TD-MPC2: Scalable, Robust World Models for Continuous Control

Nicklas Hansen[*], **Hao Su**[*†], **Xiaolong Wang**[*†]
[*]University of California San Diego, [†]Equal advising
{nihansen,haosu,xiw012}@ucsd.edu

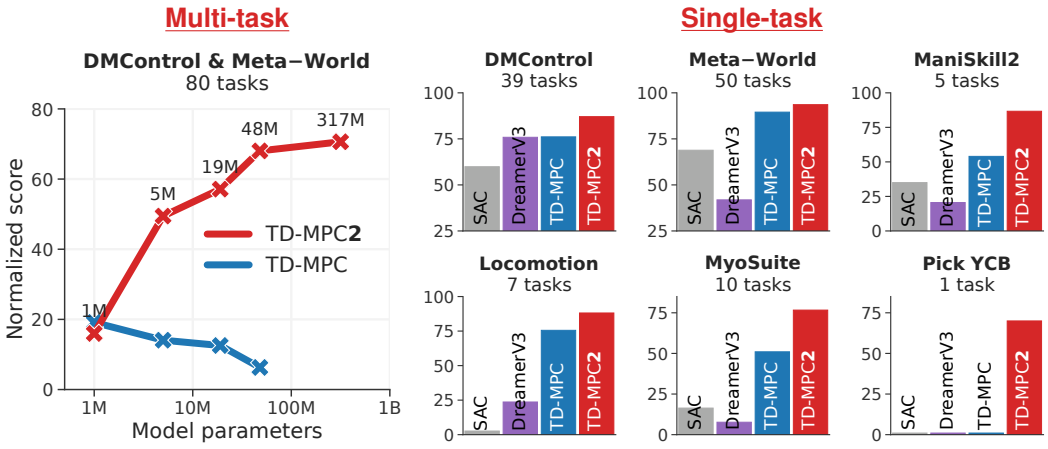

Figure 1. **Overview.** TD-MPC**2** compares favorably to existing model-free and model-based RL methods across **104** continuous control tasks spanning multiple domains, with a *single* set of hyperparameters (*right*). We further demonstrate the scalability of TD-MPC**2** by training a 317M parameter agent to perform **80** tasks across multiple domains, embodiments, and action spaces (*left*).

Abstract

TD-MPC is a model-based reinforcement learning (RL) algorithm that performs local trajectory optimization in the latent space of a learned implicit (decoder-free) world model. In this work, we present TD-MPC**2**: a series of improvements upon the TD-MPC algorithm. We demonstrate that TD-MPC**2** improves significantly over baselines across **104** online RL tasks spanning 4 diverse task domains, achieving consistently strong results with a single set of hyperparameters. We further show that agent capabilities increase with model and data size, and successfully train a single 317M parameter agent to perform **80** tasks across multiple task domains, embodiments, and action spaces.

Explore videos, models, data, code, and more at
https://nicklashansen.github.io/td-mpc2

1 Introduction

Training large models on internet-scale datasets has led to generalist models that perform a wide variety of language and vision tasks (Brown et al., 2020; He et al., 2022; Kirillov et al., 2023). The success of these models can largely be attributed to the availability of enormous datasets, and carefully designed architectures that reliably scale with model and data size. While researchers have recently

NeurIPS 2023 Workshop on Robot Learning: Pretraining, Fine-Tuning, and Generalization with Large Scale Models, New Orleans, United States of America

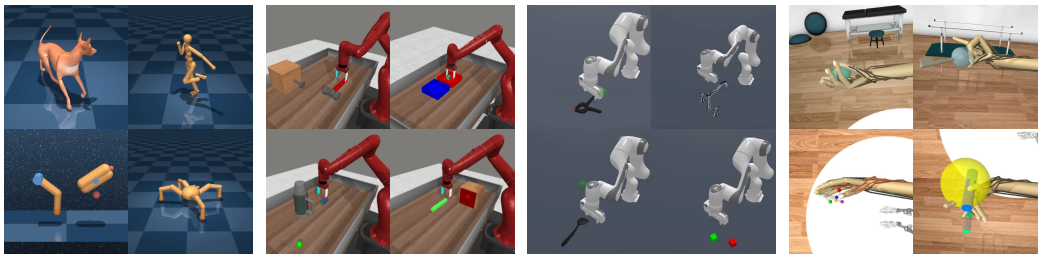

Figure 2. **Tasks.** TD-MPC**2** performs **104** diverse tasks from (left to right) DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), ManiSkill2 (Gu et al., 2023), and MyoSuite (Caggiano et al., 2022), with a *single* set of hyperparameters.

extended this paradigm to robotics (Reed et al., 2022; Brohan et al., 2023), a generalist embodied agent that learns to perform diverse control tasks via low-level actions, across multiple embodiments, from large uncurated (*i.e.*, mixed-quality) datasets remains an elusive goal. We argue that current approaches to generalist embodied agents suffer from *(a)* the assumption of near-expert trajectories for behavior cloning which severely limits the amount of available data (Reed et al., 2022; Lee et al., 2022; Kumar et al., 2022; Schubert et al., 2023; Driess et al., 2023; Brohan et al., 2023), and *(b)* a lack of scalable continuous control algorithms that are able to consume large uncurated datasets.

Reinforcement Learning (RL) is an ideal framework for extracting expert behavior from uncurated datasets. However, most existing RL algorithms (Lillicrap et al., 2016; Haarnoja et al., 2018) are designed for single-task learning and rely on per-task hyperparameters, with no principled method for selecting those hyperparameters (Zhang et al., 2021). An algorithm that can consume large multi-task datasets will invariably need to be robust to variation between different tasks (*e.g.*, action space dimensionality, difficulty of exploration, and reward distribution). In this work, we present TD-MPC**2**: a significant step towards achieving this goal. TD-MPC**2** is a model-based RL algorithm designed for learning generalist world models on large uncurated datasets composed of multiple task domains, embodiments, and action spaces, with data sourced from behavior policies that cover a wide range of skill levels, and without the need for hyperparameter-tuning.

