Generalization with a SPARC: Single-Phase Adaptation for Reinforcement Learning in Contextual Environments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generalization to unseen environments is a significant challenge in the field of robotics and control. In this work, we focus on contextual reinforcement learning, where the agent acts within environments with varying contexts, such as self-driving cars or quadrupedal robots that need to operate in different terrains or weather conditions than they were trained for. We tackle the critical task of generalizing to out-of-distribution (OOD) contexts, without access to explicit context information at test time. Recent work has addressed this problem by training a context encoder and a history adaptation module in separate stages. While promising, this two-phase approach is cumbersome to implement and train. We simplify the methodology and introduce SPARC, a **s**ingle-**p**hase **a**daptation method for **r**einforcement learning in **c**ontextual environments. We evaluate SPARC on varying contexts within *MuJoCo* environments and the high-fidelity racing simulator *Gran Turismo 7* and find that it achieves competitive or superior performance on OOD generalization.

1 Introduction

Deep reinforcement learning (RL) has demonstrated successful performance in fields such as robotics [23], nuclear fusion [7], and high-fidelity racing simulators [32]. Despite these successes, generalizing RL agents to unseen environments with varying contextual factors remains a critical challenge. In real-world applications, environmental conditions such as friction, wind speed, or vehicle dynamics can change unpredictably, often leading to catastrophic failures when the agent encounters out-of-distribution (OOD) contexts that it was not trained for.

A promising approach to tackle this issue is context-adaptive reinforcement learning [3], where agents infer and adapt to latent environmental factors by leveraging past interactions. Rapid Motor Adaptation (RMA) [15] is a notable framework in this direction, introducing a two-phase learning procedure. In the first phase, a context encoder is trained using privileged information about the environment. The second phase then employs supervised learning to train a history-based adaptation module, enabling the agent to infer latent context solely from past state-action trajectories. While effective, this two-phase approach introduces complexity during implementation and training; requiring separate optimization stages and increasing the risk of error propagation.

In this work, we introduce **SPARC** (Single-Phase Adaptation for Reinforcement learning in Contextual environments), a novel method that unifies context encoding and adaptation into a single training phase, as illustrated in Figure 1. SPARC is straightforward to implement and naturally integrates with off-policy training as well as asynchronous distributed computation on cloud-based rollout workers. Algorithms such as SPARC and RMA are advantageous when explicit context labels are unavailable at test time, a frequent limitation in real-world robotic deployment. By collapsing

Submitted to the 18th European Workshop on Reinforcement Learning (EWRL 2025). Do not distribute.

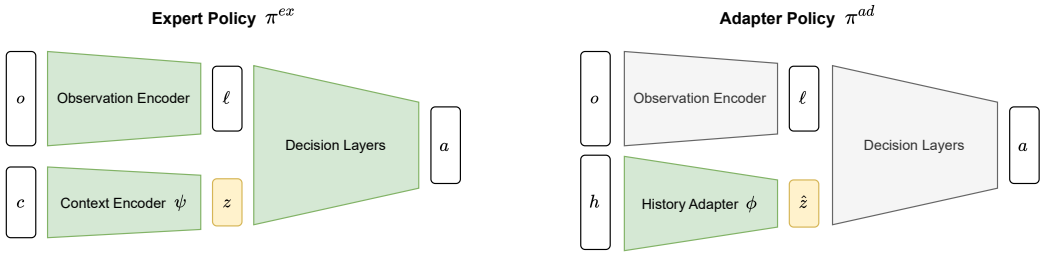


Figure 1: Overview of SPARC, which trains an expert policy π^{ex} and an adapter policy π^{ad} simultaneously in a single phase. The adapter policy does not require access to privileged contextual information, facilitating deployment in real-world scenarios. Observations o, contextual information c, and a history of recent observation-action pairs h are passed into the networks. Latent encodings ℓ and z are concatenated and passed to the final layers, producing action a. Similar to RMA [15], π^{ex} is trained with pure RL, while the History Adapter ϕ of π^{ad} is trained with supervised learning to regress its encoding $\phi(h) = \hat{z}$ to the Context Encoder's output $\psi(c) = z$. Note that since SPARC trains in one phase, the context encoding z will be a moving target, instead of a traditionally fixed target [19, 15]. Trainable modules are in green. The black modules regularly hard-copy weights from their counterpart in the expert policy.

adaptation into a single training loop, SPARC is naturally compatible with on-device continual learning—especially applicable in settings where retraining in the cloud is prohibitive due to privacy or latency constraints. In contrast, RMA is unable to perform continual learning in a straightforward manner.

We evaluate SPARC on two distinct domains: (1) a set of *MuJoCo* environments featuring strongly varying environment dynamics through the use of wind perturbations, and (2) a high-fidelity racing simulator, *Gran Turismo 7*, where agents must adapt to different car models on multiple tracks. SPARC achieves state-of-the-art generalization performance and consistently produces Pareto-optimal policies when evaluated across multiple desiderata.

Our contributions are summarized as follows.

- We introduce SPARC, a novel single-phase training method for context-adaptive reinforcement learning, eliminating the need for separate encoder pre-training.
- We empirically validate SPARC's generalization ability across OOD environments, demonstrating competitive or superior performance compared to existing approaches.
- We perform and analyze several ablation studies, examining key design choices such as history length and the selection of rollout policy during training.

2 Related Work

Generalization to out-of-distribution (OOD) environments is a fundamental challenge in reinforcement learning (RL), hindering its deployment in real-world applications, particularly in robotics and control tasks [14]. The learning dynamics of RL methods often struggle to adapt to novel environmental conditions [22]. Contextual reinforcement learning [17, 3] provides a framework to address this problem by training agents capable of adapting to varying environmental factors.

