Randomized Advantage Transformation

Natural Policy Gradients via Direct Backpropagation

Mingfei Sun

The University of Manchester | ICML 2026

Natural policy gradients are powerful — but expensive

Natural policy gradients pre-condition the gradient by the inverse **Fisher matrix** $\boldsymbol{F}^{-1}$, giving parameterization-invariant, faster-converging updates.

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{NPG}} J \;=\; \boldsymbol{F}^{-1} \nabla_{\boldsymbol{\theta}}^{\mathrm{PG}} J$$

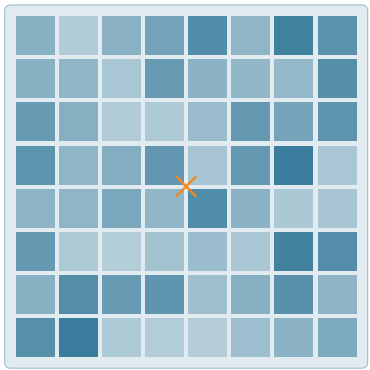
The catch

$\boldsymbol{F} \in \mathbb{R}^{|\theta| \times |\theta|}$ scales with the *number of parameters*. For deep nets it is infeasible to form or invert.

Prior fixes each pay a price:

- **Hessian-free (TRPO):** conjugate-gradient inner loops — costly, hard to tune.
- **KFAC / structured:** architecture-specific factorizations.

$\boldsymbol{F} \in \mathbb{R}^{|\theta| \times |\theta|}$



ill-conditioned
hard to invert

Key idea: move the inverse *into* the advantage

Regularized (Tikhonov) NPG is a regularized least-squares problem. Applying the **Woodbury identity** twice rewrites it so the parameter-sized inverse disappears:

$$\nabla_\theta^{\text{T-NPG}} J = (\lambda \boldsymbol{I}_p + \boldsymbol{H}^\top \Sigma \boldsymbol{H})^{-1} \boldsymbol{H}^\top \Sigma \boldsymbol{y} \;=\; \boldsymbol{H}^\top \Sigma \underbrace{(\lambda \boldsymbol{I}_n + \boldsymbol{H}\boldsymbol{H}^\top \Sigma)^{-1} \boldsymbol{y}}_{\text{transformed advantage } \bar{\boldsymbol{A}}}$$

What changed

Same form as a *vanilla* policy gradient $\boldsymbol{H}^\top \Sigma \boldsymbol{y}$ — only the advantage is replaced by $\bar{\boldsymbol{A}}$.

Why it matters

The inversion is now $n \times n$ (samples), **not** $|\theta| \times |\theta|$ (parameters). No explicit Fisher, no FVP, no architecture assumptions.

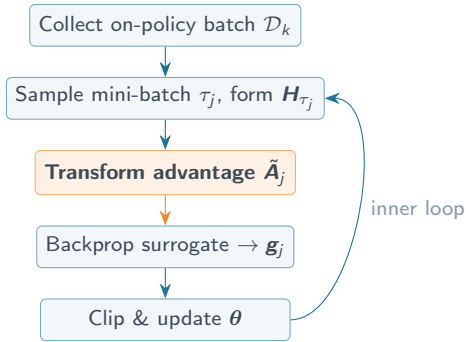
RAT: estimate it with randomized block Kaczmarz

n (state–action pairs) is still huge. RAT solves the least-squares *iteratively* on small on-policy mini-batches τ_j of size B:

$$\tilde{\boldsymbol{A}}_j(s,a) = \Big[(\lambda \boldsymbol{I} + \boldsymbol{H}_{\tau_j}\boldsymbol{H}_{\tau_j}^\top)^{-1}\big(\boldsymbol{y}_{\tau_j} - \boldsymbol{H}_{\tau_j}\boldsymbol{g}_{j-1}\big)\Big]_{(s,a)}$$

- Inversion is only $B \times B$ (when $B \ll p$).
- $\boldsymbol{H}\boldsymbol{H}^\top$ is the neural tangent kernel.
- Each step is **one backprop** through a PPO-like surrogate:

$$J_{\mathrm{RAT}}(\boldsymbol{\theta}) = \mathbb{E}\Big[\tfrac{\pi(a|s;\theta)}{\pi_{\mathbf{old}}(a|s)}\,\tilde{\boldsymbol{A}}_j(s,a)\Big]$$



Convergence guarantees

Under full column rank of $\boldsymbol{H}$ and state–action coverage, define $\mu := \lambda_{\min}(\mathbb{E}[\boldsymbol{P}_\tau]) > 0$.

Thm 1 — Exact targets

Linear convergence to the regularized NPG:

$$\mathbb{E}\|\boldsymbol{g}_j - \boldsymbol{g}^*\|_2^2 \le (1-\mu)^j \, \|\boldsymbol{g}_0 - \boldsymbol{g}^*\|_2^2$$

Thm 2 — Noisy targets (RL)

Linear convergence to an error floor:

$$\mathbb{E}\|\boldsymbol{g}_j - \boldsymbol{g}^*\|_2^2 \le (1-\mu)^j \|\boldsymbol{g}_0 - \boldsymbol{g}^*\|_2^2 + \frac{\eta^2}{\mu}$$

- Rate is set by the spectrum of $\mathbb{E}[\boldsymbol{P}_\tau]$; the error floor η^2/μ motivates **gradient-norm clipping**.
- Interleaving with policy updates = a contractive solver tracking a *slowly varying* target.

