Out-of-Distribution Generalization with a SPARC: Racing 100 Unseen Vehicles with a Single Policy

Bram Grooten,[1,2] **Patrick MacAlpine,**[1] **Kaushik Subramanian,**[1] **Peter Stone,**[1,3] **Peter R. Wurman**[1]

[1]Sony AI
[2]TU Eindhoven
[3]The University of Texas at Austin

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 agents act 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) settings, *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: **s**ingle-**p**hase **a**daptation for **r**obust **c**ontrol. We test SPARC on varying contexts within the high-fidelity racing simulator *Gran Turismo 7* and wind-perturbed *MuJoCo* environments, and find that it achieves reliable and robust OOD generalization.

Code + Appendix — https://github.com/bramgrooten/sparc

1 Introduction

Deep reinforcement learning (RL) has demonstrated successful performance in fields such as robotics (Mahmood et al. 2018), nuclear fusion (Degrave et al. 2022), and high-fidelity racing simulators (Wurman et al. 2022). 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 (Benjamins et al. 2021), where agents infer and adapt to latent environmental factors by leveraging past interactions. Rapid Motor Adaptation (RMA) (Kumar et al. 2021) 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.

In this work, we introduce **SPARC** (**s**ingle-**p**hase **a**daptation for **r**obust **c**ontrol), a novel algorithm 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 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 environments is a fundamental challenge in reinforcement learning, hindering its deployment in real-world applications, particularly in robotics and control tasks (Kirk et al. 2023). The learning