2.1 Contextual RL

Robust RL often depends on effective contextual adaptation. Recent work has explored context-aware policies that integrate contextual cues into decision-making [4, 5, 16] or employ world models to capture environment dynamics [20, 26]. In addition, several studies have focused on modifying the environment itself—such as by varying gravity or adjusting agent component dimensions—to promote the development of more versatile controllers [3, 21].

2.2 What if the Agent has No Access to Context?

In many real-world scenarios, agents are deprived of explicit contextual information during deployment. In these cases, the agent must infer the relevant environmental factors indirectly. For instance, Lee et al. [19] advanced robust legged locomotion by introducing a two-phase learning process. It first trains an expert policy, which includes a context encoder using the privileged contextual information. The second phase involves an adapter policy that tries to imitate the expert's action, while a history-based adaptation component aims to minimize the difference between its history encoding and the expert's context encoding. Rapid Motor Adaptation (RMA) [15] refines this methodology by only imitating the context encoding, not the action. The adapter policy can be deployed, as it does not require access to the privileged context. See Section 4.1 for further details.

2.3 Other Techniques for Generalization

Several complementary approaches have been proposed to enhance generalization. Domain randomization [30, 25] and procedurally generated environments [6, 9] introduce diversity during training, thereby encouraging robust policy behavior. We employ domain randomization by default in our experiments. System identification methods [33]—whether performed explicitly or through implicit online adaptation, as in SPARC and RMA—also contribute to improved performance under varying conditions. Moreover, techniques such as data augmentation [18, 12] and masking [10, 13] have been shown to further enhance generalization, particularly for pixel-based inputs.

Meta-reinforcement learning offers an alternative paradigm for learning adaptable policies [31, 27]. Foundational algorithms like Model-Agnostic Meta-Learning (MAML) [8] enable rapid task adaptation, and emerging methods using hypernetworks can generate task-specific policy parameters on the fly [2, 28, 4]. Although many of these approaches involve multi-phase training, they underscore the importance of adaptability—a principle that our single-phase approach, SPARC, aims to simplify.

3 Background

In this section, we formalize the underlying problem framework and examine the core techniques that form the foundation for SPARC, enabling context-adaptive behavior.

3.1 Problem Formulation

We consider a contextual Markov decision process (CMDP) [11, 1], redefined by Kirk et al. [14] as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{C}, R, T, O, p_s, p_c)$ where:

- $\mathcal{S}$ is the state space,
- $\mathcal{A}$ is the action space,
- $\mathcal{O}$ is the observation space,
- $\mathcal{C}$ is the context space,
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{C} \to \mathbb{R}$ is the reward function,
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{C} \to \Delta(\mathcal{S})$ defines the stochastic transition dynamics conditioned on a context $c \in \mathcal{C}$,
- $O : \mathcal{S} \times \mathcal{C} \to \mathcal{O}$ is the observation function,
- $p_s : \mathcal{C} \to \Delta(\mathcal{S})$ is the distribution over initial states s_0 given a context $c \in \mathcal{C}$, and
- $p_c \in \Delta(\mathcal{C})$ is the distribution over contexts.

During training, the agent will be exposed to a certain subset of contexts $\mathcal{C}_{\text{IND}} \subset \mathcal{C}$, which are in-distribution (IND), short for *within the training distribution*. To test generalization ability, we hold out a different subset of contexts $\mathcal{C}_{\text{OOD}} \subset \mathcal{C}$ that are out-of-distribution (OOD). We ensure that there is no overlap: $\mathcal{C}_{\text{IND}} \cap \mathcal{C}_{\text{OOD}} = \varnothing$. This separation defines two sub-CMDPs: $\mathcal{M}_{\text{IND}}$ and $\mathcal{M}_{\text{OOD}}$. We specify the context distributions to be uniform over their respective subsets:

$$p_c^i(c) = \begin{cases} \frac{1}{|\mathcal{C}_i|} & \text{if } c \in \mathcal{C}_i \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \{\text{IND, OOD}\}$.

In our setting, the agents do not observe $c \in \mathcal{C}_{\text{OOD}}$ at test time and must infer it through other means, for example from their interaction history. However, for comparison, we will also present results of an expert policy that *does* have access to the privileged context information $c \in \mathcal{C}_{\text{OOD}}$ at evaluation.

Our objective is to train a policy π that maximizes expected return across both in-distribution (IND) and out-of-distribution (OOD) contexts, while only having access to privileged contextual information $c \in \mathcal{C}_{\text{IND}}$ during training.

3.2 Pure History-based Policies

History-based policies have emerged as a powerful approach in reinforcement learning for inferring hidden environmental context from past interactions. Instead of relying solely on the current observation $o_t \in \mathcal{O}$, these policies condition action selection on a sequence of recent observation-action pairs. Let H be the history length and $\mathcal{H} = (\mathcal{O} \times \mathcal{A})^H$ the space of possible histories. For time t we define the corresponding history h_t as

$$h_t = (o_{t-H:t-1}, a_{t-H:t-1}) \in \mathcal{H}.$$

This history input results in policies of the form $\pi : \mathcal{O} \times \mathcal{H} \to \Delta(\mathcal{A})$. Including the history may enable the agent to implicitly capture latent context information $c \in \mathcal{C}$, as the context c may influence the environment dynamics.

