Natural policy gradient methods

From geometric foundations towards safe RL

Johannes Müller

Joint work with Semih Çaycı (RWTH Aachen University), Nikola Milosevic (MPI CBS & ScaDS.AI), Guido Montúfar (UCLA & MPI MiS) and Nico Scherf (MPI CBS & ScaDS.AI)

TECHNISCHE UNIVERSITÄT BERLIN

STOCHASTIC ANALYSIS IN THE SCIENCES

Reinforcement Learning (RL) and Policy Gradient Methods

Recent success stories

- Games
- Large language models
- Robotics
- Automatic proof generation

Many of these approaches use:

1. Variants of natural policy gradient methods
2. Entropy regularization

Questions (for today)

- What do the optimizers do? Which optimal policies do they find?
- What does entropy regularization do? Does it actually help?

Outlook

1. **Markov decision processes (MDPs)**
   - State-action distributions and linear programming for MDPs
   - Policy gradients
   - Entropy regularization
2. **Information geometry**
   - Entropy and KL divergence, Fisher-Rao metric
   - Natural gradients and mirror descent
   - Natural policy gradients
3. **Geometry of natural policy gradient methods**
   - Convex geometry of natural policy gradients
   - Optimal convergence rates for entropy regularization
4. **Embedding Safety into RL via Geometry**

Markov Decision Processes

Markov Decision Processes (MDPs)

Markov decision processes date back to 50s (Bellman, Blackwell, Howard) and consist of:

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition model $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
- Policy or decision rule $\pi : \mathcal{S} \to \mathcal{A}$

Goal

Maximize expected cumulative *discounted reward* (sometimes called return)

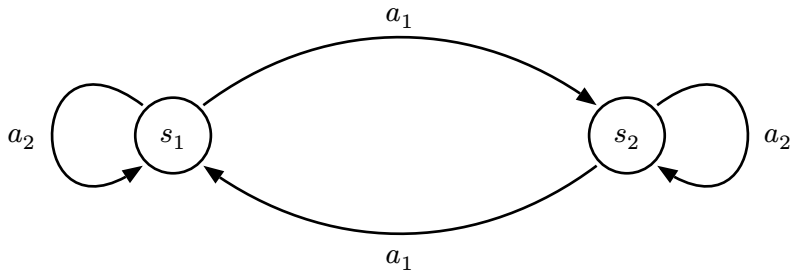
$$R(\pi) = (1-\gamma)\mathbb{E}\left[\sum_{k\in\mathbb{N}} \gamma^k r(S_k, A_k)\right].$$

Here, $\gamma \in (0,1)$ is the *discount factor*.

Stochasticity We consider stochastic transitions $P(s'|s,a)$ and policies $\pi(a|s)$.
Notation Probability simplex Δ_X and stochastic matrices Δ_X^Y.

Example

Consider the following MDP with two states and actions:



A policy is given by the stochastic matrix

$$\pi = \begin{pmatrix} \pi(a_1|s_1) & \pi(a_2|s_1) \\ \pi(a_1|s_2) & \pi(a_2|s_2) \end{pmatrix} = \begin{pmatrix} 0.4 & 0.6 \\ 0.8 & 0.2 \end{pmatrix} \in \Delta_{\mathcal{A}}^{\mathcal{S}}.$$

A possible reward is

$$r = \begin{pmatrix} r(s_1, a_1) & r(s_1, a_2) \\ r(s_2, a_1) & r(s_2, a_2) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3 & 2 \end{pmatrix} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}.$$

State-action distributions

An important concept in the theory of MDPs is the *state-action distribution* d^π induced by a policy π:

$$d^\pi(s,a) = (1-\gamma)\sum_{k\in\mathbb{N}} \gamma^k \mathbb{P}^\pi(S_k = s, A_k = a).$$

Fact The reward can be expressed as a linear function of the state-action distribution:

$$R(\pi) = \sum_{s\in\mathcal{S}, a\in\mathcal{A}} r(s,a) d^\pi(s,a) = r^\top d^\pi.$$

Theorem 1 (Linear programming for MDPs).

The set of state-action distributions is given by the polytope

$$\mathcal{D} = \left\{d^\pi : \pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}\right\} = \left\{d \in \Delta_{\mathcal{S}\times\mathcal{A}} : \ell_s(d) = 0 \quad \text{for all} \quad s \in \mathcal{S}\right\}$$

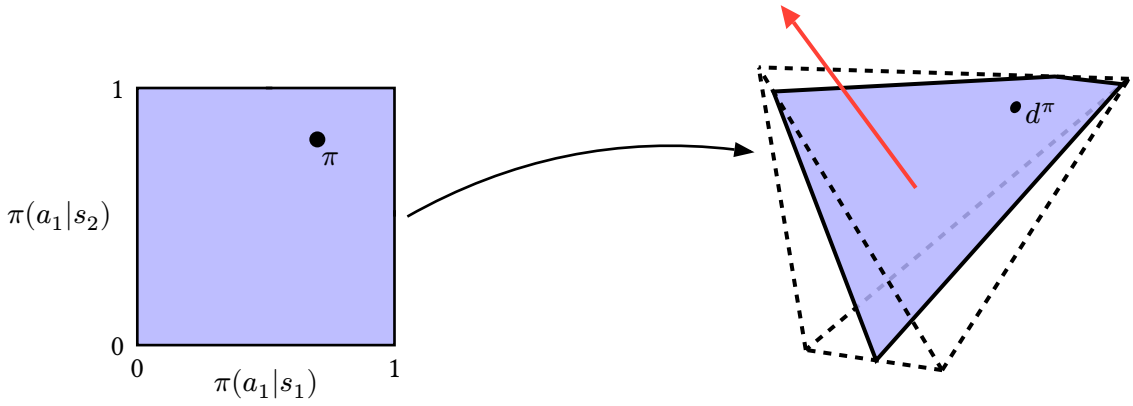
for certain linear functions ℓ_s [1]. Consequently, the MDP is equivalent to the following linear program:

$$\max r^\top d \quad \text{subject to} \quad d \in \mathcal{D}.$$

Compute the policy Given $d \in \mathcal{D}$, we can compute the corresponding policy π by $\pi(a|s) = \frac{d(s,a)}{\sum_{a'} d(s,a')}$.

