# An Investigation into Value-Implicit Pre-Training for Task-Agnostic, Sample-Efficient Goal-Conditioned Reinforcement Learning

**Samyeul Noh**
ETRI, KAIST
samuel@etri.re.kr

**Seonghyun Kim**
ETRI
kim-sh@etri.re.kr

**Ingook Jang**
ETRI
ingook@etri.re.kr

**Hyun Myung**
Electrical Engineering
KAIST
hmyung@kaist.ac.kr

## Abstract

One of the primary challenges of learning a diverse set of robotic manipulation skills from raw sensory observations is to learn a universal reward function that can be used for unseen tasks. To address this challenge, a recent breakthrough called value-implicit pre-training (VIP) has been proposed. VIP provides a self-supervised pre-trained visual representation that exhibits the capability to generate dense and smooth reward functions for unseen robotic tasks. In this paper, we explore the feasibility of VIP's goal-conditioned reward specification with the goal of achieving task-agnostic, sample-efficient goal-conditioned reinforcement learning (RL). Our investigation involves an evaluation of online RL by means of VIP-generated rewards instead of human-crafted reward signals on goal-image-specified robotic manipulation tasks from Meta-World under a highly limited interaction. We find the combination of the following three techniques: combining VIP-generated rewards with sparse task-completion rewards, policy pre-training using expert demonstration data via behavior cloning before RL training, and oversampling of the demonstrated data during the RL training, leads to a greater acceleration of online RL compared to utilizing VIP-generated rewards in isolation.

## 1 Introduction

A long-standing challenge within the field of robot learning is to learn diverse robotic manipulation skills from raw sensory observations (for example, image pixels). A key issue in addressing this challenge involves learning generalizable and scalable representations and reward functions that can be used for unseen tasks Radford et al. (2021); Nair et al. (2022). Recently, a breakthrough referred to as "value-implicit pre-training" (VIP) was proposed Ma et al. (2022a). The significance of the VIP study is underscored by its demonstration that the VIP model, pre-trained entirely on large-scale, in-the-wild human video data, can serve as zero-shot reward specifications capable of generating dense and smooth reward functions for unseen robotic tasks, without the need for fine-tuning on task-specific robot data.

In this paper, we investigate the feasibility of VIP's goal-conditioned reward specification with the goal of achieving task-agnostic, sample-efficient goal-conditioned reinforcement learning (RL). Our study involves an evaluation of online RL by means of VIP-generated rewards instead of human-crafted reward signals on goal-image-specified robotic manipulation tasks from Meta-World Yu et al. (2020)

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

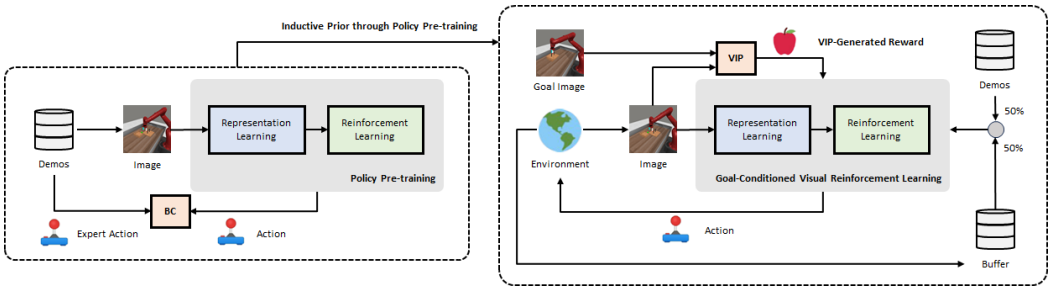



Figure 1: We incorporate four algorithms with the goal of achieving task-agnostic, sample-efficient goal-conditioned RL: (a) VIP for goal-conditioned reward specification, (b) TD-MPC for sample-efficient RL, (c) BC for pre-training a policy using expert demonstration data before RL training, and (d) oversampling of the demonstrated data during the RL training. The policy pre-training phase serves as providing an inductive prior by initializing the policy on a handful of demonstrations via BC. After initializing the policy by pre-training, oversampling of the demonstrated data results in a significant acceleration of TD-MPC. Finally, a combination of VIP-generated rewards with sparse task-completion rewards compensates for occasional inaccurate estimations of VIP's goal-conditioned reward specification.

under an extremely constrained budget of 100K environment steps. Our findings reveal that VIP's goal-conditioned reward specification allows for task-agnostic, sample-efficient goal-conditioned RL when employed in conjunction with the following three techniques: combining VIP-generated rewards with sparse task-completion rewards, policy pre-training using expert demonstration data via behavior cloning (BC) before RL training, and oversampling of the demonstrated data during the RL training, as opposed to relying solely on VIP-generated rewards. In our study, we use just 5 demonstrations for the policy pre-training.