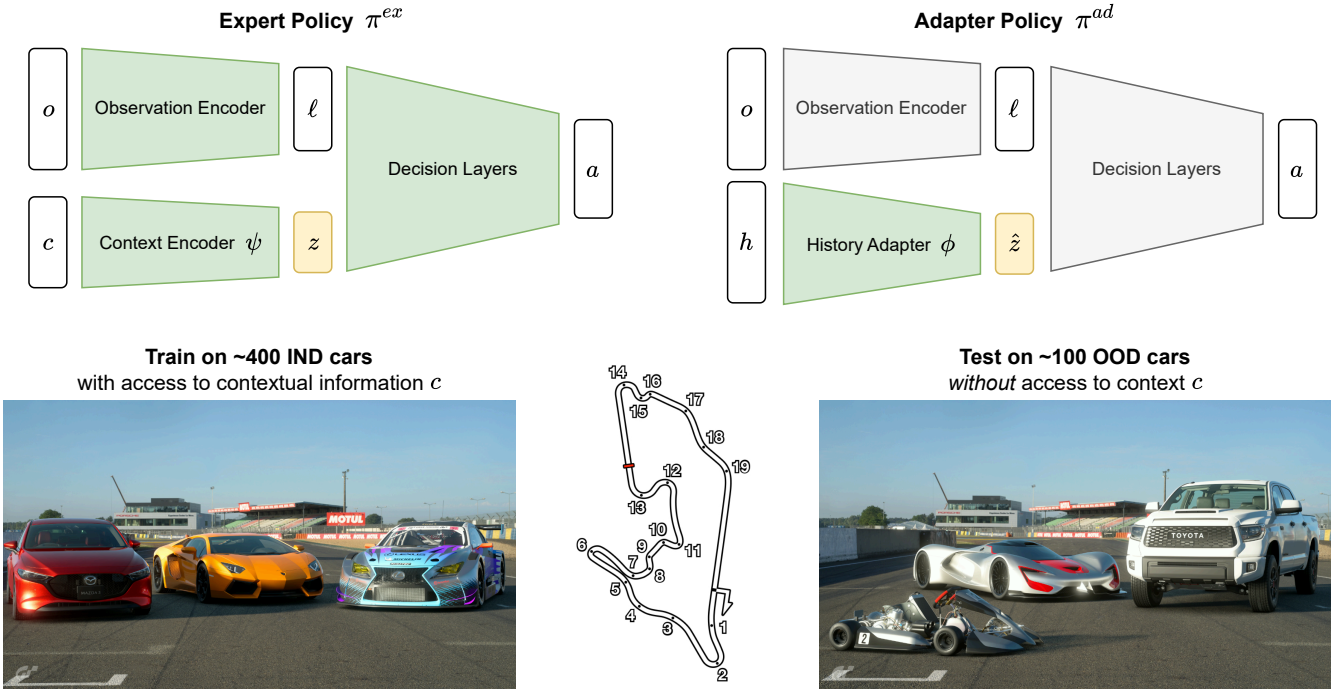


Figure 1: Overview of our algorithm SPARC (**top**) and the problem setting in Gran Turismo 7 (**bottom**). SPARC trains an expert policy π^{ex} and an adapter policy π^{ad} simultaneously in a single phase. The adapter policy does not require access to privileged contextual information, facilitating deployment to OOD 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 (Kumar et al. 2021), π^{ex} is trained with reinforcement learning, 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 is a moving target, instead of a traditionally fixed target in RMA. Trainable modules are in green. The black modules regularly copy weights from their counterpart in π^{ex}.

dynamics of RL methods often struggle to adapt to novel environmental conditions (Lyle et al. 2022). Contextual reinforcement learning (Langford 2017; Benjamins et al. 2021) 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 (Beukman et al. 2023; Chen et al. 2021; Lahmer et al. 2024) or employ world models to capture environment dynamics (Lee et al. 2020b; Prasanna et al. 2024). 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 (Benjamins et al. 2021; Leon et al. 2024).

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. (2020a) 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 (Kumar et al. 2021) 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.

2.3 Other Techniques for Generalization

Several complementary approaches have been proposed to enhance generalization. Domain randomization (Tobin et al. 2017; Peng et al. 2018) and procedurally generated environments (Cobbe et al. 2020; Gisslén et al. 2021) introduce diversity during training, thereby encouraging robust policy behavior. We employ domain randomization by default in our experiments. System identification methods (Yu et al. 2017)—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 (Laskin et al. 2020; Hansen, Su, and Wang 2021; Wang et al. 2024)

and masking (Grooten et al. 2024; Huang et al. 2022) have been shown to further enhance generalization, particularly for pixel-based inputs.

Meta-reinforcement learning offers an alternative paradigm for learning adaptable policies (Wang et al. 2016; Rabinowitz et al. 2018; Duan et al. 2016). Foundational algorithms like Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) enable rapid task adaptation, and emerging methods using hypernetworks generate task-specific policy parameters on the fly (Beck et al. 2023; Rezaei-Shoshtari et al. 2023; Beukman et al. 2023).

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) (Hallak, Di Castro, and Mannor 2015; Abbasi-Yadkori and Neu 2014), defined by Kirk et al. (2023) 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:

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

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

In our setting, the agents do not observe c 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 and out-of-distribution 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. (2020a) 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 Leveraging Contextual Information

Training with privileged contextual information—even if not available at test time—has been shown to be particularly useful for generalizing to OOD contexts. In that regard, the approaches by Lee et al. (2020a) and Kumar et al. (2021) are almost equivalent; we will focus on Rapid Motor Adaptation (RMA) (Kumar et al. 2021). 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 (Schulman et al. 2017), we employ the more sample-efficient QR-SAC, proven to work well in Gran Turismo (Wurman et al. 2022).

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 (Morrill et al. 2023), a cumbersome intermediate step that SPARC skips, as it is trained in a single phase.

The second stage of RMA trains the adapter policy

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

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:

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

The history-inferred latent context $\hat{z}_t$ is 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.

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	$\mathbf{1.0491 \pm 0.0055}$	$\mathbf{98.06 \pm 0.56}$
	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	$\mathbf{1.1199 \pm 0.0076}$	$\mathbf{89.00 \pm 0.86}$
	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	$\mathbf{100.00 \pm 0.00}$
	History Input	0.9294 ± 0.0001	100.00 ± 0.00	$\mathbf{0.9553 \pm 0.0068}$	99.33 ± 0.67
	RMA	0.9445 ± 0.0010	99.82 ± 0.18	0.9667 ± 0.0030	$\mathbf{100.00 \pm 0.00}$
	SPARC	0.9432 ± 0.0027	100.00 ± 0.00	0.9631 ± 0.0026	$\mathbf{100.00 \pm 0.00}$
	Oracle	0.9282 ± 0.0001	100.00 ± 0.00	1.1354 ± 0.0595	85.33 ± 5.81

Table 1: Performance summary on IND and OOD settings across all test tracks in Gran Turismo, 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 receives 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 of 3 tracks and completes the most laps with OOD vehicles overall.