A pure history-based approach is presented by Lee et al. [19] as a strong baseline. In their work on quadrupedal locomotion over challenging terrains, the authors demonstrate that leveraging an extended history of proprioceptive data via a temporal convolutional network (TCN) enables robust control in diverse settings.

4 Method

4.1 Making use of Contextual Information

The absence of privileged contextual information at test time does not prevent its use during the training process. Training with privileged information has been shown to be particularly useful for generalizing to OOD contexts. In that regard, the approaches by Lee et al. [19] and Kumar et al. [15] are almost equivalent; we will focus on Rapid Motor Adaptation (RMA) [15]. In RMA, two policies are trained in separate phases. First, the expert policy

$$\pi_\theta^{ex} : \mathcal{O} \times \mathcal{C} \to \Delta(\mathcal{A})$$

which includes a context encoder $\psi(\cdot)$ with access to the environment's privileged information, is trained using a reinforcement learning algorithm. While the original RMA work uses PPO [29], we make use of the more sample-efficient QR-SAC, proven to work well in Gran Turismo [32].

Once training of π_θ^{ex} has converged to a sufficient level, the best model checkpoint $\pi_{\theta^*}^{ex}$ needs to be determined. This selection requires careful evaluation across multiple dimensions [24], a cumbersome intermediate step that SPARC skips, as it is trained in a single phase.

The second stage of RMA trains the adapter policy

$$\pi^{ad} : \mathcal{O} \times \mathcal{H} \to \Delta(\mathcal{A})$$

while keeping the expert policy $\pi_{\theta^*}^{ex}$ frozen. In the adapter policy, a *history adapter* ϕ_θ processes a sequence of recent observation-action pairs h_t to produce a latent representation $\hat{z}_t = \phi_\theta(h_t)$. The history adapter is trained by minimizing the distance between $\hat{z}_t = \phi_\theta(h_t)$ and $z_t = \psi_{\theta^*}(c_t)$ through the mean squared error loss:

$$\mathcal{L}_\phi(c_t, h_t) = \mathbb{E}_{c_t, h_t}[(z_t - \hat{z}_t)^2]. \tag{1}$$

The history-inferred latent context $\hat{z}_t$ is then integrated into the policy. By conditioning on both the current observation o_t and the latent context $\hat{z}_t$, the policy can adjust its behavior to handle unseen or varying environmental conditions.

Table 1: Performance summary on IND and OOD settings across all tracks, averaged over 3 seeds. Results show the mean built-in AI (BIAI) ratio across cars (ratio = the RL agent's lap time divided by the BIAI lap time, lower is better). If an algorithm fails to complete a lap with a specific vehicle, it will receive a BIAI ratio of 2.0 for that car model. Additionally, we show the percentage of cars with a successfully completed lap ($\pm$ s.e.m.). We **bold** the best out-of-distribution results across algorithms without access to context at test time (all except Oracle, see Table 3). We include IND results for reference. SPARC achieves the fastest OOD lap times on $^2/_3$ race tracks and completes the most laps with OOD vehicles overall.

Race Track	Method	IND		OOD	
		BIAI ratio (↓)	Success % (↑)	BIAI ratio (↓)	Success % (↑)
Grand-Valley	Only Obs	0.9929 ± 0.0007	100.00 ± 0.00	1.0641 ± 0.0058	95.15 ± 0.56
	History Input	0.9904 ± 0.0001	99.68 ± 0.08	1.0826 ± 0.0203	92.56 ± 2.12
	RMA	1.0046 ± 0.0054	99.84 ± 0.16	1.0560 ± 0.0134	97.09 ± 1.12
	SPARC	0.9999 ± 0.0061	99.76 ± 0.14	**1.0491 ± 0.0055**	**98.06 ± 0.56**
	Oracle	0.9884 ± 0.0005	100.00 ± 0.00	1.1348 ± 0.0137	90.94 ± 2.27
Nürburgring	Only Obs	1.0202 ± 0.0163	95.87 ± 1.48	1.1745 ± 0.0129	81.88 ± 1.17
	History Input	0.9984 ± 0.0030	97.49 ± 0.32	1.1204 ± 0.0132	86.73 ± 1.29
	RMA	1.1085 ± 0.0195	88.03 ± 1.76	1.2995 ± 0.0306	77.99 ± 3.19
	SPARC	1.0254 ± 0.0061	95.87 ± 0.49	**1.1199 ± 0.0076**	**89.00 ± 0.86**
	Oracle	0.9804 ± 0.0027	99.27 ± 0.28	1.1182 ± 0.0215	89.64 ± 2.53
Catalunya-Rallycross	Only Obs	0.9319 ± 0.0009	100.00 ± 0.00	0.9560 ± 0.0006	**100.00 ± 0.00**
	History Input	0.9294 ± 0.0001	100.00 ± 0.00	**0.9553 ± 0.0068**	99.33 ± 0.67
	RMA	0.9445 ± 0.0010	99.82 ± 0.18	0.9667 ± 0.0030	**100.00 ± 0.00**
	SPARC	0.9432 ± 0.0027	100.00 ± 0.00	0.9631 ± 0.0026	**100.00 ± 0.00**
	Oracle	0.9282 ± 0.0001	100.00 ± 0.00	1.1354 ± 0.0595	85.33 ± 5.81

4.2 Single-Phase Adaptation

Our algorithm illustrated in Figure 1, SPARC, greatly simplifies the implementation and training of agents capable of generalizing to out-of-distribution environments without access to privileged contextual information. In SPARC, the expert policy π^{ex} and the adapter policy π^{ad} are trained simultaneously, in contrast to the two-phase approach of RMA. This means that the context encoding $\psi(c) = z$ is a moving target for the history adapter ϕ, instead of a fixed target. The results in Section 6 demonstrate that the adapter policy is able to manage these new learning dynamics.