Our algorithm, which builds upon TD-MPC (Hansen et al., 2022), performs local trajectory optimization in the latent space of a learned implicit (decoder-free) world model. While the TD-MPC family of algorithms has demonstrated strong empirical performance in prior work (Hansen et al., 2022, 2023; Yuan et al., 2022; Yang et al., 2023; Feng et al., 2023; Chitnis et al., 2023; Zhu et al., 2023; Lancaster et al., 2023), most successes have been limited to single-task learning with little emphasis on scaling. As shown in Figure 1, naïvely increasing model and data size of TD-MPC often leads to a net *decrease* in agent performance, as is commonly observed in RL literature (Kumar et al., 2023). In contrast, scaling TD-MPC**2** leads to consistently improved capabilities. Our algorithmic contributions, which have been key to achieving this milestone, are two-fold: *(1)* improved algorithmic robustness by revisiting core design choices, and *(2)* careful design of an architecture that can accommodate datasets with multiple embodiments and action spaces without relying on domain knowledge. The resulting algorithm, TD-MPC**2**, is scalable, robust, and can be applied to a variety of single-task and multi-task continuous control problems using a *single* set of hyperparameters. Refer to Appendix A for a description of the TD-MPC**2** algorithm.

We evaluate TD-MPC**2** across a total of **104** diverse continuous control tasks spanning 4 task domains: DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), ManiSkill2 (Gu et al., 2023), and MyoSuite (Caggiano et al., 2022). We summarize our results in Figure 1, and visualize task domains in Figure 2. Our results demonstrate that TD-MPC**2** consistently outperforms existing model-based and model-free methods, using the *same* hyperparameters across all tasks (Figure 1, *right*). Here, “Locomotion" and “Pick YCB" are particularly challenging subsets of DMControl and ManiSkill2, respectively. We further show that agent capabilities increase with model and data size, and successfully train a single 317M parameter world model to perform **80** tasks across multiple task domains, embodiments, and action spaces (Figure 1, *left*).

2 Experiments

We evaluate TD-MPC**2** across a total of **104** diverse continuous control tasks spanning 4 task domains: DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), ManiSkill2 (Gu et al., 2023), and MyoSuite (Caggiano et al., 2022). Tasks include high-dimensional state and action spaces (up to

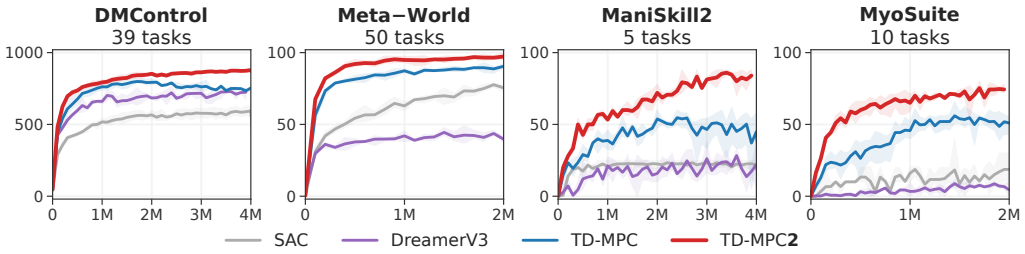

Figure 3. **Single-task RL.** Episode return (DMControl) and success rate (others) as a function of environment steps across **104** continuous control tasks spanning 4 diverse task domains. TD-MPC**2** achieves higher data-efficiency and asymptotic performance than existing methods, while using the **same** hyperparameters across all tasks. Mean and 95% CIs over 3 seeds.

$\mathcal{A} \in \mathbb{R}^{39}$), sparse rewards, multi-object manipulation, physiologically accurate musculoskeletal motor control, complex locomotion (*e.g.* Dog and Humanoid embodiments), and cover a wide range of task difficulties. In support of open-source science, **we publicly release 300+ model checkpoints, datasets, and code for training and evaluating TD-MPC2 agents, most of which has already been made available at** `https://nicklashansen.github.io/td-mpc2`.

We seek to answer three core research questions through experimentation:

- **Comparison to existing methods.** How does TD-MPC**2** compare to state-of-the-art model-free (SAC) and model-based (DreamerV3, TD-MPC) methods for data-efficient continuous control?
- **Scaling.** Do the algorithmic innovations of TD-MPC**2** lead to improved agent capabilities as model and data size increases? Can a single agent learn to perform diverse skills across multiple task domains, embodiments, and action spaces?
- **Analysis.** How do the specific design choices introduced in TD-MPC**2** influence downstream task performance? How much does planning contribute to its success? Are the learned task embeddings semantically meaningful? Can large multi-task agents be adapted to unseen tasks?

Baselines. Our baselines represent the state-of-the-art in data-efficient RL, and include *(1)* **Soft Actor-Critic** (SAC) (Haarnoja et al., 2018), a model-free actor-critic algorithm based on maximum entropy RL, *(2)* **DreamerV3** (Hafner et al., 2023), a model-based method that optimizes a model-free policy with rollouts from a learned generative model of the environment, and *(3)* the original version of **TD-MPC** (Hansen et al., 2022), a model-based RL algorithm that performs local trajectory optimization (planning) in the latent space of a learned *implicit* (non-generative) world model. SAC and TD-MPC use task-specific hyperparameters, whereas TD-MPC**2** uses the **same** hyperparameters across all tasks. We use a 5M parameter TD-MPC**2** agent in all experiments (unless stated otherwise). For reference, the DreamerV3 baseline has approx. 20M learnable parameters.

2.1 Results

Comparison to existing methods. We first compare the data-efficiency of TD-MPC**2** to a set of strong baselines on **104** diverse tasks in an online RL setting. Aggregate results are shown in Figure 3. We find that TD-MPC**2** outperforms prior methods across all task domains. The MyoSuite results are particularly noteworthy, as we did not run *any* TD-MPC**2** experiments on this benchmark prior to the reported results. See Appendix B for the full single-task RL results.

