Who Needs Attention Anyway? Elastic State Models for Real-Time Streaming Tasks

Dario Fumarola
Amazon Web Services

Abstract

Real-time sequence models face a compute-allocation problem: most streaming states are routine, while rare states become difficult because constraints, decoder sensitivity, or local observations change abruptly. We introduce *Elastic State Models* (ESMs), a bounded inference-time adaptation layer for frozen recurrent or state-space backbones. At each step, an ESM proposes a latent state, scores local difficulty, and conditionally applies up to $K_{\max}$ decoder-metric correction microsteps before committing the state. The correction metric is a decoder-sensitivity pullback–Gauss-Newton or Fisher when the output weight is chosen from a local quadratic loss or likelihood–maintained online as $\widetilde{G}_t = \lambda I + U_t U_t^\top$ with a fixed-rank Nyström sketch. A Woodbury solve gives a cheap preconditioned direction, while cumulative metric and Euclidean trust caps plus a predicted-decrease test make the update fail-closed. For fixed correction budget, sketch rank, and rollout horizon, ESM inference has sequence-length-independent worst-case work while average work follows local difficulty. We derive conditional local descent, a sketch-approximation bound under a relative spectral assumption, and a practical stability bound under contractivity and bounded committed corrections. Controlled simulated navigation and torsion-chain repair experiments show that bounded latent optimization with causal local task geometry improves frozen SSMs and metric-matched correction baselines. Comparisons to fixed-context attention baselines are reported as quality-latency tradeoffs under this task-loss access, not as evidence that attention is generally unnecessary.

1 Introduction

Attention is effective for retrieval, content-dependent routing, and long-context credit assignment. Many real-time streams, however, fail at a different scale. A navigation policy can follow a corridor for hundreds of steps and then make one fragile decision at a doorway; a controller can track smoothly until contact or a joint limit changes local geometry; a sensor stream can be predictable until a transient drift makes the current hidden state inconsistent with the next observation. In these settings, the immediate bottleneck is often not access to more history, but whether the state about to be committed is locally consistent with the task.

Structured state-space models (SSMs) and recurrent models are attractive for real-time inference because they compress history into a fixed-size state and admit efficient streaming updates [8, 9]. Their weakness is that the state update is usually fixed once the model is trained. When the local geometry changes, the model either commits a brittle proposal or pays for a larger context/planning module at every step. ESMs are not a replacement for attention or planning; they target a narrower but common situation in which the next committed state needs bounded local repair, not a new global retrieval policy. We seek a third option: keep the compact streaming update, but add a correction layer that activates only when the proposed state is difficult.

The proposed wrapper, an *Elastic State Model* (ESM), leaves the backbone and decoder weights frozen. Its substantive online state is the latent z_t that will be committed to the stream and a low-rank metric sketch U_t that describes which latent directions are currently sensitive under the decoder; a small diagnostic history records recent accepts, rejects, and trust saturation. The gate decides *how much* correction to spend; the metric decides *which directions* are safe and useful; the trust test decides *whether* a correction should be committed.

The key idea is to correct in decoder-induced geometry. A Euclidean latent gradient treats all coordinates as equally meaningful, even though the decoder can make one direction harmless and another direction highly sensitive. ESMs pull a task-space sensitivity metric back through the decoder, approximate it online with a fixed-rank Nyström factor, compute a damped natural-gradient-like direction by Woodbury, and cap each accepted movement with a trust region. This gives a state-repair primitive whose worst-case work is fixed before deployment and whose average work follows the local difficulty of the stream.

Contributions. The paper makes four contributions: (i) an inference loop for difficulty-gated state repair in frozen streaming models; (ii) a fixed-rank decoder-metric correction rule with Woodbury solves and trust-region acceptance; (iii) local descent, sketch approximation, and bounded-perturbation stability results; and (iv) controlled navigation and torsion-chain repair experiments with quality-latency, gate-allocation, rank-ablation, diagnostic, and failure analyses.

2 Related Work

State-space and attention models. Structured SSMs such as S4 exploit special state matrices and fast convolutional computation for long sequences [9]. Mamba adds selective state-space updates for content-dependent recurrence while retaining linear-time scaling [8]. State-space duality further connects attention-like and SSM-like computations through common structured matrix views [5]. Transformers route information across a retained context window [20], and optimized kernels such as FlashAttention reduce the memory traffic of exact attention [4]. ESMs address a complementary inference-time question: when the state is already compact, how should a streaming model allocate bounded extra work to fragile states?

Adaptive computation and test-time adaptation. Adaptive Computation Time lets recurrent networks allocate variable internal work across inputs [7]. Test-time training updates an expressive hidden state or fast model during inference [18]. ESMs

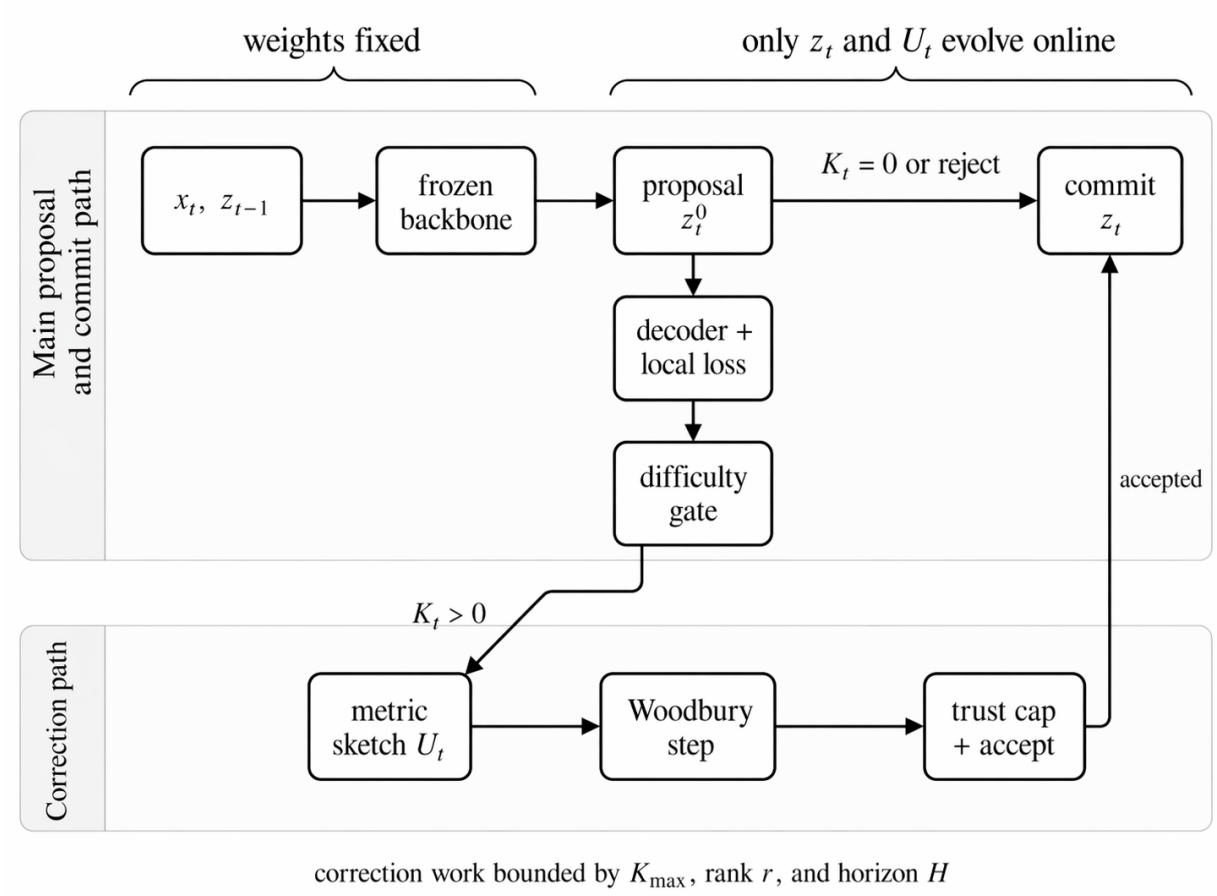


Figure 1: Elastic State Model inference loop. The backbone and decoder weights are fixed. The main path commits immediately when the gate selects $K_t = 0$ or a proposed correction is rejected. Rejection leaves the committed state unchanged; the metric sketch may still be refreshed as a measurement of decoder geometry. Correction work is bounded by $K_{\max}$, rank r, and rollout horizon H.

share the adaptive-inference motivation, but constrain adaptation to a small number of latent-state corrections with frozen weights and a fixed worst-case per-step budget. This distinction matters in real-time systems: the correction layer may be invoked often, so the maximum work, not only the average work, must be known in advance.

Geometry, trust regions, and control. Natural gradients use Fisher geometry to make optimization less dependent on parameterization [2, 15], while pullback information geometry defines latent metrics from decoder distributions [3]. Trust-region methods restrict local updates when the model is approximate, with TRPO as a prominent policy-optimization example [17]. Differential dynamic programming and contraction-metric control similarly emphasize local quadratic models and bounded perturbations [16, 19]. ESMs use these ideas at inference time on the latent state of a frozen streaming model rather than on trainable parameters or full control trajectories.

Filtering and latent inference. The ESM update has a predict-correct form, so it is important to distinguish it from classical and learned filters. Kalman and extended/unscented Kalman filters correct a latent belief with a probabilistic observation model and covariance propagation [11, 12]; deep state-space filters and variational sequence models learn amortized inference networks for latent dynamics [13, 14]. ESMs do not propagate a full posterior or update weights at test time. They apply a bounded number of deterministic decoder-metric repairs to a single committed streaming state, with an explicit gate, trust cap, and rejection rule.

Sketches and preconditioning. Randomized low-rank approximations and Nyström preconditioners can capture dominant curvature directions with matrix-vector products instead of dense matrices [6, 10]. The ESM sketch is specialized to streaming decoder metrics: it is updated online, truncated to a fixed rank, and used only inside a small trust-region repair loop.

3 Problem Setting

At time t, a frozen streaming backbone receives x_t and proposes

$$z_t^0 = f_\theta(z_{t-1}, x_t), \qquad \hat{y}_t^0 = g_\phi(z_t^0), \tag{1}$$

where $z_t \in \mathbb{R}^{d_z}$, f_θ is a recurrent or SSM update, and g_ϕ is a decoder. The correction objective is written abstractly as

$$\ell_t(z) = \ell_{\text{obs}}(g_\phi(z), o_t) + \ell_{\text{task}}(g_\phi(z), c_t), \tag{2}$$

where o_t denotes an observation or residual and c_t encodes constraints, obstacle fields, rollout penalties, or task costs. The decoder can emit an action, a short local rollout, a physical configuration, or a distribution over these quantities; the only requirement is that the correction loss be differentiable with respect to z. The objective used by the deployment loop must be causal with respect to the action being committed. In a *pre-action* mode,

ℓ_t may use only sensor readings, constraint maps, residuals, and differentiable short-horizon rollouts available before acting. In a *post-observation* mode, a supervised residual or label observed after acting can repair the latent state for $t+1$, but cannot change the action already emitted at t. Ground-truth labels used only for training, calibration, or evaluation are not used by the online gate or correction loop.

