

Trading-Off Statistical and Computational Efficiency via W -Step Markov Decision Processes: A Policy Gradient Approach

Gianmarco Tedeschi, Marco Mussi, Alberto Maria Metelli, Marcello Restelli

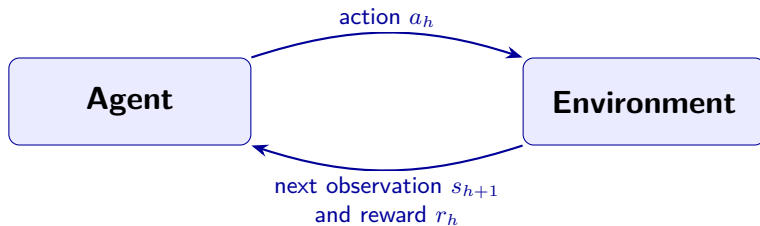
Politecnico di Milano

ECML PKDD 2026

Outline

- 1 Introduction
- 2 Motivation
- 3 W -Step Markov Decision Processes
- 4 MetaStep
- 5 Experiments
- 6 Conclusions

RL: Agent Environment Loop



In the classic interaction protocol (Sutton and Barto, 2018), at each decision step the agent:

- 1 observes the current state s_h ;
- 2 samples an action $a_h \sim \pi_{\theta}(\cdot \mid s_h)$;
- 3 receives a reward r_h and the next observation s_{h+1} .

RL: Markov Decision Process

- **MDP** $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, r, \gamma, H, \mu_0)$ (Puterman, 2014)
transition model P , reward $r \in [-R_{\max}, R_{\max}]$, discount γ , horizon H , initial-state distribution μ_0
- **Policy** $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, Markovian and possibly stochastic
- **Trajectory** $\tau = (s_0, a_0, \dots, s_{H-1}, a_{H-1})$, **return** $R(\tau) = \sum_{h=0}^{H-1} \gamma^h r(s_h, a_h)$
- **Value function** and **expected return**

$$V^\pi(s) = \mathbb{E}_{\tau \sim p(\cdot|\pi)}[R(\tau) \mid s_0 = s], \quad J(\pi) = \mathbb{E}_{s_0 \sim \mu_0}[V^\pi(s_0)]$$

Goal

Find a policy maximizing the expected return, $\pi^* \in \arg \max_{\pi} J(\pi)$.

Efficiency in RL

An RL algorithm is judged along **two axes**:

- **Statistical Efficiency**, samples needed to reach a target accuracy
- **Computational Efficiency**, training time and resource consumption

The curse of horizon

Both costs grow with the decision horizon H :

- *statistically*: longer returns to estimate $\Rightarrow$ higher uncertainty
- *computationally*: more policy evaluations and longer rollouts

Goal: strong statistical guarantees *and* a cheap training loop

Existing Solutions to Curse of Horizon

Action Persistence (Metelli et al., 2020)

- Repeat one action for several steps
- Fewer decisions per episode
- **Rigid**, the same action throughout the planning horizon

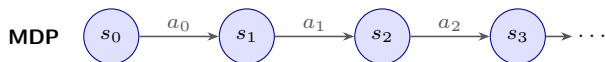
Open-loop Control (Bubeck and Munos, 2010)

- Plan a sequence with no intermediate feedback
- Very cheap at decision time
- **Fragile** in stochastic environments

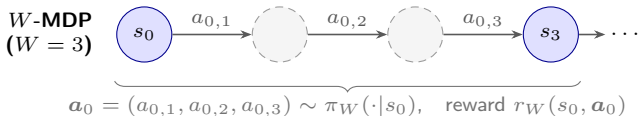
Our Idea

Bridge action persistence and open-loop control: observe the state, execute W possibly different actions open-loop, then observe again. This shortens the effective decision horizon and reduces policy queries, without adding structural complexity.