Massively multitask world models. To demonstrate that our proposed improvements facilitate scaling of world models, we evaluate the performance of 5 multitask models ranging from 1M to 317M parameters on a collection of **80** diverse tasks that span multiple task domains and vary greatly in objective, embodiment, and action space. Models are trained on a dataset of 545M transitions obtained from the replay buffers of 240 single-task TD-MPC**2** agents, and thus contain a wide variety of behaviors ranging from random to expert policies. The task set consists of all 50 Meta-World tasks, as well as 30 DMControl tasks. The DMControl task set includes 19 original DMControl tasks, as well as 11 new tasks. For completeness, we include a separate set of scaling results on the 30-task DMControl subset (345M transitions) as well. Due to our careful design of the TD-MPC**2** algorithm, scaling up is straightforward: to improve rate of convergence we use a $4\times$ larger batch size (1024) compared to the single-task experiments, but make no other changes to hyperparameters.

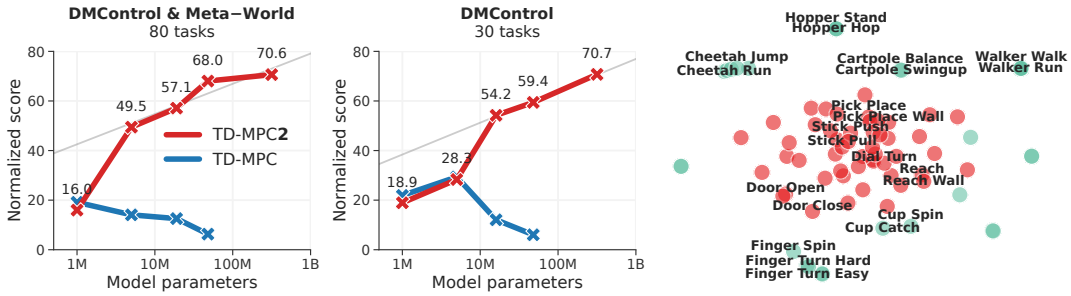

Figure 4. **Massively multi-task world models.** *(Left)* Normalized score as a function of model size on the two 80-task and 30-task datasets. TD-MPC**2** capabilities scale with model size. *(Right)* T-SNE (van der Maaten & Hinton, 2008) visualization of task embeddings learned by a TD-MPC**2** agent trained on 80 tasks from DMControl and Meta-World. A subset of labels are shown for clarity.

Scaling TD-MPC2 to 317M parameters. Our scaling results are shown in Figure 4. To summarize agent performance with a single metric, we produce a normalized score that is an average of all individual task success rates (Meta-World) and episode returns normalized to the $[0, 100]$ range (DMControl). We observe that agent capabilities consistently increase with model size on both task sets. Notably, performance does not appear to have saturated for our largest models (317M parameters) on either dataset, and we can thus expect results to continue improving beyond our considered model sizes. We refrain from formulating a scaling law, but note that normalized score appears to scale linearly with the log of model parameters (gray line in Figure 4). To better understand why multitask model learning is successful, we explore the task embeddings learned by TD-MPC**2** (Figure 4, right). Intriguingly, tasks that are semantically similar (*e.g.*, Door Open and Door Close) are close in the learned task embedding space. However, embedding similarity appears to align more closely with task *dynamics* (embodiment, objects) than objective (walk, run). This makes intuitive sense, as dynamics are tightly coupled with control.

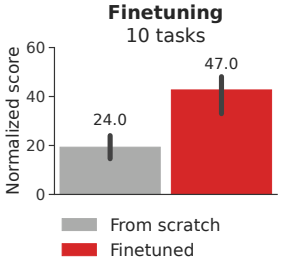

Figure 5. **Finetuning.** Score of a 19M parameter TD-MPC**2** agent trained on 70 tasks and finetuned online to each of 10 held-out tasks for 20k environment steps. 3 seeds.

Few-shot learning. While our work mainly focuses on the *scaling* and *robustness* of world models, we also explore the efficacy of finetuning pretrained world models for few-shot learning of unseen tasks. Specifically, we pretrain a 19M parameter TD-MPC**2** agent on 70 tasks from DMControl and Meta-World, and naïvely finetune the full model to each of 10 held-out tasks (5 from each domain) via online RL with an initially empty replay buffer and no changes to hyperparameters. Aggregate results are shown in Figure 5. We find that TD-MPC**2** improves $\mathbf{2}\times$ over learning from scratch on new tasks in the low-data regime (20k environment steps[1]). Although finetuning world models to new tasks is very much an open research problem, our exploratory results are promising.

See Appendix C for ablations.

[1]20k environment steps corresponds to 20 episodes in DMControl and 100 episodes in Meta-World.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *Advances in Neural Information Processing Systems*, 2016.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. Myosuite – a contact-rich simulation suite for musculoskeletal motor control. https://github.com/facebookresearch/myosuite, 2022. URL https://sites.google.com/view/myosuite.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics*, pp. 510–517, 2015. doi: 10.1109/ICAR.2015.7251504.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *International Conference on Learning Representations*, 2021.

Rohan Chitnis, Yingchen Xu, Bobak Hashemi, Lucas Lehnert, Urun Dogan, Zheqing Zhu, and Olivier Delalleau. Iql-td-mpc: Implicit q-learning for hierarchical model predictive control. *arXiv preprint arXiv:2306.00867*, 2023.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yunhai Feng, Nicklas Hansen, Ziyan Xiong, Chandramouli Rajagopalan, and Xiaolong Wang. Finetuning offline world models in the real world. *Conference on Robot Learning*, 2023.

Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiaing Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, G. Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, P. Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *ArXiv*, abs/1812.05905, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022.

Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should i run offline reinforcement learning or behavioral cloning? *International Conference on Learning Representations*, 2022.

Aviral Kumar, Rishabh Agarwal, Xinyang Geng, George Tucker, and Sergey Levine. Offline q-learning on diverse multi-task data both scales and generalizes. *International Conference on Learning Representations*, 2023.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *Conference on Learning for Decision and Control*, 2020.