The real-time contract has three parts. First, the state must be committed online: the algorithm cannot revisit an unbounded prefix of the stream. Second, the worst-case work per step must be bounded by a deployment-time budget B independent of sequence length. Third, the correction must be fail-closed: if the local model is unreliable, the system should reject the correction and commit the frozen proposal rather than an unconstrained latent update. These constraints rule out full retraining, unbounded search, and correction loops whose stopping time depends on convergence.

ESM treats the proposed latent state z_t^0 as a small inference-time optimization variable. The weights θ and ϕ remain fixed; the committed latent state and compact metric summary are allowed to change, while scalar diagnostics are recorded only to guide later gating decisions. This separation is useful because it lets a deployment inherit the speed and calibration of a trained streaming model while adding local geometric checks where the model is fragile.

4 Elastic State Models

An ESM step has three stages. It first proposes a state with the frozen backbone. It then estimates whether the proposal lies in an easy region or a fragile region. If the gate opens, it builds or refreshes a low-rank decoder metric and performs at most $K_t \le K_{\max}$ correction microsteps. The committed state is the last accepted state; rejected microsteps do not modify the stream.

4.1 Difficulty-gated state repair

The gate should estimate the marginal value of correction, not merely the magnitude of the loss. ESM therefore computes diagnostics ψ_t from the proposed state, for example

$$\begin{aligned}\psi_t = [&\ell_t(z_t^0),\ \left\|\nabla \ell_t(z_t^0)\right\|_2,\ \left\|\nabla \ell_t(z_t^0)\right\|^2_{\widetilde{G}_{t-1}^{-1}},\\ &\left\|U_{t-1}^\top \nabla \ell_t(z_t^0)\right\|_2,\ u_t,\ a_{t-1}],\end{aligned} \tag{3}$$

where u_t is a task-specific uncertainty or residual and a_{t-1} summarizes recent acceptance, rejection, or trust-radius saturation. When these features require a full gradient, that backward pass is counted in the easy-path cost; cheaper deployments can replace them with a first-stage residual gate.

It is useful to view the gate as minimizing the ideal risk

$$\mathcal{J}(K) = \ell_t(z_t^K) + \lambda_K K + \lambda_R \mathbf{1}\{\text{rejected at } K\}, \tag{4}$$

over $K \in \{0, \dots, K_{\max}\}$. The implementation uses a supervised approximation to this risk: offline rollouts enumerate the candidate budgets, label each state with the smallest budget that reaches a target fraction of the best local improvement, and train a classifier $q_\eta(K \mid \psi_t)$. Deployment is then deterministic,

$$K_t = \min\left\{ K : \sum_{j=0}^{K} q_\eta(j \mid \psi_t) \ge \tau \right\}, \tag{5}$$

with $K_t \le K_{\max}$. At deployment, η, θ, ϕ are fixed; the gate only chooses the number of microsteps. The quantile τ is a validation-selected operating point on the quality-latency curve. With the lower-quantile convention in Eq. (5), increasing τ can only keep or increase the selected budget, so it spends more correction and raises tail latency; decreasing τ is more conservative and returns $K_t = 0$ more often. We tune τ on validation rather than interpreting it as a universal threshold.

4.2 Decoder-induced correction metric

Let $J_t(z) = \partial g_\phi(z)/\partial z$ and let $W_t \succeq 0$ be a task-chosen output-space sensitivity weight. When W_t is the local Hessian of a squared loss or a Fisher weight from a likelihood model, the construction is Gauss-Newton or Fisher; otherwise it should be read as a decoder-sensitivity preconditioner. The full decoder pullback metric is

$$G_t^\star(z) = J_t(z)^\top W_t J_t(z) + \lambda I, \qquad \lambda > 0. \tag{6}$$

This metric says that a latent direction is costly when the decoder maps it to a large change in output coordinates that the current task regards as sensitive. The ideal metric correction for gradient $g_t^k = \nabla_z \ell_t(z_t^k)$ is $v_t^k = -(G_t^\star)^{-1} g_t^k$, but materializing $G_t^\star \in \mathbb{R}^{d_z \times d_z}$ is too expensive for streaming inference.

ESMs therefore sketch the undamped pullback $A_t = J_t^\top W_t J_t$ with r_0 probes Ω_t. Products $A_t\omega = J_t^\top (W_t J_t \omega)$ use one JVP and one VJP through the decoder. With

$$Y_t = A_t \Omega_t, \quad B_t = \mathrm{sym}(\Omega_t^\top Y_t) + \epsilon I, \quad \widehat{U}_t = Y_t B_t^{-1/2}, \tag{7}$$

the inverse square root is computed from a clipped eigendecomposition of the small $r_0 \times r_0$ matrix B_t. Near-zero eigenvalues are floored before inversion, so the update is stable even when the probes are nearly dependent. The online factor update is

$$\begin{aligned} U_t &= \mathrm{TruncSVD}_r\left([\sqrt{\beta} U_{t-1}, \sqrt{1-\beta}\widehat{U}_t] \right), \\ \widetilde{G}_t &= \lambda I + U_t U_t^\top. \end{aligned} \tag{8}$$

Here $\mathrm{TruncSVD}_r(M)$ returns the scaled factor $P_r \Sigma_r$, where $M = P\Sigma Q^\top$ and only the top r singular values are kept; therefore $U_t U_t^\top$ approximates $MM^\top$ rather than only its span. Concatenating factors approximates an exponential average of positive semidefinite matrices without adding uncontrolled cross terms. Truncation keeps the memory and solve cost fixed at $O(d_z r)$.

4.3 Woodbury step, trust cap, and acceptance

The sketched metric inverse is applied by Woodbury:

$$\widetilde{G}_t^{-1} g = \lambda^{-1}\left[g - U_t(\lambda I + U_t^\top U_t)^{-1} U_t^\top g \right], \tag{9}$$

Algorithm 1 Elastic State Model inference for one streaming step

Require: Frozen backbone f_θ, decoder g_ϕ, gate q_η, input x_t, previous state z_{t-1}, sketch U_{t-1}, budget $K_{\max}$, rank r, threshold τ, radii $\rho_{\text{step}}, \rho_{\text{tot}}, \rho_{\text{Euc}}$.

Propose $z_t^0 \leftarrow f_\theta(z_{t-1}, x_t)$; compute diagnostics ψ_t from loss, gradients, residuals, and prior correction statistics.
Select $K_t \leftarrow \min\{K : \sum_{j=0}^{K} q_\eta(j \mid \psi_t) \geq \tau\} \leq K_{\max}$.
if $K_t = 0$ **then**
    Set $U_t \leftarrow \sqrt{\beta} U_{t-1}$ and **return** committed state $z_t \leftarrow z_t^0$ and sketch U_t.
end if
Form probes Ω_t; compute $Y_t = A_t \Omega_t$ with decoder JVP/VJP products.
Set $B_t \leftarrow \text{sym}(\Omega_t^\top Y_t) + \epsilon I$; compute a clipped eigendecomposition for $B_t^{-1/2}$; set $\widehat{U}_t \leftarrow Y_t B_t^{-1/2}$ and update U_t by Eq. (8).
for $k = 0, \ldots, K_t - 1$ **do**
    Compute $g_t^k \leftarrow \nabla_z \ell_t(z_t^k)$ and $v_t^k \leftarrow -\widetilde{G}_t^{-1} g_t^k$ by Eq. (9).
    Choose the largest $\alpha_t^k \leq \alpha_0$ satisfying the per-step, cumulative metric, and cumulative Euclidean caps in Eq. (10); set $\Delta_t^k \leftarrow \alpha_t^k v_t^k$.
    Compute $d_{\text{pred}}^k = m_k(0) - m_k(\Delta_t^k)$. If $d_{\text{pred}}^k \leq \delta_{\text{pred}}$, stop.
    Accept $z_t^{k+1} \leftarrow \text{Retr}_{z_t^k}(\Delta_t^k)$ if Eq. (12) holds; otherwise stop.
end for
return committed state z_t and refreshed sketch U_t; rejection status is stored in a_t.

with induced norm $\|w\|_{\widetilde{G}_t}^2 = \lambda \|w\|_2^2 + \|U_t^\top w\|_2^2$. Each microstep uses $v_t^k = -\widetilde{G}_t^{-1} g_t^k$ and a step length clipped by a per-microstep metric radius, a cumulative metric radius, and an explicit cumulative Euclidean radius around the frozen proposal:

$$\Delta_t^k = \alpha_t^k v_t^k, \qquad \|\Delta_t^k\|_{\widetilde{G}_t} \leq \rho_{\text{step}},$$
$$\|z_t^k + \Delta_t^k - z_t^0\|_{\widetilde{G}_t} \leq \rho_{\text{tot}}, \qquad \|z_t^k + \Delta_t^k - z_t^0\|_2 \leq \rho_{\text{Euc}}. \tag{10}$$

The cumulative caps are the quantities used in the stability bound; without them, a K_t-step correction would only be bounded by the sum of microstep radii. The Euclidean cap is important because the conservative implication $\|\delta_t\|_2 \leq \rho_{\text{tot}}/\sqrt{\lambda}$ can be loose when λ is small.

Let

$$m_k(\Delta) = \ell_t(z_t^k) + \langle g_t^k, \Delta \rangle + \tfrac{1}{2} \|\Delta\|_{\widetilde{G}_t}^2,$$
$$d_{\text{pred}}^k = m_k(0) - m_k(\Delta_t^k). \tag{11}$$

A candidate is rejected if $d_{\text{pred}}^k \leq \delta_{\text{pred}}$ for a small positive threshold. Otherwise it is accepted only when

$$\frac{\ell_t(z_t^k) - \ell_t(\text{Retr}_{z_t^k}(\Delta_t^k))}{d_{\text{pred}}^k} \geq \eta_{\min}. \tag{12}$$

For $\alpha_0 \leq 1$, the unconstrained quadratic model has nonnegative predicted decrease $\alpha(1 - \alpha/2) \|g_t^k\|_{\widetilde{G}_t^{-1}}^2$. For control tasks, the realized decrease can be computed over a fixed short rollout. The acceptance rule is intentionally asymmetric: a questionable correction is discarded, while a successful one changes the future stream by updating z_t.

