Latent Plans for Task-Agnostic Offline Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Everyday tasks of long-horizon and comprising a sequence of multiple implicit subtasks still impose a major challenge in offline robot control. While a number of prior methods have aimed to address this setting with variants of imitation and offline reinforcement learning, the learned behavior is typically narrow and often struggles to reach configurable long-horizon goals. As both paradigms have complementary strengths and weaknesses, we propose a novel hierarchical approach that combines the strengths of both methods to learn task-agnostic long-horizon policies from high-dimensional camera observations. Concretely, we combine a low-level policy that learns latent skills via imitation learning and a high-level policy learned from offline reinforcement learning for skill-chaining the latent behavior priors. We show that our formulation enables producing previously unseen combinations of skills to reach temporally extended goals by "stitching" together latent skills through goal chaining and prove the efficacy of our proposed method in various simulated and real robot control tasks showing up to an order-of-magnitude improvement in performance upon state-of-the-art baselines.

Keywords: Offline Reinforcement Learning, Imitation Learning, Robot Learning

1 Introduction

In recent years, reinforcement learning (RL) achieved tremendous successes in a variety of domains [1, 2, 3, 4]. Especially offline RL [5, 6, 7, 8, 9, 10] and its appealing property to estimate (close-to) optimal policies from previously collected and fixed datasets yield a strong current in robot control research. However, despite the exceptional progress in this fast-moving field, current offline RL methods are often evaluated on highly specific and artificial benchmarks lacking the complexity and long-term dependencies of everyday tasks which inherently entail a sequential relationship of multiple implicit subtasks. It lies in the nature of such composite tasks that this translates to estimating optimal actions for a significant amount of consecutive decision steps, making learning of such optimal policies very difficult. In fact, Kidambi et al. [11] discovered a quadratic relationship between the horizon of a task and the worst-case accumulated error of any offline RL method. This poses a major challenge especially in case of raw and unstructured sensory inputs, as robots must be capable of learning a large repertoire of skills and combine them to perform everyday tasks acting on long time scales.

One way to alleviate the problem of long horizons is the hierarchical division of a task into high- and low-level policies, where a high-level policy is chaining executions of multiple low-level policies over primitives. Most commonly, such hierarchical structures are rather rigid and act upon a *fixed* number of low-level policies, lacking in flexibility and extendability required for most real-life settings. In addition, the *a priori* definition of useful low-level *skills* or their discovery from data is a highly non-trivial task. Related prior work attempted to solve this via a goal-conditioned reformulation [12] of *Conservative Q-learning* [6]. However, it was shown that on short and distinct robot manipulation tasks, self-supervised learning on unlabeled *play* can significantly surpass the performance of individual expert-trained behavioral-cloning policies [13] – which, on the flip-side, can be on par with computationally expensive offline RL methods in such settings [14, 15]. In this work, we thus propose to leverage *play data*, i.e. non-goal-directed collections of trajectories

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

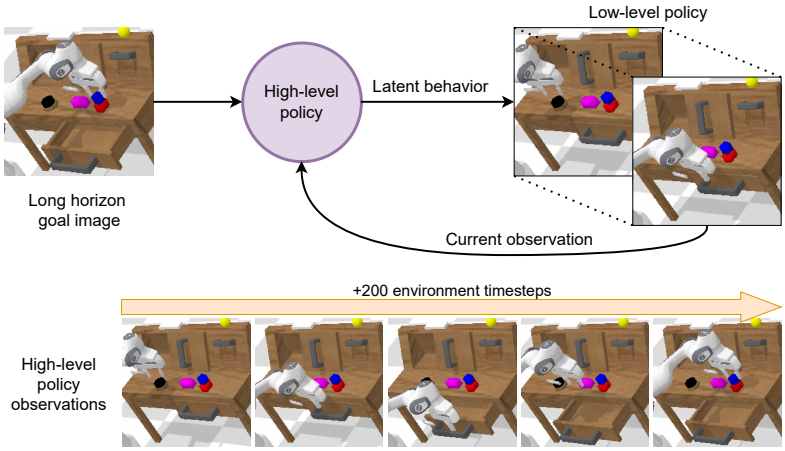

Figure 1: TACO-RL learns a single 7-DoF goal conditioned visuomotor policy from offline data. It can solve long-horizon robot manipulation tasks by using a high-level policy that divides a task into a sequence of latent behaviors that are executed by a low-level policy that interacts with the environment. It reduces the effective horizon of the high-level policy and learns to chain skills through dynamic programming.