Patrick Lancaster, Nicklas Hansen, Aravind Rajeswaran, and Vikash Kumar. Modem-v2: Visuo-motor world models for real-world robot manipulation. *arXiv preprint*, 2023.

Samuel Lavoie, Christos Tsirigotis, Max Schwarzer, Ankit Vani, Michael Noukhovitch, Kenji Kawaguchi, and Aaron Courville. Simplicial embeddings in self-supervised learning and downstream classification. *arXiv preprint arXiv:2204.00616*, 2022.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35:27921–27936, 2022.

T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Ingmar Schubert, Jingwei Zhang, Jake Bruce, Sarah Bechtle, Emilio Parisotto, Martin Riedmiller, Jost Tobias Springenberg, Arunkumar Byravan, Leonard Hasenclever, and Nicolas Heess. A generalist dynamics model for control. *arXiv preprint arXiv:2305.10912*, 2023.

R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1998.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, et al. Deepmind control suite. Technical report, DeepMind, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control using covariance variable importance sampling. *ArXiv*, abs/1509.01149, 2015.

Sizhe Yang, Yanjie Ze, and Huazhe Xu. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 2023.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *International Conference on Learning Representations*, 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*, 2019.

Yifu Yuan, Jianye Hao, Fei Ni, Yao Mu, Yan Zheng, Yujing Hu, Jinyi Liu, Yingfeng Chen, and Changjie Fan. Euclid: Towards efficient unsupervised reinforcement learning with multi-choice dynamics model. *arXiv preprint arXiv:2210.00498*, 2022.

Baohe Zhang, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. On the importance of hyperparameter optimization for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 4015–4023. PMLR, 2021.

Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. Repo: Resilient model-based reinforcement learning by regularizing posterior predictability. *arXiv preprint arXiv:2309.00082*, 2023.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 3, 2008.

A TD-MPC2

Our work builds upon TD-MPC (Hansen et al., 2022), a model-based RL algorithm that performs local trajectory optimization (planning) in the latent space of a learned implicit world model. TD-MPC**2** marks the beginning of a new era for model-based RL, in which massively multitask world models are trained and subsequently finetuned to new tasks. Specifically, we propose a series of improvements to the TD-MPC algorithm, which have been key to achieving strong algorithmic robustness (can use the same hyperparameters across all tasks) and scaling its world model to **300**$\times$ more parameters than previously. In the following, we introduce the TD-MPC**2** algorithm in detail.

A.1 Learning an Implicit World Model

Learning a generative model of the environment using a reconstruction (decoder) objective is tempting due to its rich learning signal. However, accurately predicting raw future observations (*e.g.*, images or proprioceptive features) over long time horizons is a difficult problem, and does not necessarily lead to effective control (Lambert et al., 2020). Rather than explicitly modeling dynamics using reconstruction, TD-MPC**2** aims to learn a *maximally useful* model: a model that accurately predicts *outcomes* (returns) conditioned on a sequence of actions. Specifically, TD-MPC**2** learns an *implicit*, control-centric world model from environment interaction using a combination of joint-embedding prediction (Grill et al., 2020), reward prediction, and TD-learning (Sutton, 1998), *without* decoding observations. We argue that this alternative formulation of model-based RL is key to modeling large datasets with modest model sizes. The world model can subsequently be used for decision-making by performing local trajectory optimization (planning) following the MPC framework.

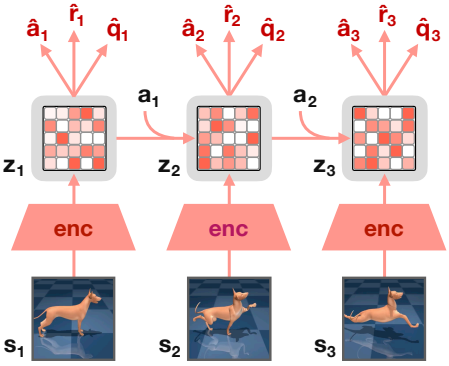

Figure 6. **The TD-MPC2 architecture.** Observations $\mathbf{s}$ are encoded into their (normalized) latent representation $\mathbf{z}$. The model then recurrently predicts actions $\hat{\mathbf{a}}$, rewards $\hat{r}$, and terminal values $\hat{q}$, *without* decoding future observations.

Components. The TD-MPC**2** architecture is shown in Figure 6 and consists of five components:

$$
\begin{array}{lll}
\text{Encoder} & \mathbf{z} = h(\mathbf{s}, \mathbf{e}) & \triangleright \text{ Maps observations to their latent representations} \\
\text{Latent dynamics} & \mathbf{z}' = d(\mathbf{z}, \mathbf{a}, \mathbf{e}) & \triangleright \text{ Models (latent) forward dynamics} \\
\text{Reward} & \hat{r} = R(\mathbf{z}, \mathbf{a}, \mathbf{e}) & \triangleright \text{ Predicts reward } r \text{ of a transition} \\
\text{Terminal value} & \hat{q} = Q(\mathbf{z}, \mathbf{a}, \mathbf{e}) & \triangleright \text{ Predicts discounted sum of rewards (return)} \\
\text{Policy prior} & \hat{\mathbf{a}} = p(\mathbf{z}, \mathbf{e}) & \triangleright \text{ Predicts action } \mathbf{a}^* \text{ that maximizes } Q
\end{array} \tag{1}
$$

where $\mathbf{s}$ and $\mathbf{a}$ are states and actions, $\mathbf{z}$ is the latent representation, and $\mathbf{e}$ is a learnable task embedding for use in multitask world models. For visual clarity, we will omit $\mathbf{e}$ in the following unless it is particularly relevant. The policy prior p serves to guide the sample-based trajectory optimizer (planner), and to reduce the computational cost of TD-learning. During online interaction, TD-MPC**2** maintains a replay buffer $\mathcal{B}$ with trajectories, and iteratively *(i)* updates the world model using data sampled from $\mathcal{B}$, and *(ii)* collects new environment data by planning with the learned model.