RAT matches empirical natural gradients

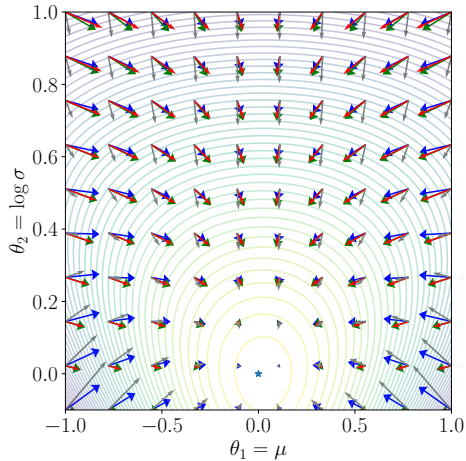
On a univariate Gaussian with a closed-form Fisher, RAT's updates align with the **empirical natural gradient**.

- Consider a policy π_θ with mean μ and log standard deviation $\log \sigma$ parameterized by θ_1 and θ_2:

$$\mathcal{N}(x|\mu, \sigma; \theta_1, \theta_2)$$

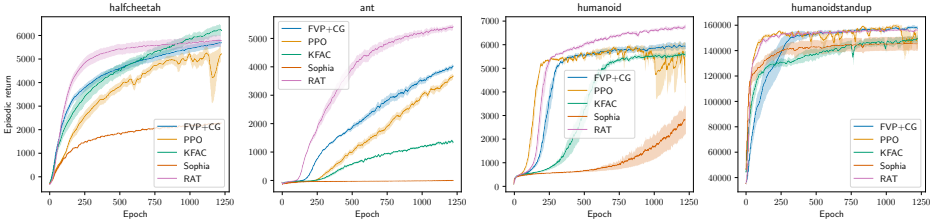
- $\mu = \theta_1$ and $\log \sigma = \theta_2$.

gray vanilla | closed-form natural | empirical natural | RAT gradients | $\star$ optimum



RAT outperforms strong baselines

RAT learns faster and higher than KFAC, FVP+CG, PPO, and Sophia:



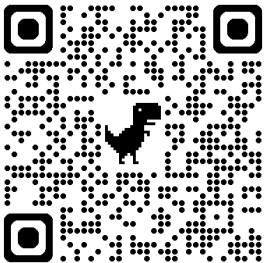
RAT applies a **pseudo-advantage** to the critic in shared actor-critic nets:

Ep. Returns	Swimmer ↑	Hopper ↑	HalfCheetah ↑	Walker2d ↑	Ant ↑	Humanoid ↑	HumanoidStandup ↑
$\mathcal{S} \times \mathcal{A}$	8×2	11×3	17×6	17×6	105×8	376×17	376×17
RAT (Ours)	$271.6^{\pm 36.3}$	$2334.6^{\pm 524.9}$	$4629.2^{\pm 287.4}$	$3156.0^{\pm 293.6}$	$2926.6^{\pm 353.1}$	$5382.7^{\pm 117.3}$	$146529.7^{\pm 2317.6}$
ACKTR	$59.1^{\pm 13.0}$	$2138.9^{\pm 171.6}$	$3630.9^{\pm 282.6}$	$2576.6^{\pm 154.6}$	$23.4^{\pm 3.2}$	$2571.7^{\pm 838.7}$	$127928.5^{\pm 5433.7}$
PPO	$191.3^{\pm 32.7}$	$2346.8^{\pm 202.7}$	$4146.0^{\pm 107.5}$	$2225.3^{\pm 303.4}$	$1373.9^{\pm 26.0}$	$5357.9^{\pm 150.9}$	$130014.2^{\pm 6463.7}$
Sophia	$57.9^{\pm 5.9}$	$1104.0^{\pm 90.6}$	$899.5^{\pm 113.2}$	$1256.0^{\pm 129.7}$	$-7.0^{\pm 1.4}$	$669.4^{\pm 56.2}$	$111212.6^{\pm 13449.9}$

- Best returns on most tasks; large gains on **Ant** and **Humanoid**.
- Ablations: both the transform *and* clipping are essential.

Takeaways

- Woodbury turns the **Fisher inverse** into a **transformed advantage** — NPG becomes a vanilla PG.
- **RAT** estimates it with randomized block Kaczmarz: **one backprop**, no CG solver, no Fisher, architecture-agnostic.
- **Linear convergence** guarantees + strong results on MuJoCo and Procgen, including shared actor–critic nets.



Code: `github.com/agent-lab/ICML2026-RAT` |

mingfei.sun@manchester.ac.uk