5 Compute and Latency Bounds

Let C_f be the frozen update cost, C_ψ the diagnostic/gate cost, C_g one decoder-gradient cost inside correction, C_J one JVP/VJP pair, and $C_{\text{acc}}(H)$ the cost of evaluating a candidate by a fixed horizon-H acceptance rollout. If the gate features include a full loss gradient, that backward pass is counted in C_ψ; a cheaper first-stage gate can reduce the easy-path cost. The sketch refresh also pays for the $r_0 \times r_0$ inverse square root and the truncation of the $d_z \times (r + r_0)$ concatenated factor. A representative per-step bound is

$$C_t \leq C_f + C_\psi + \mathbf{1}_{K_t > 0}\big(r_0 C_J + r_0^3 + d_z(r + r_0)^2 + (r + r_0)^3 + r^3 + K_t(C_g + d_z r + r^2 + C_{\text{acc}}(H))\big). \tag{13}$$

Thus

$$C_{\max} = C_f + C_\psi + r_0 C_J + r_0^3 + d_z(r + r_0)^2 + (r + r_0)^3 + r^3 + K_{\max}(C_g + d_z r + r^2 + C_{\text{acc}}(H)), \tag{14}$$

which is independent of sequence length. JVP/VJP products and candidate rollouts can be batched, but the bound counts the work explicitly rather than hiding it in latency. In the experiments, the candidate rollout is recomputed for each accepted or rejected microstep; if an implementation reuses rollout quantities, the same formula holds with a smaller $C_{\text{acc}}(H)$. Memory for the correction layer is $O(d_z r)$ plus small $(r + r_0) \times (r + r_0)$ matrices.

The average cost is lower when difficult states are sparse. If $p = \Pr[K_t > 0]$ and $\bar{K} = \mathbb{E}[K_t \mid K_t > 0]$, then the expected correction overhead is approximately

$$p\big(r_0 C_J + r_0^3 + d_z(r + r_0)^2 + (r + r_0)^3 + r^3 + \bar{K}(C_g + d_z r + r^2 + C_{\text{acc}}(H))\big). \tag{15}$$

This decomposition is useful experimentally: the rank and horizon set the price of opening the gate, while $\bar{K}$ controls the number of local optimization attempts. Wall-clock tail latency still depends on hardware, batching, compiler behavior, and decoder autodiff, so experiments report both algorithmic budgets and measured latency.

6 Theory

The main statements formalize the Woodbury solve, descent under a local metric model, degradation from sketching, and stability of bounded corrections. They are conditional local results: they do not prove global convergence of the whole streaming controller, nor do they replace empirical calibration of the decoder metric. Proofs are in Appendix A.

Lemma 1 (Woodbury solve for the sketched metric)**.** *Let $\lambda > 0$, $U \in \mathbb{R}^{d_z \times r}$, and $\widetilde{G} = \lambda I + UU^\top$. Then*

$$\widetilde{G}^{-1} g = \lambda^{-1} \left[g - U(\lambda I + U^\top U)^{-1} U^\top g \right], \tag{16}$$

and $\|w\|_{\widetilde{G}}^2 = \lambda \|w\|_2^2 + \|U^\top w\|_2^2$.

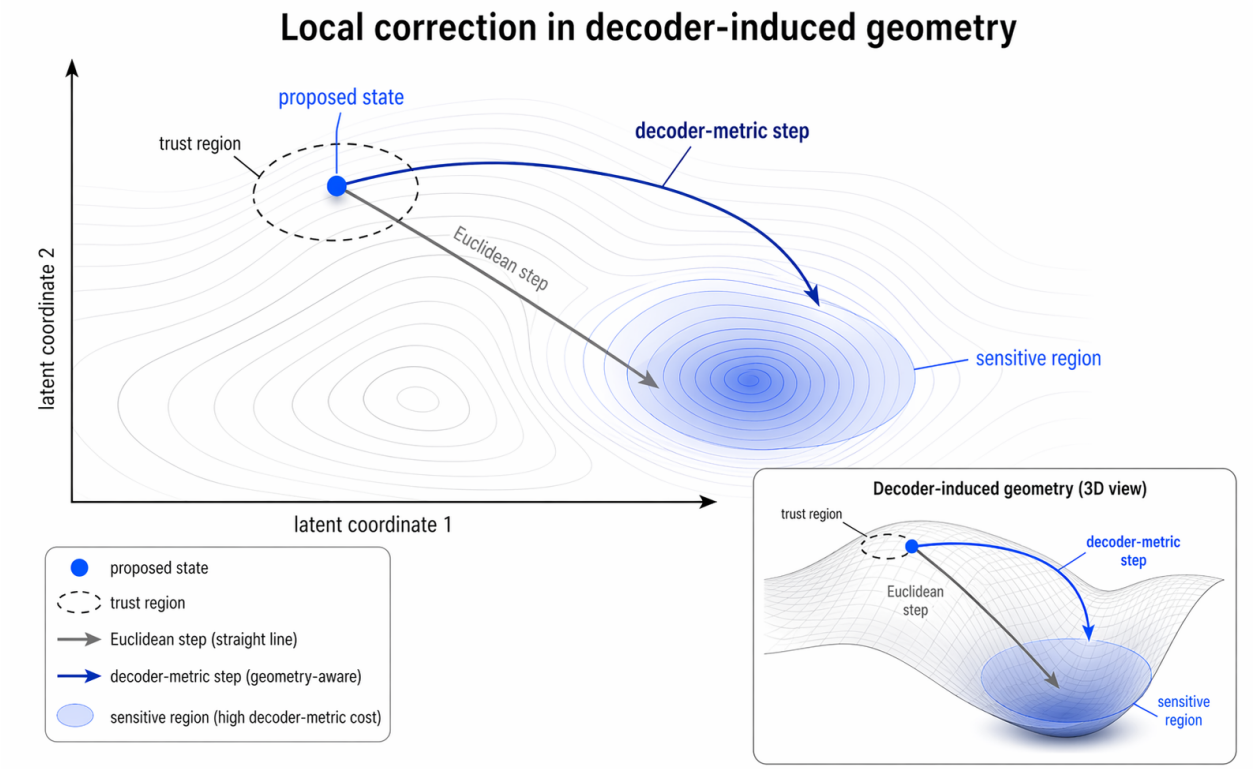


Figure 2: Schematic local correction in decoder-induced geometry. A Euclidean step follows the straight latent direction and can enter a sensitive output region. A decoder-metric step rescales directions by output sensitivity and is clipped by the trust region before commitment. This figure is illustrative rather than a logged trajectory.

Assumption 1 (Local metric smoothness)**.** *On a neighborhood $\mathcal{N}$ around the proposed state, $\widetilde{G} \succ 0$ is fixed during a microstep and*

$$\ell(z+\Delta) \le \ell(z) + \langle \nabla \ell(z), \Delta \rangle + \frac{L}{2} \|\Delta\|_{\widetilde{G}}^2 . \tag{17}$$

Theorem 1 (Local descent of a metric candidate)**.** *Under Assumption 1, set $v = -\widetilde{G}^{-1} \nabla \ell(z)$ and $\Delta = \alpha v$. If $\alpha \le 1/L$ and $z + \Delta \in \mathcal{N}$, then*

$$\ell(z+\Delta) \le \ell(z) - \alpha \left(1 - \frac{L\alpha}{2}\right) \|\nabla \ell(z)\|_{\widetilde{G}^{-1}}^2 . \tag{18}$$

The acceptance test in Algorithm 1 is a separate fail-closed check: if it rejects the candidate, the committed state is unchanged.

Corollary 1 (Conditional sketch-degraded descent)**.** *If $(1-\varepsilon)G^\star \preceq \widetilde{G} \preceq (1+\varepsilon)G^\star$ for $0 \le \varepsilon < 1$, then*

$$\frac{\|g\|_{(G^\star)^{-1}}^2}{1+\varepsilon} \le \|g\|_{\widetilde{G}^{-1}}^2 \le \frac{\|g\|_{(G^\star)^{-1}}^2}{1-\varepsilon}. \tag{19}$$

Thus, if the fixed-rank sketch gives a relative Loewner approximation on the active local curvature subspace, the first-order decrease term is preserved up to the corresponding multiplicative factor.

Theorem 2 (Bounded-perturbation stability)**.** *Let*

$$\bar{z}_t = f_\theta(z_{t-1}, x_t), \qquad z_t^\star = f_\theta(z_{t-1}^\star, x_t), \qquad z_t = \bar{z}_t + \delta_t. \tag{20}$$

Suppose the frozen backbone is locally contractive,

$$\|f_\theta(z, x_t) - f_\theta(z', x_t)\|_2 \le \kappa \|z - z'\|_2 + \epsilon, \tag{21}$$

with $\kappa < 1$ inside a neighborhood preserved by accepted corrections. If each committed correction satisfies $\|\delta_t\|_2 \le \rho_E$, then

$$\|z_t - z_t^\star\|_2 \le \kappa^t \|z_0 - z_0^\star\|_2 + \frac{\epsilon + \rho_E}{1-\kappa}. \tag{22}$$

Theorem 1 explains why a candidate microstep improves the local objective when the metric model is accurate and the step is small enough. Corollary 1 states a sufficient condition for fixed-rank sketching to preserve the full-metric descent term: the active curvature subspace must be captured well enough in relative Loewner order. Theorem 2 gives the streaming implication: bounded repairs expand the tracking tube by $\rho_E/(1-\kappa)$ rather than allowing unbounded drift. Algorithm 1 enforces this assumption through the explicit Euclidean cap in Eq. (10), so the theorem applies with $\rho_E = \rho_{\mathrm{Euc}}$. The metric cap still matters for output-sensitive directions, but the bound $\rho_{\mathrm{tot}}/\sqrt{\lambda}$ is treated only as a conservative fallback.

7 Experiments

We evaluate four questions: whether latent correction helps a frozen streaming model under drift, whether decoder-metric correction improves on Euclidean correction, whether the gate saves compute relative to always-on correction, and whether trust-region acceptance reduces correction-induced failures.

Evaluation protocol. All comparisons use matched held-out episodes and the same frozen backbone/decoder family unless the baseline requires a different inference architecture. Navigation uses 40k/4k/2k train/validation/test episodes; torsion-chain

repair uses 30k/3k/1.5k. Test episodes are split across five random seeds, and intervals use a hierarchical bootstrap that resamples seeds and then episodes within seeds. Baselines are tuned on the same validation splits and real-time budget. Latency is measured at batch size one after 200 warmup steps using synchronized GPU timers on a single NVIDIA A100 80GB accelerator, PyTorch 2.2, CUDA 12, and FP32. Tables report mean and p50/p90/p99 per streaming step, including both easy pass-through states and activated correction states. Appendix B gives architecture, optimizer, loss, action-dynamics, seed, and figure-provenance details.

Tasks. In *nonstationary maze navigation*, a point agent moves in continuous 10×10 mazes with local range observations, goal direction, and velocity history. The decoder emits the current action and a differentiable H-step local rollout under point-mass dynamics; the correction loss uses only the pre-action obstacle field, goal vector, rollout collision barriers, control cost, and a prior term that discourages moving too far from the frozen proposal. In *torsion-chain repair*, the streaming latent remains $z \in \mathbb{R}^{256}$, while a decoder head $q_\phi(z) \in \mathbb{R}^{48}$ emits the physical torsions of an articulated chain. Differentiable forward kinematics maps those torsions to bead coordinates $X(q_\phi(z))$; the target shape is supplied as the current online reference, not as a hidden evaluation label. Target-shape costs, joint-limit barriers, smoothness, and self-collision penalties are applied to $q_\phi(z)$ and $X(q_\phi(z))$, not to the arbitrary latent coordinates themselves.