[1] Cyrus Derman, *Finite State Markovian Decision Processes* (Academic Press, 1970).

Visualization of state-action distributions



Non-convex reward R

Linear program

Remark

Solving the linear program is often not tractable!

Entropy regularization

It is common to add an *entropy regularization*:

$$R_\tau(\pi) = (1-\gamma)\mathbb{E}\left[\sum_{k\in\mathbb{N}} \gamma^k \Big(r(S_k, A_k) + \tau H(\pi(\cdot\,|S_k)) \Big)\right] = R(\pi) + \tau\Psi(\pi),$$

where the *Shannon entropy* is given by $H(\mu) = -\sum_x \mu(x)\log(\mu(x))$.

Theorem 2 (Regularization error).

For any bounded regularizer, it holds that $R^\star - R(\pi_\tau^\star) \leq O(\tau)$[2].

Proof.

For any policy π we can estimate

$$R_\tau(\pi) \leq R_\tau(\pi_\tau^\star) \leq R(\pi_\tau^\star) + \tau\Psi(\pi_\tau^\star) \leq R(\pi_\tau^\star) + c\tau,$$

where we used $0 \leq H \leq \ln|\mathcal{A}|$. Rearranging and maximizing over π yields the claim.

[2] Matthieu Geist et al., "A Theory of Regularized Markov Decision Processes," in "International Conference on Machine Learning," special issue, *International Conference on Machine Learning*, 2019, 2160–69.

Overall error of natural policy gradients

Theorem 3 (Convergence of natural policy gradients).

Regularized natural policy gradient methods with step size η converge at a rate of[3]

$$O\big(e^{-\tau\eta k}\big) + O(\tau) \quad \text{giving} \quad O(\ln(k)k^{-1}).$$

Unregularized natural policy gradient methods achieve a rate of[4]

$$\tilde{O}\big(e^{-\Delta\eta k}\big).$$

Questions

- What is the convergence rate of the regularization error?
- Implicit bias of natural gradient methods?

[3] Shicong Cen et al., "Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization," *Operations Research*, INFORMS, 2021.

[4] Sajad Khodadadian et al., "On the Linear Convergence of Natural Policy Gradient Algorithm," in "2021 60th IEEE Conference on Decision and Control (Cdc)," special issue, *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, 3794–99.

Natural Policy Gradient Methods

Policy gradient methods

Parametrize the policy π_θ and optimize the parameters θ to maximize the reward:

$$\theta_{k+1} = \theta_k + \eta \nabla R(\theta_k).$$

Policy models

- *Softmax policies*

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$$

- *Gaussian policies*

$$\pi_\theta(a|s) = \mathcal{N}(u_\theta(s), \sigma^2)$$

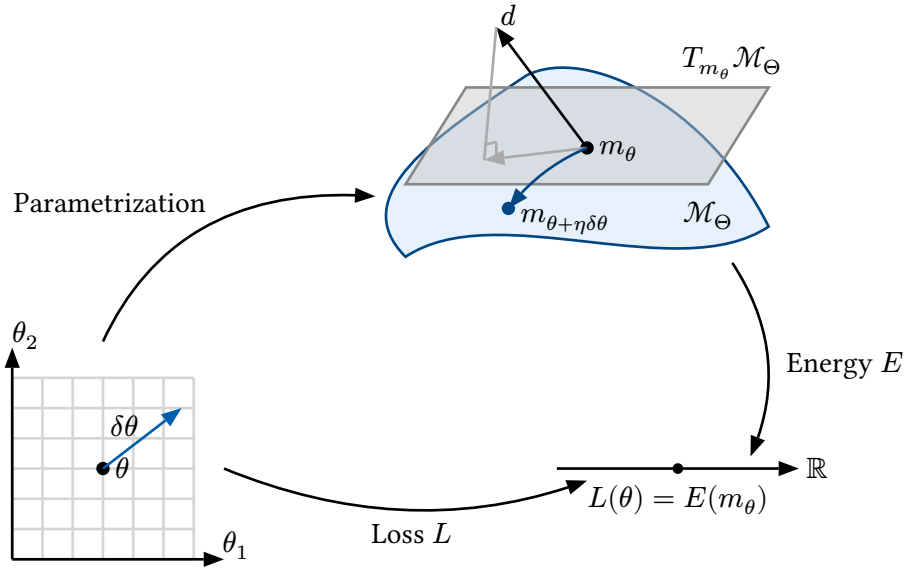
Theorem 4 (Policy gradient).

It holds that[5]

$$\nabla R(\theta) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} d^{\pi_\theta}(s) A^{\pi_\theta}(s,a) \nabla \pi_\theta(a|s).$$

[5] Richard S Sutton et al., "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in "Nips," special issue, *Nips* 99 (1999): 1057–63.

Guiding principle



Natural Gradients and Mirror Descent

Natural gradients

Given a convex function φ on $\mathcal{M}$, we call the iteration

$$\theta_{k+1} = \theta_k - \eta G(\theta_k)^{-1} \nabla L(\theta_k),$$

the *natural gradient method*[6], where the Gramian matrix is given by

$$G(\theta)_{ij} = g_{p_\theta}(\partial_{\theta_i} m_\theta, \partial_{\theta_j} m_\theta) = \partial_{\theta_i} m_\theta^\top \nabla^2 \varphi(m_\theta) \partial_{\theta_j} m_\theta.$$

Properties

- Best approximation of gradient descent while staying in the model $\mathcal{M}_\Theta$
- Reparametrization invariant[7,8]

[6] Shun-Ichi Amari, "Natural Gradient Works Efficiently in Learning," *Neural Computation* 10, no. 2 (1998): 251–76.

[7] Shun-ichi Amari, *Information Geometry and Its Applications*, vol. 194 (Springer, 2016).

[8] Jesse van Oostrum et al., "Invariance Properties of the Natural Gradient in Overparametrised Systems," *Information Geometry*, Springer, 2022, 1–17.

