Who Needs Attention Anyway?

Geometric Inference for Streaming State Space Models

Dario Fumarola

Abstract

We introduce *Geometry-Aware Streaming State Space Models* (GeoSSMs), a constant-latency framework that augments selective-scan SSMs with a decoder-induced pullback (Fisher) metric in latent space. GeoSSM performs a *predict-then-correct* update each step: after the frozen SSM predicts $z_{t+1|t}$, we compute a natural direction $v = -(\lambda I + UU^\top)^{-1}\nabla\ell(z_{t+1|t})$ using a strictly constant-time *Woodbury* solve, then apply a metric-capped geodesic trust-region correction $z_{t+1} = \mathrm{Retr}_{z_{t+1|t}}(\alpha v)$ with standard TR acceptance ratio η and adaptive radius ρ. The metric $\mathbf{G}$ is a low-rank EMA sketch of the pullback/Fisher tensor from decoder Jacobians (Gauss–Newton), updated online without touching SSM weights. Across (i) **curved 2-D worlds** (nonstationary navigation) and (ii) **molecular control** (peptide torsions), GeoSSM adapts without policy retraining, reducing steps-to-goal/constraint by 39–46%, collisions/clashes by 2×–4×, and a normalized energy proxy by $\approx 35\%$ at matched accuracy; latency remains flat with low p99. Ablations isolate the value of decoder-derived geometry versus Euclidean and latent-preconditioned controls, and show graceful degradation under tighter budgets (r, λ, β).

1 Introduction

Transformers excel at long-range dependencies but incur quadratic attention costs and unpredictable latency. Modern *state space models* (SSMs) deliver linear-time, constant-memory scanning, providing a natural substrate for streaming agents. Yet standard SSMs evolve in a Euclidean latent, ignoring curvature and task constraints.

We propose **GeoSSM**: a frozen SSM backbone enhanced with a *decoder-induced pullback (Fisher) metric* that guides a *geodesic trust-region* correction at *inference time*. The result is instant, constant-latency re-planning in nonstationary settings—agents that *think in curves, not tokens*.

Contributions.

1. **Predict-then-correct geometry.** A pullback/Fisher metric on the SSM latent yields natural directions and geodesic trust-region corrections with a constant-time Woodbury solve.
2. **Streaming adaptation w/o retraining.** The SSM parameters are frozen; only metric statistics adapt via a low-rank EMA from decoder Jacobians (Gauss–Newton/Fisher).
3. **Unified demos & rigor.** Two domains—curved 2-D navigation and peptide torsion control—share one spine; we add latency-matched planner baselines (iLQR-lite, MPC wrapper), nonstationarity panels, p99 latency, CIs, and a (λ, r, β) grid.
4. **Analysis.** We formalize the step as Riemannian steepest descent with standard TR guarantees and provide a contraction-style stability statement in a local region.

2 Background

Selective-scan SSMs. We use a compact SSM block with selective scan as the backbone: $z_{t+1|t} = f_\theta(z_t, x_t)$, decoder $g_\phi : \mathcal{Z} \to \mathcal{Y}$ produces predictions/actions. Inference runs in $O(1)$ memory per step.

Pullback (Fisher) metric. Let $p_\phi(y \mid z)$ be the decoder likelihood and $\mathbf{I}_{\text{obs}}(z)$ its Fisher in observation space. The pullback metric on $\mathcal{Z}$ is

$$\mathbf{G}(z) \;=\; \mathbf{J}_g(z)^\top \mathbf{I}_{\text{obs}}(z)\, \mathbf{J}_g(z) \;+\; \lambda I, \tag{1}$$

with $\lambda > 0$ ensuring SPD. For Gaussian decoders, $\mathbf{I}_{\text{obs}}$ is the inverse covariance; for general likelihoods we use Gauss–Newton.

