First-order Sobolev Reinforcement Learning

Fabian Schramm*
Inria - ENS Paris
PSL Research University

Nicolas Perrin-Gilbert
ISIR, CNRS
Sorbonne Université

Justin Carpentier
Inria - ENS Paris
PSL Research University

Abstract

We propose a refinement of temporal-difference learning that enforces first-order Bellman consistency: the learned value function is trained to match not only the Bellman targets in value but also their derivatives with respect to states and actions. By differentiating the Bellman backup through differentiable dynamics, we obtain analytically consistent gradient targets. Incorporating these into the critic objective using a Sobolev-type loss encourages the critic to align with both the value and local geometry of the target function. This first-order TD matching principle can be seamlessly integrated into existing algorithms, such as Q-learning or actor–critic methods (e.g., DDPG, SAC), potentially leading to faster critic convergence and more stable policy gradients without altering their overall structure.

1 Introduction

Differentiable physics simulators Freeman et al. [2021], Werling et al. [2021], Le Lidec et al. [2021] enable access to gradients of dynamics and rewards, yet standard model-based RL largely ignores this type of information, which might help to converge faster towards optimal solutions. In this work, we propose *First-Order Sobolev Reinforcement Learning*, which enforces first-order Bellman consistency by matching both values and derivatives of the Bellman target via the chain rule through differentiable dynamics. We augment temporal-difference learning with gradient-matching terms so that $Q_\phi(s, a)$ aligns with both the value and the state/action-derivative targets implied by the Bellman backup through the simulator and, when applicable, the target policy. This brings gradient information into the critic in a principled way, rather than relying on implicit derivatives of a value-only fit. Value-only TD regression does not constrain the critic's derivatives: neural networks can match targets while exhibiting arbitrary local gradients, which in turn might undermine gradient-based control Czarnecki et al. [2017]. In actor–critic methods, the policy update depends directly on $\nabla_a Q(s, a)$ Silver et al. [2014], Haarnoja et al. [2018] and can therefore benefit from improved critic's action-gradients. The resulting benefits are twofold. In Q-learning, matching Bellman-consistent derivatives accelerates learning of Q and improves greedy policies. In actor–critic, the more reliable $\nabla_a Q$ yields better policy updates and supports first-order step selection (e.g., line search or trust-region safeguards), since the local Taylor model $Q(s, a + \Delta a) \approx Q(s, a) + \nabla_a Q(s, a)^\top \Delta a$ is more trustworthy. We derive Bellman-consistent gradient targets by differentiating the backup through differentiable dynamics, introduce a simple critic objective that jointly matches values and gradients, and show how to use it in Q-learning and actor–critic methods. A 1D toy study illustrates the effect on critic accuracy and greedy control, and we extend it to continuous-control tasks with differentiable simulators.

2 Background

We consider continuous control with deterministic, differentiable dynamics $s' = f(s, a)$ and differentiable reward $r(s, a)$. A parametric critic $Q_\phi(s, a)$ and, for actor–critic, a deterministic $\mu_\theta(s)$ or stochastic $\pi_\theta(s)$ policy are trained off-policy with target networks.

*Correspondence to fabian.schramm@inria.fr

1st Workshop on Differentiable Systems and Scientific Machine Learning @ EurIPS 2025.

In Q-learning, the temporal-difference target y and critic loss L_Q are given by:

$$y_{\max}(s,a) = r(s,a) + \gamma \max_{a'} Q_{\phi_{\text{targ}}}(f(s,a),a'), \qquad L_Q^{\text{QL}}(\phi) = \mathbb{E}\big[(Q_\phi(s,a) - y_{\max}(s,a))^2\big].$$

In deterministic actor–critic (e.g., DDPG), the target uses the target actor:

$$y_\mu(s,a) = r(s,a) + \gamma\, Q_{\phi_{\text{targ}}}\big(f(s,a),\, \mu_{\theta_{\text{targ}}}(f(s,a))\big), \qquad L_Q^{\text{AC}}(\phi) = \mathbb{E}\big[(Q_\phi(s,a) - y_\mu(s,a))^2\big].$$