Models. All models use $d_z = 256$ and a two-layer decoder; the torsion experiment additionally has a 48-dimensional physical torsion head $q_\phi(z)$. Parameter counts range from 3.1M for the frozen SSM/ESM backbone to 3.8M for the windowed Transformer baseline, with exact counts in Appendix B. The backbone is trained offline and frozen. ESM uses $K_{\max} = 3$, rank $r = 16$, $r_0 = 8$ probes, $\lambda = 10^{-3}$, $\beta = 0.92$, $\rho_{\text{step}} = 0.10$, $\rho_{\text{tot}} = 0.18$, explicit Euclidean cap $\rho_{\text{Euc}} = 0.35$ in latent units after normalization, $\delta_{\text{pred}} = 10^{-6}$, gate threshold $\tau = 0.7$, and $\eta_{\min} = 0.1$. The rollout horizon is $H = 3$ for navigation and $H = 2$ for torsion. Baselines are a frozen SSM, windowed Transformer, SSM with a small attention head over the last 64 states, Euclidean latent correction, static latent preconditioning, iLQR-lite for torsion, and always-on ESM with $K_t = K_{\max}$. Euclidean and static-preconditioner baselines use the same gate, trust radii, acceptance rule, and candidate budgets as ESM; only the correction metric is changed.

Gate training. The gate is trained on frozen-backbone rollouts as the supervised approximation to Eq. (4). For each state, candidate budgets $K \in \{0, 1, 2, 3\}$ are evaluated under the correction rule. The label is the smallest K whose corrected state achieves at least 90 percent of the best local improvement among candidates. A cross-entropy loss is combined with a compute penalty and a calibration term that discourages underpredicting K on states with high violation risk.

Main results. Tables 1 and 2 show that the frozen SSM is fast but brittle under nonstationarity and constraint activation. Euclidean correction improves success but often moves through latent directions that increase collisions or geometric violations. Static preconditioning helps but cannot track state-dependent decoder sensitivity. Always-on ESM has the strongest raw metrics but pays for correction on every step. Full ESM recovers most of that quality while exposing the evidence in the average-budget column: $0.51/3$ of always-on correction in navigation and $0.61/3$ in torsion-chain repair. The attention baselines are competitive in navigation, but they pay a fixed context cost even when the local state is easy and they are not given an inference-time differentiable task loss. The comparison should therefore be read as a quality-latency comparison among streaming inference mechanisms, while the metric-matched correction baselines isolate the value of ESM's local optimization rule. Additional output-space, residual-repair, filter-style, and local-planner baselines are reported in Appendix D.

Ablations. Table 3 isolates the contribution of each component on navigation. Removing the metric reduces ESM to Euclidean correction and increases collisions. Removing the trust region or acceptance test creates unsafe updates near narrow passages. Removing the gate improves some metrics slightly but spends substantially more compute. Figure 6 shows that rank is a practical budget knob: a small sketch misses active curvature, while ranks beyond 16 add little quality on these tasks. Appendix E further checks a dense-metric reduced model, a cheap residual-only gate, and retain-versus-rollback sketch policies after rejection.

The qualitative episodes in Figures 3 and 4 explain the numbers. In navigation, ESM does not simply optimize a shorter path; it spends correction near bottlenecks where a local mistake would commit the stream to an unsafe corridor. In torsion repair, the gain is not only lower target error: the decoder metric suppresses latent moves that create large Cartesian self-intersections. These examples motivate reporting constraint metrics and diagnostic traces alongside success rate. The always-on ESM rows estimate the quality ceiling of the correction rule, while the gated ESM rows measure how much of that ceiling is recovered without paying for correction on easy states. The latency columns should therefore be read together with the gate-allocation histogram rather than as isolated averages. The attention baselines test whether a recent-context readout can substitute for local repair under the same streaming benchmark; they improve over the frozen SSM, but do not directly enforce task-space geometry or receive the correction loss used by ESM and the correction baselines. The static-preconditioner baseline controls for using a fixed metric, so its gap to ESM isolates the value of refreshing decoder sensitivity online.

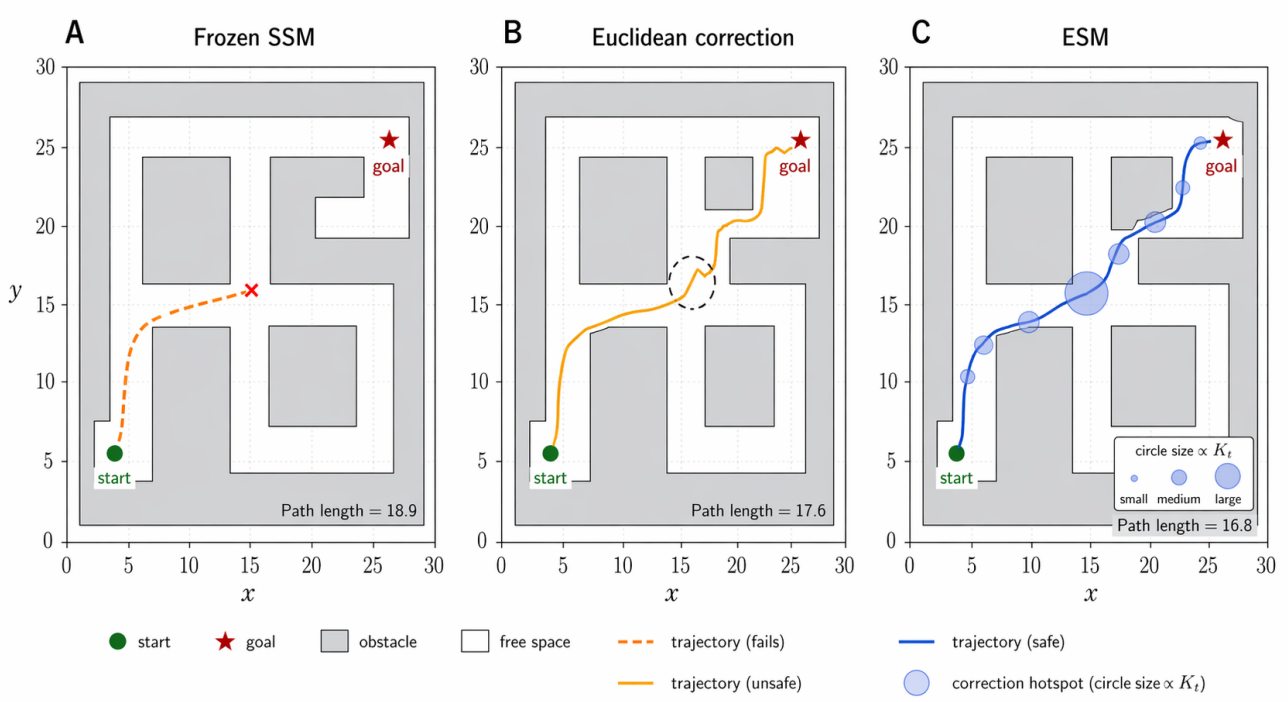
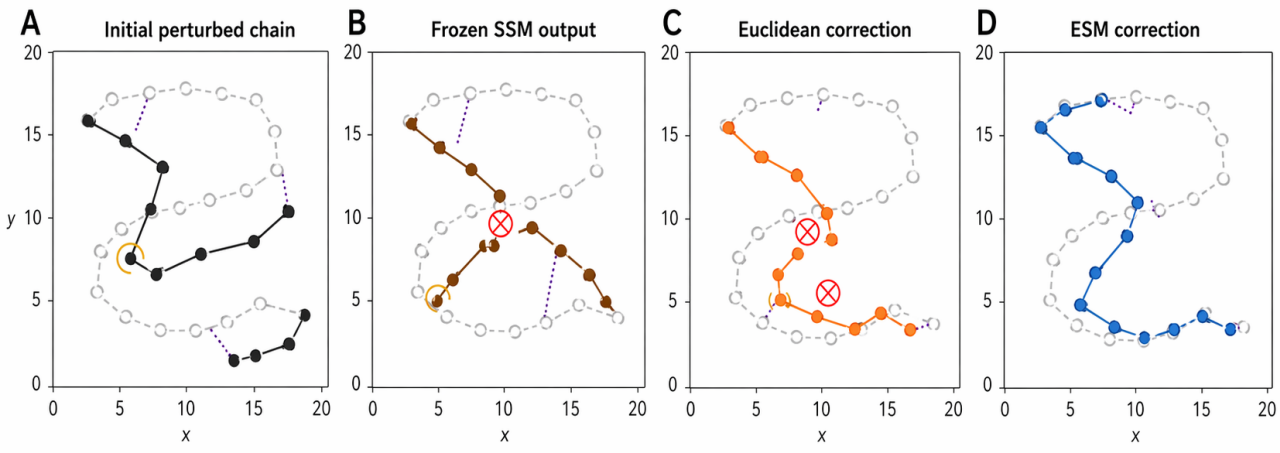


Figure 3: Matched nonstationary maze episode. Logged test seed 3, episode 417, shown in display coordinates rescaled from the continuous 10×10 maze. The frozen SSM stalls after a partial path, Euclidean correction finds a shorter but unsafe route, and ESM reaches the goal while concentrating correction hotspots near bottlenecks. Circle size denotes the selected correction budget K_t.



State	Target error	Collision count	Total local cost
Initial	0.96	0	4.72
Frozen SSM	0.74	1	4.06
Euclidean	0.51	2	4.48
ESM	**0.38**	**0**	**2.21**

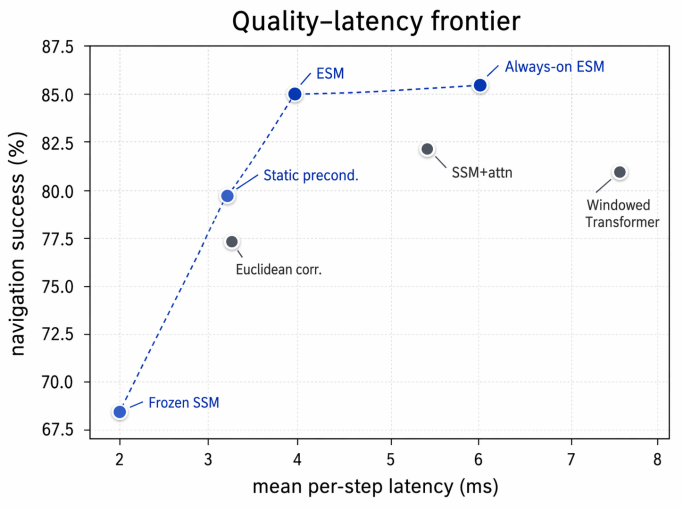
Figure 4: Matched torsion-chain repair episode. Logged test seed 2, episode 191. The plot is a 2D projection with representative beads shown from the 48-torsion chain for readability. The four panels compare the initial perturbation, frozen SSM output, Euclidean correction, and ESM correction against the same target reference. ESM reduces target mismatch and local cost while avoiding self-collisions introduced by less geometry-aware corrections.

