Representation Learning Enables Scalable Multitask Deep Reinforcement Learning

Johan Obando-Ceron[1,2] **Lu Li**[1,2] **Scott Fujimoto**[3] **Pierre-Luc Bacon**[1,2]
Aaron Courville[1,2,4] **Pablo Samuel Castro**[1,2,5]

[1]Mila – Québec AI Institute [2]Université de Montréal [3]McGill University
[4]CIFAR AI Chair [5]Google DeepMind

jobando0730@gmail.com, scott.fujimoto@mail.mcgill.ca
{lu.li, pierre-luc.bacon, courvila, pablo-samuel.castro}@mila.quebec

Abstract

Scaling reinforcement learning (RL) to diverse multitask settings remains a central challenge. While recent advances in model-based RL achieve strong performance, they rely on planning and complex training pipelines, making it unclear which components are essential for scalability. We revisit this question and argue that the primary driver of scalable multitask RL is not model-based control, but *representation learning*. In particular, we show that combining predictive, model-based representations with high-capacity value function approximation is sufficient to achieve strong performance, even without planning. We evaluate a simple model-free algorithm, MR.Q, coupled with auxiliary predictive objectives into a scalable actor-critic architecture. This approach outperforms a recent world-model-based method and a range of deep RL baselines across a diverse suite of multitask continuous control tasks, while significantly reducing computational overhead and improving wall-clock efficiency. We observe consistent improvements with increased model capacity and show through ablations that predictive representation learning is critical for performance. **Our code is available at ScaleMRL.**

"What we observe isn't nature itself, but nature exposed to our method of questioning[1]*."*
— Werner Heisenberg

1 Introduction

Deep reinforcement learning (RL) has achieved remarkable success across a wide range of domains, including games, robotics, and control [Akkaya et al., 2019, Mnih et al., 2013, Schwarzer et al., 2023]. However, much of this progress remains confined to single-task settings, where agents are trained and evaluated on narrowly defined environments, often requiring hundreds of millions of environment interactions to converge. In contrast, recent advances in machine learning, particularly in language and vision, demonstrate that scaling models across diverse data distributions enables generalization, transfer, and robustness through shared representations [Wang et al., 2022, Alayrac et al., 2022, Kojima et al., 2022, Subramanian et al., 2023, Zhou et al., 2025, Reed et al., 2022, Wiedemer et al., 2026]. Extending these principles to online deep RL remains an open challenge. Unlike supervised settings, RL involves non-stationary data, bootstrapped targets, and long-horizon credit assignment, which introduce optimization instabilities that manifest as representation collapse,

[1]In RL, what an agent "sees" depends on its representation. Our results suggest that improving representations can be more important, and significantly more efficient, than modeling environment dynamics and planning.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

loss of plasticity, and unstable value estimation. These instabilities compound the sample costs of learning and ultimately hinder progress in multitask settings [Kumar et al., 2021, Nikishin et al., 2022, Sokar et al., 2023, Nauman et al., 2024, Tang and Berseth, 2024, Castanyer et al., 2025].

Multitask RL (MTRL) seeks to train a single agent over a distribution of tasks, but doing so across increasingly diverse task distributions introduces instability, task interference, and underutilization of model capacity [Teh et al., 2017, Yu et al., 2020a, D'Eramo et al., 2020, Kong et al., 2025]. Recent work by Nauman et al. [2025] demonstrates that substantially increasing value function capacity, paired with categorical value parameterization and explicit regularization, leads to significant multitask gains. Yet scaling model size alone does not solve the problem: without the right training objectives and representation learning mechanisms, larger models simply require more data to stabilize [Taiga et al., 2023, Farebrother et al., 2024]. This points to representation quality as a central axis of progress, since better representations have been shown to reduce TD variance, accelerate learning, and stabilize training across tasks [Castro et al., 2021, Schwarzer et al., 2021, Fujimoto et al., 2023, Cetin et al., 2023, Echchahed and Castro, 2025, Obando-Ceron et al., 2026a].

Model-based RL methods pursue this goal by leveraging predictive objectives — specifically by learning latent dynamics models — to provide dense supervision that shapes representations beyond what TD learning alone can achieve. This richer learning signal is a key driver behind recent model-based advances [Hafner et al., 2020b, 2025a, Hansen et al., 2024, 2026, Fujimoto et al., 2025]. Recent large-scale systems further combine predictive representation learning, large shared architectures, and planning to achieve strong multitask performance [Xu et al., 2023, Georgiev et al., 2025, Hafner et al., 2025a, Hansen et al., 2026]. Yet because these approaches bundle multiple components together, isolating the source of their gains remains difficult. Moreover, planning itself introduces computational overhead, hyperparameter sensitivity, and compounding model errors, ultimately eroding the very efficiency gains these methods aim to provide [Zhang et al., 2021b, Talvitie, 2014, Rajeswaran et al., 2017, Clavera et al., 2018, Chua et al., 2018, Voelcker et al., 2022].

We hypothesize that much of the benefit attributed to model-based control in fact arises from the representations these methods learn, and that predictive objectives alone are sufficient to achieve competitive sample efficiency at scale [Jaderberg et al., 2017, Gelada et al., 2019, Lee et al., 2020, Anand et al., 2022]. To test this hypothesis, we study MR.Q [Fujimoto et al., 2025], a purely model-free actor–critic agent that integrates predictive objectives into TD learning. MR.Q is a natural probe for this question as it isolates the representational benefits of predictive learning from the confounds of planning, allowing us to test whether richer supervision alone drives sample efficiency gains.

While originally proposed for single-task settings, we extend MR.Q's evaluation to the multitask regime. However, previous MTRL benchmarks evaluate at 100M or more environment steps [Hansen et al., 2026], obscuring whether methods are genuinely sample-efficient or simply benefit from prolonged training. To address this, we consider a version of the benchmark that evaluates agents at 10M environment steps, where sample efficiency gains are most visible.

Across a suite of continuous control benchmarks, MR.Q outperforms a recent world-model-based method (*Newt* [Hansen et al., 2026]) while achieving substantially improved wall-clock, sample efficiency, and demonstrates performance benefits from scaling in both model size and data availability. In addition, MR.Q exhibits stronger transfer to unseen tasks than *Newt*, suggesting that representations learned through multitask training yield substantially better zero-shot initialization and faster adaptation during few-shot finetuning. Ablations further confirm that predictive objectives are critical, with performance degrading significantly when removed even at large model scales. Overall, these results support a representation-centric view of deep RL scaling, where the quality of learned representations plays a central role in enabling effective scalable multitask learning.

Contributions. We demonstrate that model-free RL with predictive representations can outperform planning-based world-model approaches across diverse multitask benchmarks, with improved sample and computational efficiency. Through controlled ablations, we disentangle the contributions of predictive objectives and online planning, showing that the former substantially improves performance while the latter provides limited additional benefits at higher computational cost. Finally, we show that predictive representation learning increases effective feature dimensionality and improves training dynamics, alongside effective scaling with model capacity and data availability, and strong transfer to unseen tasks.

2 Preliminaries

Problem setting. We consider a multitask RL (MTRL) setting in which tasks $\tau \sim p(\tau)$ are sampled from a task distribution. Each task induces a Markov decision process (MDP) $\mathcal{M}_\tau = (\mathcal{S}, \mathcal{A}, \mathcal{T}_\tau, \mathcal{R}_\tau, \gamma)$, where we assume a shared action space $\mathcal{A}$ and (typically) a shared state space $\mathcal{S}$ across tasks, while transition dynamics and rewards may vary with τ. At each time step t, the agent observes $s_t \in \mathcal{S}$, takes action $a_t \in \mathcal{A}$, and receives reward $r_t \sim \mathcal{R}_\tau(s_t, a_t)$, transitioning to $s_{t+1} \sim \mathcal{T}_\tau(\cdot \mid s_t, a_t)$. The objective is to learn a single policy $\pi(a \mid s, \tau)$ that maximizes the expected discounted return across tasks, formulated as $\mathbb{E}_{\tau \sim p(\tau), \pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. Similar to Hansen et al. [2026], when task information is available (e.g., task identifiers or language instructions), we condition the policy and value functions on a learned embedding $e(\tau)$. Otherwise, the problem reduces to a partially observable MDP, where task identity must be inferred from interaction. We assume an off-policy setting, where experience is stored in a replay buffer $\mathcal{D}$ containing tuples $(s_t, a_t, r_t, d_t, s_{t+1}, \tau)$, with $d_t \in \{0, 1\}$ indicating episode termination. We adopt an off-policy actor–critic architecture [Konda and Tsitsiklis, 1999, Fujimoto et al., 2018], where a parametric policy (actor) $\pi_\psi(a \mid s, \tau)$ is trained to maximize expected return, while a value function (critic) $Q_\theta(s, a, \tau)$ estimates the expected return of state–action pairs. The critic is optimized via temporal-difference (TD) learning using targets constructed from a slowly updated target network, while the actor is trained to maximize the critic's value estimates. In practice, we employ twin critics $Q_{\theta_1}, Q_{\theta_2}$ to mitigate overestimation bias, as in prior work on off-policy RL [Fujimoto et al., 2018, Haarnoja et al., 2018].

Predictive Information Representations. Representation learning is central to deep RL, particularly in high-dimensional and multitask settings where stability and generalization depend on the structure of learned features [Agarwal et al., 2021, Echchahed and Castro, 2025]. Because supervision from temporal-difference learning is often weak and non-stationary, predictive auxiliary objectives are commonly used to stabilize optimization and encourage latent representations to capture environment dynamics and temporal structure beyond reward signals [Nikishin et al., 2022, Hafner et al., 2020b, Hansen et al., 2024]. We consider an off-policy actor–critic operating on learned latent representations. Observations (and optionally task information) are encoded as $z_t = \phi_\xi(s_t, \tau)$, and both the policy $\pi_\psi(a \mid z)$ and twin critics $Q_{\theta_1}, Q_{\theta_2}$ operate in latent space. Critics are trained via temporal-difference learning with target networks, while the policy maximizes value estimates. To improve representation quality, we augment training with predictive modeling in latent space: models of dynamics, reward, and termination predict (z_{t+1}, r_t, d_t) from (z_t, a_t) [Fujimoto et al., 2025], and their gradients are backpropagated through the encoder ϕ_ξ. This encourages representations that are predictive of environment dynamics and task-relevant signals. Crucially, no planning is performed, the learned models are used solely to shape the representation, isolating the benefits of predictive learning without the computational overhead and instability of explicit model-based control.

3 Scaling deep RL through Representation Learning

A central challenge in deep RL is how to scale agents across tasks, model capacity, and data. Recent progress has been largely driven by model-based approaches, where agents learn predictive world models and leverage planning to improve decision-making [Hansen et al., 2024, Hafner et al., 2025a]. Methods such as Dreamer and TD-MPC2 demonstrate that combining predictive modeling with large-capacity function approximators can substantially improve performance in both single-task and multitask settings.

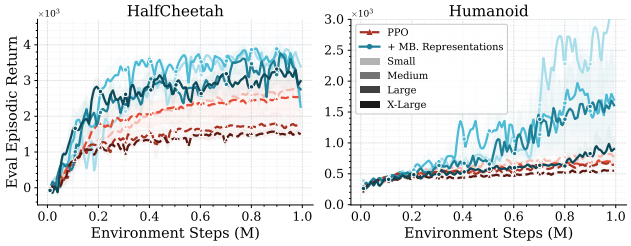

Figure 1: **Representation quality drives scaling performance in model-free RL.** We compare standard PPO with a variant augmented with model-based representations (+MB. Representations) across four network sizes (Small, Medium, Large, X-Large) on HalfCheetah and Humanoid.

At larger scales, systems such as *Newt* [Hansen et al., 2026] extend this paradigm to hundreds of tasks by training shared world models across diverse continuous-control domains, demonstrating strong multitask performance and transfer. However, these gains come with significant computational and algorithmic overhead. Model-based agents must jointly learn dynamics, reward, and value

functions while additionally performing latent rollouts or planning during training or inference. This increases wall-clock cost, memory usage, and implementation complexity, while also introducing additional sources of instability as model errors compound over imagined trajectories [Talvitie, 2014, Janner et al., 2019]. These challenges become particularly pronounced in multitask settings, where a single world model must capture diverse and potentially conflicting dynamics across environments. Additional discussion of related work is provided in App. C.

At the same time, recent work suggests that some benefits commonly attributed to model-based RL may instead arise from the representations induced by predictive learning [Schwarzer et al., 2021, Ghugare et al., 2023, Zhao et al., 2023]. In particular, methods such as MR.Q show that model-free agents augmented with auxiliary predictive objectives can achieve strong performance across diverse tasks without explicit planning [Fujimoto et al., 2025].

To isolate this effect, we study a controlled single-task setting where planning and multitask interference are absent. Fig. 1 provides evidence in a controlled setting where planning and multitask interference are absent. We compare standard PPO [Schulman et al., 2017] with a variant augmented with predictive model-based representations (+ MB. Representations) across four network sizes on HalfCheetah and Humanoid, two environments of increasing complexity and dimensionality. Without predictive representations, scaling model capacity yields little to no benefit: on HalfCheetah, larger PPO models can even underperform smaller ones, while on Humanoid, performance remains nearly flat across all model sizes. With predictive representations, however, PPO consistently outperforms standard PPO at every network size and offers increased robustness to varied capacity. These results hint that representation quality may be an important bottleneck when scaling deep RL systems. Predictive objectives provide an additional supervisory signal that appears to help larger models make more effective use of increased capacity, whereas reward-only supervision often struggles to do so.

This finding has direct implications for multitask RL, where scaling shared architectures is critical for learning transferable representations across diverse tasks [Nauman et al., 2025, Hansen et al., 2026], and where additional challenges — task interference, non-stationarity, and distributional shift — may make representation quality an even more severe bottleneck. This motivates the central question of this work: *Can model-free RL match the scalability and generalization of world-model approaches in multitask settings by focusing on representation learning alone?*

4 Multitask Model-Free RL with Structured Representations

In this section, we evaluate whether model-free RL augmented with predictive representation learning can match recent world-model approaches in multitask settings. We show that MR.Q consistently matches or surpasses the large-scale world-model baseline *Newt* across diverse multitask domains without relying on planning or latent rollouts. We further analyze how predictive representation learning impacts representation geometry and optimization stability. World models provide useful inductive biases through predictive supervision and structured latent representations [Ha and Schmidhuber, 2018, Hafner et al., 2020a, Gelada et al., 2019, Schwarzer et al., 2021]. However, many of their benefits may arise from the learned representations rather than planning itself. This motivates model-free approaches that incorporate predictive representation learning while preserving the simplicity, efficiency and scalability of model-free RL.

Baselines and Evaluation Protocol. We compare against a strong model-based baseline, *Newt* [Hansen et al., 2026], in a multitask setting under fixed interaction budgets. Our primary evaluation is conducted in a low-data regime of 10M environment steps, where sample efficiency is critical, in contrast to prior work that typically evaluates at 100M environment steps [Hansen et al., 2026]. To assess scalability, we additionally include selected longer runs. We report aggregate learning curves to evaluate sample efficiency, as well as final performance at the end of training, averaging results over five seeds and reporting 95% confidence intervals (CIs) across tasks and runs. Our results show that equipping TD3 [Fujimoto et al., 2018] with predictive representation learning objectives (MR.Q [Fujimoto et al., 2025]) enables model-free methods to match or surpass model-based approaches. All experiments follow the multitask language-conditioned training protocol introduced in *Newt* [Hansen et al., 2026]; see App. E and App. G for MR.Q and training details.