## 2 Methodology

In this section, we describe our methodology that incorporates four key components to achieve task-agnostic, sample-efficient goal-conditioned RL: (a) VIP Ma et al. (2022a) for goal-conditioned reward specification, (b) temporal difference learning for model predictive control (TD-MPC) Hansen et al. (2022) for sample-efficient RL, (c) BC Atkeson & Schaal (1997) for policy pre-training using expert demonstration data before RL training, and (d) oversampling of the demonstrated data during the RL training, in Figure 1. We provide the differences between our methodology and conventional visual RL approaches in Appendix A, and we provide an overview of the three algorithms (VIP, TD-MPC, and BC) below.

**VIP** We adopt VIP into our methodology owing to its distinctive feature: goal-conditioned reward specification. This feature empowers us to attain task-agnostic goal-conditioned RL for any task, given only a goal image. VIP, pre-trained on a subset of the expansive Ego4D dataset Grauman et al. (2022) — a comprehensive collection of ego-centric human video data — offers a self-supervised pre-trained visual representation. This representation can serve as zero-shot reward specifications, enabling the generation of dense and smooth reward functions for robotic tasks that have not been encountered previously. More details of VIP are described in Appendix B.

**TD-MPC** We incorporate TD-MPC into our methodology due to its outstanding sample efficiency. TD-MPC, as the cutting-edge model-based RL algorithm, achieves state-of-the-art performance in sample efficiency through the integration of key components, including model predictive control (MPC), a learned latent-space world model, and a terminal value function learned via temporal difference (TD) learning. More details of TD-MPC are described in Appendix C.

**BC** We utilize BC in our methodology due to its straightforward implementation while providing remarkable performance when trained with expert demonstration data. BC, a straightforward and commonly used framework for imitation learning, allows for the pre-training of a policy to predict expert actions from corresponding observations. More details of BC are described in Appendix D.

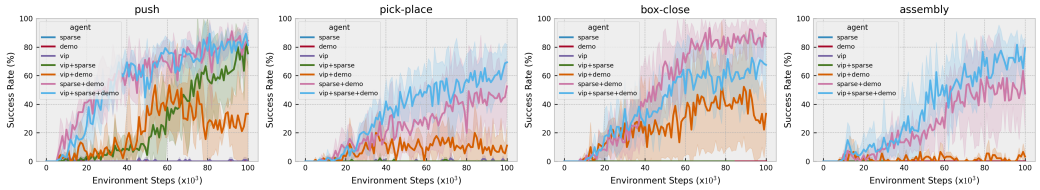



Figure 2: Main result. Success rate as a function of environment steps on four robotic manipulation tasks from Meta-World, including *push*, *pick-place*, *box-close*, and *assembly*. Mean of 3 seeds; shaded area indicates 95% CIs. Our methodology (*vip+sparse+demo*), which trains TD-MPC with the following three techniques: (a) a combination of VIP-generated rewards with sparse task-completion rewards, (b) policy pre-training using a small number of demonstrations via BC before RL training, and (c) oversampling of the demonstrated data during the RL training, demonstrates the potential of achieving task-agnostic, sample-efficient goal-conditioned RL, even under the highly limited budget of 100K environment steps.

## 3 Experiments

**Environments** We consider a subset of robotic manipulation tasks from Meta-World Yu et al. (2020). Within the context of goal-conditioned RL, we consider learning directly from RGB images, and focus on goal-image-specified tasks where each task is specified via a goal image. During training, we replace human-crafted rewards with VIP-generated rewards. The reason for this lies in the way human-crafted rewards are calculated based on coordinate states, introducing inherent complexity in the process of devising effective reward functions for a diverse set of tasks. This complexity necessitates hard coding and domain-specific knowledge. In contrast, VIP-generated rewards are computed based on embedding distances to the goal image, making it task-agnostic goal conditioned RL for any task where a goal image is specified. In this specific experimental context, we evaluate the performance of TD-MPC under an extremely constrained budget of 100K online interactions. Details of our experiments are described in Appendix E.