Model objective. The h, d, R, Q components are jointly optimized to minimize the objective

$$
\mathcal{L}(\theta) \doteq \mathop{\mathbb{E}}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')_{0:H} \sim \mathcal{B}} \left[\sum_{t=0}^{H} \lambda^t \left(\underbrace{\| \mathbf{z}'_t - \text{sg}(h(\mathbf{s}'_t)) \|_2^2}_{\text{Joint-embedding prediction}} + \underbrace{\text{CE}(\hat{r}_t, r_t)}_{\text{Reward prediction}} + \underbrace{\text{CE}(\hat{q}_t, q_t)}_{\text{Value prediction}} \right) \right], \tag{2}
$$

where sg is the `stop-grad` operator, $(\mathbf{z}'_t, \hat{r}_t, \hat{q}_t)$ are as defined in Equation 1, $q_t \doteq r_t + \bar{Q}(\mathbf{z}'_t, p(\mathbf{z}'_t))$ is the TD-target at step t, $\lambda \in (0, 1]$ is a constant coefficient that weighs temporally farther time steps less, and CE is the cross-entropy. $\bar{Q}$ used to compute the TD-target is an exponential moving average (EMA) of Q (Lillicrap et al., 2016). As the magnitude of rewards may differ drastically between tasks, TD-MPC**2** formulates reward and value prediction as a discrete regression (multi-class classification) problem in a log-transformed space, which is optimized by minimizing cross-entropy with r_t, q_t as soft targets (Bellemare et al., 2017; Kumar et al., 2023; Hafner et al., 2023).

Policy objective. The policy prior p is a stochastic maximum entropy (Ziebart et al., 2008; Haarnoja et al., 2018) policy that learns to maximize the objective

$$\mathcal{L}_p(\theta) \doteq \mathop{\mathbb{E}}_{(\mathbf{s},\mathbf{a})_{0:H}\sim\mathcal{B}}\left[\sum_{t=0}^{H}\lambda^t\left[\alpha Q(\mathbf{z}_t, p(\mathbf{z}_t)) - \beta\mathcal{H}(p(\cdot|\mathbf{z}_t))\right]\right], \ \mathbf{z}_{t+1} = d(\mathbf{z_t}, \mathbf{a_t}), \ \mathbf{z}_0 = h(\mathbf{s}_0)\,, \quad (3)$$

where $\mathcal{H}$ is the entropy of p which can be computed in closed form. Gradients of $\mathcal{L}_p(\theta)$ are taken wrt. p only. As magnitude of the value estimate $Q(\mathbf{z}_t, p(\mathbf{z}_t))$ and entropy $\mathcal{H}$ can vary greatly between datasets and different stages of training, it is necessary to balance the two losses to prevent premature entropy collapse (Yarats et al., 2021). A common choice for automatically tuning α, β is to keep one of them constant, and adjusting the other based on an entropy target (Haarnoja et al., 2018) or moving statistics (Hafner et al., 2023). In practice, we opt for tuning α via moving statistics, but empirically did not observe any significant difference in results between these two options.

Architecture. All components of TD-MPC**2** are implemented as MLPs with intermediate linear layers followed by LayerNorm (Ba et al., 2016) and Mish (Misra, 2019) activations. To mitigate exploding gradients, we normalize the latent representation by projecting $\mathbf{z}$ into L fixed-dimensional simplices using a softmax operation (Lavoie et al. (2022). A key benefit of embedding $\mathbf{z}$ as simplices (as opposed to *e.g.* a discrete representation or squashing) is that it naturally biases the representation towards sparsity without enforcing hard constraints. We dub this normalization scheme *SimNorm*. Let V be the dimensionality of each simplex $\mathbf{g}$ constructed from L partitions (groups) of $\mathbf{z}$. SimNorm then applies the following transformation:

$$\mathbf{z}^\circ \doteq [\mathbf{g}_i, \ldots, \mathbf{g}_L]\,, \ \mathbf{g}_i = \frac{e^{\mathbf{z}_{i:i+V}/\tau}}{\sum_{j=1}^{V} e^{\mathbf{z}_{i:i+V}/\tau}}\,, \quad (4)$$

where $\mathbf{z}^\circ$ is the simplicial embedding of $\mathbf{z}$, $[\cdot]$ denotes concatenation, and $\tau > 0$ is a temperature parameter that modulates the "sparsity" of the representation. As we will demonstrate in our experiments, SimNorm is essential to the training stability of TD-MPC**2**. Finally, to reduce bias in TD-targets generated by $\bar{Q}$, we learn an *ensemble* of Q-functions using the objective from Equation 2 and maintain $\bar{Q}$ as an EMA of each Q-function. We use 5 Q-functions in practice. Targets are then computed as the minimum of two randomly sub-sampled $\bar{Q}$-functions (Chen et al., 2021).

A.2 Model Predictive Control with a Policy Prior

TD-MPC**2** derives its closed-loop control policy by planning with the learned world model. Specifically, our approach leverages the MPC framework for local trajectory optimization using Model Predictive Path Integral (MPPI) (Williams et al., 2015) as a derivative-free optimizer with sampled action sequences $(\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+H})$ of length H evaluated by rolling out *latent* trajectories with the model. At each decision step, we estimate parameters μ^*, σ^* of a time-dependent multivariate Gaussian with diagonal covariance such that expected return is maximized, *i.e.*,

$$\mu^*, \sigma^* = \arg\max_{(\mu,\sigma)} \mathop{\mathbb{E}}_{(\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+H})\sim\mathcal{N}(\mu,\sigma^2)}\left[\gamma^H Q(\mathbf{z}_{t+H}, \mathbf{a}_{t+H}) + \sum_{h=t}^{H-1}\gamma^h R(\mathbf{z}_h, \mathbf{a}_h)\right], \quad (5)$$

where $\mu, \sigma \in \mathbb{R}^{H\times m}$, $\mathcal{A} \in \mathbb{R}^m$. Equation 5 is solved by iteratively sampling action sequences from $\mathcal{N}(\mu, \sigma^2)$, evaluating their expected return, and updating μ, σ based on a weighted average. Notably, Equation 5 estimates the full RL objective by bootstrapping with the learned terminal value function beyond horizon H. TD-MPC**2** repeats this iterative planning process for a fixed number of iterations and executes the first action $\mathbf{a}_t \sim \mathcal{N}(\mu_t^*, \sigma_t^*)$ in the environment. To accelerate convergence of planning, a fraction of action sequences originate from the policy prior p, and we warm-start planning by initializing (μ, σ) as the solution to the previous decision step shifted by 1. Refer to Hansen et al. (2022) for more details about the planning procedure.