Learning Across Tasks. We consider a multitask setting where a single agent is trained jointly across a diverse set of environments that share observation and action spaces but differ in dynamics

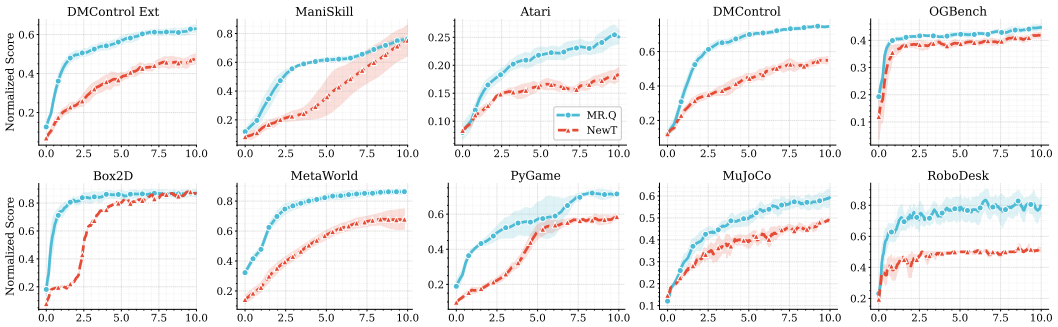

Figure 2: **Per-domain aggregate performance across all 10 MMBench domains.** Average normalized score of **MR.Q** (solid, teal) versus ***Newt*** (dashed, red) on state-based multitask benchmarks from MMBench [Hansen et al., 2026], spanning continuous control, manipulation, locomotion, and discrete game domains. MR.Q, a model-free agent with model-based representation learning, consistently matches or surpasses the model-based *Newt* baseline in sample efficiency and final performance across all domains. Shaded regions denote 95% CIs across five seeds.

and reward functions, following prior work on multitask RL [Hansen et al., 2026]. Training is performed by interleaving experience from multiple tasks under a shared set of parameters. This setup enables knowledge sharing across tasks, but introduces several challenges: *(i) non-stationarity*, as the data distribution shifts with the task mixture; *(ii) interference*, as shared representations must support multiple, potentially conflicting objectives; and *(iii) optimization difficulty*, as gradients from different tasks may not align. These challenges make representation learning a central bottleneck for scaling RL systems. These challenges are particularly relevant for evaluating predictive representation learning. If predictive objectives improve latent structure, temporal consistency, and feature reuse across tasks, they may alleviate optimization instability and reduce interference even in the absence of explicit planning. Fig. 2 shows that MR.Q consistently improves both sample efficiency and final performance across diverse multitask domains, suggesting that predictive representation learning alone can substantially improve cross-task generalization and optimization stability. Additional per-task learning curves are provided in App. O, and detailed descriptions of the multitask suites and training protocol are given in App. D. Unless otherwise specified, results are averaged over 5 seeds.

Task-Level Performance Analysis. While aggregate performance provides a summary of multitask learning, it can obscure substantial differences across individual tasks. A closer examination of the per-task results reveals that MR.Q achieves its largest improvements on families of tasks sharing common embodiments, dynamics, or control primitives. These include locomotion tasks in DMControl and Box2D, articulated-object interactions in RoboDesk, and reactive-control games in PyGame. In contrast, *Newt* remains competitive on several object-specific manipulation tasks in ManiSkill and navigation tasks in PyGame. These observations suggest that the benefits of predictive representation learning may be particularly pronounced when tasks share reusable structure, whereas the relative advantages of planning-based approaches may depend on more task-specific decision-making requirements. However, these trends are descriptive rather than causal, as MR.Q and *Newt* differ in several components beyond planning. A detailed per-task analysis, including representative successes and exceptions, is provided in Sec. O.1.

Training for Longer. While our primary evaluation focuses on the low-data regime, it is important to assess whether the observed gains persist at larger interaction budgets. To this end, we evaluate MR.Q in extended training settings, scaling up to 50M environment steps. This allows us to analyze asymptotic performance and determine whether improvements in representation learning continue to provide benefits beyond the initial learning phase. Fig. 3 shows that MR.Q maintains strong performance as the number of interactions increases, matching or

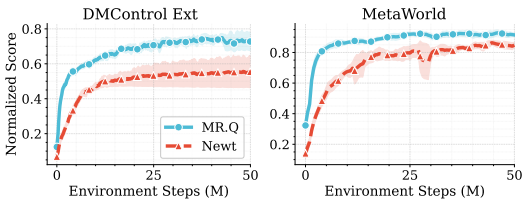

Figure 3: **Extended training performance (up to 50M environment steps).** MR.Q sustains strong performance at scale, surpassing *Newt*, indicating that gains from structured representations persist beyond the low-data regime.

surpassing model-based approaches such as *Newt* in multitask settings. This suggests that the advantages of structured representation learning are not limited to sample efficiency, but also translate to improved scalability; reinforcing the view that a model-free method equipped with structured representations provides a scalable alternative to model-based approaches, achieving strong performance across both low- and high-data regimes.

Visual Observations. While most experiments are conducted in the state-based setting, we additionally evaluate performance under high-dimensional visual inputs, following prior multitask benchmarks [Hansen et al., 2026]. We use a pretrained DINOv2 encoder [Oquab et al., 2024] to extract features from pixels, which are then used by the policy and value networks. This setup removes the burden of learning representations from scratch, allowing us to isolate the role of downstream representation adaptation. Despite strong pretrained features, representation learning remains a key bottleneck in the multitask regime. The agent must adapt shared embeddings across diverse tasks, introducing interference and instability in value learning. As shown in Fig. 4, MR.Q consistently outperforms *Newt* across all domains, achieving higher sample efficiency and final performance. These results highlight that the benefits of structured representation learning extend beyond low-dimensional settings. Even with powerful pretrained encoders, predictive objectives remain important for learning representations that support effective multitask learning under high-dimensional inputs.

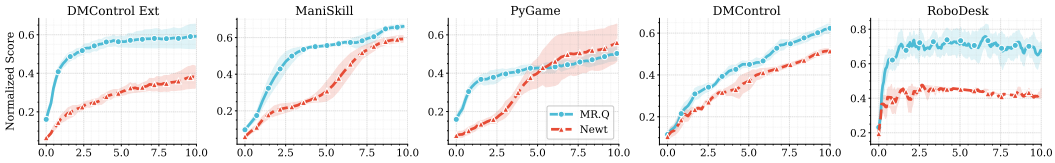

Figure 4: **Pixel-based multitask learning curves across five domains.** Average normalized score of **MR.Q**(solid) and ***Newt*** (dashed) using visual observations with a frozen DINOv2 encoder. MR.Q consistently achieves higher sample efficiency and final performance, demonstrating that its predictive auxiliary objectives yield better task-relevant representations in the high-dimensional input regime. Shaded regions denote 95% CIs across five seeds.

4.1 Analyses

To rigorously isolate the mechanisms driving performance and assess the structural integrity of the learned representations, we compare MR.Q against an encoder-free baseline (TD3) to isolate the impact of model-based representation learning. In this ablation, the encoder is removed entirely, and the actor receives a direct concatenation of the raw low-dimensional state and the 512-dimensional language instruction embedding as input, while the critic additionally receives the raw action.

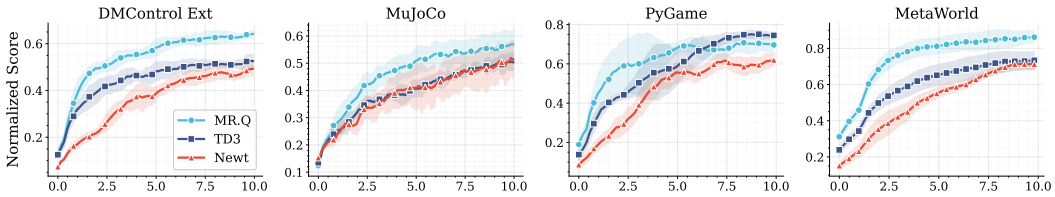

Figure 5: **Performance comparison across benchmark suites.** Per-domain aggregate performance for MR.Q, the encoder-free baseline (TD3), and *Newt* across four MMBench domains.

Performance Comparison. We evaluate the performance of MR.Q alongside the encoder-free baseline (TD3) and *Newt* as shown in Fig. 5. Our results demonstrate that MR.Q outperforms the encoder-free baseline in three out of four domains while achieving comparable results in the remaining one, demonstrating overall superior performance and sample efficiency. Interestingly, our results show that even the encoder-free baseline consistently matches or outperforms *Newt* across all four domains. This indicates that a well-tuned model-free architecture utilizing raw low-dimensional states and language instruction embeddings constitutes a highly competitive baseline while offering superior computational efficiency compared to a model-based RL approach.

Notably, this finding highlights the inherent robustness of model-free RL in multitask regimes, suggesting that explicit world-modeling may not be a strict prerequisite for handling multitask RL.

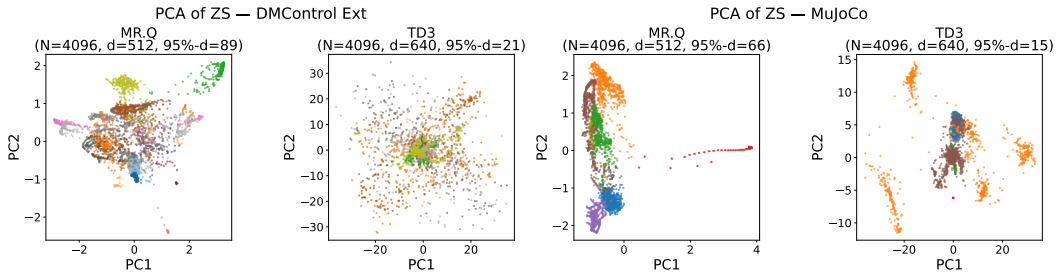

Figure 6: **PCA visualization of multitask latent representations.** Two-dimensional PCA projections of latent features extracted from multitask checkpoints trained on DMControl-Ext (left) and MuJoCo (right). Each point corresponds to an observation colored by task identity. MR.Q learns structured and well-separated task representations with substantially higher effective dimensionality (95%-d), whereas removing predictive representation learning leads to collapsed and lower-rank embeddings.

Beyond aggregate performance, the encoder-free baseline facilitates a diagnostic evaluation of how learned representations modulate underlying learning dynamics when compared against MR.Q. We analyze these effects across three key dimensions. These findings are summarized in Fig. 7 and discussed in detail below. Results averaged over five seeds, shaded areas represent 95% CIs.

Representation Geometry. We evaluate feature capacity by measuring the **SRank** [Kumar et al., 2021] of the state representations. While the encoder-free baseline TD3 uses a larger input dimensionality ($512 + d_{obs}$) than the 512-dimensional latent space of MR.Q, its SRank is significantly lower. This suggests that raw observations result in a redundant feature space when processed without specific inductive biases. In contrast, the representation learning in MR.Q enforces a high-rank manifold that has better representational capacity. We additionally perform Principal Component Analysis (PCA) on the latent features at the end of 10M training steps to quantify the variance distribution. We measure effective dimensionality by calculating the number of principal components required to explain 95% of the variance (95%-d). Consistent with the SRank collapse, removing the encoder causes a severe representational bottleneck: across the DMControl-ext and MuJoCo suites, the 95%-d drops from 89 and 66 down to merely 21 and 15, respectively. As depicted in Fig. 6, these results validate the necessity of representation learning in preserving the expressive capacity required to scale across diverse multitasks. Colors denote different tasks (12 tasks for DMControl-Ext and 6 tasks for MuJoCo); task labels are omitted in the main figure for readability, with fully annotated visualizations provided in App. M.

Training Dynamics To study how representation quality impacts optimization dynamics, we monitor the fraction of dormant neurons [Sokar et al., 2023, Liu et al., 2025b], which measures the proportion of inactive units in the network. Dormant neurons indicate underutilized capacity and reduced plasticity, both of which are particularly harmful in multitask settings where agents must continually adapt to diverse and shifting objectives. MR.Q consistently exhibits a substantially lower fraction of dormant units than the encoder-free baseline, especially in the critic network where the gap becomes pronounced throughout training. In contrast, removing predictive representation learning leads to widespread critic dormancy, suggesting that the critic fails to effectively utilize the available network capacity. This degradation is accompanied by higher value losses, indicating that collapsed or poorly structured representations make value learning significantly more difficult under multitask non-stationarity.

Overall, these results suggest that predictive representation learning not only improves representation geometry, but also preserves optimization stability and network plasticity during training [Mayor et al., 2025]. By maintaining expressive and active latent features, MR.Q enables the critic to make more effective use of model capacity, helping stabilize value learning across diverse multitask domains. However, competitive performance on fixed multitask benchmarks alone does not fully characterize scalability. In practice, scalable RL systems must continue to improve with increased task diversity, model capacity, data, and computation while remaining computationally efficient. In Sec. 5, we therefore investigate how model-free RL equipped with model-based representations behaves across these scaling axes in large multitask settings.

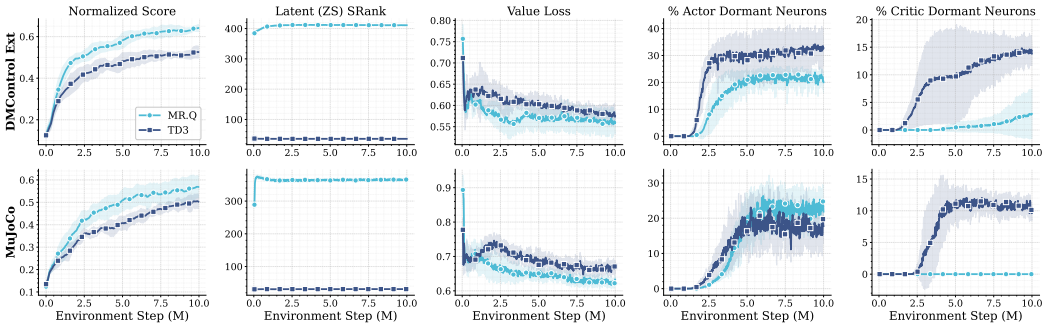

Figure 7: **Empirical analyses for the effect of representation learning.** Comparison of MR.Q against an encoder-free baseline (TD3). From left to right: aggregate return across task sets, state representation SRank, value loss, and dormant neuron fractions in the actor and critic.

5 Evaluation at Scale

A central question is whether model-free methods can scale as effectively as model-based approaches in multitask RL. We study this across multiple scaling axes, including task diversity, model capacity, data, update frequency, and computational efficiency.

Towards General Multitask RL Agents. To evaluate scalability, we train MR.Q on a large combined benchmark of 200 tasks spanning multiple domains in a unified setting. This setting stress-tests whether structured, model-free representations can scale to the diversity required of general-purpose multitask agents, where a single model must simultaneously solve locomotion, manipulation, navigation, and arcade tasks. Fig. 8 (left) reveals that MR.Q exhibits substantially higher *sample efficiency* throughout training: at 2M environment steps it achieves a normalized score of 0.11 versus 0.08 for *Newt* (+37% relative), and maintains a consistent lead of 5–8% across the range. This early advantage is practically significant in large-scale settings, where each additional interaction is costly. From a representation learning perspective, this suggests that model-free agents with structured latent spaces can match the representational expressiveness of world-model-based methods at scale, while requiring fewer environment interactions to do so.

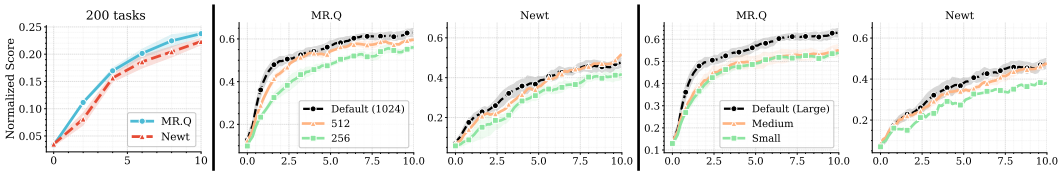

Figure 8: *(Left)* **Large-scale multitask training across 200 tasks.** Normalized score throughout training on a combined benchmark of tasks spanning multiple domains. MR.Q consistently outperforms *Newt* during training, while both methods converge to similar final performance. **Data and model scaling in multitask RL.** (*Middle*) Data scaling: performance as a function of training data for different dataset sizes. (*Right*) Model scaling: performance across model sizes. MR.Q exhibits stable scaling across both axes, while *Newt* shows sensitivity to reduced data and smaller models.