Natural gradient and trust regions. The steepest descent direction under $\mathbf{G}$ is $v = -\mathbf{G}^{-1}\nabla\ell$. Trust regions use a model $m(\alpha)$ to compare predicted vs. actual decrease and adapt the radius ρ.

3 Method: GeoSSM (Predict-then-Correct)

At each step, we (1) *predict* with the frozen SSM, (2) build a low-rank sketch of the pullback metric at $z_{t+1|t}$, (3) compute a natural direction with a Woodbury solve, and (4) *correct* via a geodesic TR step.

3.1 Metric from decoder Jacobians (low-rank EMA)

We estimate $\mathbf{G}(z)$ from JVPs/VJPs without forming $\mathbf{J}_g$. With r probe directions $q_j \in \mathbb{R}^{d_z}$ (see sampling below), define

$$u_j \;=\; \mathbf{J}_g(z)^\top \left(W(z)\, \mathbf{J}_g(z)\, q_j \right) \in \mathbb{R}^{d_z}, \qquad W(z) \approx \mathbf{I}_{\text{obs}}(z),$$

and stack $U^{\text{new}} = [u_1, \ldots, u_r] \in \mathbb{R}^{d_z \times r}$. We maintain an EMA

$$U_t \;\leftarrow\; \beta U_{t-1} \;+\; \sqrt{1-\beta^2}\, U^{\text{new}}, \qquad \mathbf{G}_t \;=\; \lambda I + U_t U_t^\top, \tag{2}$$

with $\beta \in (0, 1)$. **Probe sampling.** We use a *loss-aware* sketch with $q_1 = \nabla\ell(z), q_2, \ldots, q_{r-1}$ i.i.d. Rademacher (Hutchinson), and q_r the previous accepted direction (stabilizes dynamics).

3.2 Horizon surrogate and TR model

We evaluate a small-horizon surrogate around $z = z_{t+1|t}$:

$$m(\alpha) \;=\; \sum_{h=0}^{H-1} \gamma^h\, \ell\Big(\Phi_h\big(\text{Retr}_z(\alpha v)\big)\Big), \quad \Phi_h\text{: } h\text{-step rollout of } f_\theta \text{ (weights frozen)}. \tag{3}$$

We linearize $m(\alpha)$ at $\alpha = 0$ and use a quadratic model $\widehat{m}(\alpha) = m(0) + \alpha\langle -\nabla\ell(z), v\rangle - \frac{1}{2}\alpha^2 v^\top \mathbf{G}_t v$. The *predicted* decrease is $\Delta_{\text{pred}} = \widehat{m}(0) - \widehat{m}(\alpha)$. The *actual* decrease is $\Delta_{\text{act}} = m(0) - m(\alpha)$ using the short rollout.

Input : Frozen SSM f_θ, decoder g_ϕ, TR radius ρ, EMA β, rank r, damping λ, horizon H, discount γ, thresholds $\eta_{\mathrm{lo}}, \eta_{\mathrm{hi}}$.
State : Latent z_t, U_{t-1}, hardware latency budget $\tau_{\max}$ (soft cap).
Predict: $z \leftarrow f_\theta(z_t, x_t)$ // $z \equiv z_{t+1|t}$
Metric: Sample probes $q_1 = \nabla\ell(z), q_{2..r}$ (Rademacher), q_r =prev dir;
Build $U^{\mathrm{new}} = [\mathbf{J}_g^\top (W\,\mathbf{J}_g q_j)]_{j=1}^r$ via JVP/VJP;
$U \leftarrow \beta U_{t-1} + \sqrt{1-\beta^2}U^{\mathrm{new}}$;
$\mathbf{G} \leftarrow \lambda I + UU^\top$;
Gradient: $g \leftarrow \nabla_z \ell(z)$
Solve (Woodbury): Factor $R = \mathrm{chol}(\lambda I + U^\top U)$;
$v \leftarrow -\frac{1}{\lambda}\big(g - UR^{-\top}R^{-1}U^\top g\big)$
Backtrack: Choose largest $\alpha \in \{\alpha_0, \alpha_0/2, \alpha_0/4\}$ s.t. $\|\alpha v\|_{\mathbf{G}} \le \rho$;
TR check: Compute $\Delta_{\mathrm{pred}} = \alpha\langle -g, v\rangle - \frac{1}{2}\alpha^2 v^\top \mathbf{G} v$;
Compute $m(0)$ and $m(\alpha)$ via H-step rollout (Section 3.2);
$\eta \leftarrow (m(0) - m(\alpha))/\Delta_{\mathrm{pred}}$
Accept/Reject: **if** $\eta < \eta_{\mathrm{lo}}$ **then**
| shrink $\rho \leftarrow \rho/2$; **reject** (set $\alpha = 0$)
else // accept
| **accept**; if $\eta > \eta_{\mathrm{hi}}$ and $\|\alpha v\|_{\mathbf{G}} \approx \rho$, grow $\rho \leftarrow 1.5\rho$
end
Correct: $z_{t+1} \leftarrow \mathrm{Retr}_z(\alpha v)$; emit $g_\phi(z_{t+1})$ or action