grounded in human action execution, to estimate *short-horizon* expert policies via *imitation learning* chained via a coarse-grained high-level policy optimized by *offline RL* to account for optimal solutions over *long horizons*. By stitching latent plans extracted from unstructured demonstrations, our formulation offers the simplicity of imitation from collected play data while offering long-term optimality for sequential multi-tier tasks. Specifically, we use the collected demonstrations to learn our hierarchical policy as acquiring data with good state coverage and visual variety is important for successful applications of offline RL. Play data assumes access to an unsegmented teleoperated dataset of semantically meaningful behaviors provided by users, without a set of predefined tasks in mind. Unlike previous hierarchical approaches, that learn long-horizon tasks by performing a discrete set of tasks with a low-level policy, we are motivated by the idea of an agent capable of task-agnostic control. In this setting, we are endowing the agent with the ability to reach any possible target state from a given initial state. Specifically, we learn a low-level policy that decodes motor control actions from latent skills learned through imitation learning in a self-supervised manner. This low-level policy can perform various behaviors when present in a state by conditioning the policy with a latent plan. Consequently, we also learn a high-level policy that will order these behaviours to achieve long-horizon tasks. This high-level policy is modeled as a goal-conditioned policy that outputs latent plans to be decoded by the low-level policy. The skill-chaining of the behavior priors is learned through offline reinforcement learning by augmenting plan transitions with hindsight relabeling. This hierarchical approach constitutes a practical solution by decomposing a whole task into smaller chunks of sub-tasks. The high-level policy can learn long-horizon tasks as the effective episode horizon is reduced and it does not need to capture in detail the physics of the world which simplifies the underlying dynamics of the RL agent. The model formulation allows to design a general-purpose training objective by considering every possible state reached in the data as a potential task. Additionally our approach is effective for learning policies from large and diverse datasets which do not necessarily contain optimal behaviors.

The primary contribution of this work is an offline, self-supervised approach to solving goal-reaching tasks by combining model-free RL methods with imitation learning methods. To our knowledge, our method is the first learning system for a generalist-purpose robot that uses the benefits of hierarchical policies skill-chaining latent behaviors to reason over long-horizon tasks. We integrate our components in a unified framework, called Task-AgnostiC Offline Reinforcement Learning (TACO-RL). An overview can be seen in Figure 1. We show that our model obtains the highest success rate when tested against other state-of-the-art baselines on various long-horizon tasks of the challenging CALVIN environment [16] and that it is able to learn a single visuomotor 7-DoF policy that can perform a wide range of long-horizon manipulation tasks in both a simulated and a real-world tabletop environment.

2 Related Work

2.1 Offline Reinforcement Learning

Offline RL [5], i.e. RL from fixed and possibly mixed transition sets, constitutes a recent trend in RL and robot control research. Generally, at least in model-free settings, these techniques put a regularization on out-of-distribution actions, so as to enforce the learned policy to remain in the coverage of the dataset, since offline RL methods tend to suffer strongly from the problem of value overestimation. The simplest, yet competitive, attempt to solve this issue is a *behavioral cloning* addendum to a classical actor-critic framework [8]. *Conservative Q-learning* [6], on the other hand, imposes a penalty for actions not covered in the dataset making out-of-distribution actions non-optimal. *Fisher-BRC* parameterizes the critic as the log-behavior-policy [17] extended by a weighted offset term. *Implicit Q-learning* [10] modifies the Bellman optimality update towards a *SARSA*-like update, maximizing only over actions in the data-set. However, the rationale behind most current offline RL methods remains rather similar. While we make use of *Conservative Q-learning* in our experiments – which recently emerged as one of the most widely used benchmarks in offline RL – we want to point out that any other sophisticated improvement upon the classical offline RL objective is orthogonal to our work and could in principle be incorporated in our framework.