Model and Data Scaling. We analyze how multitask performance scales with data and model capacity. Fig. 8 (*middle*) shows performance as a function of available training data. Both methods improve with increased data, but exhibit different scaling behaviors. MR.Q shows consistent gains across data regimes, maintaining strong performance even with reduced data. In contrast, *Newt* is more sensitive to data availability, with larger performance degradation in low-data settings. Fig. 8 (*right*) shows scaling with model size. MR.Q exhibits smooth and predictable improvements as capacity increases, indicating effective utilization of additional parameters. *Newt*, however, shows weaker scaling, with smaller gains and higher sensitivity to model size. These results suggest that scaling performance is determined not only by access to more data or larger models, but also by how effectively additional capacity is utilized. MR.Q exhibits more stable scaling behavior across both axes, allowing it to better leverage increased data and model capacity.

Hyperparameter Sensitivity. We additionally examine the sensitivity of MR.Q to its main representation learning and architectural hyperparameters, including predictive loss weights, latent dimensionality, and encoder width. Across three multitask suites, MR.Q remains relatively robust to moderate hyperparameter variations, with the default configuration consistently achieving the highest aggregate performance. Reward prediction weight and encoder width exhibit relatively low sensitivity, while latent dynamics weight and latent dimensionality have a larger influence on performance. MR.Q's scaling behavior is not restricted to a narrowly tuned configuration. Detailed sensitivity analyses are provided in App. L.

Scaling with Update-to-Data Ratio. We analyze how performance scales as a function of the update-to-data (UTD) ratio, which controls the number of gradient updates performed per environment interaction. Increasing UTD effectively increases the amount of computation applied to a fixed dataset, probing how efficiently a method can extract information from available data. Fig. 11 shows performance as a function of environment steps for different UTD values. As shown in Fig. 11 (*left*), MR.Q benefits consistently from increasing UTD, with higher update regimes leading to improved performance across training. This indicates that the agent can effectively use additional gradient updates without destabilizing learning. In contrast, *Newt* (Fig. 11, *right*) exhibits weaker scaling; performance improves slowly and shows diminishing returns at higher UTD.

Few-shot finetuning. To evaluate transfer to unseen tasks, we hold out a set of 28 tasks spanning multiple domains and finetune each individually using online RL, initializing from a checkpoint trained for 10M environment steps on the remaining 200 tasks. We compare MR.Q against *Newt* under an identical finetuning budget of 200k steps. MR.Q provides a substantially stronger zero-shot initialization before any finetuning as shown in Fig. 9. MR.Q achieves an average normalized score of 0.13 versus 0.09 for *Newt*, a 50% relative advantage, indicating that multitask pretraining with MR.Q yields more transferable representations. This advantage is preserved throughout adaptation: at 100k steps MR.Q scores 0.55 versus 0.48 for *Newt* (+12.8%), and at 200k steps 0.62 versus 0.55 (+12.9%). At the individual task level, MR.Q outperforms *Newt* on 17 of 28 held-out tasks (61%) at the end of finetuning. These results suggest that MR.Q's structured, model-free representations learned during multitask pretraining transfer more effectively to novel tasks, enabling both a better starting point and faster convergence during adaptation.

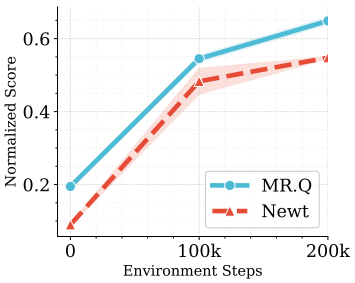

Figure 9: **Few-shot finetuning on held-out tasks.** Average normalized score across 28 unseen tasks during finetuning steps from a 10M-step multitask checkpoint. MR.Q achieves 50% higher zero-shot performance and ∼13% advantage throughout training.

Computational Impact. We evaluate wall-clock efficiency by measuring performance as a function of training time. While standard deep RL evaluations emphasize sample efficiency, methods with similar interaction budgets can differ substantially in time-to-performance. Model-based approaches incur additional overhead from learning dynamics models and performing latent rollouts, which slows down training despite their strong sample efficiency. In contrast, MR.Q avoids explicit planning and simulation, learning structured representations directly from data. As a result, MR.Q achieves faster improvement in performance per unit of time, reaching higher returns significantly earlier than model-based baselines, as shown in Fig. 10. This highlights that gains in sample efficiency for world-model approaches often come at the cost of increased computational overhead. These differences have important practical implications, as higher compute requirements translate into longer training times, increased energy consumption, and reduced accessibility.

6 Lessons and Opportunities

Scaling deep RL to large and diverse multitask settings remains a central challenge. In this work, we studied the role of model-based representations in the multitask RL setting and showed that a simple model-free approach augmented with predictive objectives can match or surpass a recent large-scale world-model baseline while substantially reducing computational overhead and improving wall-clock efficiency (see Fig. 10). Our analyses further demonstrate that predictive representation learning

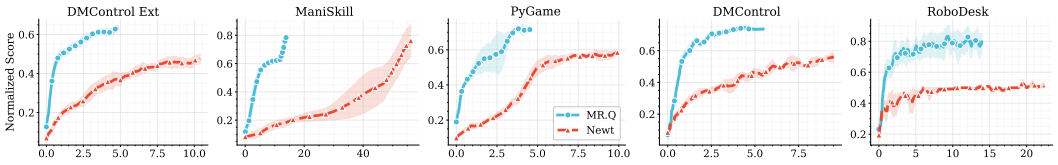

Figure 10: **Wall-clock efficiency.** Normalized score as a function of wall-clock training time (hours) on five MMBench domains. MR.Q consistently reaches higher returns earlier than *Newt*, a model-based baseline that incurs substantial overhead from world-model learning and latent rollout generation. Shaded regions denote 95% CIs. All runs use a fixed budget of 10M environment steps.

improves representation geometry, stabilizes optimization, and enables more effective utilization of model capacity as systems scale (Sec. 5).

More broadly, our results highlight the importance of developing sample- and compute-efficient multitask RL algorithms that can learn effectively across diverse tasks under a realistic interaction budget. In contrast to prior work that evaluates at substantially larger scales, our primary experiments are conducted in a challenging 10M interaction regime, where efficiency and representation quality become critical bottlenecks. This is particularly important in real-world settings, where interaction data is costly. Despite this restricted budget, our approach consistently matches or surpasses large-scale world-model baselines across multiple axes, including task diversity, model scaling, transfer, and wall-clock efficiency. These findings suggest that scalable multitask RL may depend not only on larger model size or interaction budgets, but also on learning effective representations.

Limitations and Future Work. Our study focuses primarily on continuous-control multitask benchmarks, and it remains unclear how these findings extend to more diverse domains such as long-horizon environments. In addition, although our analyses suggest that predictive objectives improve representation quality and scaling behavior, the mechanisms underlying these improvements are not yet fully understood. A more principled theoretical understanding could help guide the design of future sample-efficient multitask agents. An important direction for future work is to further investigate the relationship between predictive representation learning and planning. While MR.Q demonstrates that strong multitask performance can emerge without explicit planning, hybrid approaches that combine scalable model-free representation learning with latent planning or imagination-based rollouts may provide complementary benefits [Chang et al., 2026]. More broadly, understanding how representation learning, planning, and scaling interact in large multitask systems remains an important open problem for deep RL.

Acknowledgments

The authors would like to thank Sami Nur Islam, Walter Mayor Toro and Gopeshh Subbaraj for valuable discussions during the preparation of this work. We would also like to give special thanks to Ghada Sokar for providing valuable feedback on an early draft of the paper.

The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (https://alliancecan.ca) and Mila (https://mila.quebec). Aaron Courville acknowledges funding from the Institute of Information & Communications Technology Planning & Evaluation (IITP) through a grant funded by the Korean government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project). Pablo Samuel Castro acknowledges funding from NSERC Discovery Grant. We want to acknowledge funding support from Google and CIFAR AI. We would also like to thank the Python community [Van Rossum and Drake Jr, 1995, Oliphant, 2007] for developing tools that enabled this work, including NumPy [Harris et al., 2020], Matplotlib [Hunter, 2007], Jupyter [Kluyver et al., 2016], and Pandas [McKinney, 2013].

References

Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

Ankesh Anand, Jacob C Walker, Yazhe Li, Eszter Vértes, Julian Schrittwieser, Sherjil Ozair, Theophane Weber, and Jessica B Hamrick. Procedural generalization by planning with self-supervised world models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FmBegXJToY.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nIAxjsniDzg.

Fengshuo Bai, Hongming Zhang, Tianyang Tao, Zhiheng Wu, Yanna Wang, and Bo Xu. Picor: Multi-task deep reinforcement learning with policy correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6728–6736, 2023.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: an evaluation platform for general agents. *J. Artif. Int. Res.*, 47(1):253–279, May 2013. ISSN 1076-9757.

Avinandan Bose, Simon Shaolei Du, and Maryam Fazel. Offline multi-task transfer rl with representational penalization. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4420–4428. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/bose25a.html.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Vqj65VeDOu.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021.

Johan Samir Obando Ceron, João Guilherme Madeira Araújo, Aaron Courville, and Pablo Samuel Castro. On the consistency of hyper-parameter selection in value-based deep reinforcement learning. In *Reinforcement Learning Conference*, 2024a. URL https://openreview.net/forum?id=szUyvvwoZB.

Johan Samir Obando Ceron, Aaron Courville, and Pablo Samuel Castro. In value-based deep reinforcement learning, a pruned network is a good network. In *International Conference on Machine Learning*, pages 38495–38519. PMLR, 2024b.

Johan Samir Obando Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep rl. In *International Conference on Machine Learning*, pages 38520–38540. PMLR, 2024c.

Edoardo Cetin, Benjamin Paul Chamberlain, Michael M. Bronstein, and Jonathan J Hunt. Hyperbolic deep reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TfBHFLgv77.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. *arXiv preprint arXiv:2601.21306*, 2026.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 617–629. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/clavera18a.html.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgpv2VFvr.

Ayoub Echchahed and Pablo Samuel Castro. A survey of state representation learning for deep reinforcement learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=g0k34vUHtz. Survey Certification.

Jesse Farebrother and Pablo Samuel Castro. Cale: Continuous arcade learning environment. *Advances in Neural Information Processing Systems*, 37:134927–134946, 2024.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13049–13071. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/farebrother24a.html.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should i trust you, bellman? the bellman error is a poor replacement for value error. In *International Conference on Machine Learning*, pages 6918–6943. PMLR, 2022.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. *Advances in neural information processing systems*, 36:61573–61624, 2023.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=R1hIXdST22.

Chenxiao Gao, Haotian Sun, Na Li, Dale Schuurmans, and Bo Dai. Spectral representation-based reinforcement learning. *arXiv preprint arXiv:2512.15036*, 2025.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pages 2170–2179. PMLR, 2019.

Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. PWM: Policy learning with multi-task world models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hOELrZfg0J.

Raj Ghugare, Homanga Bharadhwaj, Benjamin Eysenbach, Sergey Levine, and Russ Salakhutdinov. Simplifying model-based RL: Learning representations, latent-space models, and policies with one objective. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MQcmfgRxf7a.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2020b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025a.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. *URL https://arxiv. org/abs/2509.24527*, 20, 2025b.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MPabX9LEds.

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8387–8406. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03): 90–95, 2007.

Haque Ishfaq, Thanh Nguyen-Tang, Songtao Feng, Raman Arora, Mengdi Wang, Ming Yin, and Doina Precup. Offline multitask representation learning for reinforcement learning. *Advances in Neural Information Processing Systems*, 37:70557–70616, 2024.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJ6yPD5xg.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5faf461eff3099671ad63c6f3f094f7f-Paper.pdf.

Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. Robodesk: A multi-task reinforcement learning benchmark, 2021.

Thomas Kluyver, Benjain Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. Jupyter Notebooks—a publishing format for reproducible computational workflows. In *IOS Press*, pages 87–90. 2016. doi: 10.3233/978-1-61499-649-1-87.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Yilun Kong, Guozheng Ma, Qi Zhao, Haoyu Wang, Li Shen, Xueqian Wang, and Dacheng Tao. Mastering massive multi-task reinforcement learning via mixture-of-expert decision transformer. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qUcUyqP1UA.

Jacob Eeuwe Kooi, Zhao Yang, and Vincent Francois-Lavet. Hadamax encoding: Elevating performance in model-free atari. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=iRQM8Ehgl9.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O9bnihsFfXU.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *Advances in Neural Information Processing Systems*, 33:11890–11901, 2020.

Jiashun Liu, Johan Samir Obando Ceron, Aaron Courville, and Ling Pan. Neuroplastic expansion in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=20qZK2T7fa.

Jiashun Liu, Zihao Wu, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Ling Pan. Measure gradients, not activations! enhancing neuronal activity in deep reinforcement learning. *arXiv preprint arXiv:2505.24061*, 2025b.

Walter Mayor, Johan Obando-Ceron, Aaron Courville, and Pablo Samuel Castro. The impact of on-policy parallelized data collection on deep reinforcement learning networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=cnqyzuZhSo.

Wes McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 1 edition, February 2013. ISBN 9789351100065. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1449319793.

Reginald McLean, Evangelos Chatzaroulas, J K Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Multi-task reinforcement learning enables parameter scaling. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=eBWwBIFV7T.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 37342–37364, 2024.

Michal Nauman, Marek Cygan, Carmelo Sferrazza, Aviral Kumar, and Pieter Abbeel. Bigger, regularized, categorical: High-capacity value functions are efficient multi-task learners. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=zhOUfuOIzA.

Tianwei Ni, Benjamin Eysenbach, Erfan SeyedSalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. In *The Twelfth International Conference on Learning Representations*, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pages 16828–16847. PMLR, 2022.

Johan Obando Ceron, Marc Bellemare, and Pablo Samuel Castro. Small batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36:26003–26024, 2023.

Johan Obando-Ceron, Walter Mayor, Samuel Lavoie, Scott Fujimoto, Aaron Courville, and Pablo Samuel Castro. Simplicial embeddings improve sample efficiency in actor–critic agents. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=mCpq1GCKxA.

Johan Obando-Ceron, Walter Mayor, Samuel Lavoie, Scott Fujimoto, Aaron Courville, and Pablo Samuel Castro. Simplicial embeddings improve sample efficiency in actor–critic agents. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=mCpq1GCKxA.

Johan S Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2021.

Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3): 10–20, 2007. doi: 10.1109/MCSE.2007.58.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ali Saheb Pasand, Johan Obando-Ceron, Aaron Courville, Pouya Bashivan, and Pablo Samuel Castro. Stable deep reinforcement learning via isotropic gaussian representations. *arXiv preprint arXiv:2602.19373*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. EPOpt: Learning robust neural network policies using model ensembles. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SyWvgP5el.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *The Nineth International Conference on Learning Representations (ICLR)*, 2021.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pages 30365–30380. PMLR, 2023.

Dmitry Shribak, Chen-Xiao Gao, Yitong Li, Chenjun Xiao, and Bo Dai. Diffusion spectral representation for reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 110028–110056, 2024.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International conference on machine learning*, pages 9767–9779. PMLR, 2021.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pages 32145–32168. PMLR, 2023.