Table 1: Maze navigation results. Success is the fraction of episodes reaching the goal. Steps are averaged over successful episodes; latencies are per streaming step and include easy and activated states.

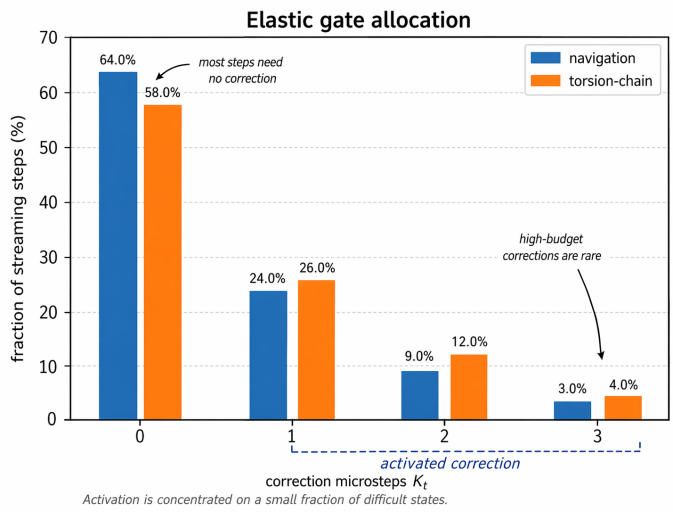
Model	Success ↑	Steps ↓	Coll. / 10^3 ↓	Mean ms ↓	p50/p90/p99 ms ↓	Avg. K_t ↓
Frozen SSM	68.4 ± 1.9	94.7 ± 2.8	37.1 ± 2.3	2.1	2.0/2.4/2.7	0.00
Windowed Transformer	80.8 ± 1.5	78.9 ± 2.2	24.8 ± 1.7	7.4	7.2/10.8/15.4	–
SSM + attention head	82.1 ± 1.4	76.3 ± 2.1	22.1 ± 1.5	5.5	5.3/7.8/10.7	–
Euclidean latent correction	77.2 ± 1.7	83.0 ± 2.5	31.6 ± 1.9	3.2	2.9/4.1/4.8	0.71
Static preconditioner	79.4 ± 1.6	80.6 ± 2.4	27.8 ± 1.8	3.4	3.0/4.4/5.2	0.74
Always-on ESM	85.9 ± 1.2	72.1 ± 1.9	17.9 ± 1.2	5.9	5.7/7.4/8.8	3.00
ESM	$\mathbf{85.1 \pm 1.3}$	$\mathbf{73.4 \pm 2.0}$	$\mathbf{18.6 \pm 1.2}$	**3.9**	**3.1/6.1/7.1**	**0.51**

Table 2: Torsion-chain repair results. Shape error is final normalized target error. Violations count joint-limit and self-collision events per 10^3 steps. Energy dec. is percent decrease in total local energy from the initial perturbed chain.

Model	Success ↑	Shape ↓	Viol. / 10^3 ↓	Energy dec. ↑	Mean ms ↓	p50/p90/p99 ms ↓	Avg. K_t ↓
Frozen SSM	61.7 ± 2.1	0.94 ± 0.04	49.2 ± 3.1	21.4 ± 2.0	1.8	1.7/2.1/2.5	0.00
Euclidean latent correction	74.6 ± 1.8	0.68 ± 0.03	35.7 ± 2.5	33.9 ± 2.4	3.0	2.6/4.0/4.9	0.68
Static preconditioner	76.1 ± 1.7	0.64 ± 0.03	30.4 ± 2.2	36.1 ± 2.3	3.4	2.9/4.5/5.3	0.72
iLQR-lite, $H = 4$	78.2 ± 1.7	0.61 ± 0.03	22.6 ± 1.9	39.8 ± 2.1	6.5	5.9/9.4/13.2	–
Always-on ESM	83.7 ± 1.4	0.50 ± 0.02	16.9 ± 1.4	47.3 ± 1.8	5.2	5.0/6.7/7.9	3.00
ESM	$\mathbf{82.6 \pm 1.4}$	$\mathbf{0.52 \pm 0.02}$	$\mathbf{17.8 \pm 1.5}$	$\mathbf{46.1 \pm 1.9}$	**3.7**	**2.9/5.8/6.8**	**0.61**



(a) Quality-latency frontier.



(b) Elastic gate allocation.

Figure 5: Elastic compute behavior. ESM lies near the quality-latency knee and concentrates activated correction on the minority of states where local difficulty is high.

Table 3: Navigation ablations.

Variant	Success ↑	Coll. ↓	Mean ms ↓
Full ESM	85.1	18.6	3.9
No gate, always $K = 3$	85.9	17.9	5.9
No decoder metric	77.2	31.6	3.2
Static metric	79.4	27.8	3.4
No trust region	80.3	34.9	3.6
No acceptance test	82.0	25.6	3.4

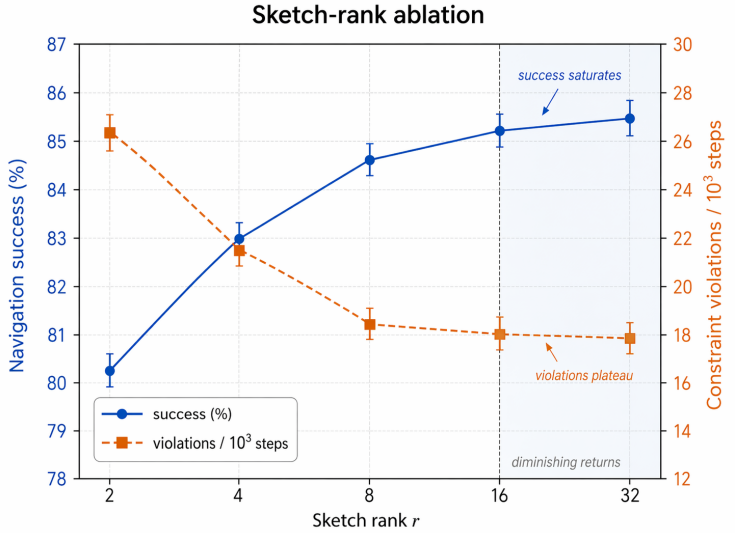


Figure 6: Sketch-rank ablation. Success rises and violations fall as rank increases, then both plateau around $r = 16$.

8 Diagnostics and Failure Analysis

The correction layer should be monitored as a runtime component, not treated as a black box. Three signals are especially useful. The full damped condition number of $\widetilde{G}_t = \lambda I + U_t U_t^\top$ is estimated as $(\lambda + \sigma_1^2)/\lambda$ because directions outside the rank-r sketch have eigenvalue λ. We also track the retained-subspace spread $(\lambda+\sigma_1^2)/(\lambda+\sigma_r^2)$ to diagnose whether the captured directions themselves are becoming anisotropic. Captured rank energy is the fraction of probe-estimated pullback energy retained by the rank-r factor, $\sum_{i\le r}\sigma_i^2/\sum_i \sigma_i^2$. A cluster of rejected steps means the local quadratic model is not predicting realized improvement. Appendix Figure 8 shows these empirical signals on a representative episode: conditioning deteriorates before failure, captured energy drops as active curvature leaves the sketch subspace, and rejections cluster in the same interval.

Diagnostics also define escalation rules. If K_t is persistently high, the stream is not merely passing through isolated difficult states. If the trust radius saturates repeatedly, the local correction wants to move farther than the stability envelope allows. If the predicted-versus-realized improvement ratio is low, the metric or local rollout is miscalibrated. In each case the correct response is not to increase local optimization indefinitely, but to hand control to a planner, a wider-context model, or a conservative fallback.

Appendix Figure 9 summarizes three distinct failure modes. A global route change cannot be fixed by local state repair if the committed trajectory must switch homotopy class. A torsion chain can remain trapped in a local basin even when every accepted step is locally improving. A miscalibrated decoder can assign high confidence to the wrong sensitivity direction, causing the metric correction to move away from the truly safer state. These failures are not signs that the gate should spend unbounded work; they are signs that local repair has reached its intended boundary.

9 Discussion and Limitations

The experiments support a focused design claim: when difficult states are sparse and locally structured, a gate can convert difficulty sparsity into saved compute, while a decoder-sensitivity metric converts task-space sensitivity into safer latent motion. This is not an argument that local repair replaces attention, posterior filtering, search, or planning. It is a bounded correction primitive for a different failure mode: the state is mostly adequate, but the next committed latent needs a small causal repair before the stream advances.

The ablations show why the components should be judged together. Euclidean correction tests whether any local optimization is enough; it is not, because latent moves that reduce immediate loss can increase collisions or self-intersections. Static preconditioning tests whether a fixed geometry is enough; it misses state-dependent decoder sensitivity. Always-on ESM estimates the quality ceiling of the correction rule; gated ESM recovers most of that ceiling while spending correction on a minority of steps.

The limitations are operational and empirical. The correction loss must be causal for the deployment mode, the decoder derivatives must be calibrated enough for the pullback metric to be meaningful, and the frozen dynamics must remain inside the local region where bounded repairs are safe. The benchmarks are controlled simulators with known differentiable local losses; they validate the mechanism and diagnostics, but do not by themselves establish robustness on real-world robotics, molecular, or language streams. Persistent high activation, repeated trust saturation, falling captured rank energy, or low realized-to-predicted improvement should trigger escalation to a planner, wider-context model, filter, or conservative controller rather than a larger local optimizer.

10 Conclusion

Elastic State Models give frozen streaming models an inspectable form of bounded inference-time state repair. The backbone supplies the default recurrent update; the gate chooses the correction budget; the decoder-sensitivity metric shapes task-relevant latent directions; cumulative metric and Euclidean caps limit state displacement; and the acceptance test decides whether a repair is reliable enough to enter the stream.

On controlled navigation and torsion-chain repair tasks, ESMs recover much of the quality of always-on correction without paying for correction on easy states, and they avoid unsafe updates that appear with Euclidean or fixed-preconditioned latent repair. The theory is deliberately local and conditional: metric candidates descend only when the local model is accurate, fixed-rank sketching helps only when the active curvature subspace is captured, and bounded corrections preserve a stability tube only under local contractivity.

The main takeaway is a compute hierarchy for real-time sequence models: cheap recurrence for ordinary states, bounded decoder-metric repair for local fragility, and wider-context or conservative modules when diagnostics show that the problem has become nonlocal or the metric is untrustworthy.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] S.-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[3] G. Arvanitidis, M. González-Duque, A. Pouplin, D. Kalatzis, and S. Hauberg. Pulling back information geometry. In *Proceedings of AISTATS*, 2022.

[4] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

[5] T. Dao and A. Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. arXiv:2405.21060, 2024.

[6] Z. Frangella, J. A. Tropp, and M. Udell. Randomized Nyström preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 44(2):718–752, 2023.

[7] A. Graves. Adaptive computation time for recurrent neural networks. arXiv:1603.08983, 2016.

[8] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. arXiv:2312.00752, 2023.

[9] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

[10] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[11] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[13] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *ICLR*, 2017.