An important detail in RMA is which model acts in the environment to collect experience. Policy π^{ex} acts in the first training phase, while π^{ad} does so in the second. This raises the question which policy should gather experience for SPARC, as both are trained together. One option would be to let the expert policy π^{ex} control the actions, since it is updated and improved through QR-SAC.

However, the expert policy, π^{ex}, is not the goal of the SPARC approach. A robust adapter policy, π^{ad}, is the overall learning target and using this policy to gather experience allows the learning algorithm to correct for any inaccuracies before final deployment. This brings the learning of π^{ad} closer to an on-policy setting, even though its history adapter ϕ is trained through supervised learning as shown in Equation 1. We perform an ablation study on this choice of rollout policy in Appendix B.

Reducing training of SPARC to one phase provides several benefits: (i) no intermediate selection of the best trained model checkpoint of the first phase is necessary, (ii) training can be easily continued indefinitely, without having to retrain the second phase, (iii) the simpler implementation facilitates the use of SPARC on asynchronous distributed systems.

5 Experimental Setup

5.1 Environments

We evaluate our approach on two distinct domains. **MuJoCo:** A suite of continuous control tasks including *HalfCheetah*, *Hopper*, and *Walker2d*. We induce contextual variability by perturbing the environment's wind speed in multiple dimensions and scales, thereby creating challenging out-of-distribution scenarios. **Gran Turismo 7:** A high-fidelity racing simulator that features diverse car

models and realistic vehicle-track dynamics. The simulator's rich contextual variability makes it an ideal testbed for assessing generalization to unseen conditions. Within Gran Turismo, we experiment on two settings: (1) generalization across car models, and (2) generalization across differing engine power and vehicle mass settings for one specific car.[1] The in-distribution (IND) training set and OOD test set are selected as follows:

(1) *Car Models*: we sort all ∼500 vehicles by their anomaly score through an isolation forest[2] on the car's contextual features such as mass, length, width, weight distribution, power source type, drive train type, wheel radius, etc. We hold out the 20% most *outlier* vehicles as a test set (OOD) and train on the 80% most *inlier* cars (IND).

(2) *Power & Mass:* for a more controlled experiment, we pick a relatively standard racing car, but tune its engine power and mass in each episode to randomly sampled values within the range [75%, 125%] of their defaults. During evaluation, we test on fixed-spaced intervals within [50%, 150%], covering IND and OOD settings.

For the wind-perturbed MuJoCo environments, we similarly train on a certain range of wind speeds, while testing on intervals twice as large. In Gran Turismo, we experiment on three different tracks, presented in Table 2. These tracks represent highly varying settings, with *Catalunya-Rallycross* even including a mixed dirt and tarmac racing path.

5.2 Training Details

All experiments are conducted using the off-policy QR-SAC algorithm [32] as the base reinforcement learning method. The *critics* present within SAC have the same architecture as the expert policy (see Figure 1), which is possible since during inference only the *actor*, or policy network, is needed. We repeat our runs with different random seeds to ensure statistical robustness: three seeds for the compute-heavy Gran Turismo simulator, and five for MuJoCo environments. Key training hyperparameters—such as the history length H, learning rates, and network architectures—are tuned through preliminary experiments with grid search. We analyze the history length in Section 7.1.

Table 2: The Gran Turismo tracks which we experiment on in the *Car Models* setting. We ensure to include varying road types and track lengths to test SPARC and the baselines on multiple settings.

Track	Length	Road Type
Grand-Valley	5.099 km	Tarmac
Nürburgring	25.378 km	Tarmac + Concrete
Catalunya-Rallycross	1.133 km	Dirt + Tarmac

We train all methods asynchronously, collecting experience on multiple distributed rollout workers. In the *MuJoCo* experiments we train for $3M$ policy updates. For Gran Turismo, in the *Power & Mass* setting we perform $6M$ updates, while across *Car Models* we train for $9M$ steps. The famously long and difficult *Nürburgring* track is an exception, where we perform additional updates: $12M$. In each training episode, a new IND setting is sampled for the environment, determining the wind speeds, a car's power & mass, or even the full car model. We further increase the generalization complexity for the *Car Model* experiment by sampling over 9 different tire types, from least traction *Comfort Hard* up to most traction *Racing Soft*.

5.3 Evaluation Protocol

We evaluate policy performance under two settings:

- **In-Distribution (IND):** Evaluation on environments with contextual parameters that lie within the training distribution.
- **Out-of-Distribution (OOD):** Contextual parameters that deviate significantly from the training set, testing the model's generalization capabilities.

During training, we evaluate the policy at fixed intervals on three IND settings. These training evaluations form a Pareto-front, from which we select the best three model checkpoints for each run. We then test these policies on a wide range of IND & OOD contexts. For the *Car Models*,

[1] This is referred to as *Balance of Power* within the Gran Turismo game.

[2] See the scikit-learn documentation for the IsolationForest algorithm.

this means all vehicles, while for *Power & Mass* and *MuJoCo* we divide the widest context ranges into fixed intervals, providing $21^2 = 441$ test environments. Results are averaged over all seeds and model checkpoints per method, along with confidence intervals to account for variance.

Performance metrics include the return for *MuJoCo* and lap times in *Gran Turismo*. However, for particularly difficult outlier cars, some algorithms may not be able to complete any laps. For this reason, we present the racing results along two dimensions: (1) percentage of cars with a completed lap, and (2) the average lap time over the cars that managed to finish. Note that (2) is a biased metric, so (1) needs to be taken into account.