A.3 Training Generalist TD-MPC2 Agents

The success of TD-MPC**2** in diverse single-task problems can be attributed to the algorithm outlined above. However, learning a large generalist TD-MPC**2** agent that performs a variety of tasks across multiple task domains, embodiments, and action spaces poses several unique challenges: *(i)* how to learn and represent task semantics? *(ii)* how to accommodate multiple observation and action spaces

without specific domain knowledge? *(iii)* how to leverage the learned model for few-shot learning of new tasks? We describe our approach to multitask model learning in the following.

Learnable task embeddings. To succeed in a multitask setting, an agent needs to learn a common representation that takes advantage of task similarities, while still retaining the ability to differentiate between tasks at test-time. When task or domain knowledge is available, *e.g.* in the form of natural language instructions, the task embedding $\mathbf{e}$ from Equation 1 may encode such information. However, in the general case where domain knowledge cannot be assumed, we may instead choose to *learn* the task embeddings (and, implicitly, task relations) from data. TD-MPC**2** conditions all of its five components with a learnable, fixed-dimensional task embedding $\mathbf{e}$, which is jointly trained together with other components of the model. To improve training stability, we constrain the ℓ_2-norm of $\mathbf{e}$ to be ≤ 1. When finetuning a multitask TD-MPC**2** agent to a new task, we can choose to either initialize $\mathbf{e}$ as the embedding of a semantically similar task, or simply as a random vector.

Action masking. TD-MPC**2** learns to perform tasks with a variety of observation and action spaces, without any domain knowledge. To do so, we zero-pad all model inputs and outputs to their largest respective dimensions, and mask out invalid action dimensions in predictions made by the policy prior p during both training and inference. This ensures that prediction errors in invalid dimensions do not influence TD-target estimation, and prevents p from falsely inflating its entropy for tasks with small action spaces. We similarly only sample actions along valid dimensions during planning.

B Single-task Experimental Results

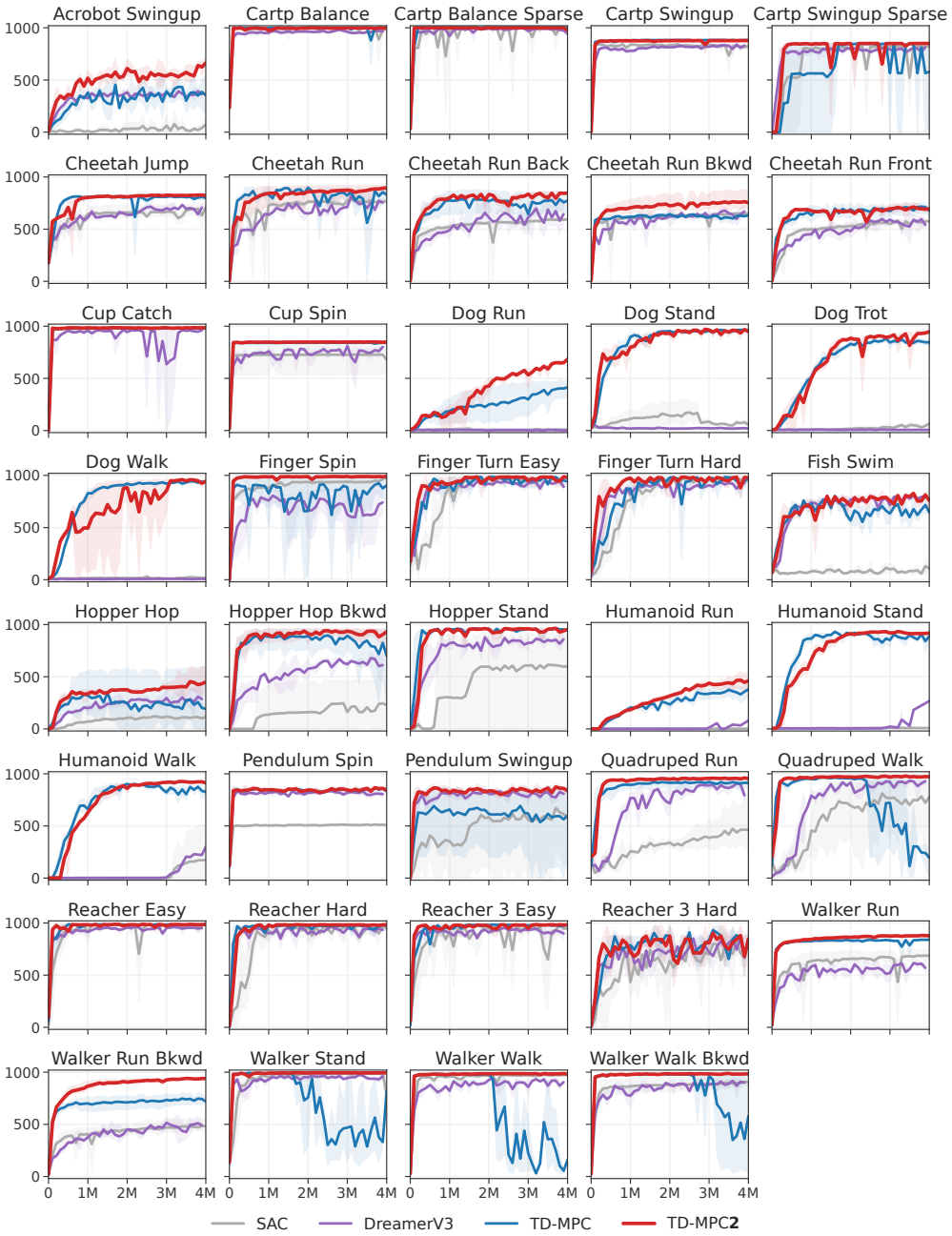

Figure 7. **Single-task DMControl results.** Episode return as a function of environment steps. The first 4M environment steps are shown for each task, although the Humanoid and Dog tasks are run for 14M environment steps; we provide those curves in Figure 10 as part of the "Locomotion" benchmark. Note that TD-MPC diverges on tasks like *Walker Stand* and *Walker Walk* whereas TD-MPC**2** remains stable. Mean and 95% CIs over 3 seeds.