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 non-stationary 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 dynamics of π^{ad} closer to an on-policy setting, even though its history adapter ϕ is trained through supervised learning as shown in Equation 2. We perform an ablation study on this choice of rollout policy in the supplementary material.

SPARC's critic networks—necessary to run QR-SAC on π^{ex}—have the same architecture as the expert policy and thus have access to the context c. We can still run inference without knowing c, because at test time only π^{ad} is needed.

Reducing training of SPARC to one phase provides several benefits: (i) no intermediate selection of the best 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

In this section, we describe the experimental setup used to evaluate SPARC, our proposed single-phase adaptation method, in comparison with several baselines.

5.1 Environments

We evaluate our approach on two distinct domains:

- **MuJoCo:** A suite of continuous control tasks including *HalfCheetah*, *Hopper*, and *Walker2d* (Todorov, Erez, and Tassa 2012). We induce contextual variability by perturbing the environment's wind speed in multiple dimensions and scales, creating challenging OOD scenarios.
- **Gran Turismo 7:** A high-fidelity racing simulator that features diverse car models and realistic vehicle-track dynamics (Wurman et al. 2022). 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. The in-distribution (IND) training set and OOD test set are selected as follows:

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

Table 2: The Gran Turismo tracks which we experiment on in the *Car Models* setting. The road type and track length pose varying challenges.

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`

Table 3: The set of inputs that each algorithm receives.

(1) *Car Models*: we sort all ∼500 vehicles by their anomaly score through an isolation forest 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 training 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 contextual 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

We repeat our runs with independent 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 train all methods asynchronously, collecting experience on distributed rollout workers. Further training details and analyses are provided in the supplementary material.

5.3 Evaluation Protocol

We evaluate policy performance under two settings:

- **In-Distribution (IND):** 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.

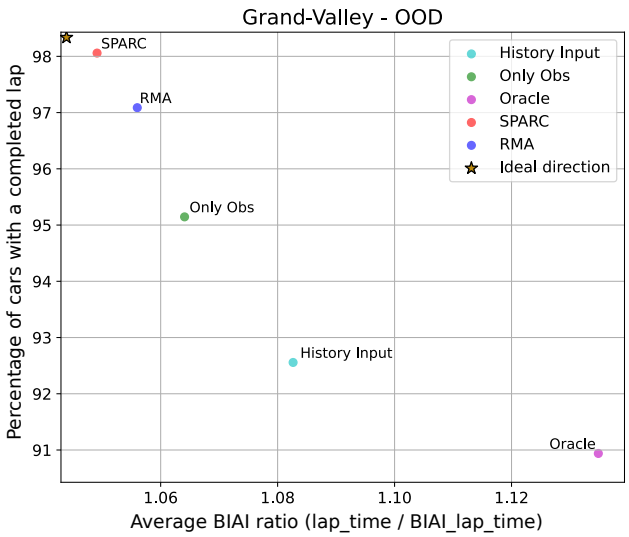


Figure 2: Results on Grand Valley averaged over three seeds. For each algorithm, we plot the percentage of cars that successfully completed laps, and the built-in AI ratio lap time. SPARC is able to complete the most and the fastest laps on out-of-distribution cars.

During training, we regularly evaluate the policy on three predetermined IND settings. These training evaluations form a Pareto-front, from which we select the best model checkpoint for each run. We then test the policy on a wide range of IND & OOD contexts. For *Car Models*, this means all unique vehicles, while for *Power & Mass* and *MuJoCo* we divide the widest context ranges into fixed intervals, providing up to $21^2 = 441$ test environments.