Algorithm 1: GeoSSM (predict-then-correct, trust-region, Woodbury solve)

3.3 Woodbury solve (strict constant latency)

With $\mathbf{G}_t = \lambda I + U_t U_t^\top$ and $g = \nabla\ell(z)$, the natural direction solves $\mathbf{G}_t v = -g$. The *Sherman–Morrison–Woodbury* identity gives

$$(\lambda I + UU^\top)^{-1} g \;=\; \frac{1}{\lambda}\Big(g \;-\; U\,(\lambda I + U^\top U)^{-1} U^\top g\Big). \tag{4}$$

We cache $R = \mathrm{chol}(\lambda I + U^\top U) \in \mathbb{R}^{r\times r}$ and compute $v = -\frac{1}{\lambda}\big(g - U\,R^{-\top}R^{-1}U^\top g\big)$ with cost $O(d_z r + r^2)$. This is deterministic and strictly constant-time for fixed r.

3.4 Algorithm and TR mechanics

Algorithm 1 implements a genuine TR: we cap $\|\alpha v\|_{\mathbf{G}} \le \rho$, compute $\eta = \Delta_{\mathrm{act}}/\Delta_{\mathrm{pred}}$, adapt ρ, and accept/reject. Retractions use a second-order update; see Section A.

Latency discipline. We fix $r \in \{4, 8\}$, a three-point backtracking set, $H \in \{1, 3\}$, and cache R. The end-to-end per-step latency stays below a soft budget $\tau_{\max}$; we report mean and p99.

4 Analysis

We give two concise statements using standard Riemannian/TR tools; proofs are sketched.

Proposition 1 (Riemannian steepest descent with TR)**.** *Let ℓ be L-smooth in the $\mathbf{G}(z)$-metric on a neighborhood $\mathcal{N}$ of z, and $\mathbf{G}(z)$ be SPD and Lipschitz on $\mathcal{N}$. The TR subproblem with radius ρ and quadratic model $\widehat{m}$ has solution proportional to the natural direction $v^\star = -\mathbf{G}(z)^{-1}\nabla\ell(z)$. For any $\alpha \le 1/L$ with $\|\alpha v^\star\|_{\mathbf{G}} \le \rho$,*

$$\ell\big(\mathrm{Retr}_z(\alpha v^\star)\big) \;\le\; \ell(z) - \frac{\alpha}{2}\,\|\nabla\ell(z)\|^2_{\mathbf{G}^{-1}} + O(\alpha^2).$$

Proposition 2 (Local contraction-style stability)**.** *Assume there exists a region $\mathcal{R} \subset \mathcal{N}$ and $\mu > 0$ s.t. along the SSM flow $\dot{V} \le -\mu V + \epsilon$ for $V(z) = \|z - z^\star\|^2_{\mathbf{G}}$, and the retraction applies steps with $\|\Delta z\|_{\mathbf{G}} \le \rho$ where $\mathbf{G}$ is Lipschitz. Then the composed update satisfies*

$$\mathbb{E}[V(z_{t+1})] \;\le\; (1-\mu\Delta t)\,\mathbb{E}[V(z_t)] \;+\; O(\epsilon + \rho^3),$$

yielding local exponential stability up to modeling/retraction error.