Natural Gradients and Mirror Descent (ii)

The *Bregman divergence* of φ is given by

$$D_\varphi(m_1, m_2) := \varphi(m_1) - \varphi(m_2) - \nabla\varphi(m_2)^\top (m_1 - m_2).$$

Mirror descent aka proximal formulation

Given a step size $\eta > 0$, the *mirror descent* or *proximal* update is given by

$$\theta_{k+1} = \arg\min_\theta \Big\{ \nabla L(\theta_k)^\top (\theta - \theta_k) + \eta^{-1} D_\varphi(m_\theta, m_{\theta_k}) \Big\}.$$

Trust region formulation

The corresponding *trust-region* udpate is given by

$$\theta_{k+1} = \arg\min_\theta \Big\{ \nabla L(\theta_k)^\top (\theta - \theta_k) : D_\varphi(m_\theta, m_{\theta_k}) \le \eta \Big\}.$$

Advantage functions

In RL, the gradient is usually replaced by a proxy $\mathbb{A}^{\pi_k}(\pi) = \mathbb{E}_{s\sim d^{\pi_k}} \left[\frac{\pi(a|s)}{\pi_k(a|s)} \cdot A^{\pi_k}(s,a) \right]$

Entropy and Kullback-Leibler (KL) Divergence

Consider the special Bregman divergence induced by the entropy $\varphi = -H$.

Definition (Kullback-Leibler divergence)

For two probability distributions p, q on a set X, the *Kullback-Leibler (KL) divergence* is defined as

$$D_{\mathrm{KL}}(p, q) = \sum_{x \in X} p(x) \log\left(\frac{p(x)}{q(x)}\right).$$

Fisher-information matrix A very important case is, when p_θ are probability measures and one chooses the *Fisher-information matrix* given by

$$F(\theta)_{ij} := \mathbb{E}_{p_\theta}\left[\partial_{\theta_i} \ln p_\theta \cdot \partial_{\theta_j} \ln p_\theta\right].$$

Theorem 5 (Natural gradients and geodesics).

The gradients $\nabla_p D_{\mathrm{KL}}(p^\star, p)$ and $\nabla_p D_{\mathrm{KL}}(p, p^\star)$ lead to interpolation and interpolation of the log-densities:

$$p_t^{(m)} = p^\star + (1 - e^{-t})(p - p^\star) \quad \text{and} \quad p_t^{(e)} \propto (p^\star)^{1-e^{-t}} p^{e^{-t}},$$

which are know as the m-geodesic *and* e-geodesic*, respectively. Both curves converge at rate $O(e^{-t})$.*

Natural Policy Gradients

Question Which model space and geometry should we choose?

Kakade's natural policy gradient

There is one main choice[9] with $m_\theta = \pi_\theta$ and

$$G(\theta)_{ij} = \mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{\pi_\theta(\cdot|s)}\left[\partial_{\theta_i} \ln \pi_\theta(a|s) \cdot \partial_{\theta_j} \ln \pi_\theta(a|s)\right]\right].$$

On the level of trust-region and proximal methods this gives[10,11]

$$D_{\mathrm{KL}}(\pi_1, \pi_2) := \mathbb{E}_{s\sim d^{\pi_\theta}}[D_{\mathrm{KL}}(\pi_1(\cdot\,|s), \pi_2(\cdot\,|s))].$$

Facts

- The function $D_{\mathrm{KL}}(\pi_1, \pi_2)$ is **not** a Bregman divergence!
- The matrix $G(\theta)$ does not come from a Hessian of a convex function in π! Candidate: entropy $\Psi(\pi)$.

[9] Sham M Kakade, "A Natural Policy Gradient," *Advances in Neural Information Processing Systems* 14 (2001).

[10] John Schulman et al., "Trust Region Policy Optimization," in "International Conference on Machine Learning," special issue, *International Conference on Machine Learning*, 2015, 1889–97.

[11] John Schulman et al., "Proximal Policy Optimization Algorithms," *Arxiv Preprint Arxiv:1707.06347*, 2017.

Geometry of Natural Policy Gradient Methods

Natural Policy Gradients as Convex Optimization

Question

What does it mean to mix the Fisher-information matrices $G(\theta)$?

Relation to path space measures[12] and generalized Čencov axiomatics[13,14].

Conditional (relative) entropy

We define the *conditional entropy* as

$$H_{A|S}(d) := \sum_{s \in \mathcal{S}} d_S(s) H(d(\cdot\,|s)) = H(d) - H(d_S).$$

Here, $d_S(s) := \sum_a d(s,a)$. Similarly, we define the conditional KL divergence as

$$D_{A|S}(d_1, d_2) := \sum_{s \in \mathcal{S}} d_S(s) D_{\mathrm{KL}}(d_1(\cdot\,|s), d_2(\cdot\,|s)).$$

[12] J. Andrew Bagnell and Jeff G. Schneider, "Covariant Policy Search," in "Ijcai," special issue, *IJCAI*, 2003, 1019–24.

[13] Guy Lebanon, "Axiomatic Geometry of Conditional Models," *IEEE Transactions on Information Theory* 51, no. 4 (2005): 1283–94.

[14] Guido Montúfar et al., "On the Fisher Metric of Conditional Probability Polytopes," *Entropy* 16, no. 6 (2014): 3207–33.

Natural Policy Gradients as Convex Optimization (ii)

Lemma 1 (Linear programming with entropy regularization, Neu et al., 2017).

It holds that $D_{A|S}(d_1, d_2) = D_{\mathrm{KL}}(\pi_1, \pi_2)$*, hence the entropy-regularized MDP is equivalent to*[15]

$$\max r^\top d + \tau H_{A|S}(d) \quad \text{subject to} \quad d \in \mathcal{D}.$$

Theorem 6 (Geometry of natural policy gradients, M. and Montúfar, 2024).