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. 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 7 to follow a preset driving line. This *BIAI ratio* of RL lap time over BIAI lap time provides an informative normalized value. When a car was unable to complete a lap, we set its BIAI ratio to 2.0 before averaging over all vehicles.

5.4 Baselines

We compare the performance of the following algorithms.

- **Only Obs:** This QR-SAC (Wurman et al. 2022) policy is trained without any context information. Only the current observation is provided as input.
- **History Input:** A baseline policy (Lee et al. 2020a) that additionally receives a history of observation-action pairs $h_t = (o_{t-H:t-1}, a_{t-H:t-1})$.

Method	Built-in-AI 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

Table 4: Performance summary of the *Power & Mass* experiments, averaged over 3 seeds. Results show the mean built-in-AI lap-time ratio (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 has the fastest laps.

- **RMA:** The two-phase approach of Rapid Motor Adaptation (Kumar et al. 2021), 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 an observation-action history and the current observation.
- **Oracle:** A policy that has access to the ground-truth unencoded contextual features, even at test time.

Benchmarking SPARC against the listed baselines allows us to isolate the benefits of our single-phase training paradigm, especially regarding implementation simplicity and OOD generalization. See Table 3 for a concise overview of the inputs per algorithm.

6 Results

We present the performance of SPARC and several baselines on *Gran Turismo* and *MuJoCo* environments, focusing on generalization to unseen contexts.

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 particularly 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.

SPARC even outperforms its two-phase counterpart RMA. We believe this occurs because SPARC avoids the brittle selection of a phase-1 checkpoint, required by RMA. Training an adapter ϕ against one checkpoint of ψ can overfit to parts of the context space, while SPARC learns against multiple strong checkpoints over time.

6.2 Gran Turismo: Power & Mass

In Figure 4 we show the difference between (a) the strongest baseline and (b) our method. 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 method. The Oracle does not receive history as part of its inputs, as opposed to other baselines. We believe that SPARC is even able to outperform the Oracle in this environment because knowledge of some history may be useful to mitigate the partial observability in our contextual MDP $\mathcal{M}$ (Section 3.1). SPARC is the most robust in this experiment—completing laps in all but one setting—and also achieves the fastest average built-in-AI lap-time ratio.

Method	HalfCheetah-v5 (↑)	Hopper-v5 (↑)	Walker2d-v5 (↑)
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	$\mathbf{10017.90 \pm 476.19}$	1348.22 ± 53.67	$\mathbf{2528.25 \pm 263.58}$
Oracle	7821.42 ± 1156.77	$\mathbf{1710.14 \pm 98.98}$	2325.30 ± 576.48