The W -Step MDP



H policy queries
effective horizon $\frac{1}{1-\gamma}$



H/W policy queries
effective horizon $\frac{1}{1-\gamma W}$

● observed state ○ state not observed

- The agent observes the state only when $h \bmod W = 0$
- The W rewards are aggregated into one discounted signal

The W -Step MDP

Definition 2.1 (W -step MDP)

Given an MDP $\mathcal{M}$ and $W \in \mathbb{N}_{\geq 1}$, $\mathcal{M}_W := (\mathcal{S}, \mathcal{A}^W, P_W, r_W, \overline{H}, \overline{\gamma}, \mu_0)$, where $\mathcal{A}^W = \{\mathbf{a} = (a_1, \dots, a_W) : a_i \in \mathcal{A}\}$ and, for $W \geq 2$ (base case $P_1 = P, r_1 = r$),

$$\begin{aligned} P_W(\cdot | s, \mathbf{a}) &= \int_{\mathcal{S}} P_{W-1}(s' | s, (a_1, \dots, a_{W-1})) P(\cdot | s', a_W) ds' \\ r_W(s, \mathbf{a}) &= r(s, a_1) + \gamma \int_{\mathcal{S}} P(s' | s, a_1) r_{W-1}(s', (a_2, \dots, a_W)) ds' \\ \overline{H} &= \lceil H/W \rceil, \quad \overline{\gamma} = \gamma^W. \end{aligned}$$

- Policies map states to *sequences*: $\pi_W : \mathcal{S} \rightarrow \Delta(\mathcal{A}^W)$, with value function

$$V_W^{\pi_W}(s) = \mathbb{E}_{\tau \sim p(\cdot | \pi_W)} \left[\sum_{h=0}^{\overline{H}-1} \overline{\gamma}^h r_W(s_{Wh}, (a_{Wh}, \dots, a_{Wh+W-1})) \mid s_0 = s \right]$$

- r_W is bounded by $\frac{1-\gamma^W}{1-\gamma} R_{\max} \rightarrow$ single rewards get larger, but the *scale* of the problem does not change
- $W = 1$ gives back $\mathcal{M}$, for $W > 1$ the effective horizon shrinks, $\frac{1}{1-\gamma^W} < \frac{1}{1-\gamma}$

What do we lose in performance by learning in $\mathcal{M}_W$ instead of $\mathcal{M}$?

Theorem 2.1 (Performance bias)

For every $s \in \mathcal{S}$,

$$V^*(s) - V_W^*(s) \leq 4R_{\max} D(P) \frac{\gamma(1 - \gamma^{W-1})}{(1 - \gamma)^2(1 - \gamma^W)}, \quad D(P) := \max_{(s,a)} \min_{s'} \{1 - P(s'|s, a)\}.$$

- $W = 1 \Rightarrow \text{bias} = 0$, we recover the original MDP
- If P is deterministic, then $D(P) = 0$, open-loop execution introduces no bias (see also Bubeck and Munos (2010))

Our Proposal: MetaStep

MetaStep changes the **agent–environment interface**, while leaving the underlying learner unchanged: the learner acts on macro-steps rather than individual transitions.

Algorithm 1 — MetaStep

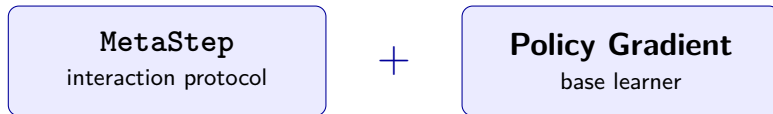
```
1: Input: horizon  $H$ , window  $W$ , policy  $\pi_W$ 
2: for  $h = 0, \dots, \lceil H/W \rceil - 1$  do
3:   Observe state  $s_h$ 
4:   Sample  $\mathbf{a}_h = (a_{h,1}, \dots, a_{h,W}) \sim \pi_W(\cdot | s_h)$ 
5:   for  $w = 1, \dots, W$  do
6:     Play  $a_{h,w}$ , collect reward  $r_{h,w}$ 
7:   end for
8: end for
```

The W rewards are aggregated into a single W -reward $\rho_h = \sum_{w=1}^W \gamma^{w-1} r_{h,w}$ and treated as one reward signal.

MetaStep as a Meta-Algorithm

MetaStep can be coupled with different RL algorithms. It changes the agent–environment loop, not the optimization process.

- MetaStep collects experience as macro-step trajectories.
- The learner updates the policy as usual, over a shorter horizon.