The mapping $\pi \to d^\pi$ *is a Riemannian* ***isometry*** *with respect to Kakade metric and the metric induced by* $H_{A|S}$*.*[16] *In other words, this means that the natural gradient matrix is given by*

$$G(\theta)_{ij} = \partial_{\theta_i} d_\theta^\top \nabla^2 H_{A|S}(d_\theta) \partial_{\theta_j} d_\theta.$$

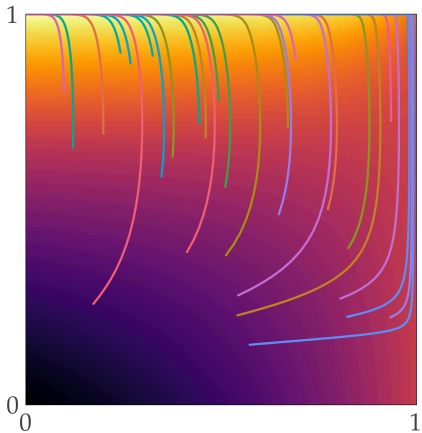
Consequently, the natural policy gradient method corresponds to the Hessian gradient flow

$$\partial_t d_t = \nabla^2 H_{A|S}(d_t)^{-1} r.$$

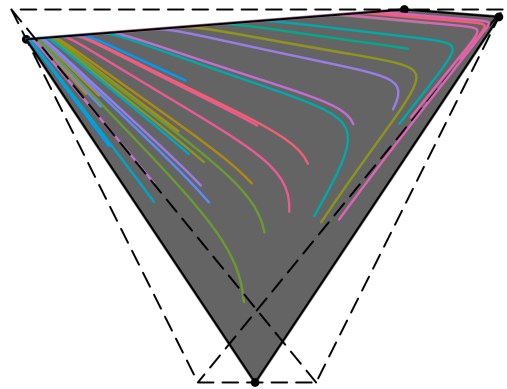
[15] Gergely Neu et al., "A Unified View of Entropy-Regularized Markov Decision Processes," *Arxiv Preprint Arxiv:1705.07798*, 2017.

[16] Johannes Müller and Guido Montúfar, "Geometry and Convergence of Natural Policy Gradient Methods," *Inf. Geom.* 7 (2024): S485–523, https://doi.org/10.1007/s41884-023-00106-z.

Natural Policy Gradients as Convex Optimization (iii)



Evolution of policies



Evolution of state-action distributions

Hessian gradient flows of linear programs

Consider a convex function φ on P and the linear program

$$\max c^\top x \quad \text{subject to} \quad x \in P$$

To understand the optimization dynamics, we study the *Hessian gradient* flow

$$\partial_t x_t = \nabla^2 \varphi(x_t)^{-1} c.$$

Lemma 2 (Hessian gradient flows and regularization paths).

It holds that[17,18]

$$x_t = \arg\max\{c^\top x - t^{-1} D_\varphi(x, x^\star) : x \in P\}.$$

Proof.

Differentiate the critical equations $\nabla\varphi(x_t) = \nabla\varphi(x^\star) + tc$.

[17] Felipe Alvarez et al., "Hessian Riemannian Gradient Flows in Convex Programming," *SIAM Journal on Control and Optimization* 43, no. 2 (2004): 477–501.

[18] Johannes Müller et al., "Fisher-Rao Gradient Flows of Linear Programs and State-Action Natural Policy Gradients," *SIAM J. Optim.* 35, no. 2 (2025): 1060–88, https://doi.org/10.1137/24M1653422.

Hessian gradient flows of linear programs (ii)

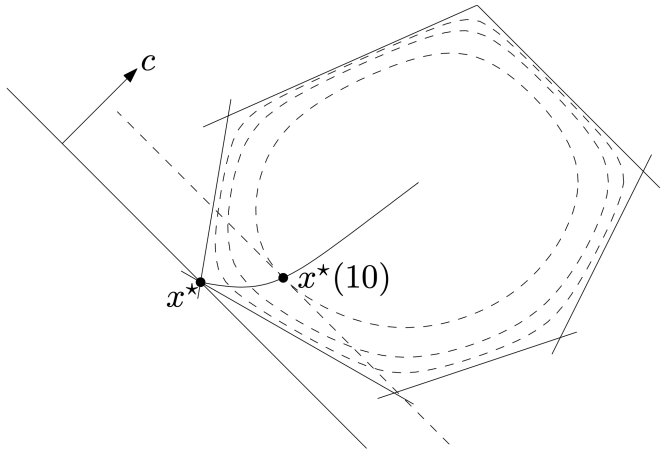


Figure 4: Visualizaton of the regularization path, also called *central path*.[19]

[19] Stephen Boyd et al., *Convex Optimization* (Cambridge university press, 2004).

Equivalence of NPG and Entropy Regularization

Theorem 7 (Central path property of natural policy gradients, M. and Çayci, 2026).

Consider the evolution of policies π_t under the natural policy gradient flow. Then, we have[20]

$$\pi_t = \arg\max_{\pi} \left\{ R(\pi) - \frac{1}{t} D_{\mathrm{KL}}(\pi, \pi_0) \right\}.$$

In particular, the policies π_t converge to the maximum entropy optimal policy

$$\pi_t \to \pi^\star = \arg\max_{\pi} \{ D_{\mathrm{KL}}(\pi, \pi_0) : R(\pi) = R^\star \}.$$

Implicit bias

- This gives a precise characterization of the implicit bias of NPG.
- Previously, $D_{\mathrm{KL}}(\pi_k, \pi^\star) = O(\ln k)$ was established[21]

[20] Johannes Müller and Semih Cayci, "Optimal Rates of Convergence for Entropy Regularization in Discounted Markov Decision Processes," *Inf. Inference* 15, no. 1 (2026): Paper No. iaaf034, 41, https://doi.org/10.1093/imaiai/iaaf034.

[21] Yuzheng Hu et al., "Actor-Critic Is Implicitly Biased Towards High Entropy Optimal Policies," in "International Conference on Learning Representations," special issue, *International Conference on Learning Representations*, 2022, https://openreview.net/forum?id=vEZyTBRPP6o.

Optimal convergence rate of entropy regularization

Theorem 8 (Optimal convergence of entropy regularization error, M. and Çayci, 2026).

Consider the problem dependent constant

$$\Delta := (1-\gamma)^{-1} \max\{A^\star(s,a) : A^\star(s,a) \neq 0, s \in \mathcal{S}, a \in \mathcal{A}\}.$$

There are polynomials p and q such that for all $\tau > 0$ we have

$$cp(\tau^{-1})e^{-\Delta\tau^{-1}} \leq R^\star - R(\pi_\tau^\star) \leq Cq(\tau^{-1})e^{-\Delta\tau^{-1}}.$$

Theorem 9 (Improved overall error bound, M. and Çayci, 2026).