Ghada Sokar, Johan Samir Obando Ceron, Aaron Courville, Hugo Larochelle, and Pablo Samuel Castro. Don't flatten, tokenize! unlocking the key to softmoe's efficacy in deep RL. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8oCrlOaYcc.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in Neural Information Processing Systems*, 36: 71242–71262, 2023.

Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G Bellemare. Investigating multi-task pretraining and generalization in reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sSt9fROSZRO.

Erik Talvitie. Model regularization for stable sample rollouts. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 780–789, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

Hongyao Tang and Glen Berseth. Improving deep reinforcement learning by reducing the chain effect of value and policy churn. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=cQoAgPBARc.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Claas A Voelcker, Victor Liao, Animesh Garg, and Amir massoud Farahmand. Value gradient weighted model-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=4-D6CZkRXxI.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22964–22984. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22u.html.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners, 2026. URL https://openreview.net/forum?id=MCWypEBtlF.

Yifan Xu, Nicklas Hansen, Zirui Wang, Yung-Chieh Chan, Hao Su, and Zhuowen Tu. On the feasibility of cross-task transfer with model-based reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KB1sc5pNKFv.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_SJ-_yyes8.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020a.

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*, pages 1094–1100. PMLR, 2020b.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=-2FCwDKRREu.

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 Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings*

of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of *Proceedings of Machine Learning Research*, pages 4015–4023. PMLR, 13–15 Apr 2021b. URL https://proceedings.mlr.press/v130/zhang21n.html.

Yi Zhao, Wenshuai Zhao, Rinu Boney, Juho Kannala, and Joni Pajarinen. Simplified temporal consistency reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42227–42246. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhao23k.html.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang.

\

texttt {TACO} : $Temporallatentaction-drivencontrastivelossforvisualreinforcementlearning.$*Advances in Neural* $48203 - -48225, 2023.$

Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé Iii, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-TACO is a few-shot policy learner: Pretraining multitask representation via temporal action-driven contrastive loss. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61413–61431. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zheng24g.html.

Yucheng Zhou, Jianbing Shen, and Yu Cheng. Weak to strong generalization for large language models with multi-capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=N1vYivuSKq.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

Appendix Contents

A The Use of Large Language Models **20**

B Impact statement **20**

C Related Work **20**

D Tasks Description **21**

E *MR.Q* algorithm: Model-based Representations for Q-learning **22**

E.1 Multitask MR.Q 23

F *Newt* algorithm: **24**

G Training Protocol **25**

H Multitask versus Single-Task Learning **26**

I Scaling with UTD **26**

J Predictive Objectives **26**

K Effect of Online Planning **27**

L Hyperparameter Sensitivity **28**

M PCA Analyses **29**

N Compute Resources **30**

O Per-tasks learning curves **30**

O.1 Task-Level Performance Analysis 30

P Finetuning: Per-tasks learning curves **40**

A The Use of Large Language Models

In this paper, LLMs were used only to polish the writing of certain paragraphs in order to improve clarity and grammar. The key ideas, theoretical analysis, method design, figures, and experimental results are entirely the result of the human authors' contributions.

B Impact statement

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

C Related Work

Representation Learning and World Models in RL. Representation learning is a central challenge in deep RL, where learned features must support value estimation, policy optimization, generalization, and stable learning under non-stationary data distributions. A large body of work studies how auxiliary objectives, contrastive learning, reconstruction, bisimulation metrics, and predictive modeling can improve learned representations [Gelada et al., 2019, Laskin et al., 2020, Yarats et al., 2021, 2022, Zhang et al., 2021a]. Recent analyses further show that poor representations can lead to feature collapse, dormant neurons, reduced plasticity, and unstable value learning [Kumar et al., 2021, Fujimoto et al., 2022, Nikishin et al., 2022, Sokar et al., 2023, Obando-Ceron et al., 2026b, Pasand et al., 2026].

Recent representation-based approaches have also investigated how learned dynamics-aware features can support control without requiring expensive model-based inference. Shribak et al. [2024] develop diffusion spectral representations for value-function learning, while Gao et al. [2025] investigate spectral representations of transition operators for reinforcement learning. These works provide complementary perspectives on learning representations that support effective control without relying on explicit trajectory optimization.

Predictive objectives are also central to modern world-model approaches. Methods such as PlaNet, Dreamer, DreamerV3, TD-MPC, and TD-MPC2 learn latent dynamics models that support imagined rollouts or latent trajectory optimization for control [Hafner et al., 2019, Łukasz Kaiser et al., 2020, Hafner et al., 2020a, 2025a, Hansen et al., 2022, 2024]. These methods demonstrate strong performance and scalability across continuous-control domains, while recent large-scale systems such as Dreamer 4 and Newt extend these principles to multitask settings [Hafner et al., 2025b, Hansen et al., 2026]. However, these approaches require jointly learning world models, value functions, and planning components, introducing substantial computational overhead and optimization complexity.

Model-Free RL with Predictive Representations. Several recent works suggest that predictive representation learning can improve RL even without explicit planning. Methods such as SPR [Schwarzer et al., 2021], BBF [Schwarzer et al., 2023], and MR.Q [Fujimoto et al., 2025] augment model-free RL with auxiliary predictive objectives that encourage temporal consistency and latent structure. Similar ideas have also been explored through self-predictive representations and latent dynamics supervision [Ni et al., 2024, Watter et al., 2015]. In these approaches, predictive models are used to shape the representation rather than to generate imagined rollouts or perform trajectory optimization.

Complementary approaches investigate action-conditioned contrastive representation learning. TACO [Zheng et al., 2023] learns temporal state–action representations through a contrastive objective, while Premier-TACO [Zheng et al., 2024] extends this approach to multitask representation pretraining for few-shot downstream policy learning. Unlike these contrastive approaches, our work investigates predictive latent dynamics and reward supervision during online multitask RL.

Our work builds on this line of research and studies whether predictive representation learning alone can recover many of the scalability and generalization benefits commonly associated with world-model methods. Unlike Dreamer, TD-MPC2, or Newt, our standard MR.Q agent does not use latent planning or imagination for policy improvement. Instead, predictive objectives are used

exclusively as auxiliary supervision for representation learning, allowing us to investigate their benefits independently of explicit model-based control.

Multitask Reinforcement Learning. Multitask RL aims to train a single agent across multiple environments while enabling transfer and representation sharing across tasks [Teh et al., 2017, Yu et al., 2020b, Sodhani et al., 2021]. Scaling RL to multitask settings introduces significant optimization challenges, including non-stationarity, gradient interference, negative transfer, and under-utilization of model capacity [Yu et al., 2020a, Taiga et al., 2023, Bai et al., 2023, Nauman et al., 2025]. Recent work further demonstrates that multitask training can benefit substantially from increased model capacity, particularly in the critic, while task diversity may help mitigate plasticity loss [McLean et al., 2025].

Theoretical studies have also investigated the benefits of shared representations for multitask transfer. Ishfaq et al. [2024] study offline multitask representation learning under low-rank MDP assumptions, establishing guarantees for downstream representation transfer. Similarly, Bose et al. [2025] investigate offline multitask transfer with representational penalization, addressing uncertainty arising from incomplete data coverage. Unlike these works, which focus on learning and transferring representations from offline data, our study investigates predictive representation learning during online multitask RL, with an emphasis on empirical scalability and computational efficiency.

Recent large-scale multitask systems such as TD-MPC2 and Newt demonstrate that world models can scale across diverse tasks and embodiments using shared architectures and task conditioning [Hansen et al., 2024, 2026]. Our work complements these approaches by showing that a model-free agent equipped with predictive representations can also achieve strong multitask performance, while avoiding the computational overhead of explicit planning.

We select *Newt* [Hansen et al., 2026] as our primary model-based baseline because it is specifically designed and evaluated for large-scale multitask reinforcement learning on MMBench. The original Newt study includes comparisons against established baselines such as PPO, FastTD3, and TD-MPC2. We therefore focus on a strong multitask world-model approach rather than attempting an exhaustive comparison against all model-based algorithms. To complement this comparison, we introduce Newt's online planning procedure directly into MR.Q, allowing us to evaluate the contribution of planning while retaining the same predictive representations and training objective.

D Tasks Description

For all experiments, we utilize the multitask suites introduced in MMBench [Hansen et al., 2026]. This benchmark encompasses 10 distinct domains and a total of 200 diverse continuous control tasks, spanning robotic manipulation, locomotion, navigation, arcade games, and classic control. A brief overview of each domain is provided below. Full task specifications and configuration details can be found in the original MMBench benchmark [Hansen et al., 2026].

MuJoCo. The MuJoCo [Todorov et al., 2012] serves as a standard benchmark for continuous control in reinforcement learning. It comprises a variety of simulated robotic locomotion tasks, ranging from lower-dimensional kinematic problems (e.g., *HalfCheetah*) to complex, high-dimensional control challenges involving severe contact dynamics (e.g. *Ant*). Following MMBench, we utilize the v4 environment configurations and disable early termination conditions to ensure consistency across all evaluated task domains.

DMControl and DMControl Extended. The DeepMind Control (DMControl) suite [Tassa et al., 2018] provides a standardized set of physics-based simulation environments, with a fixed episode length of 500 and no termination conditions. DMControl Extended is an extended task set based on the original DMControl, include 11 custom tasks previously proposed by Hansen et al. [2024].

MetaWorld. MetaWorld [Yu et al., 2020b] is a benchmark designed for multitask and meta-reinforcement learning, focusing exclusively on simulated robotic manipulation tasks. This domain consists of 50 diverse manipulation tasks that share a unified observation and action space. Due to a known simulation issue, the *Shelf Place* is excluded, yielding a final set of 49 tasks for this domain.

ManiSkill. ManiSkill3 [Tao et al., 2025] is a comprehensive physics-based benchmark focused on complex robotic control. This domain encompasses a diverse array of tasks and robotic morphologies, spanning tabletop manipulation, quadruped locomotion, whole-body humanoid control, and mobile manipulation. Additionally, it includes reimplementations of widely adopted control environments from the MuJoCo and DMControl suites.

Pygame. Pygame consists of 22 tasks spanning 14 unique arcade-style environment. These tasks exhibit significant heterogeneity in their core objectives, episode horizons, state-action dimensionalities, and underlying reward structures. MMBench enforce a fixed episode length across all tasks and disable early termination conditions.

Box2D. The Box2D suite [Brockman et al., 2016] utilizes a 2D physics engine to simulate rigid body dynamics. It encompasses a well-known set of classic control, navigation, and locomotion tasks, such as *LunarLander*. While the Box2D tasks were originally designed for low-dimensional state observations, MMBench modernizes the implementation by introducing support for high-dimensional visual observations.

RoboDesk. RoboDesk [Kannan et al., 2021] is a specialized suite of robotic manipulation tasks designed explicitly for multitask reinforcement learning research. The benchmark features 9 distinct object manipulation tasks situated within a single, unified desk-themed environment, where all tasks share a common observation and action space.

Atari. Based on the Arcade Learning Environment (ALE) [Bellemare et al., 2013], the Atari domain serves as a rigorous testbed for RL algorithms across a wide spectrum of simulated classic Atari 2600 games. More recently, Farebrother and Castro [2024] proposed a non-linear continuous-to-discrete action transformation that extends support to algorithms operating within continuous action spaces. MMBench utilizes this continuous variant of the Atari domain.

OGBench. OGBench [Park et al., 2025] is a benchmark tailored for evaluating goal-conditioned RL and offline RL. Because it was not originally designed for standard online RL, MMBench adapts these environments by introducing redefined dense reward functions and ensuring all necessary task information is fully integrated into the observation space (*e.g.* goal position).

E *MR.Q* algorithm: Model-based Representations for Q-learning

TD3 [Fujimoto et al., 2018] is a model-free off-policy actor–critic algorithm for continuous control that improves stability through twin critics, delayed policy updates, and target policy smoothing. In its standard form, TD3 operates directly on environment observations without learning an explicit latent representation encoder.

MR.Q [Fujimoto et al., 2025] extends TD3 by introducing a learned encoder together with auxiliary predictive objectives for representation learning. Observations are first encoded into a latent representation $z_t = \phi_\xi(s_t, \tau)$ using a learned encoder ϕ_ξ. The actor and twin critics then operate directly in latent space. In addition to standard temporal-difference learning, MR.Q trains auxiliary latent models to predict future latent representations, rewards, and termination signals from (z_t, a_t). The dynamics model predicts the next latent state $\hat{z}_{t+1}$, while auxiliary heads predict rewards $\hat{r}_t$ and episode termination $\hat{d}_t$. These objectives are optimized using supervised losses and backpropagated through the shared encoder.

For the MMBench experiments considered in this work, we employ latent dynamics and reward prediction objectives without an explicit termination prediction loss. The dynamics model predicts the next latent representation $\hat{z}_{t+1}$, while the reward prediction head estimates $\hat{r}_t$. Both objectives are optimized using supervised losses and contribute to learning the shared representation.

Importantly, the learned predictive models are used exclusively for representation learning. Unlike model-based RL methods such as Dreamer [Hafner et al., 2020a, 2025b], TD-MPC2 [Hansen et al., 2024], or Newt [Hansen et al., 2026], MR.Q does not use latent rollouts, trajectory imagination, or planning for policy improvement. Instead, predictive objectives provide additional supervision that

encourages representations to capture temporal structure while preserving model-free actor–critic control.

E.1 Multitask MR.Q

We extend MR.Q [Fujimoto et al., 2025] to multitask reinforcement learning by conditioning its state representations and policy on task information, while sharing the representation encoder, predictive models, actor, and critics across tasks. The underlying actor–critic optimization follows the original MR.Q algorithm.

Let τ denote a task, e_τ its frozen language embedding, s_t the observation, and a_t the action. We use $\oplus$ to denote concatenation and $\mathrm{sg}(\cdot)$ to denote stop-gradient.

Task-Conditioned Representations. The state encoder receives both the observation and task descriptor, while actions are encoded separately. The corresponding representations are

$$\begin{aligned} z_t^s &= f_{\mathrm{enc}}(s_t \oplus e_\tau), \\ z_t^a &= f_{\mathrm{za}}(a_t), \\ z_t^{sa} &= f_{\mathrm{zsa}}(z_t^s \oplus z_t^a). \end{aligned} \tag{1}$$

Here, z_t^s denotes the task-conditioned state representation, z_t^a the action representation, and z_t^{sa} the joint state–action representation. The latter is used by the predictive model and value functions. Importantly, the task embedding is explicitly provided to the state encoder and actor, but not separately concatenated to the critic or predictive heads. These components receive task information indirectly through z_t^s.

Predictive Representation Learning. MR.Q learns predictive representations through multi-step latent dynamics and reward prediction. Given the state–action representation, a shared prediction model produces

$$(\hat{z}_{t+k+1}^s, \hat{r}_{t+k}) = f_{\mathrm{dyn}}(\hat{z}_{t+k}^{sa}), \tag{2}$$

where the initial predicted state representation is $\hat{z}_t^s = z_t^s$. For subsequent prediction steps, the state–action representation is constructed as

$$\hat{z}_{t+k}^{sa} = f_{\mathrm{zsa}}\left(\hat{z}_{t+k}^s \oplus f_{\mathrm{za}}(a_{t+k})\right). \tag{3}$$

The target representation is computed using a target state encoder,

$$z_{t+k+1}^{s,-} = f_{\mathrm{enc}}^-(s_{t+k+1} \oplus e_\tau). \tag{4}$$

The predictive representation objective is

$$\mathcal{L}_e = \sum_{k=0}^{H-1} \left(\lambda_d \mathcal{D}_k + \lambda_r \mathcal{C}_k\right), \tag{5}$$

where H denotes the prediction horizon, λ_d and λ_r are the dynamics and reward loss weights, respectively, and

$$\begin{aligned} \mathcal{D}_k &= \left\| \hat{z}_{t+k+1}^s - \mathrm{sg}\left(z_{t+k+1}^{s,-}\right) \right\|_2^2, \\ \mathcal{C}_k &= \mathrm{CE}(\hat{r}_{t+k}, r_{t+k}). \end{aligned} \tag{6}$$

Here, $\mathrm{CE}(\cdot,\cdot)$ denotes the soft cross-entropy loss used for two-hot reward targets. The dynamics loss encourages the predicted latent states to match the target representations, while reward prediction provides additional supervision for learning task-relevant features. For the MMBench experiments, we use latent dynamics and reward prediction without an explicit termination prediction loss.