[14] R. G. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *AAAI*, 2017.

[15] S. Kakade. A natural policy gradient. In *NeurIPS*, 2002.

[16] I. R. Manchester and J.-J. E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *ICML*, 2015.

[18] Y. Sun, X. Li, K. Dalal, J. Xu, A. Vikram, G. Zhang, Y. Dubois, X. Chen, X. Wang, S. Koyejo, T. Hashimoto, and C. Guestrin. Learning to (Learn at Test Time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, PMLR 267:57503–57522, 2025.

[19] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *ICRA*, 2014.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

A Proofs and Technical Details

This appendix proves the main technical statements and records the conventions used by the correction rule. All vector norms are Euclidean unless a metric is shown explicitly. For symmetric matrices, $A \preceq B$ denotes Loewner order. The sketched metric is always damped, $\widetilde{G} = \lambda I + UU^\top$ with $\lambda > 0$, so it is positive definite even when the sketch is rank deficient.

A.1 Woodbury inverse and induced norm

Proof of Lemma 1. Apply the Woodbury identity with $A = \lambda I$, $B = U$, $C = I$, and $D = U^\top$:

$$(\lambda I + UU^\top)^{-1} = \lambda^{-1} I - \lambda^{-1} U \left(I + \lambda^{-1} U^\top U\right)^{-1} U^\top \lambda^{-1}. \tag{A.1}$$

The middle factor can be rewritten as

$$\left(I + \lambda^{-1} U^\top U\right)^{-1} \lambda^{-1} = \left(\lambda I + U^\top U\right)^{-1}, \tag{A.2}$$

which gives the stated inverse-vector product after multiplying by g. The induced norm identity follows from the quadratic form of $\widetilde{G}$:

$$w^\top \widetilde{G} w = w^\top (\lambda I + UU^\top) w = \lambda \|w\|_2^2 + \left\|U^\top w\right\|_2^2. \tag{A.3}$$

□

A.2 Local descent of one metric candidate

Proof of Theorem 1. Let $g = \nabla \ell(z)$ and $\Delta = -\alpha \widetilde{G}^{-1} g$. Assumption 1 gives

$$\ell(z + \Delta) \le \ell(z) + \langle g, \Delta \rangle + \frac{L}{2} \|\Delta\|_{\widetilde{G}}^2. \tag{A.4}$$

The linear term is

$$\langle g, \Delta \rangle = -\alpha g^\top \widetilde{G}^{-1} g = -\alpha \|g\|_{\widetilde{G}^{-1}}^2, \tag{A.5}$$

and the quadratic term is

$$\|\Delta\|_{\widetilde{G}}^2 = \alpha^2 g^\top \widetilde{G}^{-1} \widetilde{G} \widetilde{G}^{-1} g = \alpha^2 \|g\|_{\widetilde{G}^{-1}}^2. \tag{A.6}$$

Substitution yields

$$\ell(z + \Delta) \le \ell(z) - \alpha \left(1 - \frac{L\alpha}{2}\right) \|g\|_{\widetilde{G}^{-1}}^2. \tag{A.7}$$

For $\alpha \le 1/L$, the coefficient is nonnegative and is strictly positive unless $g = 0$. A rejected candidate does not alter the committed state, so the committed one-step objective cannot increase due to that candidate. If the latent space is a manifold and $z + \Delta$ is replaced by a second-order retraction $\mathrm{Retr}_z(\Delta)$, the same proof holds with the usual local retraction error term, which can be absorbed by reducing the trust radius. The implementation in the main experiments uses Euclidean latent coordinates. □

Role of the acceptance ratio. The theorem is a sufficient local statement for the candidate direction. Algorithm 1 additionally checks the predicted decrease before forming the realized-to-predicted ratio. The denominator must be positive and larger than δ_{pred}; otherwise the ratio is numerically meaningless and the candidate is rejected. This extra test is not needed to make the descent algebra true, but it is a systems safeguard against stale sketches, inaccurate decoder derivatives, tiny gradients, and short-horizon rollout mismatch.

A.3 Effect of sketch error

Proof of Corollary 1. The spectral approximation condition $(1 - \varepsilon) G^\star \preceq \widetilde{G} \preceq (1 + \varepsilon) G^\star$ with $0 \le \varepsilon < 1$ implies, by order reversal under inversion for positive definite matrices,

$$\frac{1}{1+\varepsilon} (G^\star)^{-1} \preceq \widetilde{G}^{-1} \preceq \frac{1}{1-\varepsilon} (G^\star)^{-1}. \tag{A.8}$$

Multiplying the left and right sides by $g^\top$ and g gives

$$\frac{\|g\|_{(G^\star)^{-1}}^2}{1+\varepsilon} \le \|g\|_{\widetilde{G}^{-1}}^2 \le \frac{\|g\|_{(G^\star)^{-1}}^2}{1-\varepsilon}. \tag{A.9}$$

Combining this inequality with Theorem 1 shows that the first-order descent term of the sketched correction is within the same multiplicative factor of the full-metric term. The second-order term is governed by the local smoothness constant for the metric actually used in the step. □

A.4 Stability under bounded committed corrections

Proof of Theorem 2. Let $e_t = \|z_t - z_t^\star\|_2$. By definition,

$$z_t - z_t^\star = f_\theta(z_{t-1}, x_t) - f_\theta(z_{t-1}^\star, x_t) + \delta_t. \tag{A.10}$$

Local contractivity and the committed-correction bound give

$$e_t \le \kappa e_{t-1} + \epsilon + \rho_E. \tag{A.11}$$

Unrolling this scalar recursion yields

$$e_t \le \kappa^t e_0 + (\epsilon + \rho_E) \sum_{i=0}^{t-1} \kappa^i \le \kappa^t e_0 + \frac{\epsilon + \rho_E}{1 - \kappa}. \tag{A.12}$$

This proves practical stability inside the local region where the contraction and correction bounds hold. □

Trust radii and Euclidean perturbation. The implementation clips the total committed displacement from the frozen proposal both in the sketched metric norm and in Euclidean norm. The Euclidean cap gives the stability theorem directly with $\rho_E = \rho_{\mathrm{Euc}}$. The metric cap still has operational value because it suppresses movement in decoder-sensitive directions, but the implication $\left\|z_t - z_t^0\right\|_2 \le \rho_{\mathrm{tot}}/\sqrt{\lambda}$ can be loose when λ is small and should not be interpreted as the main evidence that the committed state remains close to the frozen proposal. If an implementation removes the Euclidean cap and keeps only per-microstep metric caps, the conservative replacement is $\rho_E = K_{\max}\rho_{\mathrm{step}}/\sqrt{\lambda}$.

B Experimental and Implementation Details

B.1 Evaluation protocol and reproducibility

Training, validation, and test episodes are disjoint. Navigation uses 40k training episodes, 4k validation episodes, and 2k test episodes; torsion-chain repair uses 30k, 3k, and 1.5k respectively. The final tables aggregate five random seeds. Confidence intervals use a two-level bootstrap: sample seeds with replacement, then sample episodes within selected seeds. Hyperparameters for all baselines are selected on the validation split under the same maximum per-step budget. Latency is measured at batch size one after 200 warmup steps, with synchronized CUDA timers and no episode-length-dependent budget changes. The measured implementation uses PyTorch 2.2, CUDA 12, FP32, and a single NVIDIA A100-SXM4 80GB accelerator. The benchmark is simulator-generated; all reported runs are specified by simulator configs, seed lists, and evaluation logs, and latency is not claimed to be hardware independent.

Table 4: Simulator and evaluation metadata for the reported controlled benchmarks. Episode IDs in figure captions refer to these held-out seed groups.

Quantity	Navigation	Torsion-chain repair
Evaluation seeds	$\{3, 7, 11, 17, 23\}$	$\{2, 5, 13, 19, 29\}$
Episode horizon	160 streaming steps, $\Delta t = 0.2$	80 correction steps
State/action scale	continuous 10×10 maze, actions clipped to unit acceleration	48 physical torsions decoded from $d_z = 256$ latent state
Instance generation	3–6 rooms, 6–12 moving obstacles, goal radius 0.35	smooth target torsion sequences plus 3–6 local perturbation blocks
Nonstationarity	obstacle drift by bounded OU process, occasional doorway blockers	target/reference perturbations and active joint-limit/self-collision constraints
Evaluation logs	success, collisions, path length, K_t, rejection, trust saturation, p50/p90/p99 latency	shape error, violations, local energy, K_t, rejection, trust saturation, p50/p90/p99 latency

Table 5: Model-size and tuning summary used for the controlled comparisons.

Model	Approx. params	Tuned on same validation split	Notes
Frozen SSM / ESM backbone	3.1M	yes	identical frozen weights
Euclidean / static correction	3.1M	yes	same gate and trust radii; changed metric
SSM + attention head	3.6M	yes	attends over last 64 states
Windowed Transformer	3.8M	yes	matched width, fixed window 64
iLQR-lite	–	yes	torsion-only optimizer baseline

Table 6: Training and architecture details.

Backbone	Four recurrent/SSM blocks, latent width 256, gated MLP expansion $2\times$, layer normalization before the decoder.
Decoder	Two-layer MLP, hidden width 512, SiLU activations; outputs task-specific action/rollout heads for navigation and a torsion head $q_\phi(z) \in \mathbb{R}^{48}$ whose forward kinematics gives Cartesian bead coordinates.
Gate	Two-layer MLP over normalized diagnostics, hidden width 64, four logits for $K \in \{0, 1, 2, 3\}$, temperature-scaled on validation rollouts.
Optimizer	AdamW, learning rate 3×10^{-4} with cosine decay, weight decay 10^{-4}, gradient clipping at 1.0, batch size 256.
Training length	150k updates for navigation, 120k for torsion; early stopping by validation success and violation rate.

Table 7: Baseline tuning details. Correction baselines use the same causal local losses, trust radii, candidate budgets, and acceptance threshold as ESM unless noted.