2.2 Hierarchical policy learning

Hierarchical policy learning involves learning a hierarchy of policies where a low-level policy performs motor control actions and a high-level policy directs the low-level policy to solve a task. While some works [18, 19, 20] learn a discrete set of lower-level policies, each behaving as a primitive skill, this is not appropriate for a general-purpose robot that accomplishes a continuum of behaviors. Our approach follows a similar rationale to *BELT* [21], however, we alleviate the necessity of a model by estimating the high-level policy via model-free offline RL as opposed to model-based planning. Other works [22, 23] address this problem by learning a continuous space of primitive skills representing the lower-level policy. Although these methods have mostly been applied in online settings, there have been some recent works that extract skills from a fixed prior dataset and use it for either performing tasks directly [13] or learning downstream tasks with online RL [24]. In TACO-RL, we use a single offline dataset to learn the hierarchical policy. In contrast to prior works that focus on a single downstream task and do not use an image-based policy [25], we learn a general-purpose visuomotor agent that has a high zero-shot generalization even for complex long-horizon tasks.

3 Mathematical Foundation

In this section, we introduce notation and define the problem setting. We model the interaction between and environment and a goal-conditioned policy as a goal-augmented Markov decision process $M = (S, A, p, r, G, p_0, \gamma)$ where S represents the state-observation space, A represents the action space, $p(s'|s, a)$ is a state-transition probability function, $r(s, a, s')$ represents the reward function, $G \subseteq S$ specifies the goal space, $p_0(s)$ is an initial state distribution, and $\gamma \in (0, 1)$ represents the discount factor. We note that the agent does not have access to the true state of the environment, but to visual observations. We learn in an offline manner by assuming to use a large, unlabeled, and undirected fixed dataset $\mathcal{D} = \{(s_1, a_1), (s_2, a_2), ..., (s_T, a_T)\}$. We then relabel this long temporal state-action stream to produce a dataset of trajectories $D = \{(\tau_i = (s_t, a_t)_{t=0}^{k}\}_{i=1}^{N}$ that can be used to learn both the low-level and high-level policy without access to a model.

4 Offline goal-conditioned RL with TACO-RL

In this section, we elaborate on TACO-RL. Our proposed method considers the bottom-up approach; we start by training a low-level policy and we use it to provide a higher-level action space for a high-level policy that, due to this task division, is ideally facing an easier learning problem. First, we describe our unsupervised objective, which learns a continuous space of latent-conditioned behaviors $\pi_\omega(a|s_c, z)$ from D. Afterward, we detail how to learn the high-level policy with offline RL by hindsight relabeling sub-trajectories with the aid of the previously learned low-level policy. This reduces our effective task horizon, making it easier to learn long-horizon tasks. Additionally, the low-level policy will predict actions close to the offline data distribution, bringing stability to the whole learning pipeline. A graphical overview of TACO-RL is shown in Figure 2.