Critic Optimization. The critic is trained using clipped double Q-learning with target-policy smoothing. For the target state representation at time $t+H$, the smoothed target action is

$$a_{t+H}^- = \tanh\left(f_\pi^-(z_{t+H}^{s,-} \oplus e_\tau) + \epsilon\right), \tag{7}$$

where

$$\epsilon \sim \mathrm{clip}\left(\mathcal{N}(0,\sigma), -c, c\right). \tag{8}$$

The corresponding target state–action representation is

$$z^{sa,-}_{t+H} = f_{\text{zsa}}\left(z^{s,-}_{t+H} \oplus f_{\text{za}}(a^{-}_{t+H})\right). \tag{9}$$

The target value is obtained from the minimum of the target critics,

$$Q_{\text{targ}} = \min_{i\in\{1,2\}} Q^{-}_{i}\,(z^{sa,-}_{t+H}). \tag{10}$$

Using the H-step return, the critic target is

$$R^{(H)}_{t} = \sum_{k=0}^{H-1} \gamma^{k} r_{t+k} + \gamma^{H} Q_{\text{targ}}. \tag{11}$$

The value functions are trained by minimizing the average Huber loss,

$$\mathcal{L}_{v} = \frac{1}{N_q} \sum_{i=1}^{N_q} \mathcal{H}_{i}, \tag{12}$$

where

$$\mathcal{H}_{i} = \text{Huber}\left(Q_{i}(z^{sa}_{t}), R^{(H)}_{t}\right), \tag{13}$$

and N_q denotes the number of value functions.

Actor Optimization. The actor is explicitly conditioned on both the task-conditioned state representation and the frozen task embedding:

$$a_{t} = \tanh\left(f_{\pi}(\text{sg}(z^{s}_{t}) \oplus e_{\tau})\right). \tag{14}$$

The actor maximizes the corresponding critic value, together with the pre-activation regularization used in the original MR.Q algorithm [Fujimoto et al., 2025].

Optimization Procedure. The predictive representation, critic, and actor objectives are optimized sequentially, with the remaining modules held fixed during each corresponding update. This preserves the optimization structure of MR.Q while introducing explicit task conditioning into its state representations and policy.

The standard MR.Q agent uses its predictive models exclusively for representation learning, rather than planning or imagined policy improvement. We separately investigate the effect of introducing online planning in App. K.

F *Newt* algorithm:

Newt [Hansen et al., 2026] builds upon TD-MPC2 [Hansen et al., 2024], a model-based RL framework that combines latent world models with trajectory optimization for control. The central idea is to learn a compact latent dynamics model that supports both value estimation and planning directly in latent space.

Given an observation s_t, TD-MPC2 first encodes it into a latent representation

$$z_{t} = h_{\theta}(s_{t}),$$

where h_θ is a learned encoder. A latent dynamics model then predicts future latent states conditioned on actions:

$$\hat{z}_{t+1} = f_{\theta}(z_{t}, a_{t}).$$

Additional prediction heads estimate rewards and state values:

$$\hat{r}_{t} = r_{\theta}(z_{t}, a_{t}), \qquad \hat{V}_{t} = V_{\theta}(z_{t}).$$

The world model is trained using supervised consistency objectives across imagined latent rollouts. TD-MPC2 optimizes a multi-step latent prediction objective of the form

$$\mathcal{L}_{\text{model}} = \sum_{t,k} \Big(\|z_{t+k} - \hat{z}_{t+k}\|^{2} + \|r_{t+k} - \hat{r}_{t+k}\|^{2} + \|V_{t+k} - \hat{V}_{t+k}\|^{2} \Big),$$

where latent states are recursively imagined through the learned dynamics model. Unlike reconstruction-based world models, TD-MPC2 operates entirely in latent space without pixel reconstruction, improving scalability and computational efficiency.

A key difference from standard actor–critic methods is that TD-MPC2 performs explicit planning using the learned latent model. At decision time, candidate action sequences $a_{t:t+H}$ are optimized using model predictive control (MPC) by maximizing predicted future returns over imagined latent trajectories:

$$\max_{a_{t:t+H}} \sum_{k=0}^{H} \gamma^k \hat{r}_{t+k} + \gamma^{H+1} \hat{V}_{t+H+1}.$$

This planning procedure repeatedly rolls out trajectories inside the learned world model and selects actions according to the highest predicted return.

Newt extends these principles to massively multitask settings by training a single language-conditioned world model jointly across hundreds of tasks and embodiments. The resulting system jointly optimizes latent dynamics learning, value estimation, reward prediction, policy learning, and trajectory optimization within a shared multitask architecture.

In contrast, MR.Q [Fujimoto et al., 2025] uses predictive latent modeling exclusively for representation learning rather than planning. Similar to TD-MPC2, observations are encoded into latent representations and auxiliary models predict future latent states, rewards, and termination signals. However, MR.Q does not perform latent rollouts for control or trajectory optimization. The predictive objectives are instead used solely as auxiliary supervision to shape the latent representation:

$$\mathcal{L}_{\text{MR.Q}} = \mathcal{L}_{\text{TD}} + \lambda \mathcal{L}_{\text{predictive}},$$

where $\mathcal{L}_{\text{predictive}}$ includes latent dynamics, reward, and termination prediction losses. Policy improvement remains entirely model-free and is performed through standard actor–critic optimization rather than planning.

G Training Protocol

For all experiments, we follow the multitask language-conditioned training protocol introduced in MMBench [Hansen et al., 2026]. A single shared agent is trained jointly across tasks spanning multiple domains and embodiments using a unified multitask architecture. Task identity is provided through language instruction embeddings [Radford et al., 2021], allowing the policy and value functions to condition behavior on the current task while sharing representations across environments. Following the official benchmark implementation, language embeddings are concatenated with state or latent features and used as additional conditioning signals throughout training.

Training is performed in an off-policy setting using replay buffers that store transitions collected across all tasks. During training, minibatches are sampled uniformly from the shared replay buffer and used to jointly optimize the actor, critics, and auxiliary predictive objectives. Unless otherwise specified, all results are averaged over five random seeds.

For visual-observation experiments, we follow prior work [Hansen et al., 2026, Oquab et al., 2024] and use the same frozen pretrained DINOv2 encoder [Oquab et al., 2024] for both MR.Q and *Newt*. This ensures a common visual feature extractor across methods, allowing us to compare their downstream representation learning and control mechanisms without introducing differences in visual encoder pretraining. Consequently, these experiments do not evaluate end-to-end visual representation learning, which remains outside the scope of this study.

Our primary evaluation focuses on a challenging low-data regime of 10M environment interactions, substantially smaller than the budgets commonly used in prior large-scale multitask world-model systems. Additional experiments investigate extended training, model scaling, transfer, and update-to-data (UTD) scaling. The controlled planning and hyperparameter sensitivity experiments reported in App. K and App. L, respectively, are evaluated up to 11M environment interactions. Evaluation follows the normalized-score protocol introduced in MMBench, aggregating performance across tasks within each benchmark suite.

H Multitask versus Single-Task Learning

To investigate whether joint multitask training provides benefits over independent task-specific learning, we compare multitask MR.Q against separately trained single-task MR.Q agents under a matched total environment interaction budget. We conduct experiments on DMControl, DMControl-Ext, and RoboDesk, using five random seeds. For each suite, the multitask agent is trained jointly across its constituent tasks, while separate single-task agents are trained independently. The total number of environment interactions allocated across the single-task agents matches the interaction budget of the multitask agent. This comparison evaluates whether sharing parameters and representations across tasks improves learning efficiency under the same aggregate interaction budget.

Results. Across the three suites, multitask MR.Q achieves an aggregate normalized score of 0.76, compared to 0.69 for independently trained single-task agents, representing a 10.1% relative improvement. Multitask training also exhibits faster learning throughout training, indicating that its advantages are not restricted to final performance. All these results suggest that shared representation learning enables beneficial transfer across tasks, despite the potential interference introduced by joint optimization. Together with the held-out finetuning experiments, they provide evidence that multitask training can improve both learning on the training tasks and adaptation to previously unseen environments.

I Scaling with UTD

The update-to-data ratio (UTD) controls the number of gradient updates performed per environment interaction and serves as an important scaling axis for evaluating data reuse efficiency. Increasing UTD effectively increases the amount of optimization performed on a fixed dataset, testing whether a method can efficiently extract information from available experience without destabilizing learning.

Fig. 11 compares scaling behavior across different UTD values. MR.Q consistently benefits from larger UTD regimes, achieving improved performance as additional gradient updates are applied per interaction. In contrast, *Newt* exhibits weaker gains and greater sensitivity to increased update frequency. These results suggest that predictive representation learning enables more stable and effective reuse of replay data, allowing model-free methods to better exploit additional computation under fixed interaction budgets.

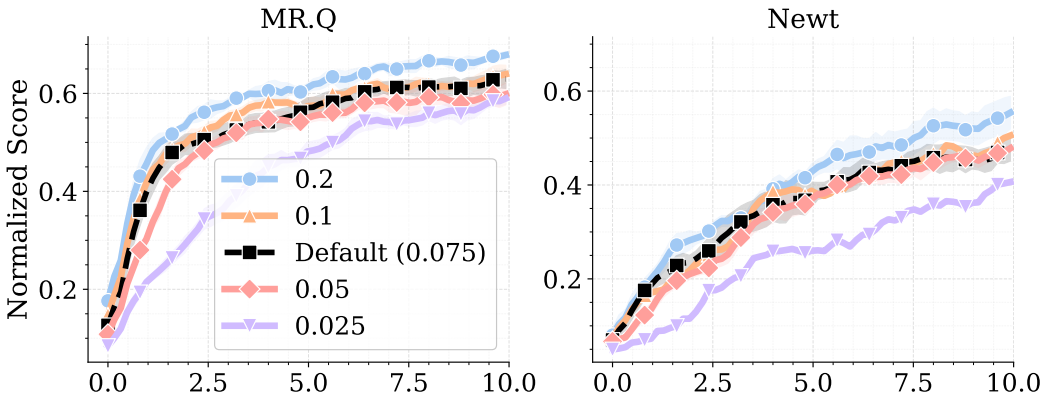

Figure 11: **Scaling with UTD.** Normalized score across five multitask suites. MR.Q benefits more from higher UTD than *Newt*, better data reuse.

J Predictive Objectives

To disentangle the contributions of the individual predictive objectives used by MR.Q, we conduct controlled ablations of latent dynamics prediction and reward prediction. We evaluate three variants: removing latent dynamics prediction, removing reward prediction, and removing both objectives simultaneously. All other architectural components and optimization hyperparameters are held fixed.

Experiments are conducted on three multitask suites (DMControl, DMControl-Ext, and RoboDesk), using five random seeds. We report aggregate performance across the three suites, together with the relative degradation compared to the full MR.Q agent.

Table 1: **Contribution of predictive objectives to multitask performance.** Aggregate performance across DMControl, DMControl-Ext, and RoboDesk, evaluated over five random seeds. Relative degradation is measured with respect to the full MR.Q agent.

Method	Score ↑	Degradation ↓
MR.Q (Full)	**0.684**	–
w/o Dynamics Prediction	0.525	23.2%
w/o Reward Prediction	0.614	10.2%
w/o Both	0.391	42.8%

As shown in Tab. 1, both predictive objectives contribute to the performance of MR.Q. Removing latent dynamics prediction results in a 23.2% relative degradation, more than twice the degradation observed when removing reward prediction (10.2%). Removing both objectives leads to the largest performance reduction (42.8%).

These findings identify latent dynamics prediction as the more influential individual auxiliary objective in the evaluated settings, while reward prediction provides additional complementary supervision. The substantial degradation observed when removing both objectives further supports the importance of predictive representation learning for effective multitask control.

K Effect of Online Planning

To directly examine the contribution of explicit planning, we augment MR.Q with the model predictive control (MPC) planner used by *Newt* [Hansen et al., 2026]. Specifically, we apply population-based trajectory optimization with iterative refinement using MR.Q's learned latent dynamics and reward models. The planning and non-planning variants share the same representation learning objectives, latent dynamics model, policy architecture, optimizer, and training procedure. The intervention is the use of MPC during action selection.

Experimental Setup. We evaluate both variants on DMControl, DMControl-Ext, and RoboDesk, using five random seeds per configuration. To avoid relying on a single planning configuration, we perform independent sweeps over the planning horizon and planning budget.

We consider planning horizons of $H \in \{3, 5, 7, 10\}$ and three planning-budget presets (small, default, and large). The planning-budget presets jointly vary the population size, number of elites, policy-seeded trajectories, and number of refinement iterations. All other hyperparameters are kept fixed. For each suite, we report the best-performing planning configuration identified by the sweep. We evaluate aggregate performance at 1M, 3M, 6M, and 10M environment steps, together with the total wall-clock training time under the same hardware configuration.

Table 2: **Effect of online planning on DMControl-Ext.** Aggregate normalized performance over five seeds at different environment interaction budgets. Wall-clock time denotes total training duration.

Method	1M	3M	6M	10M	Time
MR.Q	0.421 ± 0.024	0.543 ± 0.018	0.594 ± 0.017	0.668 ± 0.022	5h 21m
MR.Q + Planning	0.365 ± 0.027	0.521 ± 0.022	0.562 ± 0.021	0.599 ± 0.026	14h 07m

Table 3: **Effect of online planning on DMControl.** Aggregate normalized performance over five seeds at different environment interaction budgets. Wall-clock time denotes total training duration.

Method	1M	3M	6M	10M	Time
MR.Q	0.450 ± 0.010	0.640 ± 0.025	0.740 ± 0.020	0.760 ± 0.018	5h 25m
MR.Q + Planning	0.490 ± 0.032	0.630 ± 0.028	0.680 ± 0.024	0.710 ± 0.022	15h 12m

Table 4: **Effect of online planning on RoboDesk.** Aggregate success rate over five seeds at different environment interaction budgets. Wall-clock time denotes total training duration.

Method	1M	3M	6M	10M	Time
MR.Q	0.580 ± 0.035	0.720 ± 0.028	0.740 ± 0.025	0.770 ± 0.022	16h 57m
MR.Q + Planning	0.540 ± 0.038	0.660 ± 0.032	0.710 ± 0.028	0.700 ± 0.030	24h 00m

Results. Across all three multitask suites, MR.Q without planning achieves higher final aggregate performance than its planning-based counterpart. On DMControl-Ext, introducing MPC reduces the final normalized score from 0.668 to 0.599, while increasing wall-clock training time by $2.64\times$. A similar trend emerges on DMControl, where final performance decreases from 0.76 to 0.71, accompanied by a $2.80\times$ increase in training time. On RoboDesk, MR.Q without planning achieves a final success rate of 0.77, compared to 0.70 with planning, while the latter requires $1.41\times$ more wall-clock time.

Interestingly, planning provides a modest early advantage on DMControl at 1M environment steps (0.49 versus 0.45), but this advantage disappears with further training. Across the evaluated suites, online planning therefore provides no consistent improvement in sample efficiency or final aggregate performance, despite its substantially higher computational cost.