When averaging raw lap times, slower cars have a larger impact on the average. To avoid skewed results, we divide by the built-in AI (BIAI) lap time for each specific car. The BIAI is a classical control method implemented in Gran Turismo to follow a preset driving line. This *BIAI ratio* of RL lap time over BIAI lap time gives us a useful normalized value.

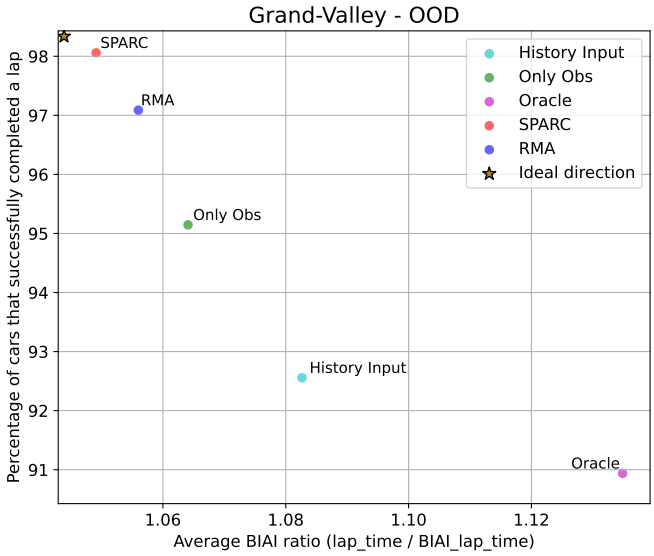


Figure 2: Results on Grand-Valley averaged over three seeds. For each algorithm, we plot the percentage of cars that successfully completed laps, and the built-in AI ratio lap time. SPARC finishes the most and the fastest laps on OOD cars.

5.4 Baselines

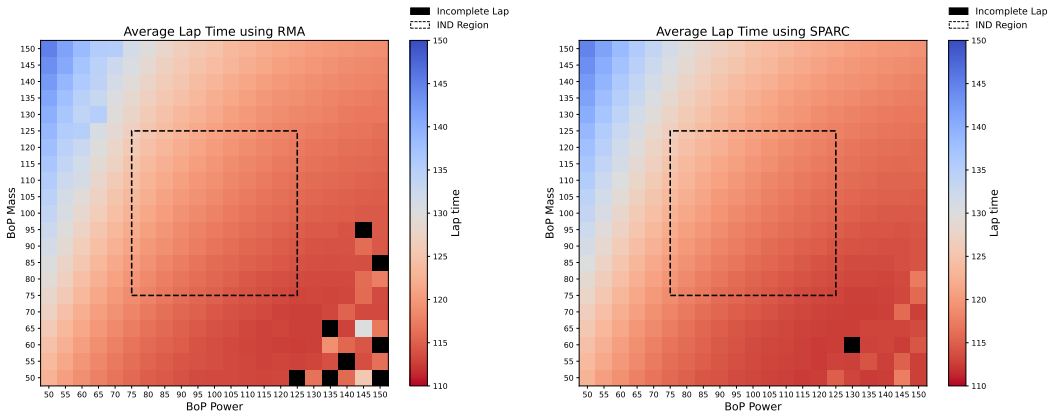
We compare the performance of the following algorithms.

- **Only Obs:** A simple QR-SAC [32] policy trained without any context information. Only the current observation is provided as input.
- **History Input:** A strong baseline policy [19] that additionally receives a history of observation-action pairs.
- **RMA:** The two-phase approach of Rapid Motor Adaptation [15], first trains an expert policy with context input, then learns the adapter policy from history.
- **SPARC:** Our single-phase adaptation technique introduced in this work. At test time it only receives observation-action history and the current observation.
- **Oracle:** A policy that has access to the current observation *and* the ground-truth unencoded contextual features, even at test time.

These baselines allow us to isolate the benefits of the single-phase training paradigm of SPARC, especially regarding implementation simplicity and OOD generalization. See Table 3 for a concise overview of the inputs per algorithm. We are interested in OOD generalization without access to contextual settings at test time, but include the Oracle for reference.

Table 3: The inputs that each algorithm receives.

Method	Inputs during **Training**	Inputs at **Test Time**
Only Obs	`obs`	`obs`
History Input	`obs, history`	`obs, history`
RMA	`obs, history, context`	`obs, history`
SPARC	`obs, history, context`	`obs, history`
Oracle	`obs, context`	`obs, context`



(a) RMA (b) SPARC

Figure 3: Lap times on the *Power & Mass* experiment. We show the average lap time over 3 seeds, and color a square black if at least one seed does not complete a lap in that setting. Even though both algorithms are trained only on settings within the IND region, SPARC is able to handle challenging OOD settings in the bottom right corner (high power and low mass).

6 Results

6.1 Gran Turismo: Car Models

The scatterplot in Figure 2 summarizes the performance of each algorithm averaged over all out-of-distribution cars on the race track Grand-Valley. The results indicate that SPARC outperforms the baselines across unseen vehicles during training. SPARC completes laps with the most cars and with the fastest average built-in AI ratio lap time.

Table 1 provides a quantitative summary of our findings across all three tracks. On IND settings, SPARC is competitive, but it is specially designed to handle OOD dynamics. When racing untrained cars, SPARC is the fastest of all algorithms without access to context at test time on 2 out of 3 tracks. Furthermore, our method manages to complete laps with the most OOD vehicles on aggregate.