With $\tau = \sqrt{\frac{2\Delta}{\eta k}}$ it holds for the entropy-regularized NPG method that

$$R^\star - R(\pi_k) \leq O\left(p(\eta k)e^{-\sqrt{\frac{\Delta\eta k}{2}}}\right).$$

Comparisons

- Compare this to the guarantee for the unregularized NPG method: $O\big(e^{-\Delta\eta k}\big)$.
- Compare this to the old $O\big(\ln(k)k^{-1}\big)$ guarantee for $\eta = \ln(k)k^{-1}$.

Embedding Safety into RL via Geometry

Constrained MDPs

Constrained MDPs

Extension of MDPs such that for $i = 1, ..., m$ we have *cost functions* $c_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and thresholds b_i. It is now the goal to maximize the reward subject to the constraints

$$\max_{\pi} R(\pi) \quad \text{subject to} \quad (1-\gamma)\mathbb{E}_{\pi}\left[\sum_{k\in\mathbb{N}} c_i(S_k, A_k)\right] \leq b_i \quad \text{for all} \quad i = 1, ..., m.$$

Common approaches

- *Lagrangian methods* Use $\max_{\pi} \min_{\lambda \geq 0} \mathcal{L}(\pi, \lambda)$, where $\mathcal{L}(\pi, \lambda) = R(\pi) - \lambda(C(\pi) - b)$.
  - Can be unstable, allows for unsafe policies during optimization
- *Projection methods*
  - Computationally expensive, sacrifice in performance
- *Penalty methods*
  - Introduce bias

A safe geometry

The safe state-action distributions are given by the

$$\mathcal{D}_{\text{safe}} := \{d \in \mathcal{D} : c_i^\top d \leq b_i \quad \text{for all} \quad i = 1, ..., m\}.$$

Legendre or mirror functions

Given a convex set $C \subseteq \mathbb{R}^d$ we call $\varphi : \text{int}(C) \to \mathbb{R}$ *Legendre function* if

$$\|\nabla \varphi(x)\| \to +\infty \quad \text{for} \quad x \to \partial C.$$

This corresponds to an infinite curvature at the boundary.

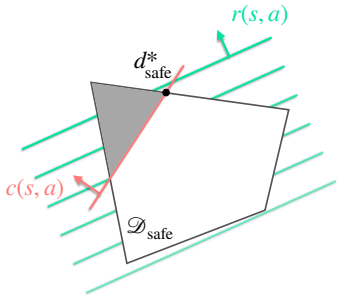
Why are Legendre functions important?

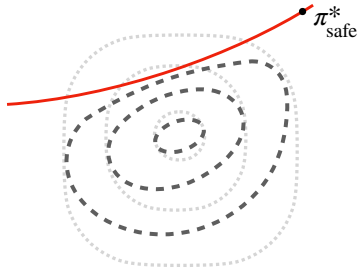
Ensure that natural gradients, mirror descent, and trust-region methods are guaranteed to stay in $\text{int}(C)$.

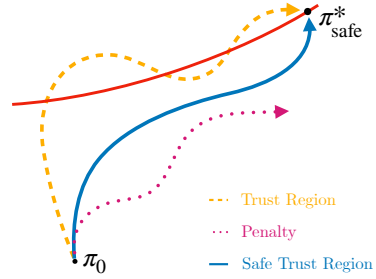
Examples Consider the convex set $C = \{x : f_i(x) \leq d_i \quad \text{for} \quad i = 1, ..., m\}$ for convex functions f_i. Then:

- *Entropy penalty* $\varphi(x) = \sum_{i=1}^m (d_i - f_i(x)) \ln(d_i - f_i(x))$
- *Logarithmic barrier* $\varphi(x) = \sum_{i=1}^m -\ln(d_i - f_i(x))$

A safe geometry (ii)







Legendre functions for constrained MDPs

$$\varphi_{\mathrm{C}}(d) = H_{A|S}(d) + \beta \sum_{i=1}^{m} (b_i - c_i^\top d) \ln(b_i - c_i^\top d).$$

Constrained Trust Region Policy Optimization (C-TRPO)

Constrained trust region policy optimization (C-TRPO)

Extension of TRPO, where we use the safe trust region[22]. This yields

$$\pi_{k+1} = \arg\max_{\pi} \mathbb{A}_r^{\pi_k}(\pi) \quad \text{sbj to} \quad D_{\mathrm{KL}}(\pi, \pi_k) + \beta D_{\mathrm{B}}(\pi, \pi_k) \leq \eta.$$

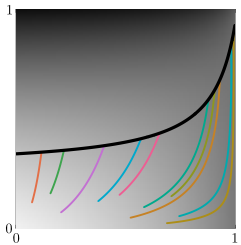


Figure 10: $\beta = 0.0001$

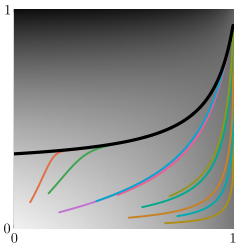


Figure 11: $\beta = 0.01$

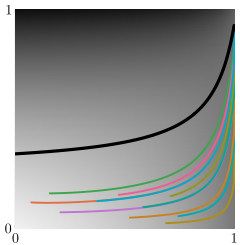


Figure 12: $\beta = 1.0$

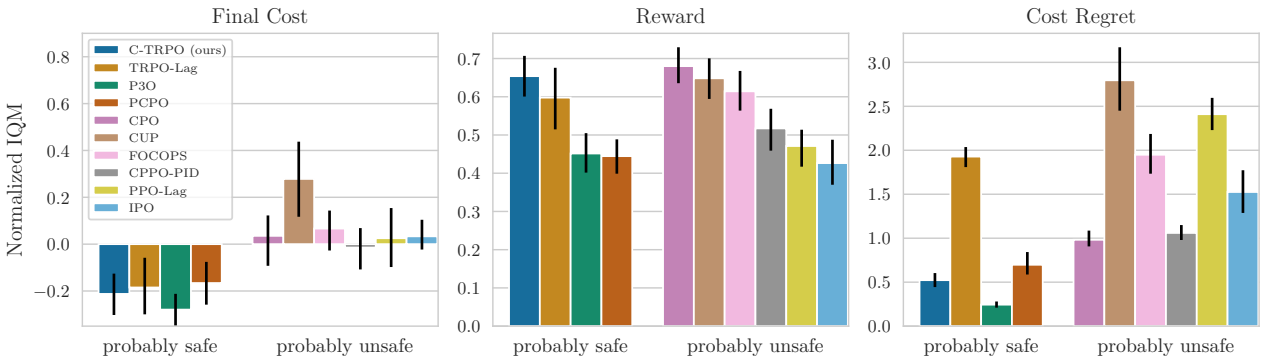
[22] Nikola Milosevic et al., "Embedding Safety into RL: A New Take on Trust Region Methods," in "Forty-Second International Conference on Machine Learning," special issue, *Forty-Second International Conference on Machine Learning*, 2025, https://openreview.net/forum?id=4zRb89SbzG.