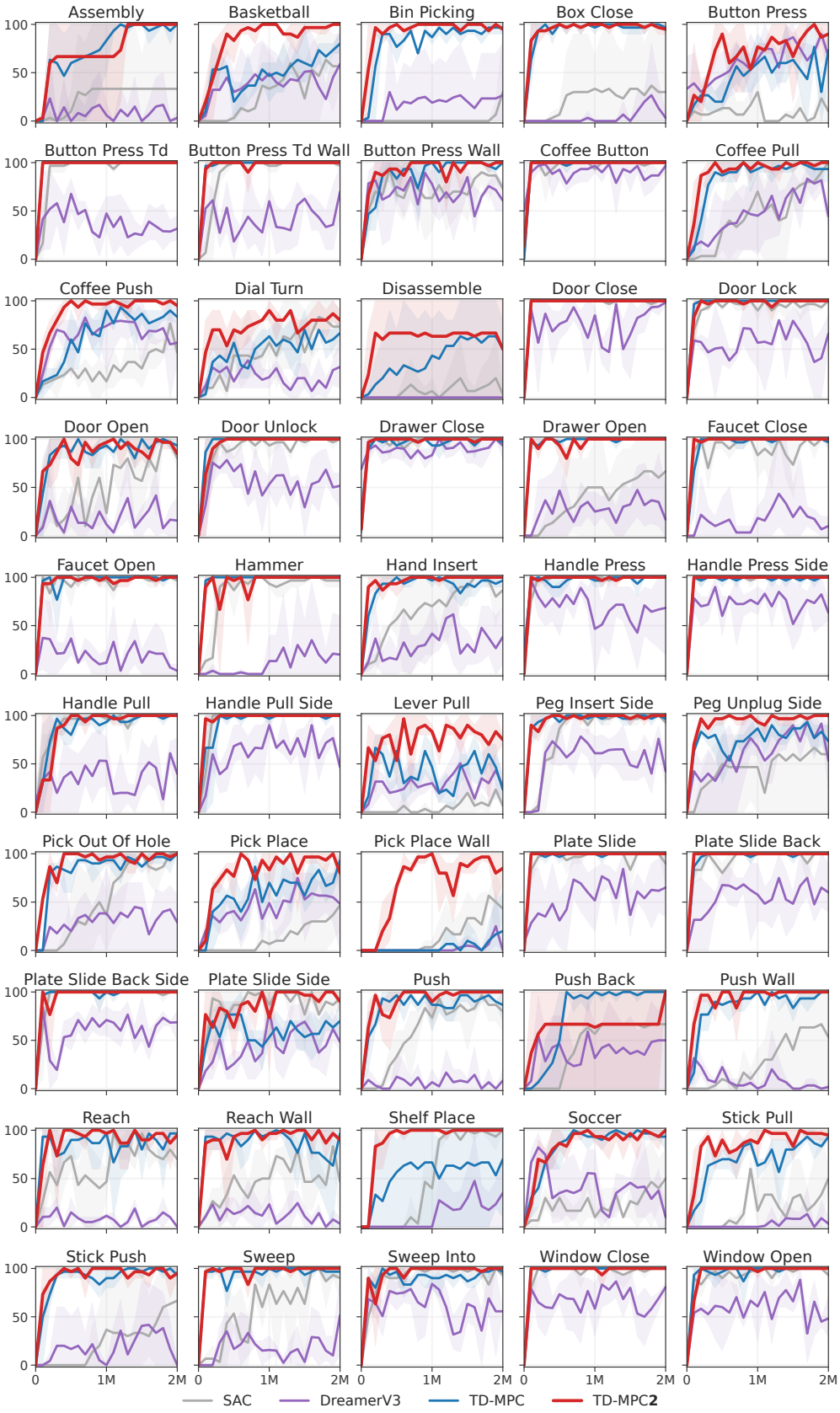

Figure 8. **Single-task Meta-World results.** Success rate (%) as a function of environment steps. TD-MPC**2** performance is comparable to existing methods on easy tasks, while outperforming other methods on hard tasks such as *Pick Place Wall* and *Shelf Place*. DreamerV3 often fails to converge.

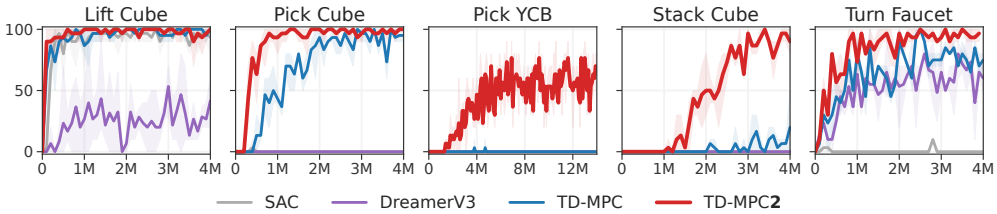

Figure 9. **Single-task ManiSkill2 results.** Success rate (%) as a function of environment steps on 5 object manipulation tasks from ManiSkill2. *Pick YCB* is the hardest task and considers manipulation of all 74 objects from the YCB (Calli et al., 2015) dataset. We report results for this tasks at 14M environment steps, and 4M environment steps for other tasks. TD-MPC**2** achieves a $> 60\%$ success rate on the Pick YCB task, whereas other methods fail to learn within the given budget. Mean and 95% CIs over 3 seeds.