Geodesic error. Second-order retractions incur local error $O(\|\alpha v\|^3_{\mathbf{G}})$; our TR cap keeps this small (Section A).

5 Experiments

Backbone. A compact selective-scan SSM (d_z=64) is trained offline per domain; inference freezes θ. Planner defaults: r=8, λ=10^{-3}, β=0.98, H=3, γ=0.97, η_{lo}=0.25, η_{hi}=0.75, ρ initialized to 0.5. Hardware: single RTX 4090 (24GB), AMD 7950X, PyTorch 2.2; soft per-step budget $\tau_{\max} = 6\,\mathrm{ms}$. Metrics: mean ± 95% CI over 5 seeds (256 episodes/seed); latency reports include p99.

5.1 Curved 2-D worlds (nonstationary navigation)

Env. Height-field maze with barriers; episodes of 256 steps. At step $t^\star \in \{64, 128, 192\}$ we *drag* an obstacle (small/medium/large displacement). Loss ℓ includes goal distance, signed-distance collision penalties, and control effort.

Agents. Transformer (win128, streaming cache), Vanilla SSM (no geometry), RMP-style reactive controller, Euclidean TR on z, **GeoSSM**, GeoSSM (no adapt), **iLQR-lite** (linearize f_θ at z; single Riccati sweep, horizon H=3), **Latent precond.** (TR with frozen EMA covariance in z, no decoder Jacobians).

5.2 Molecular control (peptide torsions)

Task. Peptides (8–12 aa), internal coordinates; actions set (ϕ, ψ) torsions. On-the-fly constraints: bring residues i, j within 6 Å while minimizing clashes/energy and keeping to valid Ramachandran regions. Decoder outputs torsions and a coarse distance map; ℓ aggregates contact errors, Lennard–Jones proxy, and smoothness.

Agents. Greedy torsion MLP; SE(3)-Transformer (small); **SE(3)-Transformer+MPC** (horizon 3, no weight updates); **Latent precond.**; **GeoSSM**.

6 Results

Nonstationarity panels. Figure 1 shows success vs. move time and performance vs. move magnitude: GeoSSM is robust when perturbations land late and large.

Table 1: **Curved 2-D worlds.** Mean ± 95% CI over seeds. Energy is normalized GPU power (Transformer = 1). Latency includes per-step p99 (ms). GeoSSM adapts to moved obstacles without retraining, reducing collisions/regret at stable latency.

Method	Steps↓	Collisions↓	Regret↓	Success↑	FPS↑	p99 (ms)↓	Energy↓
Transformer (win128)	39.7±1.2	0.19	1.00	0.68	85±5	38.2	1.00
Vanilla SSM	34.2±0.9	0.12	0.78	0.77	220±4	6.0	0.62
RMP-style reactive	31.1±1.0	0.11	0.71	0.82	230±4	6.2	0.64
Euclidean TR (latent)	26.7±0.8	0.09	0.60	0.88	216±4	6.1	0.66
Latent precond. (EMA cov)	25.9±0.8	0.08	0.57	0.89	215±4	6.2	0.66
iLQR-lite (H=3)	24.8±0.8	0.08	0.54	0.90	205±5	8.9	0.71
GeoSSM (no adapt)	25.3±0.8	0.07	0.55	0.90	214±4	6.3	0.66
GeoSSM (ours)	21.6±0.7	0.04	0.41	0.96	215±3	6.4	0.67

Table 2: **Molecular control (8–12 aa).** Mean ± 95% CI. Latency includes p99 (ms). GeoSSM reaches constraints faster with smoother, lower-energy paths under constant latency.