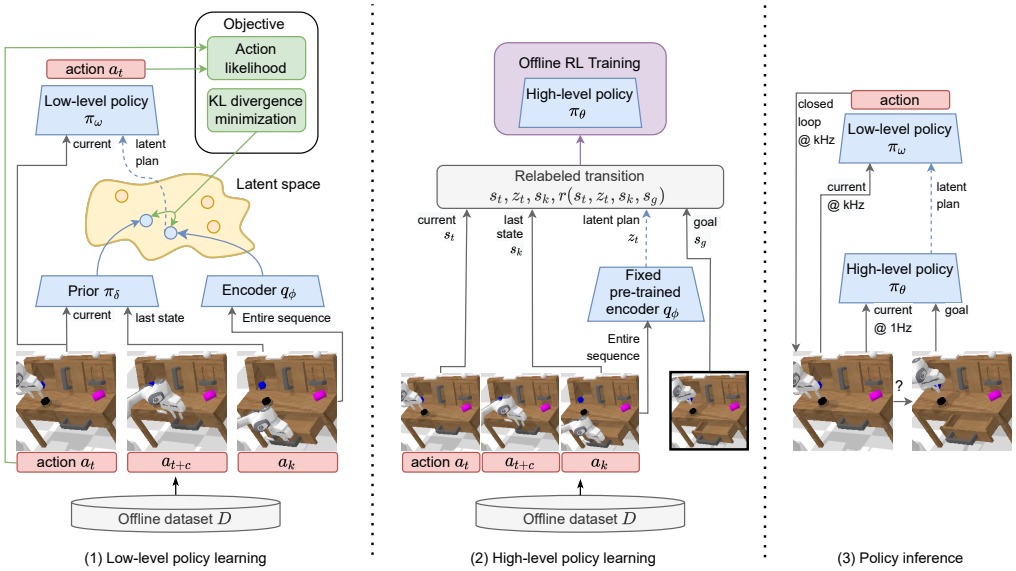

Figure 2: TACO-RL: A self-supervised single general-purpose model learned from an offline dataset of robot interactions, it generalizes to a wide variety of long-horizon manipulation tasks. (1) Low-level policy: Trained to recognize and organize a repertoire of behaviors executed throughout the interaction dataset in a latent plan space. (2) High-level policy learning: Hindsight relabeling a sampled random window of experience into a reward-augmented latent plan transition. Learned with offline RL, this allows the high-level policy to stitch plans together to achieve complex long-horizon tasks. (3) Inference: the learned hierarchical model is used to perform goal-conditioned rollouts in robot manipulation tasks.

4.1 Learning the low-level policy

We would like to extract a continuous space of primitives that propose meaningful behaviors for an agent to take within a given state. We learn a low-level policy $\pi_\omega(a|s_c, z)$ from the offline, unstructured dataset D that is able to decode a latent plan z to its respective motor-control actions a. After learning an action decoder policy, we can later use the latent plans as an action space for the high-level policy to learn reaching temporally extended goals by "stitching" together latent skills through goal chaining.

In our fixed static dataset D, it is expected to find different valid behaviors achieving the same outcome in a scene, e.g. closing a drawer quickly or slowly. We address this inherent multi-modality by auto-encoding contextual demonstrations through a latent plan space with a sequence-to-sequence conditional variational auto-encoder (seq2seq CVAE) [13, 26]. Conditioning the action decoder on the latent plan allows the policy to use the entirety of its capacity for learning uni-modal behavior. Consequently, we propose the following objective for learning the low-level policy $\pi_\omega(a|s_c, z)$:

$$\min_{\omega,\phi} \mathbb{E}_{\tau\sim D, z\sim q_\phi(z|\tau)} \left[-\sum_{t=0}^{|\tau|} \log(\pi_\omega(a_t|s_t, z)) \right], \tag{1}$$

where $\mathbb{E}$ indicates empirical expectation and $q_\phi(z|\tau)$ may be interpreted as the latent plan encoder.

As an additional component of the algorithm, we enforce consistency in the latent variables predicted by encoder $q_\phi(z|\tau)$ and prior $\pi_\delta(z|s_t, s_g)$. Since our goal is to obtain a latent plan z that captures a temporal sequence of actions for a given trajectory $\tau = (s_0, a_0, ..., s_k, a_k)$, we utilize a regularization that enforces the distribution $q_\phi(z|\tau)$ to be close to just predicting the primitive or the latent variable z given the initial and last state of this sub-trajectory, i.e. $\pi_\delta(z|s_c, s_g)$. The Evidence Lower Bound (ELBO) [27] for the CVAE can be written as:

$$\log p(x|s) \geq -\mathrm{KL}\left(q(z|x, s) \parallel p(z|s)\right) + \mathbb{E}_{q(z|x,s)}\left[\log p(x|z, s)\right]. \tag{2}$$