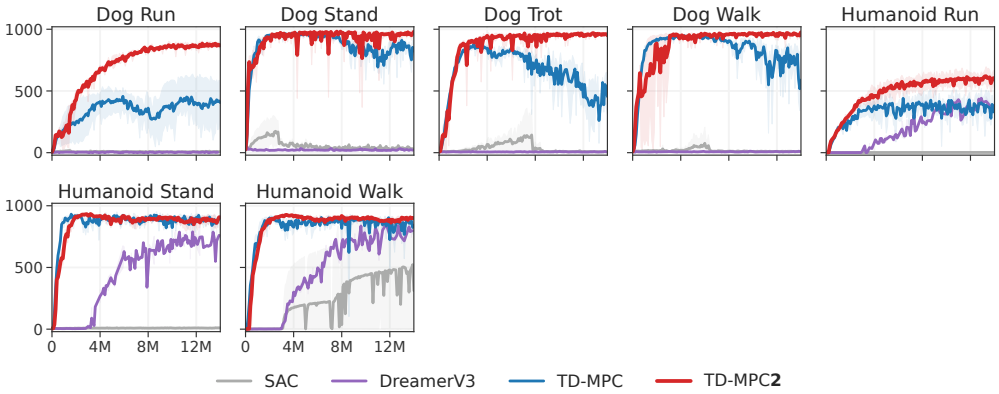

Figure 10. **Single-task high-dimensional locomotion results.** Episode return as a function of environment steps on all 7 "Locomotion" benchmark tasks. This domain includes high-dimensional Humanoid ($\mathcal{A} \in \mathbb{R}^{21}$) and Dog ($\mathcal{A} \in \mathbb{R}^{38}$) embodiments. Mean and 95% CIs over 3 seeds.

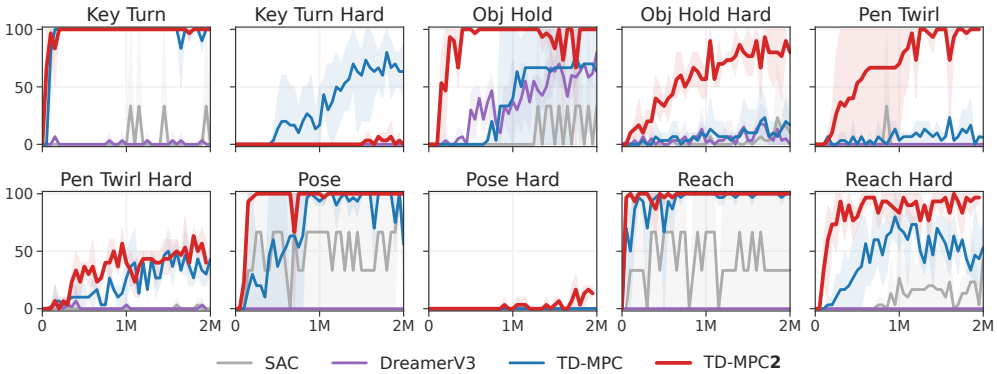

Figure 11. **Single-task MyoSuite results.** Success rate (%) as a function of environment steps. This task domain includes high-dimensional contact-rich musculoskeletal motor control ($\mathcal{A} \in \mathbb{R}^{39}$) with a physiologically accurate robot hand. Goals are randomized in tasks designated as "Hard". TD-MPC**2** achieves comparable or better performance than existing methods on all tasks from this benchmark, except for *Key Turn Hard* in which TD-MPC succeeds early in training.

C Ablations

We ablate most of our design choices for TD-MPC**2**, including choice of actor, various normalization techniques, regression objective, and number of Q-functions. Our ablations, shown in Figure 12, are conducted on three of the most difficult online RL tasks, as well as large-scale multitask training (80 tasks). We observe that all of our proposed improvements contribute meaningfully to the robustness and strong performance of TD-MPC**2** in both single-task RL and multi-task RL. Interestingly, we find that the relative importance of each design choice is consistent across both settings.

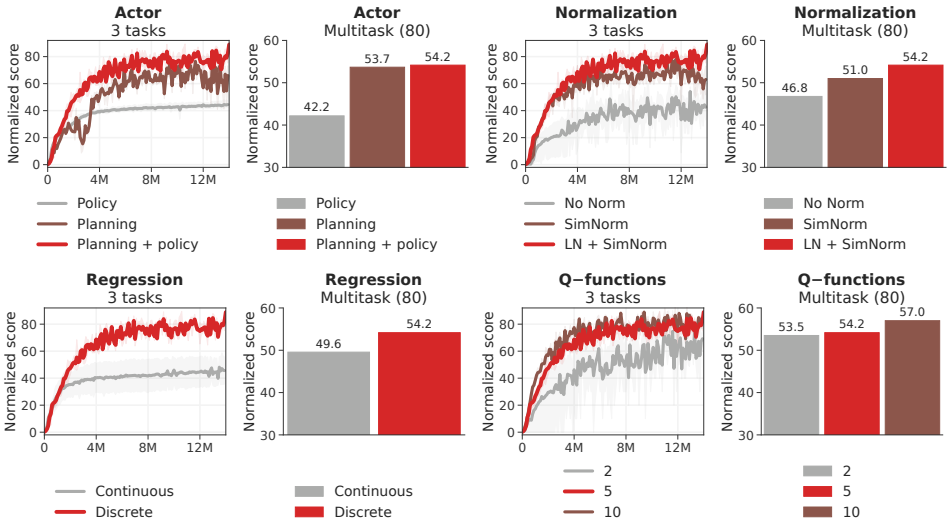

Figure 12. **Ablations.** *(Curves)* Normalized score as a function of environment steps, averaged across three of the most difficult tasks: *Dog Run*, *Humanoid Walk* (DMControl), and *Pick YCB* (ManiSkill2). Mean and 95% CIs over 3 random seeds. *(Bars)* Normalized score of 19M parameter multitask (80 tasks) TD-MPC**2** agents. Our ablations highlight the relative importance of each design choice; **red** is the default formulation of TD-MPC**2**.