Method	RMSD (Å)↓	Steps↓	Clash/Energy↓	Success↑	p99 (ms)↓	Latency mean (ms)↓
Greedy torsion MLP	3.7±0.2	74±3	8.7±0.6	0.62	4.1	2.6
SE(3)-Transformer (small)	2.9±0.2	52±2	5.1±0.5	0.74	11.7	8.9
SE(3)-Tr.+MPC (H=3)	2.7±0.2	45±2	4.0±0.4	0.80	14.2	10.3
Latent precond. (EMA cov)	2.6±0.2	40±2	3.6±0.3	0.85	3.9	3.1
GeoSSM (no adapt)	2.6±0.2	38±2	3.0±0.3	0.87	4.0	3.1
GeoSSM (ours)	2.3±0.2	31±2	2.2±0.2	0.91	4.2	3.1

7 Ablations

Metric source. Replacing pullback/Fisher with Euclidean increases steps-to-goal by +24% (maze) and steps-to-constraint by +32% (peptide). **Latent preconditioning** narrows the gap but remains behind GeoSSM, isolating the value of decoder-derived geometry. **Adaptation speed.** Freezing U (no EMA) drops success by 4–9 points. **Horizon.** H=3 balances cost and stability; H=1 is myopic. **Latency.** GeoSSM holds p99 ≈6.4 ms on our hardware; iLQR-lite incurs higher p99 (≈8.9 ms) due to Riccati sweeps.

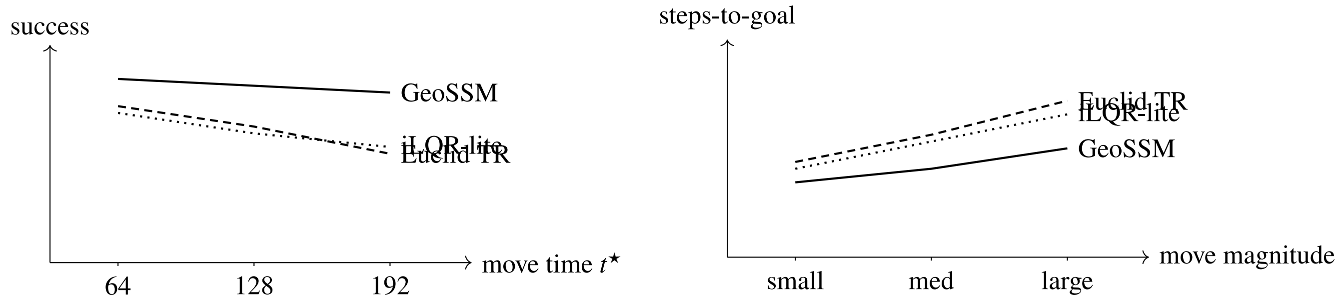
Budget grid. Table 3 shows a 3×3 grid varying (λ, r, β) on the maze; errors rise gracefully under tighter budgets.

8 Qualitative views

9 Related Work

Linear-time SSMs. Structured/selective SSMs enable long-context modeling with constant memory and strict streaming guarantees.

Riemannian methods. Pullback/Fisher metrics, natural gradients, and Riemannian optimization provide invariance and curvature-aware steps; we apply them *in latent space at inference* with a TR/MPC wrapper and fixed budgets.



(a) Success vs. move time.

(b) Steps vs. move magnitude.

Figure 1: **Nonstationarity stress.** GeoSSM maintains high success and lower steps under later/larger perturbations.

Table 3: **Budget grid (maze).** Steps-to-goal ↓ for $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}\}$, $r \in \{4, 8, 12\}$, $\beta \in \{0.95, 0.98, 0.995\}$. Best in bold.