6.2 Gran Turismo: Power & Mass

In Figure 3 we show the difference between the RMA baseline and SPARC. SPARC is able to complete laps in almost all OOD contextual settings, while RMA struggles in the most difficult scenarios of lightweight cars with high engine power. Table 4 provides a summary of the average results across all OOD contexts, indicating that SPARC outperforms all baselines, including the oracle which has access to context features at test-time. SPARC is the most robust—completing laps in all but one setting—and also achieves the fastest average built-in-AI lap-time ratio.

Table 4: Performance summary of the *Power & Mass* experiments, averaged over 3 seeds. Results show the mean built-in-AI lap-time ratios (2.0 if no lap completed) across all OOD power & mass settings, and the percentage of these settings with a successfully completed lap ($\pm$ s.e.m.). SPARC completes the most and the fastest laps.

Method	BIAI lap-time ratio (↓)	Success % (↑)
Only Obs	1.0131 ± 0.0136	98.75 ± 1.25
History Input	1.0135 ± 0.0013	98.33 ± 0.10
RMA	1.0004 ± 0.0030	99.17 ± 0.28
SPARC	$\mathbf{0.9907 \pm 0.0011}$	$\mathbf{99.90 \pm 0.10}$
Oracle	0.9962 ± 0.0067	99.27 ± 0.58

6.3 MuJoCo Results

Table 5 presents results for all baselines and MuJoCo enviroments. Again, SPARC presents strong generalization ability to unseen contexts. On Hopper the Oracle performs best; note that this baseline has access to true context at test-time, in contrast to all others (see Table 3). Appendix A shows the difference between SPARC and RMA in each wind perturbation tested. Overall, SPARC beats RMA in significantly more IND and OOD settings, demonstrating robust performance across contexts.

Table 5: Performance across MuJoCo environments, averaged over 5 seeds. Results show the mean return over all out-of-distribution wind perturbations (± s.e.m.). SPARC outperforms all baselines—including the Oracle—in 2 out of 3 environments.

Method	HalfCheetah (↑)	Hopper (↑)	Walker2d (↑)
Only Obs	5724.51 ± 1624.98	1274.13 ± 133.78	2495.77 ± 220.69
History Input	8760.12 ± 161.53	1367.09 ± 67.79	1534.86 ± 144.26
RMA	9033.87 ± 634.11	1307.96 ± 45.65	2306.23 ± 222.09
SPARC	**10017.90** ± **476.19**	1348.22 ± 53.67	**2528.25** ± **263.58**
Oracle	7821.42 ± 1156.77	**1710.14** ± **98.98**	2325.30 ± 576.48

7 Analysis and Ablation Studies

To further understand the contributions of our design choices, we perform a deeper analysis on essential settings, such as the optimal history length (Section 7.1) and the ideal rollout policy (Appendix B). Furthermore, we present an analysis of transferability between distinct environment dynamics in Appendix C.

7.1 Optimal History Length

We perform a sensitivity analysis of SPARC to different lengths of the observation-action history H. Recall from Figure 1 that SPARC's History Adapter ϕ uses this recent experience to recognize its current contextual environment. The results in Figure 4 show that a history length of 50 timesteps seems to be optimal for SPARC. Too few observation-action pairs do not provide enough information to robustly distinguish between contexts, whereas too many may distract the agent and waste computational resources.

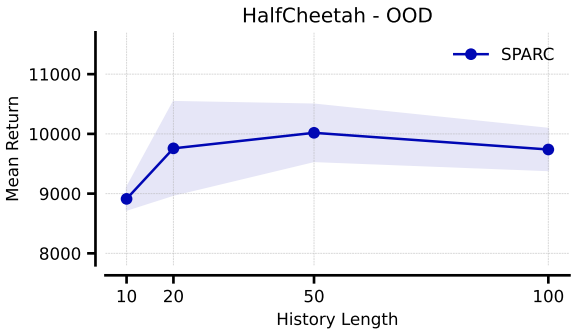


Figure 4: Analysis of the optimal history length for SPARC, averaged over all OOD settings and 5 seeds (± s.e.m.).

8 Conclusion

This paper introduces SPARC, a novel **s**ingle-**p**hase **a**daptation method for **r**einforcement learning in **c**ontextual environments that unifies context encoding and history-based adaptation into one streamlined training procedure. By eliminating the need for separate phases—commonly required in approaches such as Rapid Motor Adaptation—SPARC not only simplifies implementation but also facilitates continual training and deployment in real-world scenarios.

Our extensive experiments in both the high-fidelity Gran Turismo 7 simulator and various MuJoCo tasks demonstrate that SPARC achieves competitive or superior performance in both in-distribution and out-of-distribution settings. In particular, SPARC excels at generalizing to unseen contexts while maintaining robust control, a critical capability for robotics applications where explicit contextual information is unavailable during deployment.

While our results are promising, the work also highlights opportunities for future research. In particular, testing SPARC on physical robotic platforms and further optimizing its training efficiency remain important next steps. Training with other base methods instead of QR-SAC is a promising direction, which we expect to work well as our approach is agnostic to the underlying reinforcement learning algorithm. Overall, SPARC represents a significant advance toward practical, adaptable agents that can thrive in dynamic and uncertain environments.