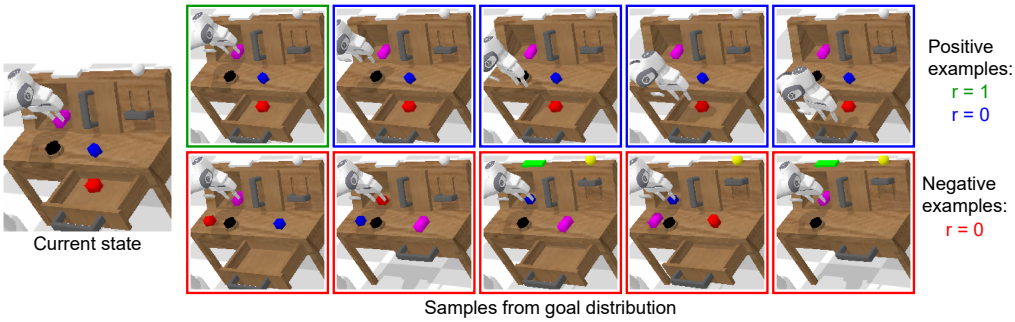

Figure 3: We relabel sampled trajectories into reward augmented transitions by sampling goal states that can be reached after executing a sequence of behaviors. In this image, we depict examples of possible goal states that could be sampled by our approach. With green border, we have the frame found at the end of the sampled trajectory. As this state will be reached after executing the latent behavior, the reward for this transition is 1. With blue border, we find future states that occur after the sampled sequence. These goals are necessary for chaining behaviours. The reward for these transitions is 0. Finally, with red border, we present images with similar proprioceptive information to the final state in the sampled trajectory, but a different scene arrangement. The reward for these transitions is 0.

The conditioning of the prior $\pi_\delta(z|s_c, s_g)$ on the initial and final state regularizes the distribution $q_\phi(z|\tau)$ to not overfit to the complete sub-trajectory τ. In practice, rather than solving the constrained optimization directly, we implement the KL-constraint as a penalty, weighted by an appropriately chosen coefficient β. Thus, one may interpret our unsupervised objective as using a sequential β-VAE [28].

4.2 Offline RL with Hindsight relabeling

After distilling learned behaviors from D in terms of an encoder $q_\phi(z|\tau)$, a latent behavior policy $\pi_\omega(a|s_c, z)$, and a prior $\pi_\delta(z|s_c, s_g)$, TACO-RL then applies these behaviors to learn a general-purpose agent with offline RL.

We formulate a goal augmented MDP by augmenting environment trajectories with a reward function. Thereby, we sample a trajectory from the dataset $(\tau = (s_t, a_t)_{t=0}^k) \sim D$. Then, we use this trajectory to represent a high-level policy transition by using the pre-trained fixed encoder. For this, we sample a latent plan from the policy encoder $z_t \sim q_\phi(z|\tau)$ and we generate an interaction transition using the sampled latent plan (s_t, z_t, s_k). To augment this transition with a reward, we use hindsight relabeling with a sparse reward as follows $r(s_t, z_t, s_k, s_g) = \mathbb{1}_{s_k = s_g}$. In this formulation, we assume that during inference we can decode the latent plan using the low-level policy and we will reach the final trajectory state s_k.

We add a discount factor of γ and consider episodes terminated once they reach the goal state s_g. Note that s_t, s_{t+1}, and s_g all represent images, and the reward is only given when the next state and goal images exactly match. During training, goals are sampled according to a distribution $s_g \in S$, which we will discuss later.

Our Q-learning approach corresponds to the following Bellman error optimization objective:

$$\min_\lambda \mathbb{E}_{\tau \sim D, z \sim q_\phi(z|\tau), s_g \sim S} \left[Q_\lambda(s_t, z_t, g) - \left(\mathbb{1}_{s_k = s_g} + \gamma \mathbb{1}_{s_k \neq s_g} \max_{z_k} Q_\lambda(s_k, z_k, s_g) \right) \right]^2 . \quad (3)$$

We can then learn a high-level policy $\pi_\theta(z|s_c, s_g)$ with an off-the-shelf offline actor-critic method. In TACO-RL, we use Conservative Q-Learning (CQL) [6] where we initialize the actor weights with the previously learned prior policy $\pi_\delta(z|s_c, s_g)$, as the prior policy is already a good starting point that is able to solve short-horizon tasks.