**Results** The experimental results are summarized in Figure 2. **(sparse)** Although TD-MPC provides state-of-the-art performance in sample efficiency within online RL algorithms, we observe that it cannot succeed the given tasks within an extremely limited budget of 100K environment steps. **(demo)** We observe that a small number of expert demonstration data cannot succeed the given task via BC. This difficulty arises primarily from issues such as covariance shift Rajeswaran et al. (2017) and the intricacies involved in visual representation learning Parisi et al. (2022); Nair et al. (2022). **(vip)** We observe that training TD-MPC using only VIP-generated rewards do not succeed the given tasks within 100K environment steps even if VIP-generated rewards are sufficiently dense and smooth. **(vip+sparse)** We observe that training TD-MPC by means of a combination of VIP-generated rewards and sparse task-completion rewards has the potential to enhance task success rates compared to relying solely on VIP-generated rewards. This enhancement is particularly noticeable in relatively simple tasks, as evident in the success rates for the *push* task in Figure 2. **(vip+demo)** We observe that VIP's goal-conditioned reward specification can accelerate online RL when used in conjunction with two techniques: policy pre-training with expert demonstration data and oversampling of the demonstrated data. **(sparse+demo)** Even with just a few demonstrations, we observe a remarkable improvement in task success rates when TD-MPC is subsequently trained by means of oversampling of the demonstrated data after pre-training a policy via BC. **(vip+sparse+demo)** We observe that a significant improvement in task success rates when TD-MPC is trained using the following three techniques: a combination of VIP-generated rewards and sparse task-completion rewards, policy pre-training via BC before RL training, and oversampling of the demonstrated data during the RL training. We visualize the final states of the trajectories generated by the policies learned by the seven different training methods: *sparse*, *demo*, *vip*, *vip+sparse*, *vip+demo*, *sparse+demo*, and *vip+sparse+demo*, in Figure 3.

**Discussion** Our attempts to train TD-MPC using VIP-generated rewards exclusively, with a goal image provided for each task, did not meet our initial expectations within a limited number of environment steps. These results can be thought of as a result of the following reason: VIP is trained on ego-centric data, while in this study, we train TD-MPC using bird-view camera data due to the

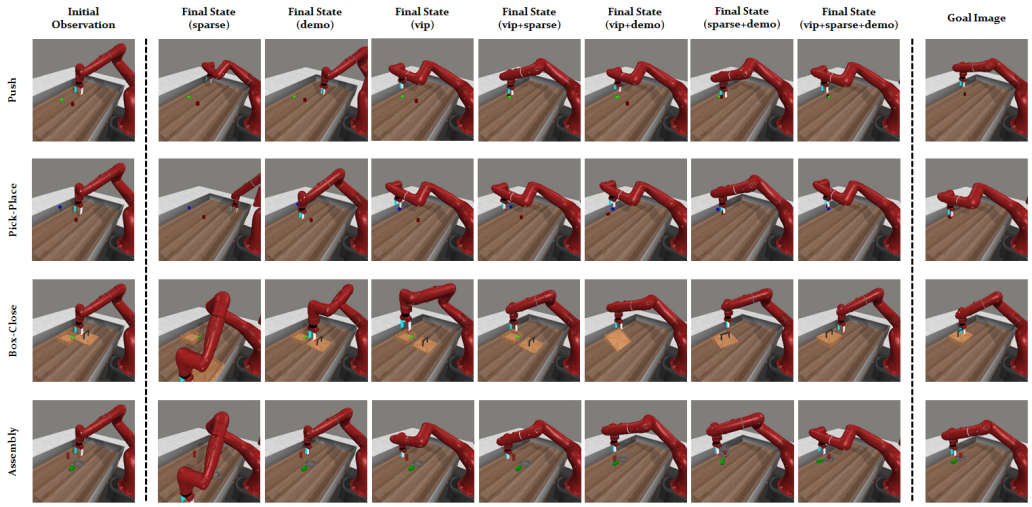



Figure 3: We visualize the final states of the trajectories generated by the policies learned by the seven different training methods: *sparse*, *demo*, *vip*, *vip+sparse*, *vip+demo*, *sparse+demo*, and *vip+sparse+demo*.