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

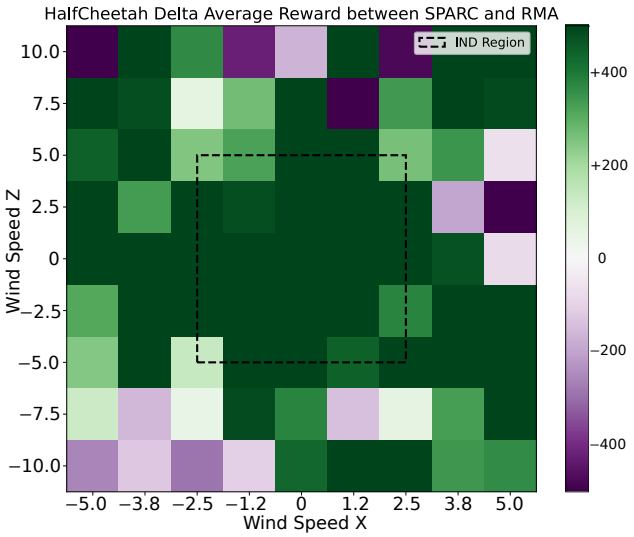
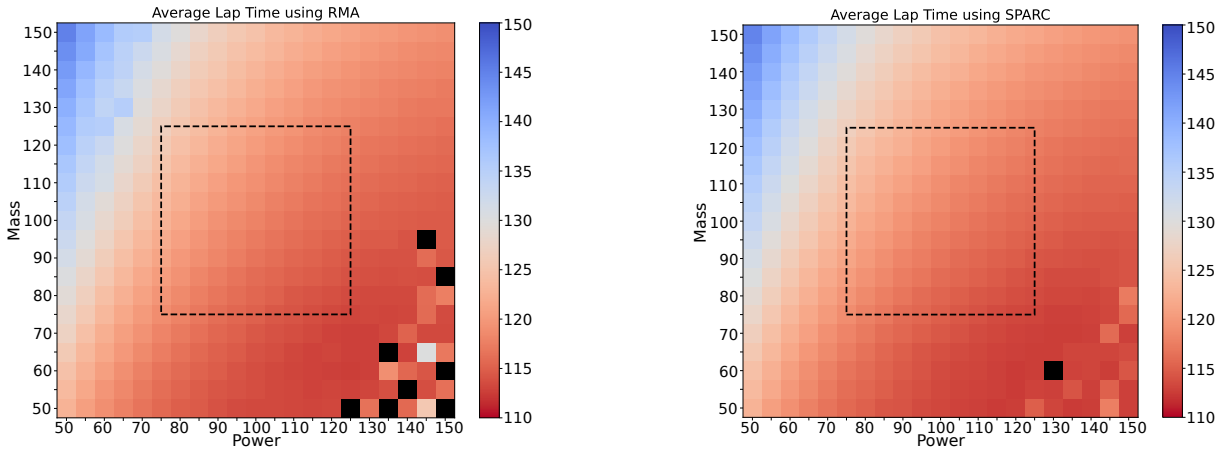


Figure 3: Difference in average return of SPARC versus RMA with varying wind perturbations over 5 seeds. In green SPARC is better in that wind setting, while in purple RMA scores higher. Our method outperforms the two-phase baseline across many IND and OOD contextual settings.

6.3 MuJoCo: Wind Perturbations

In Figure 3, we present results on HalfCheetah by calculating the difference in performance between SPARC and its main baseline RMA, in each wind perturbation tested. The green squares show SPARC outperforming RMA, while purple indicates the opposite. Overall, SPARC beats RMA in significantly more IND and OOD settings, demonstrating a robust performance across varying contexts.



(a) RMA (b) SPARC

Figure 4: Lap times on the *Power & Mass* experiment. Colours denote average lap time over 3 seeds (red = fast, blue = slow); black squares indicate at least one unfinished lap. Even though both algorithms are trained only on settings within the IND region (dashed box), SPARC is able to handle challenging OOD settings in the bottom right corner (high power and low mass).

Table 5 shows the OOD 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).

6.4 Transferability to Updated Game Dynamics

The *Gran Turismo* developers regularly deploy game updates, where the simulation physics can be adjusted. Reinforcement learning agents that are trained on previous game dynamics may struggle to adapt. In this section we evaluate policies on the newest game dynamics, while they were solely trained on a previous version of Gran Turismo.

In Figure 5, we show that SPARC outperforms all baselines in OOD generalization, not only across different car models, but also across other unseen dynamics. The Oracle is not able to finish laps with around 10% of the OOD 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 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 comprehensive contextual details, e.g., when training in simulation and transferring to a real-world environment.

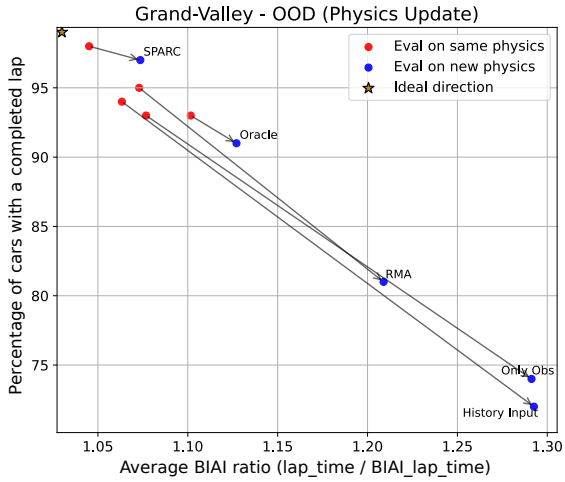


Figure 5: Performance difference between old and new game dynamics. These algorithms have only been trained on old physics settings, and are tested zero-shot on the new physics after a game update of Gran Turismo. SPARC shows the best OOD generalization, with only slightly slower lap times on new dynamics, while other methods degrade significantly.