Policy Gradient

- Search over a **parametric policy class** $\Pi_{\Theta} = \{\pi_{\theta} : \theta \in \Theta \subseteq \mathbb{R}^d\}$ and maximize $J(\theta)$ (Sutton et al., 1999a)
- At each iteration k , collect a batch of N trajectories with π_{θ_k} and take a gradient ascent step

$$\theta_{k+1} \leftarrow \theta_k + \eta \hat{\nabla}_{\theta} J(\theta_k)$$

GPOMDP estimator (Baxter and Bartlett, 2001)

$$\hat{\nabla}_{\theta} J(\theta_k) = \frac{1}{N} \sum_{i=1}^N \sum_{h=0}^{H-1} \gamma^h r(s_{i,h}, a_{i,h}) \sum_{u=0}^h \nabla_{\theta} \log \pi_{\theta_k}(a_{i,u} | s_{i,u})$$

- **Rewards-to-go**: an action is credited only with the rewards that follow it $\Rightarrow$ lower variance than REINFORCE (Williams, 1992)
- Both the variance and the cost of one gradient estimate still scale with H

Each episode is split into $\lceil H/W \rceil$ macro-steps; the standard estimator is applied to the W -MDP:

$$\hat{\nabla}_{\theta} J(\theta; W) = \frac{1}{N} \sum_{l=1}^N \sum_{h=0}^{\lceil H/W \rceil - 1} \left(\sum_{u=0}^h \nabla_{\theta} \log \pi_{W, \theta}(\mathbf{a}_{l,u} | s_{l,u}) \right) \gamma^{Wh} \rho_{l,h}$$

Computationally

- Sum over H/W terms instead of H
- Approximately $W \times$ fewer policy queries (each query outputs a richer action sequence)
- Primitive environment transitions remain H

Statistically

- Discount γ^W , horizon H/W
- Under assumptions, the variance upper bound decreases with W

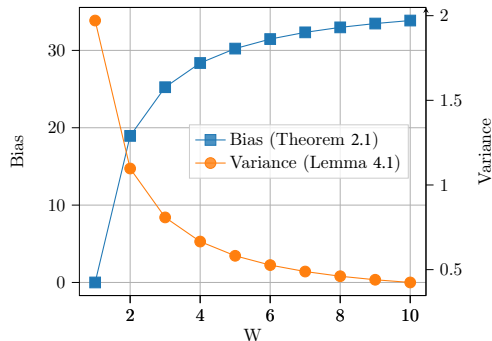
Variance Reduction and the Bias–Variance Trade-Off

Lemma 4.1 (Variance of the estimator)

$$\text{Var}[\hat{\nabla}_{\theta} J(\theta; W)] \leq \frac{V(W)}{N},$$

$$V(W) = \frac{R_{\max}^2 G^2 (1 - \gamma^H)}{(1 - \gamma)^2 (1 - \gamma^W)} \left(1 - \frac{H}{W} (\gamma^H - \gamma^{H+W}) - \gamma^H \right)$$

- $V(W)$ is **non-increasing** in W (Lemma A.5)



Sample Complexity

Define

$$\Phi_H(W) := 1 - \frac{H}{W} (\gamma^H - \gamma^{H+W}) - \gamma^H.$$

Theorem 4.1 (Convergence to a stationary point)

Under assumptions, a suitable step size guarantees $\mathbb{E}[\|\nabla_{\theta} J(\theta_U; W)\|] \leq \epsilon$ with

$$NK \leq \tilde{O}\left(\frac{R_{\max}^3(1 - \gamma^H)\Phi_H(W)^2}{\epsilon^2(1 - \gamma)^3(1 - \gamma^W)^2}\right)$$

Theorem 4.2 (Global convergence)

Under assumptions, a suitable step size guarantees $\mathbb{E}[J^*(W) - J(\theta_K; W)] \leq \epsilon + \beta$ with

$$NK \leq \tilde{O}\left(\frac{R_{\max}^3(1 - \gamma^H)\Phi_H(W)^2}{\epsilon^3(1 - \gamma)^3(1 - \gamma^W)^2}\right).$$

The rates in ϵ match standard policy-gradient analyses (Montenegro et al., 2024). For $W \geq 1$, $\gamma^W \leq \gamma$, so the improvement appears in the horizon-dependent constants