The actor maximizes $J(\theta) = \mathbb{E}_s[Q_\phi(s,\mu_\theta(s))]$ with the deterministic policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_s\Big[\nabla_a Q_\phi(s,a)\big|_{a=\mu_\theta(s)}\, \nabla_\theta \mu_\theta(s)\Big],$$

hence accurate $\nabla_a Q_\phi$ is essential for stable and efficient policy improvement. For stochastic policies, as in SAC, we optimize an entropy-regularized objective $J_{\text{SAC}}(\theta) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta}[Q_\phi(s,a) - \alpha \log \pi_\theta(a|s)]$. The critic reads

$$y_{\text{SAC}}(s,a) = r(s,a) + \gamma\, \mathbb{E}_{a'\sim\pi_{\theta_{\text{targ}}}}\big[Q_{\phi_{\text{targ}}}(f(s,a),a') - \alpha \log \pi_{\theta_{\text{targ}}}(a'|f(s,a))\big].$$

Gradients w.r.t. s and a propagate through this target analogously to the deterministic case, including the derivative of the entropy term when computing $\nabla_a y_{\text{SAC}}$.

3 Methodology

We leverage differentiable simulators to enforce *first-order Bellman consistency* in the critic. For actor–critic targets, set $a' = \mu_{\theta_{\text{targ}}}(s')$; for max-target Q-learning, set $a' = \arg\max_b Q_{\phi_{\text{targ}}}(s',b)$ and treat a' as a stop-gradient variable.

3.1 First-order Bellman targets

We define the Bellman target $y(s,a) \in \{y_\mu(s,a),\, y_{\max}(s,a)\}$. Differentiating the backup via the chain rule yields gradients that are consistent with the Bellman operator:

$$\begin{aligned}
\nabla_s y &= \nabla_s r + \gamma\Big[\nabla_{s'} Q_{\phi_{\text{targ}}}(s',a')\,\tfrac{\partial f}{\partial s} + \nabla_{a'} Q_{\phi_{\text{targ}}}(s',a')\,\tfrac{\partial a'}{\partial s'}\,\tfrac{\partial f}{\partial s}\Big],\\
\nabla_a y &= \nabla_a r + \gamma\Big[\nabla_{s'} Q_{\phi_{\text{targ}}}(s',a')\,\tfrac{\partial f}{\partial a} + \nabla_{a'} Q_{\phi_{\text{targ}}}(s',a')\,\tfrac{\partial a'}{\partial s'}\,\tfrac{\partial f}{\partial a}\Big].
\end{aligned}$$

For actor–critic methods, $\frac{\partial a'}{\partial s'} = \frac{\partial \mu_{\theta_{\text{targ}}}}{\partial s'}$; in max-target Q-learning, we set $\frac{\partial a'}{\partial s'} = 0$ to respect the stop-gradient. For SAC, an additional term accounts for the derivative of the entropy term.

3.2 Sobolev critic

We train the critic using a Sobolev-type objective that matches both values and gradients:

$$L_Q(\phi) = \mathbb{E}\big[(Q_\phi - y)^2 + \lambda_s\|\nabla_s Q_\phi - \nabla_s y\|^2 + \lambda_a\|\nabla_a Q_\phi - \nabla_a y\|^2\big].$$

This enforces first-order Bellman consistency and improves the local Taylor approximation of Q_ϕ. In actor–critic methods, the actor update remains $\nabla_\theta J(\theta) = \mathbb{E}[\nabla_a Q_\phi(s,a)\, \nabla_\theta \mu_\theta(s)]$, but the improved $\nabla_a Q_\phi$ yields more reliable policy gradients and supports first-order step selection.