Selecting goals for relabeling transitions. Our high-level policy is learned via reinforcement learning as we want to learn through dynamic programming how to chain skills to reach long-horizon

goals. For this we need to create a formulation that sample goal states s_g that can be reached after executing a sequence of plans. Naively choosing s_g, say by sampling random states uniformly from the dataset, will provide an extremely sparse reward signal, as two random state images will rarely be identical. The sparse reward problem can be mitigated by selectively sampling as goals the states that were reached in future time steps along the same trajectory as s_t [29]. Given that we want to sample states that are reached after executing a plan we can assume an arbitrary window size k. More precisely, to sample goals for a transition at time step t, we sample a discrete-time offset $\Delta \sim Geom(p)$, where $p \in [0, 1]$ is a hyper-parameter, and use the state at time $t + \Delta * (k - 1)$ as the goal. Note that if we assume consistent transitions from latent plan decoding, then if $\Delta = 1$, the reward for this transition is 1. This is because after the low-level policy executes the plan it will actually reach the specified state, avoiding the sparsity issue.

However, relabeling all transitions in this way creates an issue: since the distance function would only be trained on goals that were reached, it would systematically underestimate the distance to unreachable goals. We need a procedure to select "negative" goals that are distant yet relevant. Randomly selecting dataset states will lead to pairs of images that are distant with high probability (e.g., pairs in which all objects and the robot have been moved), but not necessarily relevant. We would like a goal sampling procedure that produces less obvious examples of distant states, which are more informative for training. Similar to Tian et al. [30] we sample negative goal states s_g which have a similar proprioceptive state. This constraint allows the Q-function to learn to focus on the under-actuated aspects (e.g. the objects) of the scene, as they are likely to have distinct positions. Hence, these timesteps can serve as informative hard negatives that encourage the model to pay more attention to the scene. This sampling approach is computationally inexpensive, as we can query a precomputed k-nearest neighbors structure. It is also practical, as actuated components of the state can typically be measured through encoders on the actuator.

5 Experimental Results

We evaluate TACO-RL for learning a general-purpose robot in both simulated and real-world environments. The goals of these experiments are to investigate: (i) whether our hierarchical model is effective in performing complex long-horizon skills, (ii) how TACO-RL compares with alternative goal-conditioned policies, (iii) if our model scales to be used in real-world robotics.

5.1 Experimental Setup

We evaluate our approach in both simulated and real-world environments. We first evaluate our algorithm in the Maze2D environments from D4RL [31]. We then investigate learning 7-DoF visuomotor robot skills in the CALVIN environment [16]. We train on the environment D of CALVIN, which contains 6 hours of unstructured play data collected via teleoperating a Franka Emika Panda robot arm to manipulate objects in a 3D tabletop environment.

As we aim to combine the complementary strengths of both paradigms, imitation and offline reinforcement learning, we compare TACO-RL to representatives of the two extrema of the spectrum: the offline RL method *Conservative Q-learning* [6] extended by hindsight relabeling (CQL+HER) and the imitation learning method *Play-supervised Latent Motor Plan* (LMP) [13]. CQL+HER is trained on the derived reward as explained in Section 4.2 and introduces a penalty for out-of-distribution actions so as to limit the respective values of unseen actions. LMP, on the other hand, consists of a plan recognition module, a plan proposal module and a goal-conditioned policy to reproduce actions demonstrated via play. It hence resembles the low-level policy in TACO-RL, however, without any long-term optimality guarantees which TACO-RL accounts for by offline RL of a higher-level plan-selective policy.

5.2 Experiments in Simulation

We start by evaluating the validity of our approach in a low dimensional observation space. Specifically, we train our model in the Maze2D environments. These tasks are designed to provide a simple test that the model is capable of stitching together various subtrajectories such that the agent is capable of reaching the goal in the smallest amount of steps. For TACO-RL, we learn the policy by relabeling the transitions instead of using the rewards given by the dataset. Additionally, given that the scene does not change through time, we only add positive examples in our goal sampling

approach. We perform the rollouts of both LMP and TACO-RL by exposing the desired target position. TACO-RL outperforms both LMP and CQL as it is able to stitch plans through dynamic programming as shown in Table 1.

Environment	TACO-RL	LMP [13]	CQL [6]
maze2d-umaze	**110**$\pm$2.22	81.96$\pm$14.91	5.7
maze2d-medium	**88.9**$\pm$2	77.05$\pm$18.05	5.0
maze2d-large	**76.73**$\pm$9.78	20.13$\pm$29.34	12.5

Table 1: The average normalized score for three different random seeds. The CQL results were obtained from the D4RL whitepaper [31].