These findings suggest that MR.Q's predictive representations already provide a strong foundation for effective model-free control, with explicit planning offering limited additional benefits in the evaluated settings. Importantly, this does not imply that planning is ineffective in general, as its benefits may depend on the underlying agent, task characteristics, and planning procedure. An important direction for future work is therefore to investigate when and how planning can be leveraged to achieve meaningful performance improvements that justify its additional computational cost.

L Hyperparameter Sensitivity

To investigate the robustness of MR.Q to its representation learning and architectural hyperparameters, we conduct independent one-dimensional sweeps over four parameters: the latent dynamics prediction loss weight, reward prediction loss weight, latent dimension, and encoder width. All experiments are conducted on DMControl, DMControl-Ext, and RoboDesk using five random seeds and a training budget of 10M environment steps. For each sweep, we vary one parameter while keeping all remaining hyperparameters fixed at their default values.

Table 5: **Hyperparameter sensitivity.** Default settings and values explored in independent one-dimensional sweeps.

Hyperparameter	Default	Values explored
Dynamics loss weight	1.0	0.1, 0.5, 1.0, 2.0
Reward loss weight	0.1	0.05, 0.1, 0.2, 0.4
Latent dimension	512	256, 384, 512, 768
Encoder width	1024	512, 768, 1024, 1536

The default configuration corresponds to the main experiments and is reused in the comparisons rather than being independently rerun for each sweep. Across all four sweeps, the default configuration achieves the highest aggregate performance. Reward prediction weight and encoder width exhibit relatively low sensitivity, with alternative configurations remaining within approximately 2–4% of the default performance. Latent dynamics prediction weight and latent dimensionality have a larger influence, with the weakest configurations exhibiting approximately 8–13% lower aggregate performance relative to the default. These results suggest that MR.Q is reasonably robust to moderate hyperparameter variations, although its performance remains sensitive to the strength of latent dynamics supervision and the dimensionality of its learned representations.

M PCA Analyses

To further analyze the geometry of the learned multitask representations, we visualize latent features using Principal Component Analysis (PCA). We project latent representations extracted from trained checkpoints onto their top two principal components and color points according to task identity.

Across both DMControl-Ext and MuJoCo suites, MR.Q learns substantially more structured and separated latent representations than the encoder-free baseline. Predictive representation learning produces higher-rank embeddings with improved task separation and greater effective dimensionality, whereas removing representation learning leads to collapsed feature spaces with substantially reduced variance across dimensions. These results complement the quantitative analyses presented in the main paper. Together with the SRank measurements and dormant-neuron analyses, the PCA visualizations suggest that predictive auxiliary objectives improve representation diversity and preserve expressive capacity in large multitask settings.

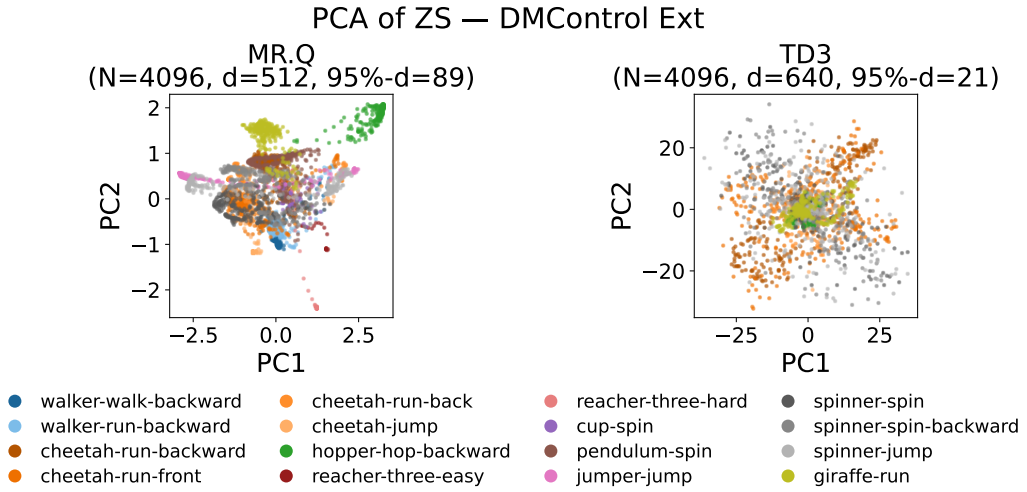

Figure 12: **PCA visualization on DMControl-Ext.** Two-dimensional PCA projections of multitask latent representations learned by MR.Q and the encoder-free baseline (TD3). Predictive representation learning produces substantially more structured and separated task representations.

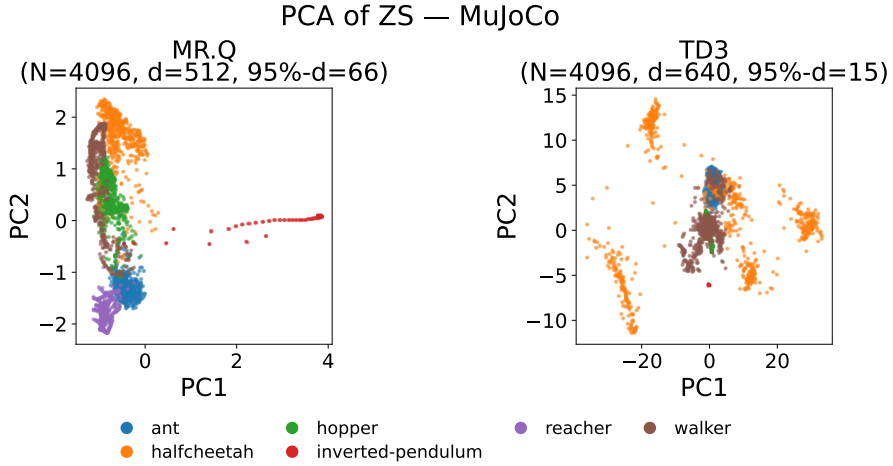

Figure 13: **PCA visualization on MuJoCo.** Latent representations learned by MR.Q exhibit higher diversity and improved task separation compared to the encoder-free baseline (TD3), indicating more expressive multitask representations.

N Compute Resources

All experiments were conducted on NVIDIA A100 GPUs using distributed Slurm-based compute clusters. Most multitask experiments were trained on a single GPU with approximately 24–48 GB of memory. Depending on the benchmark and model size, training required approximately 12–60 hours per run. Results are averaged over five seeds.

Beyond reducing training time, the computational efficiency of model-free agents equipped with predictive model-based representations has practical implications for how multitask RL systems are developed and studied. By avoiding explicit planning and latent rollout generation, our approach lowers the cost of experimentation and enables faster iteration cycles during development and finetuning. This can make large-scale multitask RL more accessible under limited compute budgets [Obando-Ceron and Castro, 2021], allowing researchers to explore architectures [Ceron et al., 2024c,b, Sokar et al., 2025, Liu et al., 2025a, Kooi et al., 2026], hyperparameters [Andrychowicz et al., 2021, Obando Ceron et al., 2023, Ceron et al., 2024a], and adaptation strategies without repeatedly incurring the cost of expensive model-based training pipelines.

These efficiency gains may create opportunities to scale multitask RL beyond the model sizes and experimental regimes commonly explored today. Since additional compute is not spent on planning procedures, resources can instead be allocated toward larger networks, broader task distributions, or more extensive scaling studies.

O Per-tasks learning curves

In addition to the aggregate results presented in the main paper, we provide per-task learning curves for all benchmark suites. These plots offer a more fine-grained view of training dynamics across individual environments.

O.1 Task-Level Performance Analysis

While aggregate results characterize overall multitask performance, they can obscure differences across task families and individual environments. We therefore examine the per-task learning curves to identify where MR.Q achieves its largest improvements and where *Newt* remains competitive. Rather than uniform gains across all tasks, several consistent patterns emerge.

DMControl and DMControl-Ext. MR.Q achieves particularly strong improvements on coordinated locomotion and continuous-control tasks, including Walker, Quadruped, Cheetah, Hopper, Jumper, and Finger. Across these environments, MR.Q generally learns faster and often reaches higher final performance. In contrast, both methods achieve comparable results on several simpler balancing, reaching, and periodic-control tasks, including Cartpole Balance, Reacher, Pendulum Spin, and Spinner Spin.

Box2D. MR.Q learns the Bipedal Walker terrain variants substantially faster, although *Newt* approaches comparable final performance on several variants. One notable exception is LunarLander Land, where *Newt* achieves relatively stronger performance. This suggests that the advantages of MR.Q are not uniform even within the same multitask domain.

ManiSkill. The results exhibit greater variability across manipulation tasks. MR.Q performs particularly well on tasks involving reaching, pushing, peg insertion, locomotion, cube manipulation, and several pick-and-place variants. In contrast, *Newt* becomes relatively more competitive on a subset of object-specific pick-and-lift tasks involving apples, bananas, mugs, oranges, and tennis balls. However, this does not constitute a general advantage for model-based control in manipulation, as MR.Q remains stronger on several other tasks, including Pick Cube, Pick Sponge, Lift Peg, and Stack Cube.

RoboDesk. MR.Q achieves substantial improvements on articulated-object interaction tasks, including Open Drawer, Open Slide, and Flat Block in Bin. Both methods exhibit more comparable performance on simpler color-conditioned pushing tasks.

PyGame. MR.Q performs particularly well on reactive-control games, including Air Hockey, Bird Attack, Chase Evade, Landing, Pong, and Spaceship. Conversely, *Newt* remains relatively competitive on several PointMaze navigation variants and other temporally structured tasks.

Discussion. These differences are not explained solely by reward sparsity, as MR.Q performs strongly on several sparse-reward tasks, while *Newt* remains competitive across both dense- and sparse-reward settings. Instead, the observed trends appear more closely associated with the degree of shared structure across tasks. MR.Q exhibits its largest advantages on task families with reusable dynamics, embodiments, or control primitives, consistent with the potential benefits of predictive representation learning for transferring information across related environments.

In contrast, *Newt* remains competitive on a smaller subset of tasks involving more specialized manipulation or navigation behavior. These observations should not be interpreted as establishing a causal relationship between task characteristics and algorithmic performance, since the two methods differ in multiple architectural and optimization components. Nevertheless, they highlight important differences that are not captured by aggregate performance and motivate further investigation of when predictive representations and explicit planning provide complementary advantages.

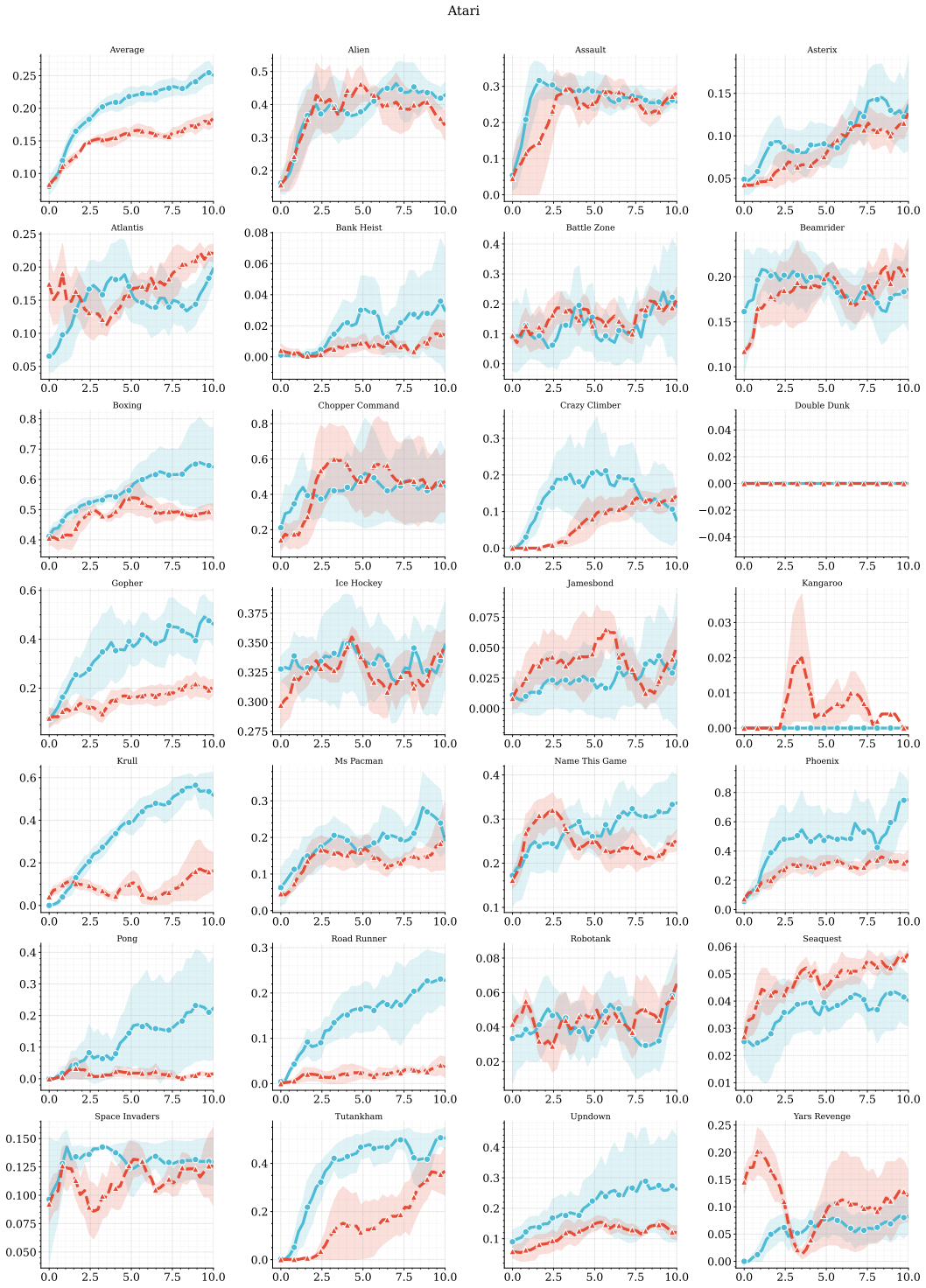

Figure 14: **Atari per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across Atari tasks. Shaded regions denote 95% confidence intervals (CIs).

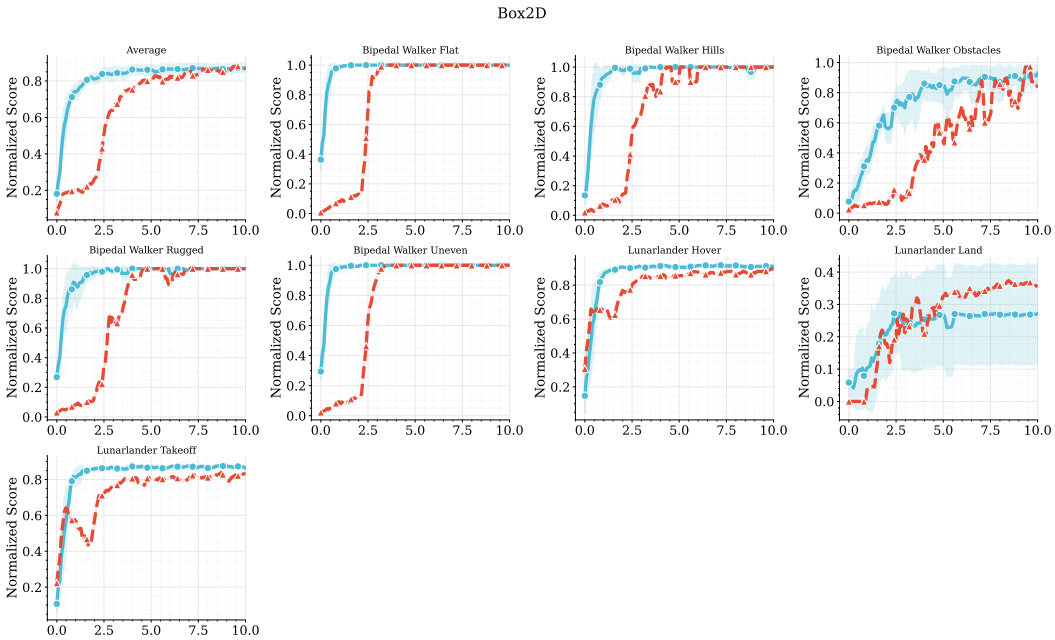

Figure 15: **Box2D per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across Box2D tasks. Shaded regions denote 95% confidence intervals (CIs).