unavailability of ego-centric viewpoints in Meta-World. As a result, VIP may generate rewards based on variations in robot motion that affect a significant number of image pixels, rather than alterations related to small-sized target objects, which involve a relatively limited number of pixels. These VIP-generated rewards encourage the learning of a policy that instructs the robot to achieve the same pose depicted in the goal image without the need to manipulate the target object. In order to address these challenges, we have integrated VIP's goal-conditioned reward specification with three key techniques: a combination of VIP-generated rewards and sparse task-completion rewards, policy pre-training with expert demonstration data via BC before RL training, and oversampling of the demonstrated data during the RL training. As shown in Figure 2, task-completion rewards can compensate for inaccuracies in VIP-generated rewards, even if they occur rarely. Remarkably, in addition to a combination of VIP-generated rewards and sparse task-completion rewards, a substantial enhancement in task success rates is achieved when TD-MPC is trained using the following dual techniques: policy pre-training with expert demonstration data and oversampling of the demonstrated data. This favorable result can be attributed to the influence of policy pre-training and oversampling of the demonstrated data, which provides a strong inductive priors both before and during the RL training process.

## 4 Conclusion

In this study, we have investigated the potential of VIP's goal-conditioned reward specification in pursuit of achieving task-agnostic, sample-efficient goal-conditioned RL. Our investigation involved evaluating online RL by means of VIP-generated rewards instead of human-crafted reward signals on goal-image-specified robotic manipulation tasks from Meta-World within an extremely constrained interaction. Our results indicate that VIP's goal-conditioned reward specification can achieve task-agnostic, sample-efficient goal-conditioned RL when employed alongside three key components: a combination of VIP-generated rewards and sparse task-completion rewards, policy pre-training using expert demonstration data via BC before RL training, and oversampling of the demonstrated data during the RL training.

In future work, we will further explore the feasibility of VIP's goal-conditioned reward specification. The further investigation will involve training TD-MPC with VIP-generated rewards only based on ego-centric data in a benchmark simulator that supports ego-centric viewpoints. Furthermore, we will consider the inclusion of another advanced backbone RL algorithm besides TD-MPC such as DrQ-v2 Yarats et al. (2021), the state-of-the-art model-free RL algorithm, in the context of this investigation.

## Acknowledgments

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZR1120, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways].

## References

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pp. 12–20. Citeseer, 1997.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

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. *arXiv preprint arXiv:2210.00030*, 2022a.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How far i'll go: Offline goal-conditioned reinforcement learning via $f$-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022b.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

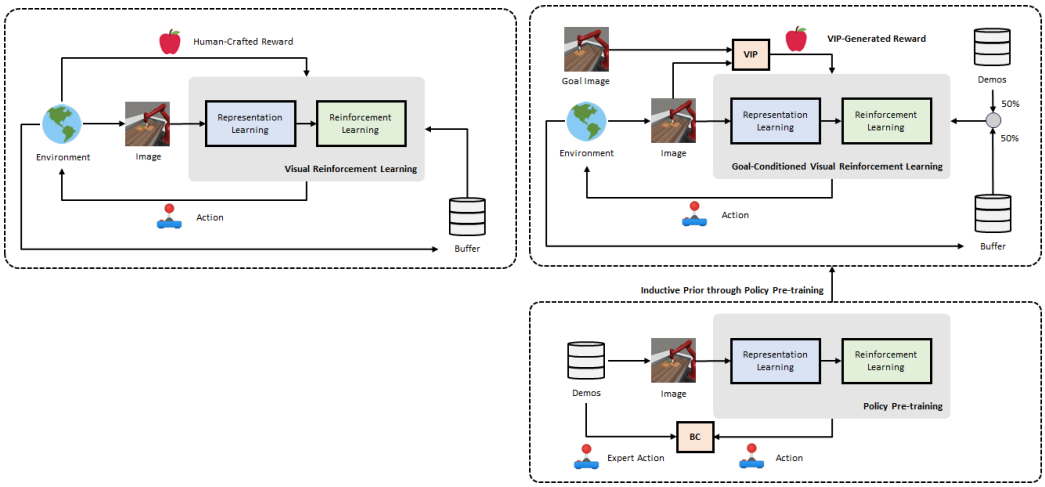
Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

## A Differences from Conventional Visual RL Approaches

In this section, we describe differences between our methodology and existing visual RL approaches. As shown in Figure 4a, conventional visual RL approaches use human-crafted rewards from the environment to learn an agent to solve a given task. On the other hand, our methodology achieves task-agnostic goal-conditioned RL by utilizing the VIP model Ma et al. (2022a), which provides dense and smooth rewards for any task given a goal image, as shown in Figure 4b. In addition, it achieves a substantial acceleration in online RL by pre-training a policy with a limited set of expert demonstration data and oversampling this demonstration data during the RL learning process. Consequently, it can achieve task-agnostic, sample-efficient goal-conditioned RL through a combination of the following four components: (a) VIP for goal-conditioned reward specification, (b) TD-MPC for sample-efficient RL, (c) BC for policy pre-training on expert demonstrations before RL training, and (d) oversampling of the demonstrated data during the RL training.