We then tested our approach in the CALVIN environment [16]. This is a more realistic environment as the scene changes through time and we act by using only RGB images of a static camera as input. As there is no predefined reward signal in this dataset, we relabel the transitions analogously as we do with TACO-RL. In the CALVIN environment, we investigate if our method is capable of performing complex long-horizon tasks in a robot control setting. We first attempt to solve 500 unique chains of 5 image-based goals queried in a row. For each subtask in a row the policy is conditioned on the current sub-goal image instruction and transitions to the next sub-goal only if the agent successfully completes the current task or if 180 timesteps have passed without reaching a success. Quantitative results are shown in Table 2.

Method	LH-VisMTLC					
	No. Instructions in a Row (500 chains)					
	1	2	3	4	5	Avg. Len.
Ours	**95.4%**$\pm$2	**82%**$\pm$7.3	**57.8%**$\pm$15	**32.7%**$\pm$10	**6.9%**$\pm$3.1	**2.7**$\pm$0.3
LMP [13]	91.4%$\pm$2.3	63.3%$\pm$3.5	23.1%$\pm$2.2	3.6%$\pm$0.9	0.2%$\pm$0.08	1.8$\pm$0.08
CQL+HER	65.5%$\pm$12.7	25.3%$\pm$11	5.6%$\pm$3.2	0.6%$\pm$0.2	0%	0.9$\pm$0.2

Table 2: The average success rate of goal-conditioned models running 500 chains per three different random seeds, using intermediate sub-goal images.

We then evaluate the capacity of our model to perform 1000 rollouts of two sequential tasks using a single goal image. For this, we allow the agent to perform actions until 300 timesteps has passed. We record the success rate of all models in Table 3. TACO-RL successfully performs long-horizon tasks that require reasoning over sequential behaviors with an average final success rate of 27% which corresponds to an order of magnitude improvement upon the LMP and CQL+HER baselines, proving its effectiveness in chaining skills through dynamic programming.

Method	LH-VisMTLC		
	No. Instructions in a Row (1000 chains)		
	1	2	Avg. Len.
Ours	**67.9%**$\pm$3.93	**27%**$\pm$2.57	**0.94**$\pm$0.06
LMP [13]	34.3%$\pm$3.20	2.7%$\pm$0.17	0.37$\pm$0.03
CQL+HER	35.2%$\pm$3.48	2.4%$\pm$0.86	0.37$\pm$0.04

Table 3: The average success rate of goal-conditioned models running 1000 chains per three different random seeds conditioned only on the last goal image.

We further test the capacity of the models to perform a single task when the goal image does not contain the robot performing the task but the end effector appears in another position after the task was performed (cf. Table 4). We run 50 rollouts for each task. Each rollout uses a different goal image which was not seen throughout training. These harder tasks require reasoning about changes in the scene, TACO-RL was able to outperform the baselines, which can be explained through the skill-chaining abilities of our model and the negative goal mining used to train the critic network of the high-level policy. On the other hand, CQL+HER is not able to achieve the desired goal image and LMP has a strong bias towards the end-effector position ignoring the changes in the environment.

Method \Task	Place block in drawer	Open drawer	Move slider left	Turn on lightbulb
Ours	**94%**±8.71	**87.3%**±5.03	**79.3%**±11.71	**94%**±4
LMP [13]	78.6%±6.11	12%±2	12%±5.29	10%±2
CQL+HER	67.6%±20.88	8.6%±5.03	17.3%±8.08	4%±4

Table 4: The average success rate of goal-conditioned models running 50 rollouts where the goal image does not contain the end effector performing the task. Three models trained from different random seeds were used to perform the rollouts.

5.3 Real-World Experiments

For the real-world experiments, we investigate to learn a single policy capable of performing 5 goal conditioned tasks: closing the drawer, opening the drawer, lifting the block, pushing the block and rotating the block. Images of the robot task setup are shown in Figure 4.

Figure 4: We evaluate the capacity of our model in different real-world tasks: closing the drawer, opening the drawer, lifting the block, pushing the block and rotating the block.