References

[1] Yasin Abbasi-Yadkori and Gergely Neu. Online learning in MDPs with side information. *arXiv preprint arXiv:1406.6812*, 2014. (Cited on page 3)

[2] Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in Meta-Reinforcement Learning. In *Conference on Robot Learning*, pages 1478–1487. PMLR, 2023. (Cited on page 3)

[3] Carolin Benjamins, Theresa Eimer, Frederik Schubert, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. CARL: A Benchmark for Contextual and Adaptive Reinforcement Learning. *Eco. Theory RL, NeurIPS*, 2021. (Cited on page 1, 2)

[4] Michael Beukman, Devon Jarvis, Richard Klein, Steven James, and Benjamin Rosman. Dynamics Generalisation in Reinforcement Learning via Adaptive Context-Aware Policies. *Neural Information Processing Systems*, 2023. (Cited on page 2, 3)

[5] Baiming Chen, Zuxin Liu, Jiacheng Zhu, Mengdi Xu, Wenhao Ding, Liang Li, and Ding Zhao. Context-Aware Safe Reinforcement Learning for Non-Stationary Environments. In *Int. Conf. on Robotics and Automation*. IEEE, 2021. (Cited on page 2)

[6] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Int. Conf. on Machine Learning*, 2020. (Cited on page 3)

[7] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419, 2022. (Cited on page 1)

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. (Cited on page 3)

[9] Linus Gisslén, Andy Eakins, Camilo Gordillo, Joakim Bergdahl, and Konrad Tollmar. Adversarial Reinforcement Learning for Procedural Content Generation. In *2021 IEEE Conference on Games (CoG)*. IEEE, 2021. (Cited on page 3)

[10] Bram Grooten, Tristan Tomilin, Gautham Vasan, Matthew E Taylor, A Rupam Mahmood, Meng Fang, Mykola Pechenizkiy, and Decebal Constantin Mocanu. MaDi: Learning to Mask Distractions for Generalization in Visual Deep Reinforcement Learning. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2024. (Cited on page 3)

[11] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov Decision Processes. *arXiv preprint arXiv:1502.02259*, 2015. (Cited on page 3)

[12] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. *Neural Information Processing Systems*, 34: 3680–3693, 2021. (Cited on page 3)

[13] Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum Random Masking for Generalization in Image-based Reinforcement Learning. *Neural Information Processing Systems*, 35, 2022. (Cited on page 3)

[14] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023. (Cited on page 2, 3)

[15] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid Motor Adaptation for Legged Robots. *Robotics: Science and Systems*, 2021. (Cited on page 1, 2, 3, 4, 7)

[16] Seyyidahmed Lahmer, Federico Mason, Federico Chiariotti, and Andrea Zanella. Fast context adaptation in cost-aware continual learning. *IEEE Transactions on Machine Learning in Communications and Networking*, 2024. (Cited on page 2)

[17] John Langford. Contextual Reinforcement Learning. In *2017 IEEE International Conference on Big Data*, pages 3–3, 2017. (Cited on page 2)

[18] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020. (Cited on page 3)

[19] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), 2020. (Cited on page 2, 3, 4, 7)

[20] Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware Dynamics Model for Generalization in Model-Based Reinforcement Learning. In *International Conference on Machine Learning*, 2020. (Cited on page 2)

[21] Borja G Leon, Francesco Riccio, Kaushik Subramanian, Peter R Wurman, and Peter Stone. Discovering Creative Behaviors through DUPLEX: Diverse Universal Features for Policy Exploration. In *Neural Information Processing Systems*, 2024. (Cited on page 2)

[22] Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. Learning Dynamics and Generalization in Reinforcement Learning. *Int. Conf. on Machine Learning*, 2022. (Cited on page 2)

[23] A Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra. Benchmarking Reinforcement Learning Algorithms on Real-World Robots. In *Conf. on Robot Learning*, pages 561–591. PMLR, 2018. (Cited on page 1)

[24] Dustin Morrill, Thomas J Walsh, Daniel Hernandez, Peter R Wurman, and Peter Stone. Composing Efficient, Robust Tests for Policy Selection. In *Uncertainty in Artificial Intelligence*, pages 1456–1466. PMLR, 2023. (Cited on page 4)

[25] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2018. (Cited on page 3)

[26] Sai Prasanna, Karim Farid, Raghu Rajan, and André Biedenkapp. Dreaming of Many Worlds: Learning Contextual World Models Aids Zero-Shot Generalization. *Reinforcement Learning Conference*, 2024. (Cited on page 2)

[27] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine Theory of Mind. In *International Conference on Machine Learning*, pages 4218–4227. PMLR, 2018. (Cited on page 3)

[28] Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R Hogan, Gregory Dudek, and David Meger. Hypernetworks for Zero-shot Transfer in Reinforcement Learning. In *The AAAI Conference on Artificial Intelligence*, 2023. (Cited on page 3)

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. (Cited on page 4)

[30] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017. (Cited on page 3)

[31] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016. (Cited on page 3)

[32] Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602 (7896):223–228, 2022. (Cited on page 1, 4, 6, 7)

[33] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. *Robotics: Science and Systems*, 2017. (Cited on page 3)

Appendix

A Delta Results on MuJoCo

In Figure 5, we show results on HalfCheetah by calculating the difference in performance between SPARC and its main baseline RMA, in each contextual setting that we tested. The green squares show SPARC outperforming RMA, while purple indicates the opposite. Overall, SPARC beats RMA in significantly more IND and OOD contexts, demonstrating a robust performance across wind perturbations.