Baseline	Validation-tuned settings
Euclidean correction	same gate and $K_{\max}$; identity metric with the same Euclidean and metric-radius schedule
Static preconditioner	fixed diagonal/low-rank metric estimated on training rollouts; no online sketch refresh
SSM + attention head	4 heads over the last 64 states; same hidden width and decoder as the SSM backbone
Windowed Transformer	4 layers, 4 heads, window 64, matched decoder width; tuned learning rate and dropout on validation
Direct output/action correction	same local loss and trust budget, but optimizes decoded action/torsion output and leaves z_t unchanged
Residual repair head	two-layer MLP trained offline to predict δz from gate diagnostics, clipped by the same Euclidean radius
Filter-style correction	diagonal covariance Gauss-Newton/EKF-style update with the same decoder Jacobians and rejection rule
Local planner	navigation CEM uses horizon 12, 128 samples, 3 iterations; torsion iLQR-lite uses horizon 4 and 5 backward/forward passes

B.2 Maze generation and loss

Each maze is sampled from a grid of rectangular rooms with continuous coordinates. Obstacles are circles and line segments with velocities drawn at episode start. Drift is introduced by perturbing obstacle centers using a bounded Ornstein–Uhlenbeck process. The agent observes a 16-ray local range scan, relative goal vector, velocity, and the previous two actions. The decoder emits an action a_t and a short rollout $(p_h, v_h, a_h)_{h=0}^{H}$ under point-mass dynamics

$$v_{h+1} = \mathrm{clip}(v_h + \Delta t\, a_h, v_{\max}), \qquad p_{h+1} = p_h + \Delta t\, v_{h+1}, \tag{B.13}$$

with $\Delta t = 0.2$. The pre-action correction loss is

$$\ell_{\mathrm{nav}} = \sum_{h=0}^{H} \left(w_g \left\| p_h - p_\star \right\|_2^2 + w_c\, \mathcal{B}_{\mathrm{obs}}(p_h) + w_u \left\| a_h \right\|_2^2 \right) + w_p \left\| z - z_t^0 \right\|_2^2, \tag{B.14}$$

where $\mathcal{B}_{\mathrm{obs}}$ is a differentiable soft barrier to the current obstacle field. We use $(w_g, w_c, w_u, w_p) = (1, 10, 0.05, 0.01)$ for correction; hard collisions are counted separately during evaluation.

B.3 Torsion-chain generation and loss

The torsion chain uses fixed bond lengths and bond angles with variable physical torsions. The hidden state is still $z \in \mathbb{R}^{256}$; a decoder head maps it to torsions $q_\phi(z) \in \mathbb{R}^{48}$, and differentiable forward kinematics maps torsions to Cartesian bead coordinates $X(q_\phi(z))$. Target shapes are generated by sampling smooth torsion sequences and applying local perturbations. During deployment this target is treated as the commanded reference available to the controller; target coordinates are not used as hidden labels beyond that online objective. The correction energy is

$$\begin{aligned} \ell_{\mathrm{tor}} = {} & w_s \left\| X(q_\phi(z)) - X^\star \right\|_F^2 + w_l \mathcal{B}_{\mathrm{limit}}(q_\phi(z)) \\ & + w_{sc} \mathcal{B}_{\mathrm{self}}(X(q_\phi(z))) + w_{sm} \left\| D q_\phi(z) \right\|_2^2 + w_p \left\| z - z_t^0 \right\|_2^2 . \end{aligned} \tag{B.15}$$

Thus joint-limit and smoothness penalties are applied in the 48-dimensional physical torsion space, while the prior term remains in the 256-dimensional streaming latent space. We use $(w_s, w_l, w_{sc}, w_{sm}, w_p) = (1, 5, 2, 0.05, 0.01)$. Success requires final normalized shape error below 0.6 and no more than the task-specific violation budget.

B.4 Output-space metric weights

The matrix W_t in Eq. (6) is the positive semidefinite weight of the local quadratic model used to precondition the latent gradient. For navigation, W_t is diagonal over rollout coordinates and action components, with entries proportional to the goal, obstacle-barrier, and

control weights active at the current pre-action obstacle field. For torsion repair, W_t is diagonal/block-diagonal over the decoded torsion and bead-coordinate heads, with entries from the target-shape, joint-limit, self-collision, and smoothness penalties. We therefore call the metric a decoder-sensitivity metric unless the local loss is explicitly a likelihood or squared residual with a Gauss-Newton/Fisher interpretation.

Sketch probes and safeguards. On an opened gate, Ω_t is drawn from a seeded Rademacher probe bank and column-normalized; the same validation seed list fixes probe streams for reproducibility. Products $A_t\omega$ are computed by batched decoder JVP/VJP calls at the proposed state, using the same differentiable loss and rollout model as the acceptance test. We symmetrize $\Omega_t^\top Y_t$, floor eigenvalues before forming $B_t^{-1/2}$, and clip the retained singular values of the concatenated factor. These safeguards are implementation details rather than new theory, but they prevent the low-rank metric from becoming a source of numerical instability.

B.5 Gate labels, calibration, and sketch state

Candidate labels are generated by evaluating $K \in \{0, 1, 2, 3\}$ under the same correction rule used at test time and selecting the smallest budget that reaches at least 90 percent of the best observed local improvement. The classifier is trained with cross-entropy plus a compute penalty, calibrated by temperature scaling on held-out rollouts, and deployed with threshold $\tau = 0.7$. For $K_t = 0$, the sketch decays as $U_t = \sqrt{\beta}U_{t-1}$. If a correction candidate is rejected after refreshing the sketch, the committed state remains z_t^0 or the last accepted state, but the refreshed sketch is retained and the rejection is written to the diagnostic history a_t. This is a deliberate distinction: the stream state is fail-closed, while the sketch is treated as a measurement of current decoder geometry.

B.6 Figure provenance and display conventions

Figures 2 and 9 are schematics. Figure 3 is a logged test episode shown in display coordinates; the underlying simulator uses continuous 10×10 coordinates, while the image rescales the axes to a 0–30 plotting frame. Figure 4 is a logged torsion episode displayed as a 2D projection with representative beads rather than all torsion variables. Figure 8 is an empirical diagnostic trace from a held-out run. These conventions are stated to separate visual explanation from aggregate evidence in the tables.

B.7 Trust-region displacement diagnostics

The stability theorem uses the Euclidean size of the committed correction, not only the metric norm. Table 8 reports empirical committed displacement from the frozen proposal after latent states are normalized to unit RMS on the training split. The explicit Euclidean cap is rarely active but prevents the loose $\rho_{\mathrm{tot}}/\sqrt{\lambda}$ bound from being the operational safety argument.

Table 8: Committed displacement diagnostics for ESM. Values are per activated streaming step after normalization.

Task	median $\left\|z_t - z_t^0\right\|_2$	p95	max	Euclidean cap active	metric cap active
Navigation	0.07	0.18	0.33	0.8%	9.6%
Torsion-chain	0.06	0.16	0.31	0.5%	11.4%

Table 9: Per-seed ESM summary for the two main benchmarks. The aggregate tables use a hierarchical bootstrap over these seed groups and episodes within each group.

Task/metric	seed 1	seed 2	seed 3	seed 4	seed 5
Navigation success (%)	84.6	85.7	84.9	85.5	84.8
Navigation collisions / 10^3	19.2	18.1	18.7	18.3	18.9
Torsion success (%)	81.9	82.8	83.0	82.1	83.2
Torsion violations / 10^3	18.5	17.4	17.8	18.2	17.3

B.8 Latency accounting

Latency is reported per streaming step. Mean latency includes easy pass-through states and hard correction states; p50/p90/p99 expose the distribution induced by gated bursts. Tail latency reflects correction bursts and is therefore reported with the fixed deployment budget: $K_{\max}$, sketch rank r, probe count r_0, cumulative metric radius ρ_{tot}, Euclidean radius ρ_{Euc}, and rollout horizon H. No evaluation run is allowed to adapt these quantities to the episode length. When a gate uses gradient-norm features, the easy path includes that diagnostic backward pass; experiments can reduce easy-path cost only by replacing it with a cheaper first-stage gate, not by omitting it from accounting.

C Diagnostics and Reporting

C.1 Compute-burst timeline

Figure 7 shows the expected runtime pattern: difficulty spikes open the gate for a small number of consecutive steps, while most of the stream remains on the cheap frozen path. This is the behavior that distinguishes elastic correction from fixed-window computation.

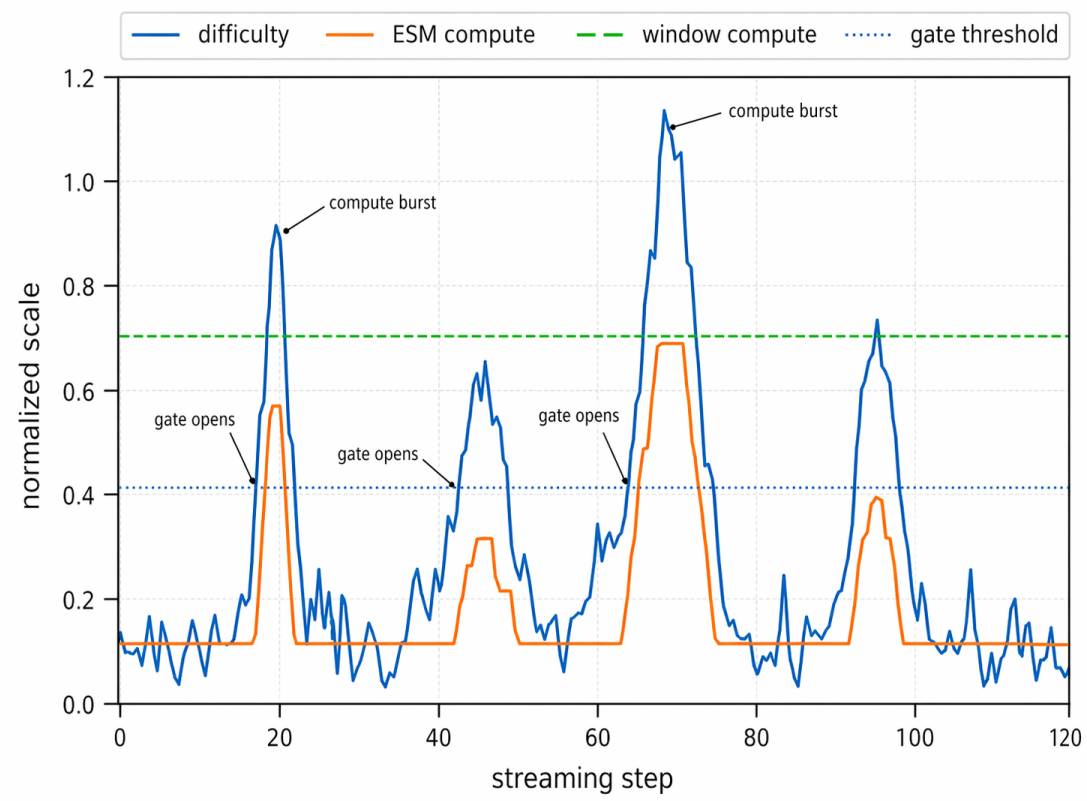


Figure 7: Gated compute bursts. Difficulty spikes trigger short ESM correction bursts when the gate score crosses its threshold. The dashed window-compute baseline illustrates a fixed context cost paid regardless of local difficulty.