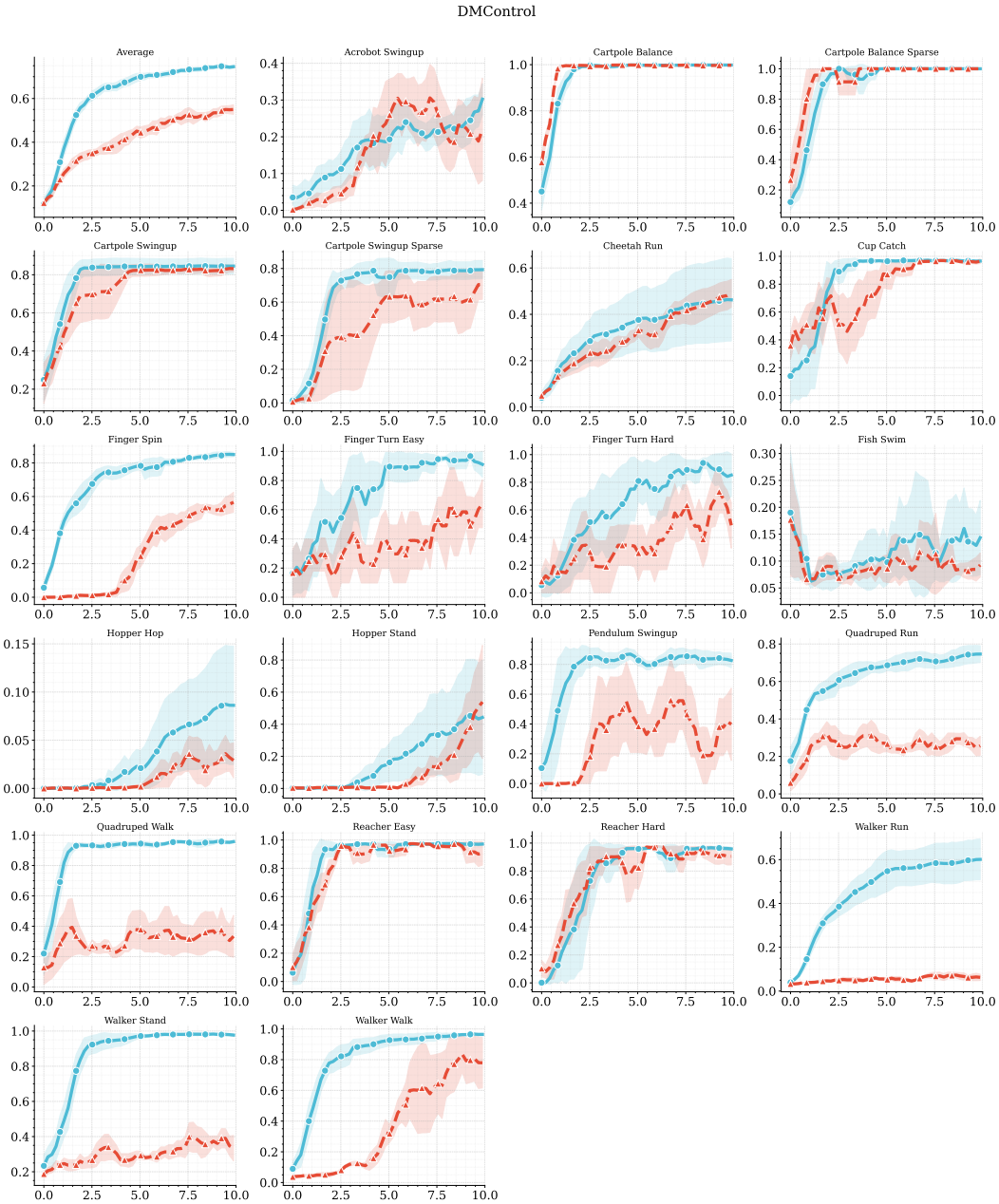

Figure 16: **DMControl per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across DMControl tasks. Shaded regions denote 95% confidence intervals (CIs).

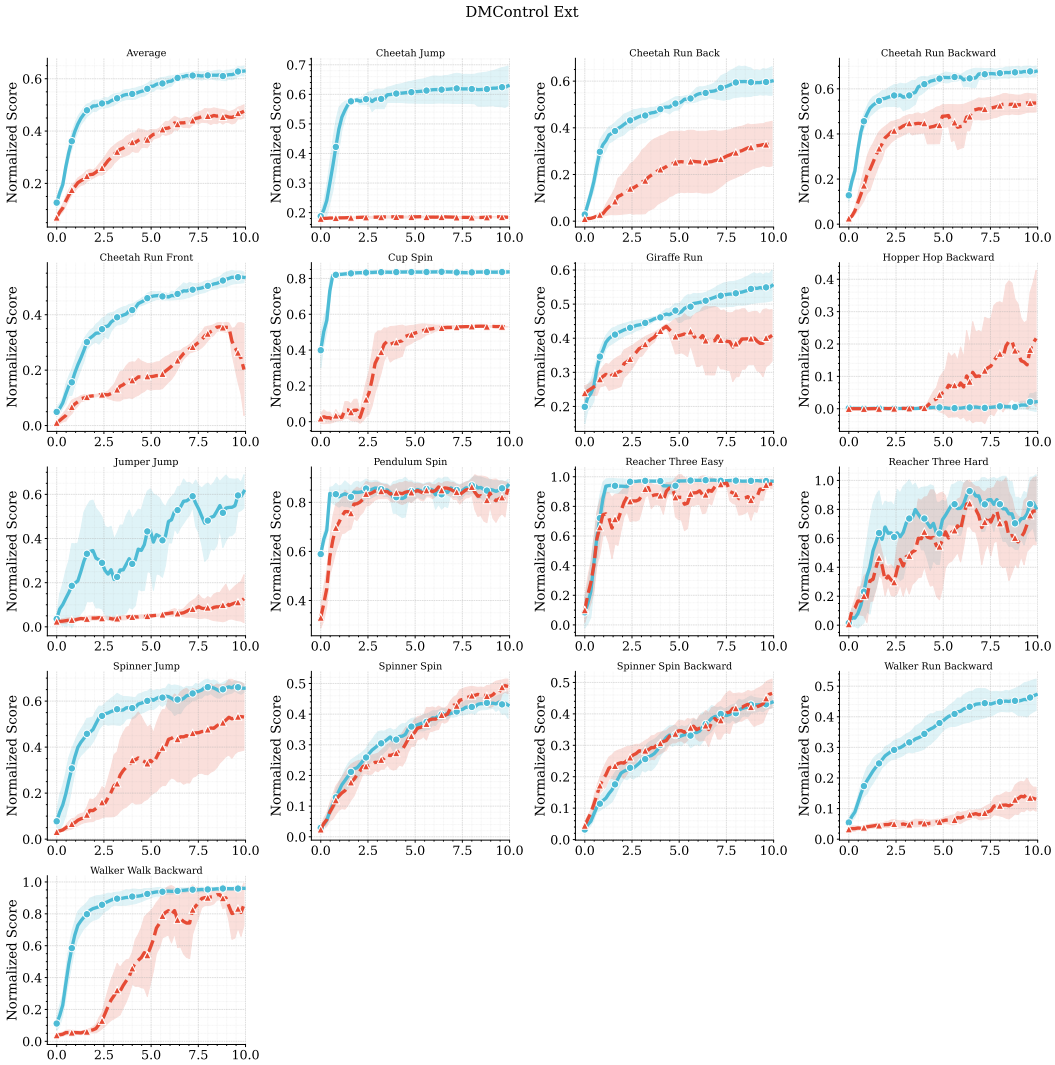

Figure 17: **DMControl-Ext per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across DMControl-Ext tasks. Shaded regions denote 95% confidence intervals (CIs).

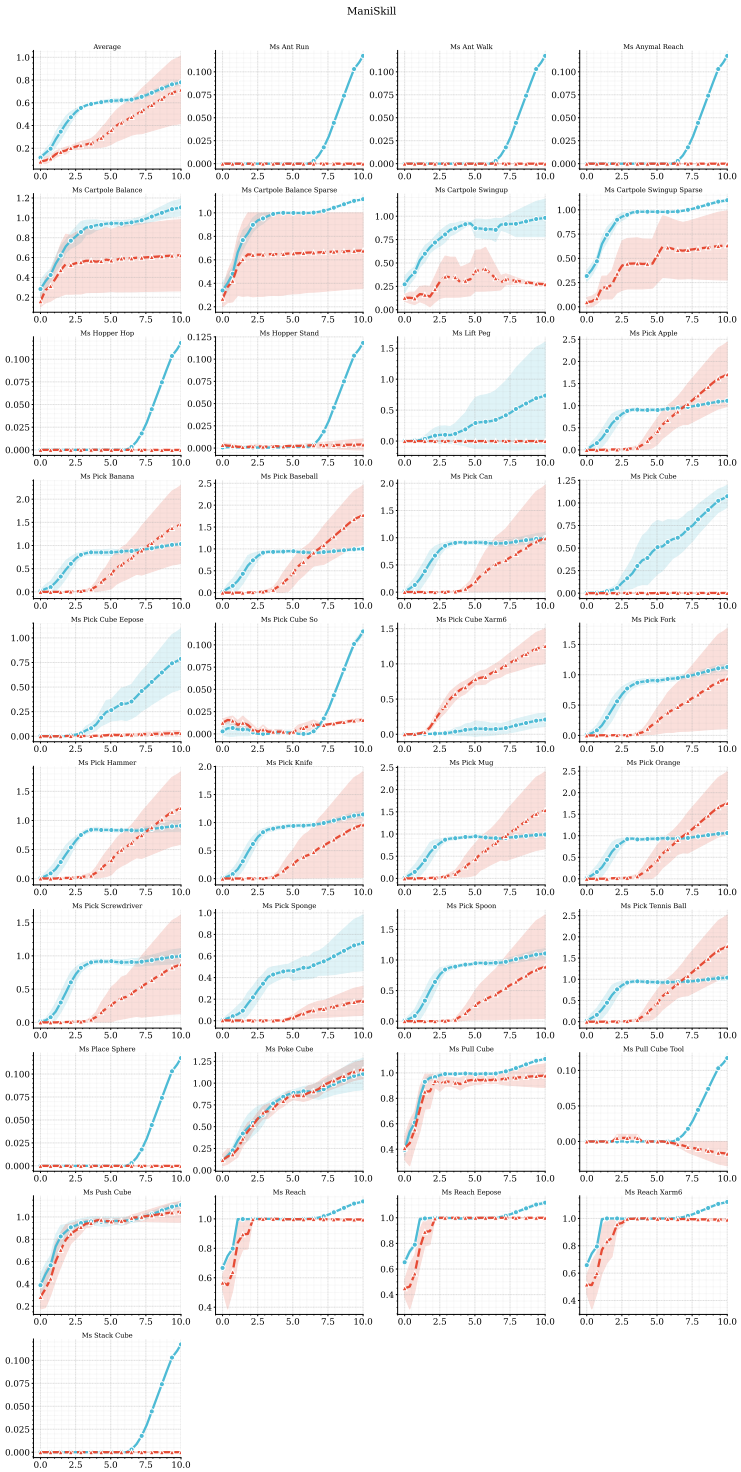

Figure 18: **ManiSkill per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across ManiSkill tasks. Shaded regions denote 95% confidence intervals (CIs).

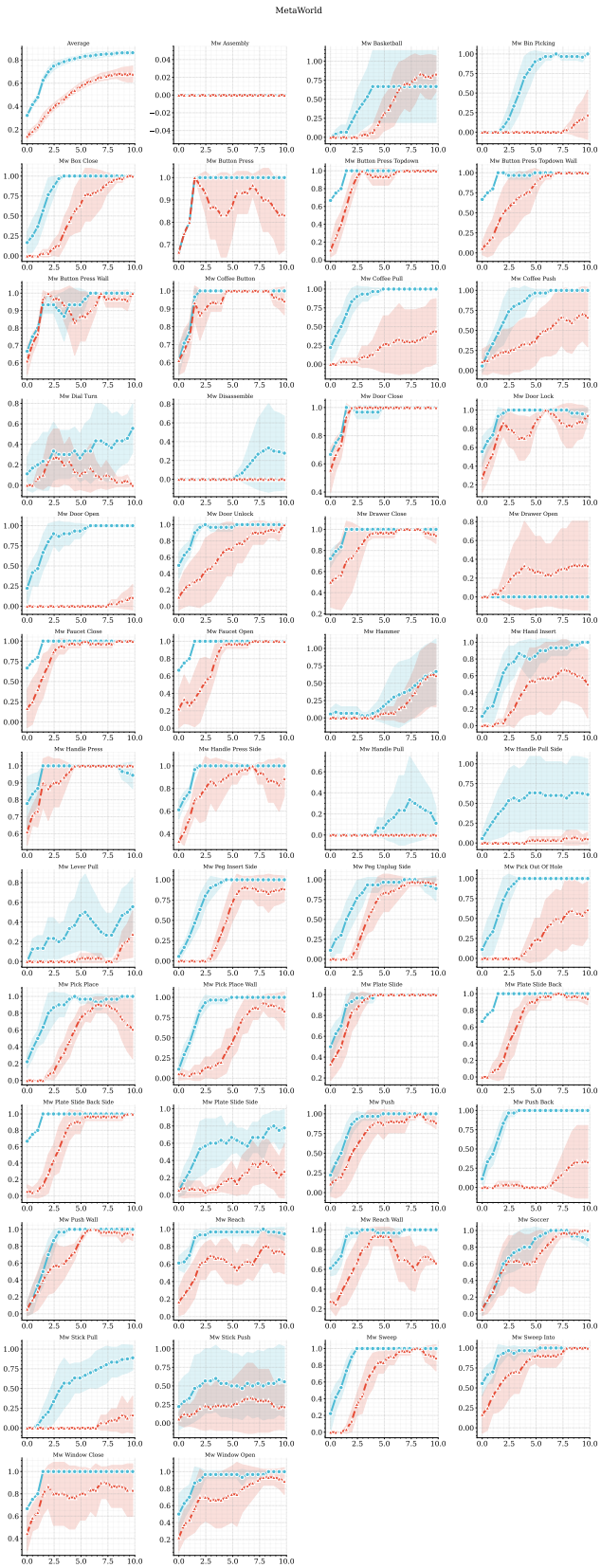

Figure 19: **MetaWorld per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across MetaWorld tasks. Shaded regions denote 95% confidence intervals (CIs).