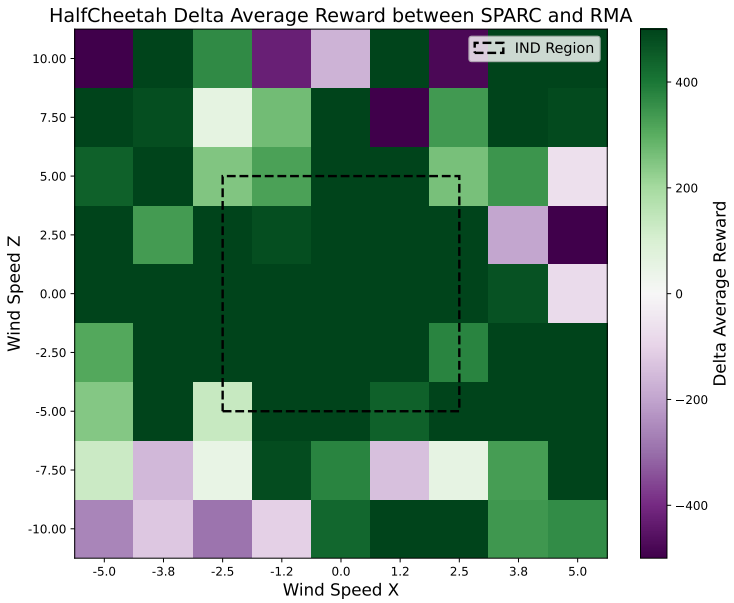


Figure 5: Difference in average return of SPARC versus RMA with varying wind perturbations over 5 seeds. In green: SPARC is better in that wind setting, while in purple: RMA scores higher. Our method outperforms the two-phase baseline across many IND and OOD contextual settings.

B Ablation Study on the Rollout Policy

In this ablation we compare performance when experience is collected with the expert policy (π^{ex}) versus the adapter policy (π^{ad}). As detailed in Section 4.2, SPARC performs rollouts with the adapter policy to ensure a more on-policy style of learning.

Table 6 presents results across all OOD cars on three tracks. Although the differences are small, the results demonstrate that naively using π^{ex} for rollouts leads to slower lap times on 2 out of 3 tracks. Moreover, SPARC finishes every track with at least as many cars as the naive scheme.

C Transferability to Updated Game Dynamics

The *Gran Turismo* developers regularly deploy game updates, where the simulation physics can be adjusted.[3] Reinforcement learning agents that are trained on previous game dynamics generally struggle to adapt to the new physics. We present results on an experiment where we evaluate policies trained on a previous version of *Gran Turismo*, but tested for zero-shot generalization on the newest game dynamics.

In Figure 6 we show that SPARC outperforms all baselines in OOD generalization, this time not only across different car models, but also across other unseen environment dynamics. The oracle

[3] See https://www.gran-turismo.com/us/gt7/news/00_3399040.html for details on the game update we discuss here.

Table 6: Ablation on rollout-policy. Results across all OOD cars and 3 seeds. Mean ± s.e.m. of lap-time BIAI ratio and % successful laps. Naively collecting experience with π^{ex} does not perform as well as directly using π^{ad}.

Race Track	Method	BIAI ratio (↓)	Success % (↑)
Grand-Valley	SPARC-naive	**1.0417 ± 0.0024**	**98.06 ± 0.00**
	SPARC	1.0491 ± 0.0055	**98.06 ± 0.56**
Nürburgring	SPARC-naive	1.1531 ± 0.0158	85.44 ± 1.12
	SPARC	**1.1199 ± 0.0076**	**89.00 ± 0.86**
Catalunya	SPARC-naive	0.9659 ± 0.0032	**100.00 ± 0.00**
	SPARC	**0.9631 ± 0.0026**	**100.00 ± 0.00**

policy with access to ground-truth context is not able to finish laps with around 10% of the cars, while SPARC reduces this to less than 5%, with significantly faster lap times. Note that the context $c \in \mathcal{C}$ that we provide to the oracle policy contains information about the car model only, as the exact simulator physics adjustments are unknown to us. This missing information highlights the importance of SPARC's ability to adapt to unseen contexts without access to all the exact contextual details, e.g., when training in simulation and transferring to a real-world environment.

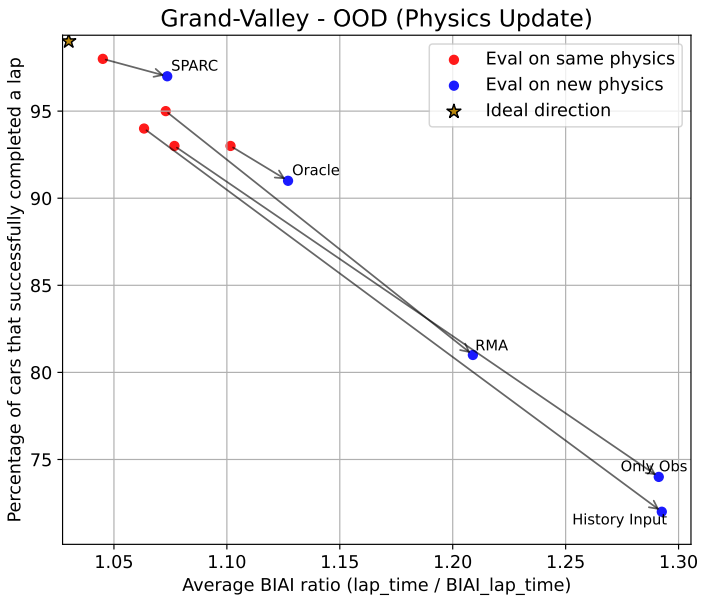


Figure 6: Performance difference between old and new game dynamics. These algorithms have only been trained on old physics settings, and are tested zero-shot on the new physics after a game update of Gran Turismo. SPARC shows the best OOD generalization, driving only slightly slower lap times on the new dynamics, while other algorithms decrease significantly in performance.