Constrained Trust Region Policy Optimization (C-TRPO) (ii)

Algorithm 1: Constrained trust region policy optimization (C-TRPO)

▷ **Input** Safety parameter $\beta > 0$, recovery parameter $b_{\mathrm{H}} \in (0, b]$
for $k = 0, 1, 2, \ldots$ **do**
  ▷ Sample trajectories from $\pi_k = \pi_{\theta_k}$
  if $\pi_k \in \Pi^{\mathrm{H}}_{\mathrm{safe}}$ **then**
    $\mathbb{A} \leftarrow \mathbb{A}_r$
    $D \leftarrow D_{\mathrm{KL}} + \boldsymbol{\beta D_{\mathrm{C}}}$
  else
    $\boldsymbol{\mathbb{A} \leftarrow -\mathbb{A}_c}$
    $D \leftarrow D_{\mathrm{KL}}$
  end
  ▷ Compute π_{k+1} with one TRPO step with A and D
end

Constrained Trust Region Policy Optimization (C-TRPO) (iii)



Takeaway

Higher reward with increased safety!

Central Path Proximal Policy Optimization (C3PO)

Goal

Improve scalablity as C-TRPO relies on a linear solve (CG typically) for the trust-region update!

Reminder of PPO

TRPO maximizes the surrogate

$$\mathbb{A}(\theta) = \mathbb{E}_{s,a\sim d^{\pi_k}}\left[\frac{\pi_\theta(a|s)}{\pi_k(a|s)} \cdot A^{\pi_k}(s,a)\right] = \mathbb{E}_{s,a\sim d^{\pi_k}}[\rho_\theta(s,a) \cdot A^{\pi_k}(s,a)].$$

Proximal policy optimization (PPO) maximizes the clipped surrogate

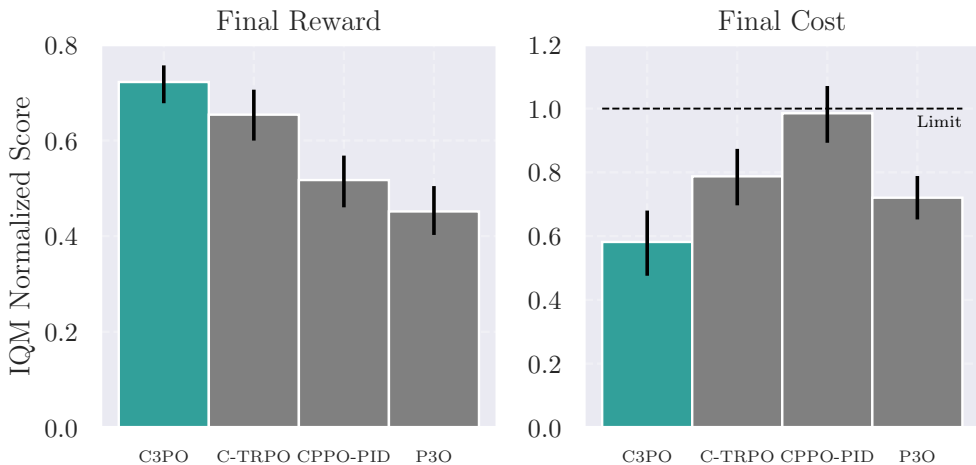
$$L_{\text{PPO}}(\theta) = \mathbb{E}_{s,a\sim d^{\pi_k}}[\min(\rho_\theta A^{\pi_k}(s,a), \text{clip}(\rho_\theta, 1-\varepsilon, 1+\varepsilon) \cdot A^{\pi_k}(s,a))].$$

Approach

Central path proximal policy optimization (C3PO) adds a second term L_{C3PO}[23].

[23] Nikola Milosevic et al., "Central Path Proximal Policy Optimization," in "The Exploration in AI Today Workshop at ICML 2025," special issue, *The Exploration in AI Today Workshop at ICML 2025*, 2025, https://openreview.net/forum?id=2cvUHCgZbF.

Central Path Proximal Policy Optimization (C3PO) (ii)



Conclusion

Conclusion

What can we learn from the geometry of RL?

- *Implicit bias:* Unregularized NPG $\longleftrightarrow$ entropy regularization
- Optimal exponential convergence $\tilde{O}(e^{-\Delta\tau^{-1}})$ for entropy regularization error
- C-TRPO and C3PO: Guaranteed safety, no penalty error

Outlook

- Extension to other settings in RL: unsupervised, contextual, multi-agent, ...
- Improved convergence rates for regularized natural policy gradients
- Improved sample complexity bounds for regularized natural policy gradients
- Continuous state and action spaces
- Flow-based generative models: three different levels with u^θ, p^θ, $\mathbb{P}^\theta$

References

Alvarez, Felipe, Jérôme Bolte, and Olivier Brahic. “Hessian Riemannian Gradient Flows in Convex Programming.” *SIAM Journal on Control and Optimization* 43, no. 2 (2004): 477–501.

Amari, Shun-ichi. *Information Geometry and Its Applications*. Vol. 194. Springer, 2016.

Amari, Shun-Ichi. “Natural Gradient Works Efficiently in Learning.” *Neural Computation* 10, no. 2 (1998): 251–76.

Bagnell, J. Andrew, and Jeff G. Schneider. “Covariant Policy Search.” In “Ijcai.” Special issue, *IJCAI*, 2003, 1019–24.