Experimental Setup

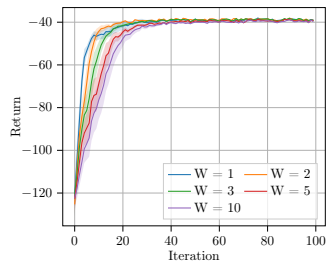
MetaStep on GPOMDP (with optimal baseline (Peters and Schaal, 2008)), $W \in \{1, 2, 3, 5, 10\}$

Task	d_S	d_A	H	γ	N	K
Dam control (Castelletti et al., 2010)	7	1	1825	0.999	10	100
Swimmer (Todorov et al., 2012)	8	2	700	0.995	100	500
Half-Cheetah (Todorov et al., 2012)	17	6	200	0.999	10	500

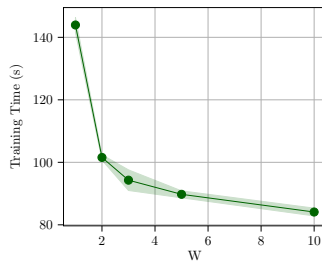
Three views per task (10 runs, mean $\pm$ 95% C.I.):

- (a) return over *iterations* $\rightarrow$ statistical complexity
- (b) training time vs. W $\rightarrow$ computational complexity
- (c) return over *wall-clock time* $\rightarrow$ the two combined

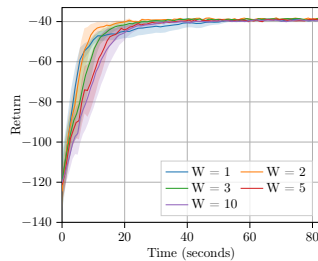
Dam Control



(a) Return over iterations

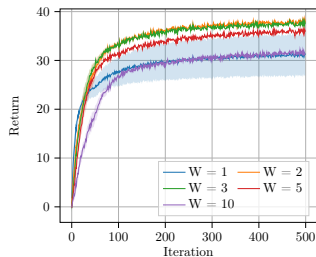


(b) Training time

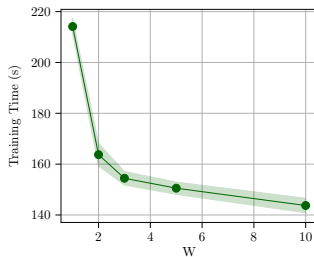


(c) Return over time

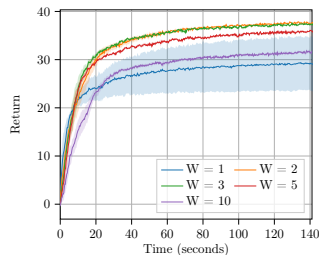
- Long horizon ($H = 1825$, five years of daily data), $\gamma = 0.999$
- Training time falls monotonically with W



(a) Return over iterations



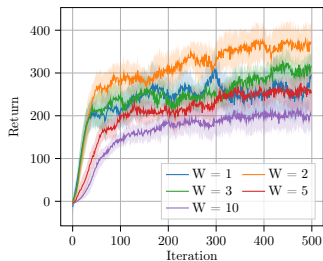
(b) Training time



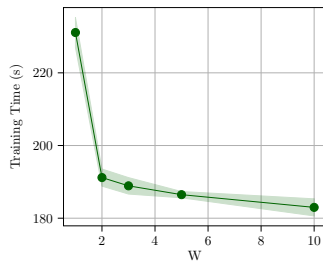
(c) Return over time

- Windows bigger than 1 improve wall-clock convergence while remaining competitive over iterations
- The results suggest that optimizing over Π_W changes the optimization landscape and can help avoid plateaus

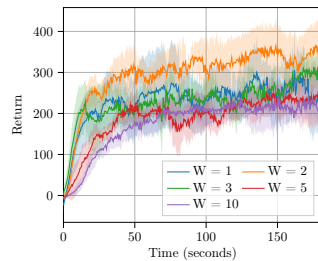
Half-Cheetah



(a) Return over iterations



(b) Training time



(c) Return over time

- Hardest setting: $d_{\mathcal{A}} = 6$ and small batch ($N = 10$) $\Rightarrow$ noisy gradients
- $W = 10$ degrades as the action space $\mathcal{A}^W$ becomes too large to optimize

Limitations and Future Work

Limitations

- W is fixed and chosen by hand
- A fixed open-loop window is too restrictive for tasks needing fine-grained control

Future work

- *Adaptive* window sizes, possibly state-dependent
- W -MDPs for value-based RL (temporal-difference methods)
- MetaStep on other state-of-the-art algorithms
- Large-scale and real-world domains (robotics, control of complex systems)

Thank you!