3.3 Algorithm and practicalities

We provide the full algorithm in Alg. 1. Training requires simulator Jacobians $\frac{\partial f}{\partial s}, \frac{\partial f}{\partial a}$, reward gradients $\nabla_s r, \nabla_a r$, and autograd for $\nabla_{s'} Q_{\phi_{\text{targ}}}, \nabla_{a'} Q_{\phi_{\text{targ}}}$, as well as either $\frac{\partial \mu_{\theta_{\text{targ}}}}{\partial s'}$ (actor–critic) or a stop-gradient through a' (max-target Q-learning). The simulator Jacobians and reward gradients can be stored in an extended replay buffer after rollout. Training stability can benefit from Polyak averaging of target networks, stopping gradients through all target-network paths, and using moderate coefficients λ_s, λ_a (optionally warmed up from zero). The same first-order Bellman gradient consistency applies whether the target y is deterministic or entropy-regularized (SAC).

Algorithm 1 First-Order Sobolev RL (Q-learning or Actor–Critic)

Initialize critic Q_ϕ; if actor–critic, initialize actor μ_θ
Initialize target networks $Q_{\phi_{\text{targ}}} \leftarrow Q_\phi$; if actor–critic, $\mu_{\theta_{\text{targ}}} \leftarrow \mu_\theta$
repeat
  Collect transitions $(s, a, r, s' = f(s, a))$ off-policy and store in replay buffer
  Optionally store simulator Jacobians $(\frac{\partial f}{\partial s}, \frac{\partial f}{\partial a})$ and reward gradients $(\nabla_s r, \nabla_a r)$ in the buffer
  Sample a minibatch (s_i, a_i, r_i, s'_i) (and cached derivatives, if available)
  for each i **do**
    If actor–critic: $a'_i = \mu_{\theta_{\text{targ}}}(s'_i)$; else: $a'_i = \arg\max_b Q_{\phi_{\text{targ}}}(s'_i, b)$ with stop-gradient
    $y_i = r_i + \gamma\, Q_{\phi_{\text{targ}}}(s'_i, a'_i)$
    Compute $\nabla_s y_i, \nabla_a y_i$ via the chain rule through f (and $\mu_{\theta_{\text{targ}}}$ if applicable)
  end for
  Update critic by minimizing

$$\frac{1}{B} \sum_i \left[(Q_\phi(s_i, a_i) - y_i)^2 + \lambda_s \|\nabla_s Q_\phi - \nabla_s y_i\|^2 + \lambda_a \|\nabla_a Q_\phi - \nabla_a y_i\|^2 \right]$$

  If actor–critic: update μ_θ with $\nabla_\theta J(\theta) = \frac{1}{B} \sum_i \nabla_a Q_\phi(s_i, a)|_{a=\mu_\theta(s_i)} \nabla_\theta \mu_\theta(s_i)$
  Optionally Polyak-update targets
until convergence

4 Experiments

1D toy problem. We study a simple control problem to isolate the effect of Sobolev training on the critic. The state and action spaces are $s, a \in [-1, 1]$, dynamics are $s' = f(s, a) = a$, the reward is $r(s, a) = 0.2\, a - (a - s)^2$, and the discount factor is $\gamma = 0.9$. We use Q-learning with the max target

$$y(s, a) = r(s, a) + \gamma \max_{a' \in [-1,1]} Q_{\phi_{\text{targ}}}(s', a'), \qquad s' = a,$$

where the maximization over a' is approximated on a dense grid for training and evaluation. This system admits a closed-form solution for the optimal value, Q-function, and policy. The unconstrained optimum is $a^*(s) = s + 1$, which becomes piecewise after clipping to the action bounds, and allows analytical computation of Q^* and V^*. These analytic solutions serve as ground truth for quantitative evaluation and for visual comparison. We compare two parameterizations for Q_ϕ. First, a quadratic model $Q_\phi(s, a) = \theta_0 + \theta_1 s + \theta_2 a + \theta_3 s^2 + \theta_4 sa + \theta_5 a^2$ with only six parameters and second, a neural critic implemented as a three-layer MLP with 128 units per layer and leaky ReLU activations. Both are trained using Adam with a learning rate of 10^{-4} and a batch size of 50. At each iteration, we sample (s, a) uniformly from $[-1, 1]^2$, compute $s' = a$ and $r(s, a)$, and evaluate the $\max_{a'}$ target on a 100-point grid. The Sobolev critic minimizes