Boyd, Stephen, Stephen P Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Cen, Shicong, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. “Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization.” *Operations Research*, INFORMS, 2021.

Derman, Cyrus. *Finite State Markovian Decision Processes*. Academic Press, 1970.

References (ii)

Geist, Matthieu, Bruno Scherrer, and Olivier Pietquin. “A Theory of Regularized Markov Decision Processes.” In “International Conference on Machine Learning.” Special issue, *International Conference on Machine Learning*, 2019, 2160–69.

Hu, Yuzheng, Ziwei Ji, and Matus Telgarsky. “Actor-Critic Is Implicitly Biased Towards High Entropy Optimal Policies.” In “International Conference on Learning Representations.” Special issue, *International Conference on Learning Representations*, 2022. https://openreview.net/forum?id=vEZyTBRPP6o.

Kakade, Sham M. “A Natural Policy Gradient.” *Advances in Neural Information Processing Systems* 14 (2001).

Khodadadian, Sajad, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. “On the Linear Convergence of Natural Policy Gradient Algorithm.” In “2021 60th IEEE Conference on Decision and Control (Cdc).” Special issue, *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, 3794–99.

Lebanon, Guy. “Axiomatic Geometry of Conditional Models.” *IEEE Transactions on Information Theory* 51, no. 4 (2005): 1283–94.

References (iii)

Milosevic, Nikola, Johannes Müller, and Nico Scherf. “Central Path Proximal Policy Optimization.” In “The Exploration in AI Today Workshop at ICML 2025.” Special issue, *The Exploration in AI Today Workshop at ICML 2025*, 2025. https://openreview.net/forum?id=2cvUHCgZbF.

Milosevic, Nikola, Johannes Müller, and Nico Scherf. “Embedding Safety into RL: A New Take on Trust Region Methods.” In “Forty-Second International Conference on Machine Learning.” Special issue, *Forty-Second International Conference on Machine Learning*, 2025. https://openreview.net/forum?id=4zRb89SbzG.

Montúfar, Guido, Johannes Rauh, and Nihat Ay. “On the Fisher Metric of Conditional Probability Polytopes.” *Entropy* 16, no. 6 (2014): 3207–33.

Müller, Johannes, and Semih Cayci. “Optimal Rates of Convergence for Entropy Regularization in Discounted Markov Decision Processes.” *Inf. Inference* 15, no. 1 (2026): Paper No. iaaf034, 41. https://doi.org/10.1093/imaiai/iaaf034.

Müller, Johannes, and Guido Montúfar. “Geometry and Convergence of Natural Policy Gradient Methods.” *Inf. Geom.* 7 (2024): S485–523. https://doi.org/10.1007/s41884-023-00106-z.

References (iv)

Müller, Johannes, Semih Çayci, and Guido Montúfar. "Fisher-Rao Gradient Flows of Linear Programs and State-Action Natural Policy Gradients." *SIAM J. Optim.* 35, no. 2 (2025): 1060–88. https://doi.org/10.1137/24M1653422.

Neu, Gergely, Anders Jonsson, and Vicenç Gómez. "A Unified View of Entropy-Regularized Markov Decision Processes." *Arxiv Preprint Arxiv:1705.07798*, 2017.

Oostrum, Jesse van, Johannes Müller, and Nihat Ay. "Invariance Properties of the Natural Gradient in Overparametrised Systems." *Information Geometry*, Springer, 2022, 1–17.

Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust Region Policy Optimization." In "International Conference on Machine Learning." Special issue, *International Conference on Machine Learning*, 2015, 1889–97.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal Policy Optimization Algorithms." *Arxiv Preprint Arxiv:1707.06347*, 2017.

Sutton, Richard S, David A McAllester, Satinder P Singh, Yishay Mansour, and others. "Policy Gradient Methods for Reinforcement Learning with Function Approximation." In "Nips." Special issue, *Nips* 99 (1999): 1057–63.

Appendix

Value and advantage functions

The *value function* of a policy π is given by

$$V^\pi(s) := \mathbb{E}_\pi \left[\sum_{k \in \mathbb{N}} \gamma^k r(S_k, A_k) \mid S_0 = s \right].$$

The *action-value function* or *Q-function* of a policy π is given by

$$Q^\pi(s,a) := \mathbb{E}_\pi \left[\sum_{k \in \mathbb{N}} \gamma^k r(S_k, A_k) \mid S_0 = s, A_0 = a \right].$$

The *advantage function* of a policy π is given by

$$A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s).$$

We define the optimal value function $V^\star$ and optimal action-value function $Q^\star$ as

$$V^\star(s) := \max_\pi V^\pi(s) \quad \text{and} \quad Q^\star(s,a) := \max_\pi Q^\pi(s,a)$$

and set

$$A^\star(s,a) := Q^\star(s,a) - V^\star(s).$$

Reminder on TRPO

Reward proxy

We consider the approximation of the reward

$$\mathbb{A}_r^{\pi_k}(\pi) := \mathbb{E}_{s,a\sim d^{\pi_k}}\left[\frac{\pi(a|s)}{\pi_k(a|s)} \cdot A_r^{\pi_k}(s,a)\right] \approx \mathbb{E}_{s,a\sim d^{\pi}}\left[\frac{\pi(a|s)}{\pi_k(a|s)} \cdot A_r^{\pi_k}(s,a)\right] = R(\pi).$$

Trust region policy optimization (TRPO)

TRPO performs the update

$$\max_{\pi} \mathbb{A}_r^{\pi_k}(\pi) \quad \text{subject to} \quad D_{\mathrm{KL}}(\pi, \pi_k) \leq \eta.$$

Here, the trust-region is usually **approximated quadratically** and then a **CG** is performed to solve the corresponding qudratic program. This is equivalent to a **natural policy gradient** step up to the step size:

$$\delta\theta_{\mathrm{TRPO}} = \sqrt{\frac{2\eta}{g^\top G(\theta_k)^{-1} g}} \cdot G(\theta_k)^{-1} g = \sqrt{\frac{2\eta}{g^\top G(\theta_k)^{-1} g}} \cdot \delta\theta_{\mathrm{NPG}}.$$

Theoretical guarantees for C-TRPO

Lemma 3 (Almost improvement).