(a) Conventional visual RL approaches (b) Our methodology

Figure 4: Differences between conventional visual RL approaches and our methodology. (a) Conventional visual RL approaches rely on human-crafted rewards from the environment for online RL. (b) Our methodology integrates four key components with the aim of realizing task-agnostic, sample-efficient goal-conditioned RL: (i) VIP for goal-conditioned reward specification, (ii) TD-MPC for sample-efficient RL, (iii) BC for policy pre-training using expert demonstration data before RL training, and (iv) oversampling of the demonstrated data during the RL training.

## B Background: VIP

In this section, we provide a concise overview of VIP Ma et al. (2022a). VIP is designed to learn a visual representation capable of generating dense and smooth reward functions for unseen tasks in a self-supervised manner on passive human videos, utilizing temporal value-function optimization. In other words, VIP learns the optimal goal-conditioned value function on the basis of the dual offline goal-conditioned RL formulation Ma et al. (2022b) as follows:

$$\begin{aligned}\mathcal{L}(\phi) = \; & \mathbb{E}_{p(g)}\big[(1-\gamma)\mathbb{E}_{\mu_0(o;g)}\big[-\mathcal{S}(\phi(o);\phi(g))\big] \\ & + \log \mathbb{E}_{(o,o';g)\sim D}\big[\exp\big(\mathcal{S}(\phi(o);\phi(g)) + 1 - \gamma\mathcal{S}(\phi(o');\phi(g))\big)\big]\big] \end{aligned} \tag{1}$$

where $g$ denotes a goal image; $\mu_0(o;g)$ denotes the goal-conditioned distribution of initial image; and $D(o,o';g)$ denotes the goal-conditioned distribution of two consecutive intermediate images $(o,o')$ in dataset $D$. The goal-conditioned value function is implicitly parameterized as a similarity metric (for example, $L_2$ distance) in the embedding space. This formulation can be denoted as $V(o;g) := \mathcal{S}(\phi(o);\phi(g))$, and it serves a dual purpose as both a representation learning and a reward learning algorithm. Because VIP does not rely on actions, it is amenable to pre-training on large-scale human video datasets. The resultant implicit value function fulfills a twofold role: firstly, as a visual representation for unseen robotic tasks, and secondly, as a dense reward specification conditioned on

goals. More specifically, when provided with a goal image $g$, VIP assigns a potential-based reward at each time step $t$:

$$R(o_t, o_{t+1}; g) := S(\phi(o_{t+1}; \phi(g))) - S(\phi(o_t; \phi(g))) \tag{2}$$

In this study, we use VIP as goal-conditioned reward specification capable of generating dense and smooth reward functions for unseen robotic tasks, which making it task-agnostic RL. To clarify, we train TD-MPC by using VIP-generated rewards instead of human-crafted reward signals, in relation to goal-image-specified robotic manipulation tasks from Meta-World within the context of goal-conditioned RL. Further details of VIP are available on Ma et al. (2022a).

## C Background: TD-MPC

In this section, we provide a brief overview of TD-MPC Hansen et al. (2022), the current state-of-the-art model-based RL algorithm. TD-MPC provides the most up-to-date performance in terms of sample efficiency within online RL methods by effectively combining the following key components: model predictive control (MPC), a learned latent-space world model, and a terminal value function learned via temporal difference (TD) learning. In detail, TD-MPC learns the following five components: (a) a representation $z = h_\theta(o)$ that maps the high-dimensional state $o$ into a compact representation $z$, (b) a dynamics model in this latent space $z' = d_\theta(z, a)$, (c) an instantaneous reward $r = R_\theta(z, a)$, (d) a state-action value $Q_\theta(z, a)$, and (e) an action $a \sim \pi_\theta(z)$. Here, the policy $\pi_\theta$ plays a crucial role in guiding planning toward trajectories with high expected returns, and is optimized to maximize temporally weighted Q-values. The remaining components are jointly optimized to minimize TD-errors, reward prediction errors, and latent state prediction errors. The objective is as follows:

$$\begin{aligned}\mathcal{L}_{\text{TD-MPC}}\big(\theta; (o, a, r, o')_{t:t+H}\big) = \sum_{i=t}^{t+H} \lambda^{i-t} \big[ \, \| Q_\theta(z_i, a_i) - (r_i + \gamma Q_{\bar{\theta}}(z'_i, \pi_\theta(z'_i)))\|_2^2 \\ + \| R_\theta(z_i, a_i) - r_i \|_2^2 + \| d_\theta(z_i, a_i) - h_{\bar{\theta}}(o'_i) \|_2^2 \, \big] \end{aligned} \tag{3}$$

$$z_t = h_\theta(o_t), \quad z_{i+1} = d_\theta(z_i, a_i)$$

where $\bar{\theta}$ is an exponential moving average of $\theta$. In the course of interacting with the environment, TD-MPC employs a sample-based planning method Williams et al. (2015) in combination with the learned latent world-model $z' = d_\theta(z, a)$ and state-action value function (critic) $Q_\theta(z, a)$ for action selection.

In this study, we use TD-MPC as an online RL approach for sample-efficient RL. Further details of TD-MPC are available on Hansen et al. (2022).

## D Background: BC

In this section, we offer a brief summary of BC Atkeson & Schaal (1997), a straightforward and commonly utilized approach in imitation learning. BC relies entirely on demonstration data and presents an ideal scenario if successful policies can be trained, as it entails zero interaction sample complexity. BC trains a parameterized policy, denoted as $\pi_\theta : \mathcal{O} \mapsto \mathcal{A}$, with the objective of predicting the demonstrated action from the corresponding observation. Typically, BC cannot surpass the capabilities of the expert because it lacks a concept of task success. This factor motivates the need for combining demonstrations with sample-efficient RL.

In this study, we use BC to pre-train a policy on a handful of expert demonstration data (just 5 demonstrations) before training TD-MPC by means of VIP-generated rewards. While policy pre-training does not produce successful policies under a limited number of demonstrations, it can provide an inductive prior through initialization. Notably, oversampling of the demonstrated data within the RL learning process significantly accelerates TD-MPC.

## E Experimental Details

**Environments** From a diverse set of robotic manipulation tasks from Meta-World Yu et al. (2020), we select the following four tasks: *box-close* at the medium-level of difficulty, and *push*, *pick-place*,

and *assembly* at the hard-level of difficulty. Here, the classification of difficulty levels is defined in Seo et al. (2023). For our evaluation, we follow the experimental setup outlined in Seo et al. (2023) for Meta-World, with the exception of image size. Observations are a stack of the two most recent $224\times224$ RGB images from a bird-view camera. For policy pre-training, we collect expert demonstration data by employing MPC with a ground truth model.

**Network Architecture** We follow the network architecture of TD-MPC (see Hansen et al. (2022) for more details).

**Hyper-parameters** We follow the original hyper-parameters of TD-MPC, with the exception of the parameters associated with reward specification and policy pre-training, including reward type, number of demonstrations, and oversampling ratio. The main hyper-parameters are describe in Table 1. Further details are available on Hansen et al. (2022).

Table 1: An overview of hyper-parameters.

| Hyper-parameter | Value |
|---|---|
| Discount factor | 0.99 |
| Seed frames | 5,000 |
| Replay buffer capacity | Unlimited |
| Latent dimension | 50 |
| MLP hidden size | 512 |
| MLP activation | ELU |
| Sampling technique | PER ($\alpha$=0.6, $\beta$=0.4) |
| Planning horizon | 5 |
| Initial parameters ($\mu^0$, $\sigma^0$) | (0, 2) |
| Population size | 512 |
| Elite fraction | 64 |
| Learning rate | $10^{-3}$ |
| Optimizer | Adam |
| Temporal coefficient ($\lambda$) | 0.5 |
| Reward loss coefficient ($c_1$) | 0.5 |
| Value loss coefficient ($c_2$) | 0.1 |
| Consistency loss coefficient ($c_3$) | 2 |
| Exploration schedule ($\epsilon$) | $0.5 \rightarrow 0.05$ (25K steps) |
| Planning horizon schedule | $1 \rightarrow 5$ (25K steps) |
| Mini-batch size | 256 |
| Frame stack | 2 |
| Action repeat | 2 |
| Steps per gradient update | 1 |
| Soft-update frequency | 2 |
| Reward type | VIP (or VIP+Sparse) |
| Number of demos | 5 |
| Oversampling ratio | 50% |