Email `gianmarco.tedeschi@polimi.it`

Code `github.com/gianmtedeschi/metastep`

References I

- Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
- Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.
- Sébastien Bubeck and Rémi Munos. Open loop optimistic planning. In *Annual Conference on Learning Theory (COLT)*, pages 477–489. Omnipress, 2010.
- Andrea Castelletti, Stefano Galelli, Marcello Restelli, and Rodolfo Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(9), 2010.
- Shivaram Kalyanakrishnan, Siddharth Aravindan, Vishwajeet Bagdawat, Varun Bhatt, Harshith Goka, Archit Gupta, Kalpesh Krishna, and Vihari Piratla. An analysis of frame-skipping in reinforcement learning. *arXiv preprint arXiv:2102.03718*, 2021.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- Alberto Maria Metelli, Flavio Mazzolini, Lorenzo Bisi, Luca Sabbioni, and Marcello Restelli. Control frequency adaptation via action persistence in batch reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6862–6873, 2020.
- Alessandro Montenegro, Marco Mussi, Alberto Maria Metelli, and Matteo Papini. Learning optimal deterministic policies with stochastic policy gradients. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.
- Alessandro Montenegro, Federico Mansutti, Marco Mussi, Matteo Papini, and Alberto Maria Metelli. Reusing trajectories in policy gradients enables fast convergence. *arXiv preprint arXiv:2506.06178*, 2025.
- Gábor Paczoly, Matteo Papini, Alberto Maria Metelli, István Harmati, and Marcello Restelli. Sample complexity of variance-reduced policy gradient: Weaker assumptions and lower bounds. *Machine Learning*, 113(9):6475–6510, 2024.

References I

- Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirota, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4026–4035, 2018.
- Matteo Papini, Matteo Pirota, and Marcello Restelli. Smoothing policies and safe policy gradients. *Machine Learning*, 111(11):4081–4137, 2022.
- Shubham Pateria, Budhitama Subagdja, Ah-Hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys*, 54(5), 2022.
- Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.
- Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2014.
- Luca Sabbioni, Luca Al Daire, Lorenzo Bisi, Alberto Maria Metelli, and Marcello Restelli. Simultaneously updating all persistence values in reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 9668–9676, 2023.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018.
- Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1057–1063, 1999a.
- Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999b.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.
- Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

References III

- Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In *International Conference on Learning Representations (ICLR)*, 2020.
- Huizhuo Yuan, Xiangru Lian, Ji Liu, and Yuren Zhou. Stochastic recursive momentum for policy gradient methods. *arXiv preprint arXiv:2003.04302*, 2020.
- Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3332–3380, 2022.

Backup: Positioning

Approach	Fewer samples	Less computation
Variance-reduced PG (SVRPG, SRVR-PG, RT-PG)	yes	no : added machinery
Open-loop planning	—	yes, but fragile if stochastic
Action persistence / frame skip	yes	yes, but rigid
Hierarchical RL / options	yes	no : hierarchies to learn
MetaStep (W-MDPs)	yes	yes

SVRPG (Papini et al., 2018), SRVR-PG (Xu et al., 2020), STORM-PG (Yuan et al., 2020), DEF-PG (Paczolay et al., 2024), RT-PG (Montenegro et al., 2025); open-loop planning (Bubeck and Munos, 2010); action persistence (Metelli et al., 2020; Sabbioni et al., 2023); frame skipping (Kalyanakrishnan et al., 2021); options and hierarchical RL (Sutton et al., 1999b; Bacon et al., 2017; Pateria et al., 2022).