7 Conclusion

This paper introduces SPARC, a novel **s**ingle-**p**hase **a**daptation method for **r**obust **c**ontrol in contextual environments. The algorithm unifies context encoding and history-based adaptation into one streamlined training procedure. By eliminating the need for separate phases, SPARC not only simplifies implementation but also facilitates continual learning 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 OOD settings. In particular, SPARC excels at generalizing to unseen contexts while maintaining robust control, a critical capability for robotics applications where explicit contexts are unknown 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. Overall, SPARC represents a significant advance toward practical, adaptable agents that can thrive in dynamic environments.

Ethical Statement

This work advances core machine learning capabilities by improving the performance and generalizability of reinforcement learning agents. While we focus on algorithmic improvements, we acknowledge that, like most technical advances in ML, this work may have various societal impacts. We encourage thoughtful consideration of these implications when building upon this research.

Acknowledgements

We are grateful to Florian Fuchs for initiating this research project. Further appreciation to everyone at Sony AI—especially the *Gran Turismo* team members—for many fruitful discussions. Finally, we thank all anonymous reviewers who helped to improve the quality of our work.

References

Abbasi-Yadkori, Y.; and Neu, G. 2014. Online learning in MDPs with side information. *arXiv preprint arXiv:1406.6812*.

Beck, J.; Jackson, M. T.; Vuorio, R.; and Whiteson, S. 2023. Hypernetworks in Meta-Reinforcement Learning. In *Conference on Robot Learning*, 1478–1487. PMLR.

Benjamins, C.; Eimer, T.; Schubert, F.; Biedenkapp, A.; Rosenhahn, B.; Hutter, F.; and Lindauer, M. 2021. CARL: A Benchmark for Contextual and Adaptive Reinforcement Learning. *Eco. Theory RL, NeurIPS*.

Beukman, M.; Jarvis, D.; Klein, R.; James, S.; and Rosman, B. 2023. Dynamics Generalisation in Reinforcement Learning via Adaptive Context-Aware Policies. *Neural Information Processing Systems*.

Chen, B.; Liu, Z.; Zhu, J.; Xu, M.; Ding, W.; Li, L.; and Zhao, D. 2021. Context-Aware Safe Reinforcement Learning for Non-Stationary Environments. In *Int. Conf. on Robotics and Automation*. IEEE.

Cobbe, K.; Hesse, C.; Hilton, J.; and Schulman, J. 2020. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Int. Conf. on Machine Learning*.

Degrave, J.; Felici, F.; Buchli, J.; Neunert, M.; Tracey, B.; Carpanese, F.; Ewalds, T.; Hafner, R.; Abdolmaleki, A.; de las Casas, D.; et al. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv preprint arXiv:1611.02779*. URL: https://arxiv.org/abs/1611.02779.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, 1126–1135. PMLR.

Gisslén, L.; Eakins, A.; Gordillo, C.; Bergdahl, J.; and Tollmar, K. 2021. Adversarial Reinforcement Learning for Procedural Content Generation. In *2021 IEEE Conference on Games (CoG)*. IEEE.

Grooten, B.; Tomilin, T.; Vasan, G.; Taylor, M. E.; Mahmood, A. R.; Fang, M.; Pechenizkiy, M.; and Mocanu, D. C. 2024. MaDi: Learning to Mask Distractions for Generalization in Visual Deep Reinforcement Learning. *Autonomous Agents and Multiagent Systems*.

Hallak, A.; Di Castro, D.; and Mannor, S. 2015. Contextual Markov Decision Processes. *arXiv preprint arXiv:1502.02259*.

Hansen, N.; Su, H.; and Wang, X. 2021. Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. *Neural Information Processing Systems*, 34: 3680–3693.

