathaniel D. Daw. Hippocampal Contributions to Model-Based Planning and Spatial Memory. *Neuron*, 102(3):683–693, 2019.

[71] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, et al. BridgeData V2: A Dataset for Robot Learning at Scale. In *Conference on Robot Learning*, pp. 1723–1736. 2023.

[72] Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. In *International Conference on Machine Learning*, pp. 36411–36430. 2023.

[73] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning Invariant Representations for Reinforcement Learning without Reconstruction. In *International Conference on Learning Representations*. 2021.

[74] Tianjun Zhang, Benjamin Eysenbach, Ruslan Salakhutdinov, Sergey Levine, and Joseph E Gonzalez. C-Planning: An Automatic Curriculum for Learning Goal-Reaching Tasks. In *International Conference on Learning Representations*. 2022.

[75] Zichen Zhang, Yunshuang Li, Osbert Bastani, Abhishek Gupta, Dinesh Jayaraman, Yecheng Jason Ma, and Luca Weihs. Universal Visual Decomposer: Long-Horizon Manipulation Made Easy. 2023. arXiv:2310.08581.

A Implementation Details

A.1 Dataset Curation

We use an augmented version of BridgeData. We augment the dataset by generating 5 additional paraphrased instruction per language instruction. During training process, we randomly sample the instructions for each trajectory to ensure an equal coverage of texts.

During data loading process, for each observation that is being sampled with timestep k, we also sample $\min(k + x, \text{trajectory length}), x \sim \text{Geom}(1 - \gamma)$, we load the new observation along with the previous data. We employ random cropping, resizing, and hue changes during training process image robustness.

A.2 Policy Training

We use a ResNet-34 architecture to model the policy $\pi(a|s, z)$. We use a multivariate Gaussian distribution to model the action of the policy. We train our policy with one Google V4-8 TPU VM instance. We train the policy for 150,000 steps, which takes a total of 20 hours to train the policy. We use a learning rate of 3e-4, 2000 linear warmup steps, and a MLP head of 3 layers of 256

dimensions after encoding the observation representations as well as goal representations. During inference, we use the argmax policy, that is, we use the mode of the distribution instead of random sampling during evaluation.

B Baseline Implementation Details

B.1 Octo

We use an Octo-base 1.5 model publicly availble 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.

B.2 Behavior Cloning

We train a Goal-Conditioned Behavior Cloning and Language-Conditioned Behavior Cloning agent using the same procedure as GRIF, with major reductions to remove the contrastive training between image and language. During the training process, only the behavior cloning loss is used for optimization, and we use the same hyperparameters as TRA during the training process.

B.3 AWR

In order to train an AWR agent without separately implementing a reward critic, we implement a surrogate for advantage using the following formula:

$$\mathcal{A}(s_t) = \mathcal{L}_{\text{NCE}}(\text{Enc}(s_t), \text{Enc}(g)) - \mathcal{L}_{\text{NCE}}(\text{Enc}(s_{t+1}), \text{Enc}(g)) \tag{7}$$

In which Enc could 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 the same algorithm described in AWR:

$$\pi \leftarrow \arg\max_{\pi} \mathbb{E}_{s,a\sim\mathcal{D}}[\log \pi(a|s,z) \exp(\frac{1}{\beta}\mathcal{A}(s))] \tag{8}$$

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

C 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 or 10 trials on each task, depending on whether this task is important within each scene.

C.1 List of Tasks

Table 4 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.

C.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.

Table 4: 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"

C.3 Validation MSE

In addition to rolling out the policy on real-world robot settings, we additionally collected 9 additional tasks that are compositionally OOD for 5 trajectories each, and we use 3 randomly selected seeds to train policies to evaluate the MSE on the validation trajectories.

D 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.

E 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.