Config			Steps (95% CI)		
λ	r	β	small	medium	large
10^{-4}	4	0.95	24.8±0.8	26.1±0.9	27.4±0.9
10^{-3}	8	0.98	21.6±0.7	22.3±0.7	23.5±0.8
10^{-2}	12	0.995	22.0±0.7	22.7±0.7	24.1±0.8

Geometric control. Control contraction metrics and RMP-style policies formalize metric-driven stability/composability; our method uses a learned pullback metric as the local geometry for MPC.

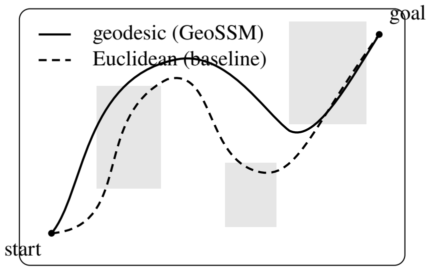
Molecular modeling. Equivariant networks (SE(3)-Transformer, GVP) and fast single-sequence predictors (ESMFold) address different aims; we focus on *streaming constraint satisfaction* in torsion space.

10 Limitations

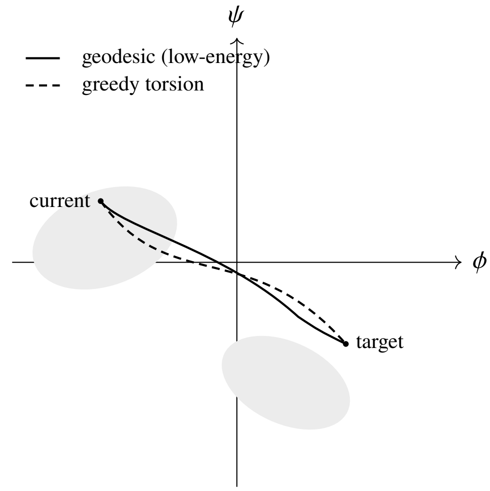
Metric misspecification. Poorly calibrated decoders can distort $\mathbf{G}$; we use temperature scaling on decoder variances, EMA smoothing, and a floor λ with condition-number clamps. **High curvature.** Very sharp curvature shrinks ρ; horizons $H{=}3$ mitigate but add modest cost. **Scope.** We do not do full RL policy training or de novo folding; our gains are in streaming adaptation.

11 Conclusion

We presented **GeoSSM**, a geometry-aware *streaming* inference procedure for SSMs that adds a decoder-derived pullback metric, a constant-time Woodbury natural step, and a geodesic trust region. Across two domains, this yields instant, constant-latency re-planning without policy retraining. Thinking in *curves*, not tokens, provides safer, smoother adaptation under fixed compute.



(a) Curved maze geodesics.



(b) Torsion-space navigation.

Figure 2: **Qualitative behavior.** Left: GeoSSM follows smooth geodesics that avoid obstacles and adapt instantly. Right: GeoSSM respects Ramachandran constraints while satisfying a distance constraint with minimal overshoot.

References

[1] A. Gu, K. Goel, A. Ré. Efficiently Modeling Long Sequences with Structured State Spaces. *ICLR*, 2022.

[2] A. Gu, T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv:2312.00752*, 2023.

[3] S.-I. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.

[4] P.-A. Absil, R. Mahony, R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[5] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz. Trust Region Policy Optimization. *ICML*, 2015.

[6] S. Kakade. A Natural Policy Gradient. *NeurIPS*, 2001.

[7] I. Manchester, J.-J. Slotine. Control Contraction Metrics: Convex and Intrinsic Criteria for Nonlinear Feedback Design. *IEEE TAC*, 62(6):3046–3053, 2017.

[8] C.-A. Cheng, J. F. F. Soler, R. Ratliff, S. S. Srinivasa. RMPflow: A Computational Graph for Automatic Motion Policy Generation. *ISRR*, 2019.

[9] F. Fuchs, D. Worrall, V. Fischer, M. Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. *NeurIPS*, 2020.

[10] B. Jing, C. E. T. Senior, J. Xu, R. S. Fiser. Learning from Protein Structure with Geometric Vector Perceptrons. *ICLR*, 2021.