$$L_Q(\phi) = \mathbb{E}\big[(Q_\phi - y)^2 + \lambda_s \|\nabla_s Q_\phi - \nabla_s y\|^2 + \lambda_a \|\nabla_a Q_\phi - \nabla_a y\|^2\big],$$

with $\lambda_s = \lambda_a = 1$, while the value-only baseline uses $\lambda_s = \lambda_a = 0$. We evaluate performance in terms of mean-squared error (MSE) to the analytic Q^*, the gradient MSE with respect to $\nabla_a Q^*$, and the return of the greedy policy $\pi(s) = \arg\max_a Q_\phi(s, a)$, estimated via Monte Carlo rollouts. Fig. 1 visualizes learned Q-slices against Q^*, which reveals how Sobolev training accelerates the convergence of both function shape and derivatives. Tab. 1 shows the quantitative results: Sobolev training reduces both value and gradient errors for both model classes. Enforcing first-order Bellman consistency enhances the geometric accuracy of the learned Q-function and quality of the induced policy.

Table 1: Quantitative results (avg. over 5 seeds) on the 1D control problem. Reported are mean-squared errors (MSE) with respect to Q^*, $\nabla_a Q^*$, and the induced policy.

Model	Method	Q^* MSE	$\nabla_a Q^*$ MSE	Policy error
Quadratic	Baseline	$4.10_{\pm 0.8} \times 10^{-2}$	$1.93_{\pm 0.4} \times 10^{-1}$	$3.80_{\pm 0.1} \times 10^{-2}$
	Sobolev	$1.05_{\pm 0.2} \times 10^{-2}$	$3.16_{\pm 0.8} \times 10^{-2}$	$5.03_{\pm 0.4} \times 10^{-3}$
MLP	Baseline	$4.45_{\pm 2.8} \times 10^{-2}$	$4.33_{\pm 1.6} \times 10^{-1}$	$1.67_{\pm 1.1} \times 10^{-2}$
	Sobolev	$1.25_{\pm 2.6} \times 10^{-2}$	$2.14_{\pm 2.2} \times 10^{-2}$	$1.35_{\pm 2.0} \times 10^{-3}$

Continuous control task using differentiable simulator. To assess the applicability beyond toy settings, we train SAC with and without Sobolev critics on a differentiable MuJoCo Ant task using `Rewarped` Xing et al. [2025]. The simulator provides differentiable dynamics f and reward gradients $\nabla_s r, \nabla_a r$, allowing computation of first-order Bellman targets as in Section 3.1. Both agents share

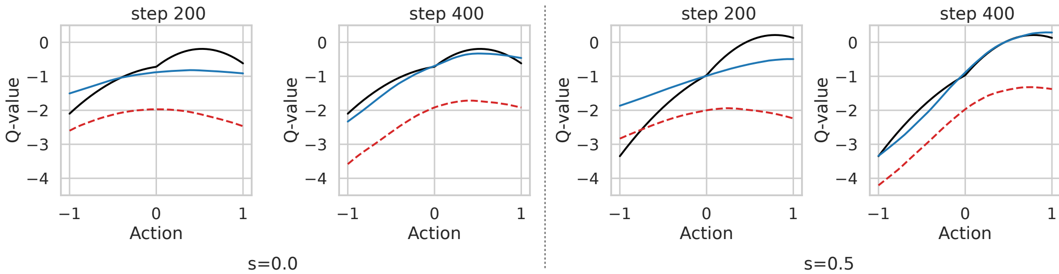
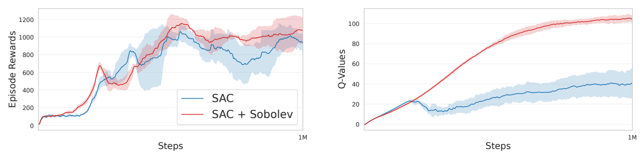


Figure 1: **Comparison of Q-function slices:** ground-truth (black), Sobolev Q-learning, and dashed default Q-learning after 200 and 400 training step for two different states $s = 0.0$ and $s = 0.5$.

identical network architectures and hyperparameters; the only difference is the additional gradient-matching terms in the critic loss. Training uses learning rate 5×10^{-3}, target smoothing $\tau = 0.99$, batch size 2048, and discount $\gamma = 0.99$. For Sobolev critics, $\lambda_s = 1$; $\lambda_a = 0.1$, gradients through target networks are stopped and results are averaged over 5 seeds.