- W -step sequences act as macro-actions, but no hierarchical structure has to be discovered or learned
- The only knob is W , and $W = 1$ recovers the base algorithm exactly

Backup: Assumptions

Assumption 4.1 (Papini et al., 2022; Yuan et al., 2022)

$$\mathbb{E}_{a \sim \pi_{W, \theta}(\cdot | s)} [\|\nabla_{\theta} \log \pi_{W, \theta}(a | s)\|^2] \leq G^2 \text{ for all } s \in \mathcal{S}.$$

Assumption 4.2 (Papini et al., 2022; Yuan et al., 2022)

$$\mathbb{E}_{a \sim \pi_{W, \theta}(\cdot | s)} [\|\nabla_{\theta} \log \pi_{W, \theta}(a | s)\|] \leq F \text{ for all } s \in \mathcal{S}.$$

Assumption 4.3 (weak gradient domination) (Montenegro et al., 2024)

$$J^*(W) - J(\theta; W) \leq \alpha \|\nabla_{\theta} J(\theta; W)\| + \beta.$$

Why is G independent of W ? Increasing W grows the output layer linearly ($d_A W X$ parameters), but a fully connected network is dominated by its hidden layers (ZX^2). The quadratic term outweighs the linear expansion, so treating G as constant in W is a reasonable approximation.

Backup: Proof Sketch of Theorem 2.1

- 1 Introduce an auxiliary *non-stationary, partly deterministic* model $\bar{P}_W$: the true kernel P when $h \bmod W = 0$, and the deterministic $\arg \max_{s'} P(s'|s, a)$ otherwise.
- 2 Under $\bar{P}_W$, acting open-loop costs nothing: $Q_{1, \bar{P}_W}^* = Q_{W, \bar{P}_W}^*$, since intermediate transitions carry no information.
- 3 Decompose $Q_{1, P}^* - Q_{W, P}^* = (Q_{1, P}^* - Q_{1, \bar{P}_W}^*) + (Q_{W, \bar{P}_W}^* - Q_{W, P}^*)$, i.e. two same-policy comparisons under different kernels.
- 4 Bound each by a telescoping/simulation-lemma argument; each mismatched step contributes at most $\max_{s,a} \|P(\cdot|s, a) - \bar{P}_W(\cdot|s, a, v)\|_1 = 2D(P)$, and only when $v \bmod W \neq 0$.
- 5 Sum the geometric series over the $W - 1$ mismatched steps per window.

Backup: Choosing the Window Size W

Three forces to balance:

- ↓ **Effective horizon** $\frac{1}{1-\gamma^W}$ shrinks with W
⇒ lower variance, less computation
- ↑ **Optimality bias** grows with W and with the stochasticity $D(P)$ (Thm. 2.1)
- ↑ **Action space** $\mathcal{A}^W$ grows with W ⇒ a harder optimization problem

In practice

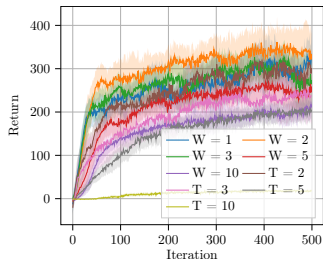
The theory identifies the competing effects but does not select W automatically. Treat it as a hyperparameter; in our experiments, **moderate values such as 2 or 3 are a strong starting point.**

Backup: Full Experimental Details

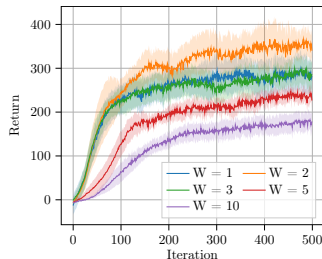
Environment	σ	η	Step size
Dam control	0.5	0.005	Adam
Swimmer	1	0.001	Constant
Half-Cheetah	0.1	0.001	Adam

- Gaussian policy, mean parametrized by a network with two hidden layers (50 and 25 units)
- GPOMDP with the optimal baseline (Peters and Schaal, 2008); adaptive step sizes with Adam (Kingma and Ba, 2015)
- Sweeps in the appendix: $H \in \{200, 500, 700\}$, $N \in \{10, 100\}$, $\gamma \in \{0.995, 0.999\}$
- Hardware: MacBook Pro, Apple M2 Pro (10 cores), 16 GB RAM; runs over 8 CPU cores

Backup: Action Persistence and REINFORCE



(a) MetaStep (W) vs. action persistence (T) on Half-Cheetah



(b) MetaStep on top of REINFORCE

- On Half-Cheetah, MetaStep dominates action persistence (Metelli et al., 2020) for almost every W : letting actions *vary* inside the window beats repeating one action
- MetaStep is not tied to GPOMDP: with the higher-variance REINFORCE estimator (Williams, 1992), $W = 2$ converges faster than the closed-loop baseline