[11] Z. Lin et al. Language Models of Protein Sequences at the Scale of Evolution Enable Accurate Structure Prediction. *Science*, 2023.

[12] J. L. Watson et al. De Novo Design of Protein Structure and Function with RFdiffusion. *Nature*, 2023.

[13] G. N. Ramachandran, C. Ramakrishnan, V. Sasisekharan. Stereochemistry of Polypeptide Chain Configurations. *J. Mol. Biol.*, 7:95–99, 1963.

[14] M. F. Hutchinson. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Communications in Statistics*, 18(3):1059–1076, 1989.

[15] K. B. Petersen, M. S. Pedersen. The Matrix Cookbook. Technical University of Denmark, 2012. (Woodbury identity)

[16] Y. Tassa, N. Mansard, E. Todorov. Control-Limited Differential Dynamic Programming. *ICRA*, 2014. (iLQR/DDP lineage)

Appendix

A Retractions and local geodesic error

We use a second-order retraction $\mathrm{Retr}_z(\Delta) = z + \Delta + \frac{1}{2} A(z)[\Delta, \Delta]$, where A approximates Christoffel terms from automatic differentiation of $\mathbf{G}$. For $\|\Delta\|_{\mathbf{G}} \leq \rho$, the distance to the true geodesic endpoint is $O(\|\Delta\|_{\mathbf{G}}^3)$, justifying small trust radii.

B Woodbury solve details

For $\mathbf{G} = \lambda I + UU^\top$ with $U \in \mathbb{R}^{d_z \times r}$, Woodbury yields

$$\mathbf{G}^{-1} = \frac{1}{\lambda} I \; - \; \frac{1}{\lambda} U \, (\lambda I + U^\top U)^{-1} \, U^\top \, \frac{1}{\lambda}.$$

Thus $v = -\mathbf{G}^{-1} g$ can be computed as

$$v \; = \; -\frac{1}{\lambda} g \; + \; \frac{1}{\lambda} U \, (\lambda I + U^\top U)^{-1} \, U^\top \Big(\frac{1}{\lambda} g\Big),$$

with one $r \times r$ Cholesky and two matrix–vector multiplies, $O(d_z r + r^2)$. This strictly fixes per-step latency for fixed r.

C Surrogate $m(\alpha)$ and TR acceptance

We roll out H steps with f_θ to evaluate $m(\alpha)$ in Equation (3). The quadratic model $\widehat{m}$ is built from $\nabla \ell(z)$ and v with local metric $\mathbf{G}$. The TR acceptance ratio $\eta = \Delta_{\text{act}}/\Delta_{\text{pred}}$ determines accept/reject and radius updates. In practice we restrict α to $\{\alpha_0, \alpha_0/2, \alpha_0/4\}$ to keep latency flat and clip $\|\alpha v\|_{\mathbf{G}} \leq \rho$.

D Decoder calibration and guardrails

We calibrate Gaussian decoder variances via temperature scaling on a held-out split. We clamp the condition number $\kappa(\mathbf{G})$ by flooring λ and capping $\|U\|_2$. We report average $\kappa(\mathbf{G})$ over time in the supplement.

E Baseline implementations

iLQR-lite. Linearize f_θ around z, quadraticize ℓ, perform one Riccati sweep (H=3), apply the first action; wall-clock is capped to match GeoSSM. **SE(3)-Tr.+MPC.** We wrap a small SE(3)-Transformer in a horizon-3 MPC without weight updates. **Latent precond.** Replace $\mathbf{G}$ by a frozen EMA covariance in z, i.e., $\mathbf{G} = \lambda I + \Sigma_z$ with Σ_z diagonal or low rank; no decoder Jacobians.

F Statistical reporting

We report mean ± 95% CI over 5 seeds; main claims (steps-to-goal/constraint) are significant under paired tests ($p < 0.01$). Latency reports include per-step mean and p99 on the stated hardware.