The Sobolev critic leads to faster and smoother initial learning compared to the standard SAC baseline. Q-values for a held-out set of 1000 states converge earlier and fluctuate less, suggesting more stable Bellman updates (see Fig. 2). However, overall performance remains comparable, indicating that while the modified critic improves early dynamics, it does not yet translate into consistently higher returns. Future work should explore adaptive scheduling of the first-order weighting and inclusion of a line search to balance gradient and value consistency better.



(a) Episode Reward. (b) Mean Q-Values.

Figure 2: **Comparison on MuJoCo Ant**.

5 Related Work

Recent efforts have explored incorporating first-order information directly into reinforcement learning updates by differentiating through dynamics. SHAC Xu et al. [2021] and PODS Mora et al. [2021] exploit differentiable simulators to compute analytic gradients of returns, enabling efficient on-policy optimization in smooth environments. However, such fully differentiable on-policy approaches often suffer from exploding or vanishing gradients over long horizons, and become unstable when encountering non-smooth transitions or contact dynamics Georgiev et al. [2024].

Off-policy methods, by contrast, are generally more sample-efficient Fujimoto et al. [2018], Lillicrap et al. [2015], Haarnoja et al. [2018] and can reuse experience across updates, but so far have primarily relied on value matching rather than enforcing gradient consistency. Our work aims to bridge this gap by introducing first-order Bellman consistency, a local, single-step gradient matching principle that leverages simulator Jacobians without requiring complete trajectory differentiation. This eliminates gradient accumulation issues, remains compatible with standard off-policy algorithms, and provides a stable route toward first-order reinforcement learning in complex continuous-control domains.

6 Conclusion

Enforcing first-order Bellman consistency yields critics that better align values and derivatives, improving local accuracy and gradient usefulness for control. Preliminary results show smoother learning dynamics and more stable value estimates, suggesting that gradient consistency can complement standard value matching in off-policy RL. This opens a promising direction for developing first-order off-policy methods that better exploit differentiable simulators, for instance, through adaptive weighting or line-search strategies to balance value and derivative objectives.

Acknowledgments

This work has received support from the French government, managed by the National Research Agency, under the France 2030 program with the references Organic Robotics Program (PEPR O2R) and "PR[AI]RIE-PSAI" (ANR-23-IACL-0008). This research was funded, in part, by l'Agence Nationale de la Recherche (ANR), projects RODEO (ANR-24-CE23-5886) and PEPR O2R – PI3 ASSISTMOV (ANR-22-EXOD-0004). The European Union also supported this work through the ARTIFACT project (GA no.101165695) and the AGIMUS project (GA no.101070165). The Paris Île-de-France Région also supported this work in the frame of the DIM AI4IDF. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the funding agencies.

References

Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. *Advances in neural information processing systems*, 30, 2017.

C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax-a differentiable physics engine for large scale rigid body simulation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Ignat Georgiev, Krishnan Srinivasan, Jie Xu, Eric Heiden, and Animesh Garg. Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation, 2024.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

Quentin Le Lidec, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Differentiable simulation for physical system identification. *IEEE Robotics and Automation Letters*, 6(2):3413–3420, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Miguel Angel Zamora Mora, Momchil Peychev, Sehoon Ha, Martin Vechev, and Stelian Coros. Pods: Policy optimization via differentiable simulation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7805–7817. PMLR, 2021.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. *International conference on machine learning*, pages 387–395, 2014.

Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C Karen Liu. Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints. In *Robotics: Science and Systems*, 2021.

Eliot Xing, Vernon Luk, and Jean Oh. Stabilizing reinforcement learning in differentiable multiphysics simulation. In *International Conference on Learning Representations*, 2025.

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations*, 2021.