Huang, Y.; Peng, P.; Zhao, Y.; Chen, G.; and Tian, Y. 2022. Spectrum Random Masking for Generalization in Image-based Reinforcement Learning. *Neural Information Processing Systems*, 35.

Kirk, R.; Zhang, A.; Grefenstette, E.; and Rocktäschel, T. 2023. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76: 201–264.

Kumar, A.; Fu, Z.; Pathak, D.; and Malik, J. 2021. RMA: Rapid Motor Adaptation for Legged Robots. *Robotics: Science and Systems*.

Lahmer, S.; Mason, F.; Chiariotti, F.; and Zanella, A. 2024. Fast Context Adaptation in Cost-Aware Continual Learning. *IEEE Transactions on Machine Learning in Communications and Networking*.

Langford, J. 2017. Contextual Reinforcement Learning. In *2017 IEEE International Conference on Big Data*, 3–3.

Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; and Srinivas, A. 2020. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895.

Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; and Hutter, M. 2020a. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47).

Lee, K.; Seo, Y.; Lee, S.; Lee, H.; and Shin, J. 2020b. Context-aware Dynamics Model for Generalization in Model-Based Reinforcement Learning. In *International Conference on Machine Learning*.

Leon, B. G.; Riccio, F.; Subramanian, K.; Wurman, P. R.; and Stone, P. 2024. Discovering Creative Behaviors through DUPLEX: Diverse Universal Features for Policy Exploration. In *Neural Information Processing Systems*.

Lyle, C.; Rowland, M.; Dabney, W.; Kwiatkowska, M.; and Gal, Y. 2022. Learning Dynamics and Generalization in Reinforcement Learning. *Int. Conf. on Machine Learning*.

Mahmood, A. R.; Korenkevych, D.; Vasan, G.; Ma, W.; and Bergstra, J. 2018. Benchmarking Reinforcement Learning Algorithms on Real-World Robots. In *Conf. on Robot Learning*, 561–591. PMLR.

Morrill, D.; Walsh, T. J.; Hernandez, D.; Wurman, P. R.; and Stone, P. 2023. Composing Efficient, Robust Tests for Policy Selection. In *Uncertainty in Artificial Intelligence*, 1456–1466. PMLR.

Peng, X. B.; Andrychowicz, M.; Zaremba, W.; and Abbeel, P. 2018. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *Int. Conf. on Robotics and Automation (ICRA)*. IEEE.

Prasanna, S.; Farid, K.; Rajan, R.; and Biedenkapp, A. 2024. Dreaming of Many Worlds: Learning Contextual World Models Aids Zero-Shot Generalization. *Reinforcement Learning Conference*.

Rabinowitz, N.; Perbet, F.; Song, F.; Zhang, C.; Eslami, S. A.; and Botvinick, M. 2018. Machine Theory of Mind. In *International Conference on Machine Learning*, 4218–4227. PMLR.

Rezaei-Shoshtari, S.; Morissette, C.; Hogan, F. R.; Dudek, G.; and Meger, D. 2023. Hypernetworks for Zero-shot Transfer in Reinforcement Learning. In *The AAAI Conference on Artificial Intelligence*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *International Conference on Intelligent Robots and Systems (IROS)*, 23–30. IEEE.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wang, J. X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J. Z.; Munos, R.; Blundell, C.; Kumaran, D.; and Botvinick, M. 2016. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, S.; Wu, Z.; Hu, X.; Wang, J.; Lin, Y.; and Lv, K. 2024. What Effects the Generalization in Visual Reinforcement Learning: Policy Consistency with Truncated Return Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6): 5590–5598.

Wurman, P. R.; Barrett, S.; Kawamoto, K.; MacGlashan, J.; Subramanian, K.; Walsh, T. J.; Capobianco, R.; Devlic, A.; Eckert, F.; Fuchs, F.; et al. 2022. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602(7896): 223–228.

Yu, W.; Tan, J.; Liu, C. K.; and Turk, G. 2017. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. *Robotics: Science and Systems*.