Set $\varepsilon_r = \max_s |\mathbb{E}_{a\sim\pi_{k+1}} A_r^{\pi_k}(s,a)|$. *The expected reward of a policy updated with C-TRPO is bounded from below by*

$$V_{r(\pi_{k+1})} \geq V_{r(\pi_k)} - \frac{\sqrt{2\delta}\gamma\varepsilon_r}{1-\gamma}.$$

Lemma 4 (C-TRPO worst-case constraint violation).

Let $\overline{\{D\}}_{C(\pi_{k+1} \parallel \pi_k)} \leq \delta$ *with* $\delta > 0$ *and set* $\varepsilon_c = \max_s |\mathbb{E}_{a\sim\pi_{k+1}} A_c^{\pi_k}(s,a)|$. *It holds that*

$$V_c(\pi_{k+1}) \leq V_c(\pi_k) + \mathbb{A}_c^{\pi_k}(\pi_{k+1}) + \frac{\sqrt{2\delta(\beta)}\gamma\varepsilon_c}{1-\gamma},$$

where $\delta(\beta) = \delta - \beta D_\varphi(\pi_{k+1}, \pi_k) \leq \delta$ *is decreasing in* $\beta > 0$, $\lim_{\beta\to 0}\delta(\beta) = \delta$, *and* $\delta(\beta) \to 0$ *for*

$$\beta \to \frac{\delta D_{\mathrm{C}}(\pi_{k+1}, \pi_k)}{D_\varphi}(\pi_{k+1}, \pi_k).$$

Theoretical guarantees for C-TRPO (ii)

Theorem 10 (Safety during training).

Assume that $\varphi : \mathbb{R}_{>0} \to \mathbb{R}$ satisfies $\varphi'(x) \to +\infty$ for $x \to 0$ and consider a regular policy parameterization. Then the set Θ_{safe} is invariant under the natural policy gradient flow

$$\partial_t \theta_t = G_C(\theta_t)^{-1} \nabla R(\theta_t).$$

Theorem 11.

Assume that $\varphi'(x) \to +\infty$ for $x \to 0$, set $V^\star_{r,C} := \max_{\pi \in \Pi_{\text{safe}}} V_{r(\pi)}$ and denote the set of optimal constrained policies by $\Pi^\star_{\text{safe}} = \left\{\pi \in \Pi_{\text{safe}} : V_{r(\pi)} = V^\star_{r,C}\right\}$, consider a regular policy parametrization and let $(\theta_t)_{t \geq 0}$ solve the natural policy gradient flow. It holds that $V_{r(\pi_{\theta_t})} \to V^\star_{r,C}$ and

$$\lim_{t \to +\infty} \pi_t = \pi^\star_{\text{safe}} = \arg\min\{D_C(\pi^\star, \pi_0) : \pi^\star \in \Pi^\star_{\text{safe}}\}.$$

Implicit bias

The resulting policy has the least amount of active constraint violation among all optimal policies.

Details for C3PO

Consider the slight variation of the C-TRPO update

$$\max_{\pi} \mathbb{A}_r^{\pi_k}(\pi) \quad \text{subject to} \quad D_{\mathrm{B}}(\pi, \pi_k) \leq \delta_{\mathrm{B}} \quad \text{and} \quad D_{\mathrm{KL}}(\pi, \pi_k) \leq \delta_{\mathrm{KL}}.$$

Instead of solving this constrained problem directly, we consider the penalized problem given by

$$\max_{\pi} \mathbb{A}_r^{\pi_k}(\pi) - \kappa \max\{0, D_{\mathrm{B}}(\pi, \pi_k) - \delta_{\mathrm{B}}\} \quad \text{subject to} \quad D_{\mathrm{KL}}(\pi, \pi_k) \leq \delta_{\mathrm{KL}}.$$

Theorem 12 (Exactness).

Let λ be the Lagrange multiplier vector for the optimizer doubly constrained upate. Then for $\kappa \geq |\lambda|$, the solution sets of problem the doubly constrained and penalized problem agree.

We define the ratio $\rho(\theta) = \frac{\pi_\theta(a|s)}{\pi_k(a|s)}$ and set

$$\alpha_{\text{clipped}}(\theta) := \mathbb{E}_{a,s\sim d^{\pi_k}}\left[\max\Big(\rho(\theta)\hat{A}_c(s,a), \text{clip}(\rho(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_c(s,a)\Big)\right]$$

and define the C3PO objective as

$$L_{\text{C3PO}}(\theta) := \text{ReLU}(\alpha_{\text{clipped}}(\theta) - \min\{b, wb\},$$

where b is the threshold and $w \in (0,1)$ is a hyper-parameter.

Reparametrization invariance of GN

Consider the problem $\min_w f(P(w))$ and the corresponding Gauss-Newton flow

$$\partial_t w_t = -\left(\nabla J_t^\top J_t\right)^{-1} J_t^\top \nabla f,$$

where $J_t = DP(w_t)$ and $\nabla f(P(w_t)) = J_t^\top \nabla f$.

Now, consider a reparametrization $w = \psi(v)$ for an invertible ψ. Then, the Gauss-Newton flow for the reparametrized problem $\min_v f(P(\psi(v)))$ becomes

$$\partial_t v_t = -\left(\tilde{J}_t^\top \tilde{J}_t\right)^{-1} \tilde{J}_t^\top \nabla f,$$

where $\tilde{J}_t = DP(\psi(v_t))$. Now, considering the dynamics of $P(w_t)$ we get

$$\partial_t P(w_t) = J_t \partial_t w_t = -J_t \left(\nabla J_t^\top J_t\right)^{-1} J_t^\top \nabla f.$$

To understand the dynamics of $P(\psi(v_t))$, the chain rule gives $\tilde{J}_t = J_t B_t$ (for $B_t = D\psi(v_t)$), we have

$$\partial_t P(\psi(v_t))) = \tilde{J}_t \partial_t v_t = -\tilde{J}_t \left(\tilde{J}_t^\top \tilde{J}_t\right)^{-1} \tilde{J}_t^\top \nabla f = -J_t B_t \left(B_t^\top J_t^\top J_t B_t\right)^{-1} B_t^\top J_t^\top \nabla f.$$

We see that the B_t cancels.