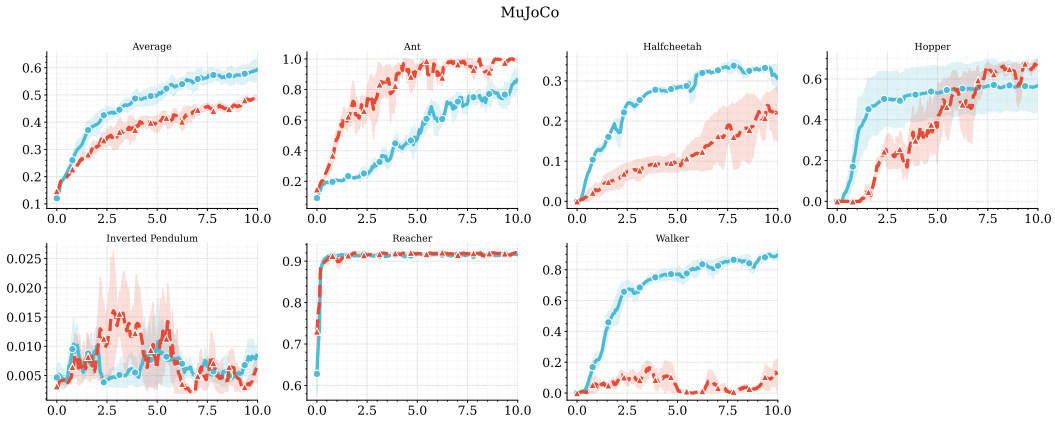

Figure 20: **MuJoCo per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across MuJoCo tasks. Shaded regions denote 95% confidence intervals (CIs).

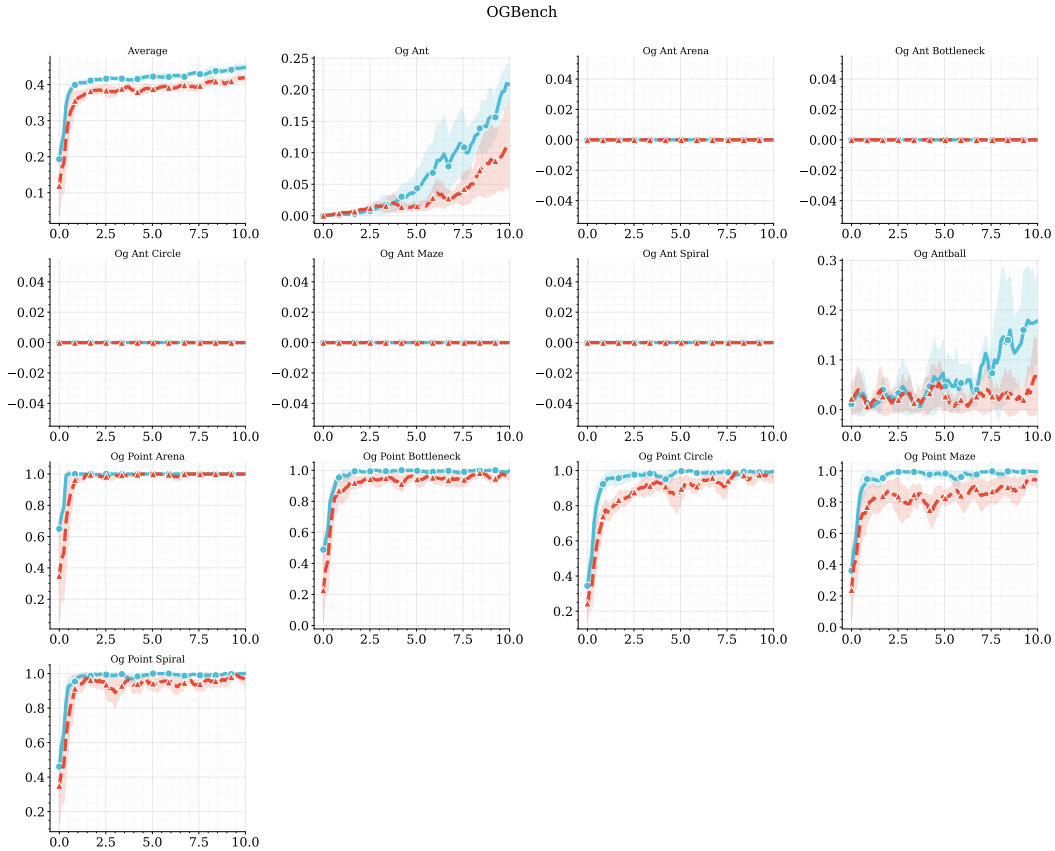

Figure 21: **OGBench per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across OGBench tasks. Shaded regions denote 95% confidence intervals (CIs).

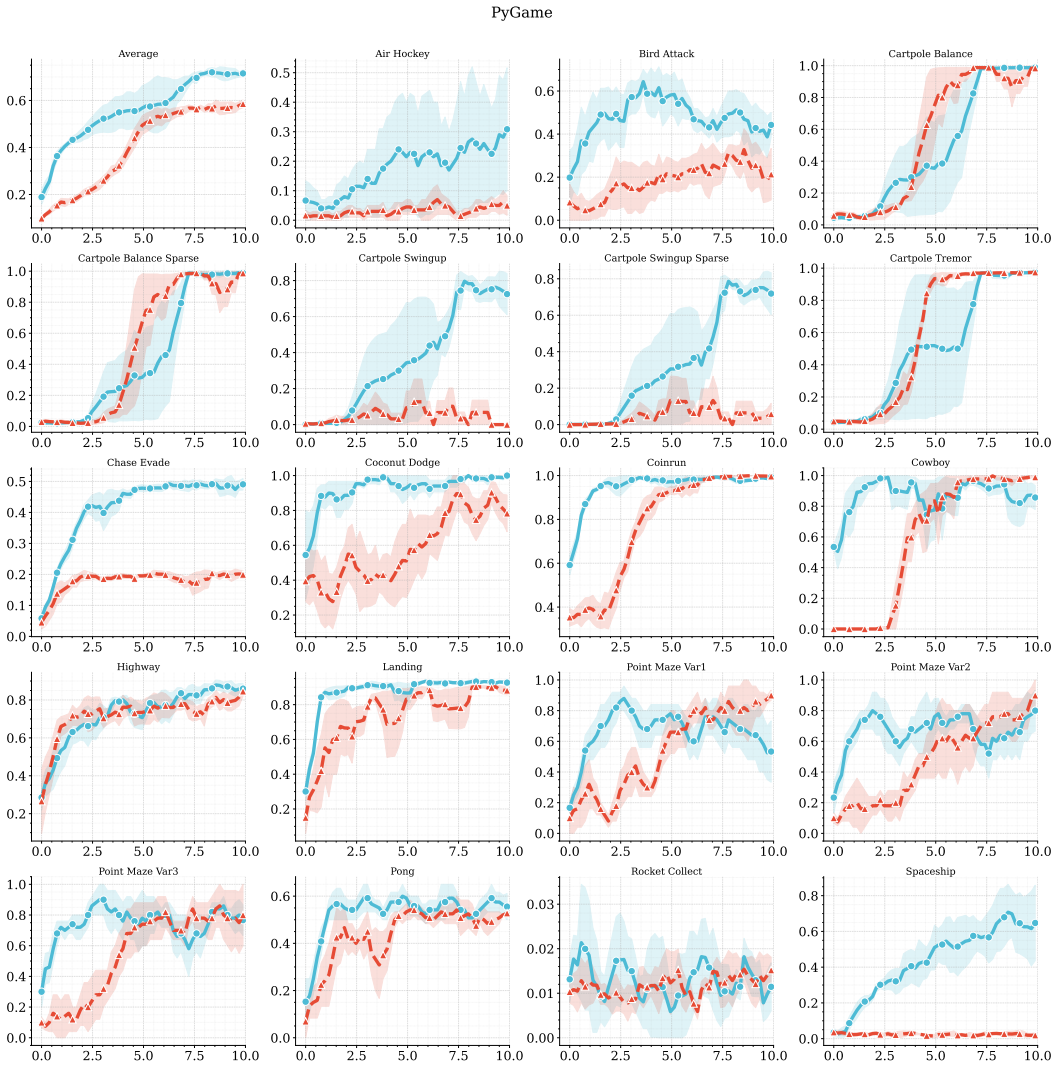

Figure 22: **PyGame per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across PyGame tasks. Shaded regions denote 95% confidence intervals (CIs).

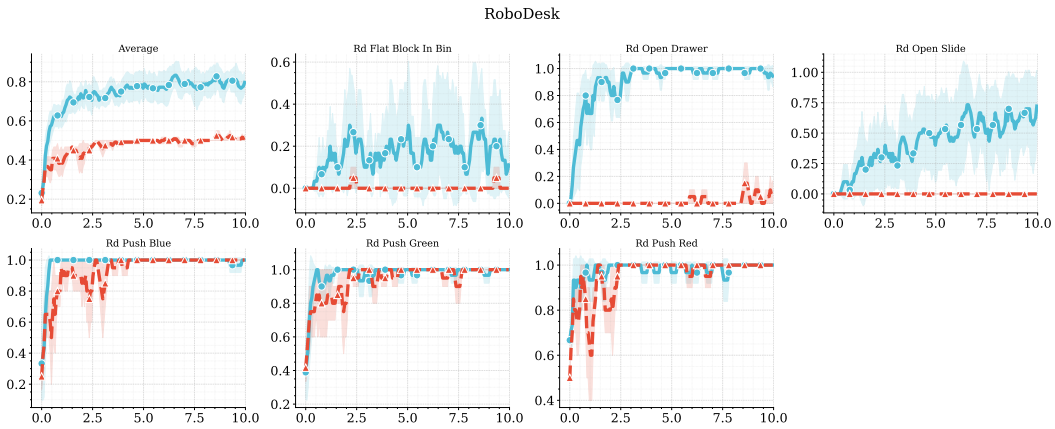

Figure 23: **RoboDesk per-game learning performance.** MR.Q, a model-free agent augmented with predictive model-based representations, consistently matches or surpasses the world-model-based approach *Newt* across RoboDesk tasks. Shaded regions denote 95% confidence intervals (CIs).

P Finetuning: Per-tasks learning curves

To evaluate transfer to unseen tasks, we finetune pretrained multitask checkpoints on held-out environments using online RL. All experiments are initialized from the same multitask checkpoint and finetuned under identical interaction budgets. These experiments evaluate whether the representations learned during multitask pretraining transfer effectively to novel tasks and support rapid adaptation under limited additional experience. See Sec. 5 for more details.

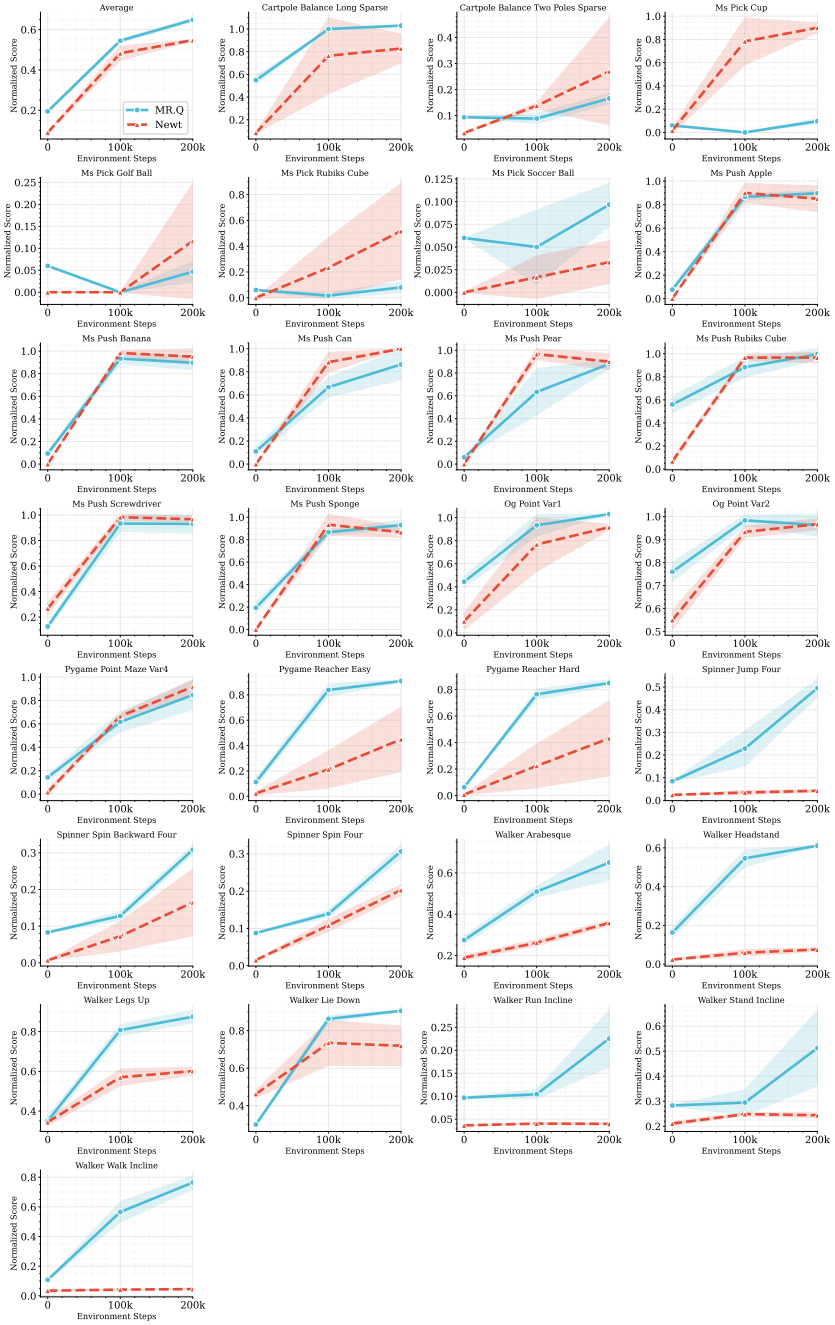

Figure 24: **Per-task finetuning performance on held-out environments.** Learning curves during online finetuning from pretrained multitask checkpoints. MR.Q consistently achieves stronger zero-shot initialization and faster adaptation across the majority of held-out tasks, indicating improved transfer and representation reuse. Shaded regions denote 95% confidence intervals (CIs).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes]

   Justification: The abstract and introduction accurately describe the scope and contributions of the paper. In particular, the paper evaluates whether predictive representation learning alone can enable scalable multitask model-free RL without planning, and supports these claims through large-scale empirical evaluation across multitask domains, scaling analyses, transfer experiments, and wall-clock efficiency comparisons.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification: The paper explicitly discusses limitations and future directions in the "Limitations and Future Work" paragraph of Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The contribution is primarily empirical, focusing on large-scale experimental analyses of multitask reinforcement learning.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the experimental setup, multitask training protocol, baselines, evaluation metrics, interaction budgets, model scaling settings, and training procedures necessary to reproduce the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper states that code will be released anonymously alongside the submission to support reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All primary experimental results are averaged across five random seeds and reported with 95% confidence intervals. Error bars and confidence regions are consistently shown in plots throughout the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports results averaged over five independent random seeds and includes 95% confidence intervals to characterize variability in performance. These intervals are shown as shaded regions in the learning curves.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports wall-clock efficiency comparisons and interaction budgets, detailed hardware specifications and total compute usage are included in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses the importance of developing sample- and compute-efficient multitask RL systems, which can reduce computational cost and improve accessibility of large-scale RL research. The work is primarily foundational research in reinforcement learning and does not directly target deployment settings. Potential risks associated with scalable autonomous decision-making systems are acknowledged in the broader context of RL research.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk generative models, scraped datasets, or systems that pose substantial misuse risks. The work focuses on reinforcement learning algorithms evaluated in standard simulated environments.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly cites and credits all benchmark suites, pretrained models, and prior methods used throughout the experiments, including MMBench, DINOv2, MR.Q, and Newt. Publicly available assets are used in accordance with their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce new datasets or benchmark suites. The contribution focuses on algorithmic and empirical analyses built on top of existing multitask RL environments.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or experiments with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or data collection requiring IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed methodology, experiments, or scientific contributions. Language models were only used for minor writing and grammar assistance and therefore do not require declaration under the NeurIPS policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.