To generate the training dataset we collected 8 hours of play data collected via teleoperating a Franka Emika Panda robot arm with a VR controller. After training the models with the offline dataset we performed 10 rollouts for each task using multiple goal images and start positions. TACO-RL was able to outperform the baselines consistently, especially when evaluated from a start position far away from the goal image or that required extended reasoning. We recorded the success rate of each model in Table 5.

Method \Task	closing drawer	opening drawer	lifting the block	pushing the block	rotating the block
Ours	**90%**	**70%**	**90%**	**70%**	**70%**
LMP [13]	70%	50%	70%	70%	40
CQL+HER	20%	10%	30%	10%	10%

Table 5: The average success rate of goal-conditioned models running roll-outs in the real world.

6 Limitations

Our method enables learning diverse goal-reaching tasks even reasoning over long-horizon tasks, however it does have a number of limitations. Specifying a task to the goal-conditioned policy requires providing a suitable goal image at test-time, which should be consistent with the current scene. An exciting direction for future work is to use natural language processing techniques to command the robot policy. In addition, recent results [14] show that, in general, pure offline RL methods tend to offer better data-efficiency compared to behavioral cloning, which could be counted as one general limitation of imitation learning methods and derivatives of it, as the approach introduced in this work. However, since one of the most appealing properties of play lies in the simplicity of data collection, this limitation appears to be rather minor.

7 Conclusion

In this paper, we introduced *Task-Agnostic Offline Reinforcement Learning* (TACO-RL). It exploits a latent plan representation estimated from human teleoperated play data in order to effectively limit the horizon of a high-level offline RL policy acting upon this latent plan space. By dividing sequential multi-tier tasks into chunks of implicit subtasks solved by imitation learning, TACO-RL showed up to an order-of-magnitude improvement in performance compared to LMP and CQL+HER baselines in both, simulated and real robot control tasks. The results underline the practicality and possible future impact of combining imitation and offline RL methods in the area of robot learning.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[3] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. van de Wiele, V. Mnih, N. Heess, and J. T. Springenberg. Learning by playing solving sparse reward tasks from scratch. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4344–4353. PMLR, 10–15 Jul 2018.

[4] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine. Learning to walk via deep reinforcement learning. In *Proceedings of Robotics: Science and Systems*, 2019.

[5] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[6] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[7] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

[8] S. Fujimoto and S. Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

[9] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020.

[10] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

[11] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline reinforcement learning. In *NeurIPS*, 2020.

[12] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, B. Eysenbach, R. C. Julian, C. Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *International Conference on Machine Learning*, pages 1518–1528. PMLR, 2021.

[13] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[14] A. Kumar, J. Hong, A. Singh, and S. Levine. Should i run offline reinforcement learning or behavioral cloning? In *International Conference on Learning Representations*, 2022.

[15] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, 2021.

[16] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. CALVIN: a benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv:2112.03227*, 2021.

[17] I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 5774–5783. PMLR, 2021.

[18] A. Gupta, C. Lynch, B. Kinman, G. Peake, S. Levine, and K. Hausman. Demonstration-bootstrapped autonomous practicing via multi-task reinforcement learning. *arXiv*, 2022.

[19] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[20] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[21] B. Ichter, P. Sermanet, and C. Lynch. Broadly-exploring, local-policy trees for long-horizon task planning. In *CoRL*, volume 164 of *Proceedings of Machine Learning Research*, pages 59–69. PMLR, 2021.

[22] O. Nachum, M. Ahn, H. Ponte, S. S. Gu, and V. Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. In *Conference on Robot Learning*, pages 110–121. PMLR, 2020.

[23] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

[24] T. Shankar and A. Gupta. Learning robot skills with temporal variational inference. In *International Conference on Machine Learning*, pages 8624–8633. PMLR, 2020.

[25] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum. OPAL: offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021.

[26] L. Wang, X. Meng, Y. Xiang, and D. Fox. Hierarchical policies for cluttered-scene grasping with latent plans. *IEEE Robotics and Automation Letters*, 2022.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[29] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[30] S. Tian, S. Nair, F. Ebert, S. Dasari, B. Eysenbach, C. Finn, and S. Levine. Model-based visual planning with self-supervised functional distances. In *International Conference on Learning Representations*, 2021.

[31] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.