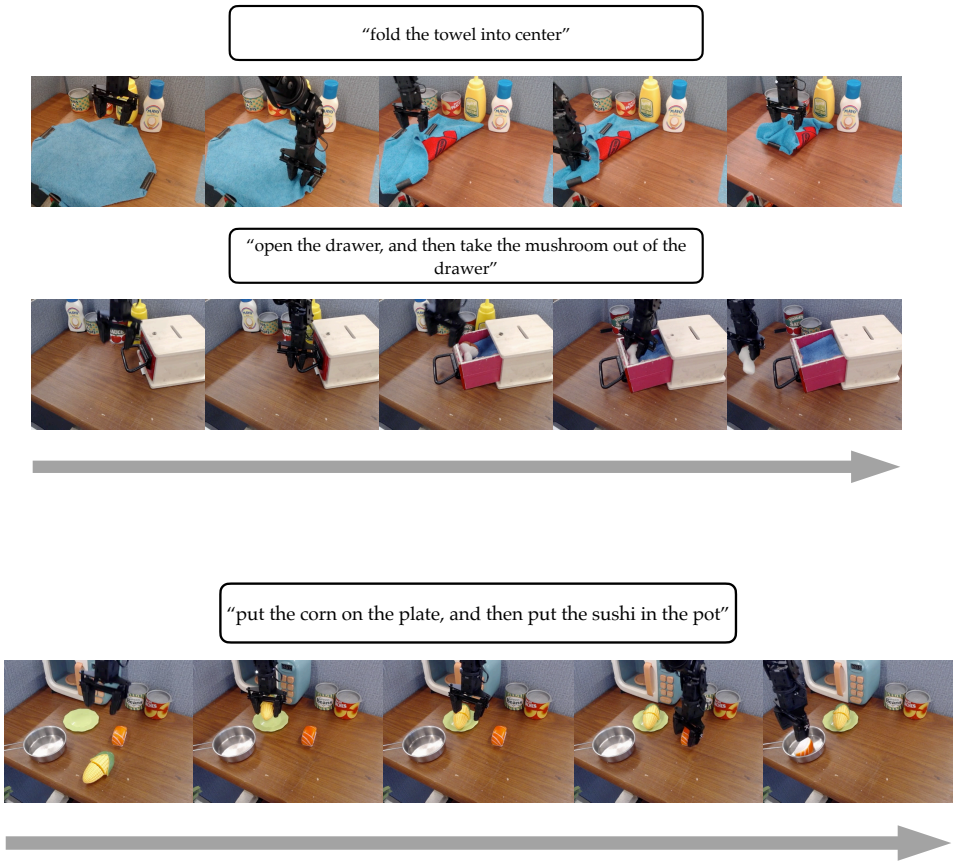


Figure 5: In these figures, we see that TRA is able to perform good compositional generatlization over a variety of tasks seen within BridgeData

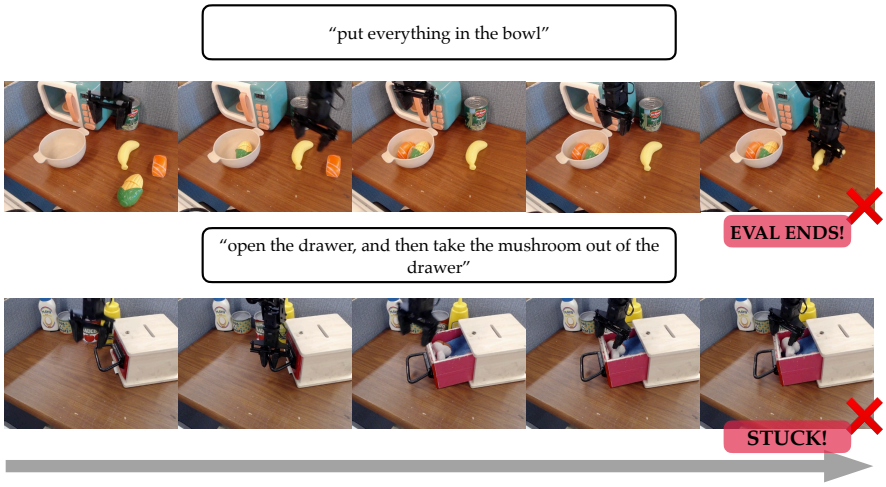


Figure 6: Most of the failure cases came from the fact that a policy cannot learn depth reasoning, causing early grasping or late release, and it has trouble reconciling with multimodal behavior