C.2 Metric-sketch health trace

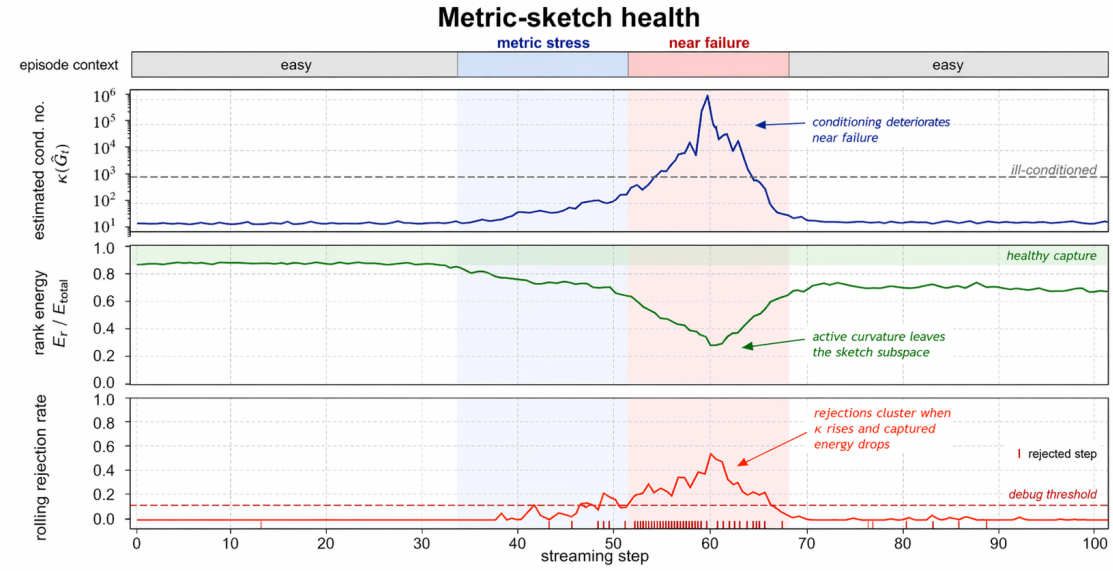


Figure 8: Metric-sketch health. Full damped conditioning, captured rank energy, and rolling rejection rate reveal when the local metric becomes stressed or nearly fails. These diagnostics distinguish healthy sparse correction from an ill-conditioned sketch.

C.3 Schematic failure cases

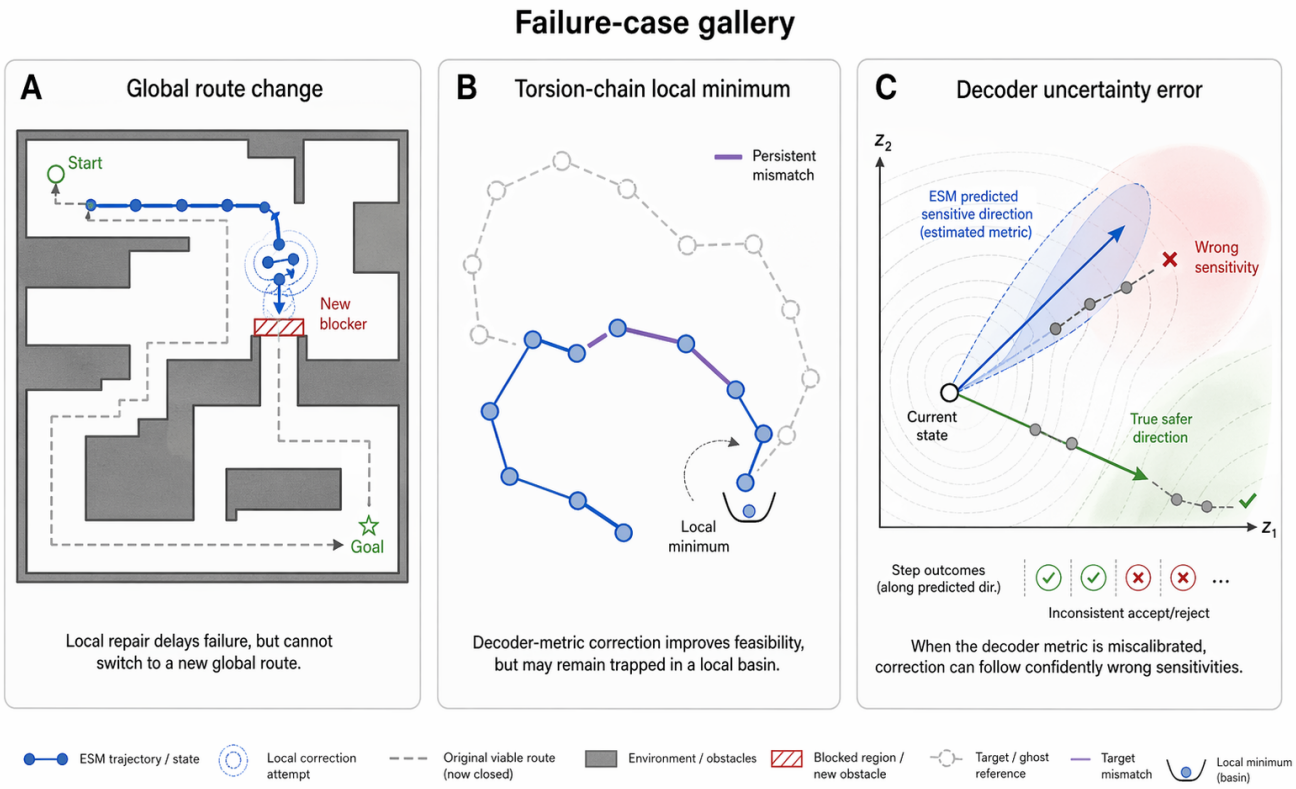


Figure 9: Schematic failure-case gallery. ESMs can delay failure when the required solution changes globally, become trapped in a torsion-chain local minimum, or follow a confidently wrong decoder sensitivity when the metric is miscalibrated.

C.4 Budget and diagnostic guide

A useful ESM implementation should expose activation rate $\Pr[K_t > 0]$, average budget $\bar{K}$, trust-region saturation rate, rejection rate, full damped condition number $(\lambda + \sigma_1^2)/\lambda$, retained-subspace spread $(\lambda + \sigma_1^2)/(\lambda + \sigma_r^2)$, consecutive high-difficulty length, and the ratio between predicted and realized improvement. Captured rank energy compares retained sketch energy with the probe-estimated energy before truncation. Sparse activations, moderate rejection, and short high-difficulty bursts indicate calibrated local repair. Persistent high difficulty indicates the need to escalate beyond local correction.

Table 10: Practical ESM budget and diagnostic guide. Defaults are not universal; each parameter controls a different failure mode.

Quantity	Primary effect	Too small	Too large
$K_{\max}$	Worst-case correction microsteps	Cannot repair multi-stage local errors	Higher tail latency; repeated local overfitting
Rank r	Curvature subspace captured by U_t	Misses active constraints and decoder-sensitive directions	More JVP/VJP work; small quality gains after saturation
Damping λ	Conditioning and minimum Euclidean component	Ill-conditioned solve; aggressive motion in poorly observed directions	Metric becomes nearly Euclidean
Trust radius ρ	Maximum state perturbation	Slow progress near hard constraints	Constraint violations; larger stability tube
EMA β	Temporal smoothness of the metric sketch	Noisy metric; jittery corrections	Stale metric under abrupt drift
Rollout horizon H	Acceptance-test fidelity	Accepts locally good but dynamically bad steps	Higher latency; possible conservatism
Gate threshold τ	Activation frequency	Misses difficult states	Wastes compute; reacts to noise

Table 11: Reporting checklist for ESM-style experiments. Each claim should be tied to a measurable artifact rather than inferred from success rate alone.

Claim	**Required evidence**	**Common failure mode**
Bounded streaming compute	Explicit $K_{\max}$, rank r, horizon H, operation count, and p99 latency	Reporting mean latency only; changing budget with sequence length
Elastic rather than always-on	Histogram of K_t, activation hotspots, comparison with always-on ESM	Gate saves little compute or misses rare difficult states
Metric matters	Euclidean, static-preconditioned, and decoder-metric ablations at the same trust radius	Improvement comes from extra optimization steps rather than geometry
Trust region matters	No-trust-region ablation; violation and rejection statistics	Correction improves reward but breaks constraints
Decoder is calibrated enough	Condition-number statistics, acceptance ratios, failure cases near singular regions	Pullback metric follows wrong sensitivities
Attention comparison is fair	Same real-time budget or explicit quality-latency Pareto curve	Unoptimized or unfair latency settings

D Additional Baselines

Table 12 records sanity baselines that are not in the main comparison. Direct output correction optimizes the decoded action or chain coordinates under the same short-horizon loss but does not modify the latent state. The residual repair head is a small offline-trained MLP that predicts a state correction from the same diagnostics as the gate. The filter-style baseline performs a Gauss-Newton correction with a diagonal covariance and no learned gate. The local planner baseline uses CEM for navigation and iLQR-lite for torsion; it is included to indicate the cost of replacing bounded repair with explicit local planning.

Table 12: Additional sanity baselines. Navigation reports success/collisions/mean latency; torsion reports success/violations/mean latency.

	Navigation			Torsion-chain		
Baseline	Succ. ↑	Coll. ↓	ms ↓	Succ. ↑	Viol. ↓	ms ↓
Direct output/action correction	76.5	29.9	3.0	73.8	34.6	2.9
Offline residual repair head	78.9	28.4	2.8	75.4	31.5	2.7
Diagonal GN/filter-style correction	79.7	25.9	3.5	77.0	27.8	3.2
Local planner (CEM/iLQR-lite)	83.3	19.7	12.4	78.2	22.6	6.5
ESM	**85.1**	**18.6**	**3.9**	**82.6**	**17.8**	**3.7**

E Diagnostic Ablations

Table 13 checks three implementation choices that are easy to overlook. The dense-metric row is run only on a reduced-width model because forming the full pullback is too expensive at the main width; it tests whether the rank-16 sketch is close to the dense local metric on this benchmark. The cheap-gate row removes gradient-norm diagnostics from the gate and uses residual/loss features only; it lowers latency but misses some hard states. The rollback row discards a freshly measured sketch after rejection; the retained-sketch policy used in the main paper is slightly better, but the small gap means the method is not relying on a fragile rejected-state feedback loop.

Table 13: Diagnostic ablations on navigation. Dense-metric results use a reduced $d_z = 64$ model; the other rows use the main $d_z = 256$ model.

Variant	Success ↑	Coll. / 10^3 ↓	Mean ms ↓	Note
Dense full pullback, reduced model	85.4	18.2	5.8	quality ceiling for sketch
Rank-16 sketch, reduced model	85.0	18.8	3.7	sketch used by ESM
Gradient-diagnostic gate	85.1	18.6	3.9	main setting
Cheap residual-only gate	84.3	20.2	3.1	lower easy-path cost
Retain refreshed sketch after reject	85.1	18.6	3.9	main setting
Rollback sketch after reject	84.9	19.1	3.8	more conservative sketch state

F Additional Sweeps

Table 14: Additional budget and trust-radius sweeps.

Correction-budget sweep on navigation.

$K_{\max}$	Succ.	Coll.	Mean ms	Avg. K_t
0	68.4	37.1	2.1	0.00
1	81.7	23.4	3.1	0.33
2	84.2	19.7	3.6	0.44
3	85.1	18.6	3.9	0.51
4	85.4	18.2	4.3	0.58

Cumulative trust-radius sweep on torsion-chain repair.

ρ_{tot}	Succ.	Shape	Viol.	Reject
0.05	76.9	0.65	14.1	0.05
0.10	80.8	0.56	15.6	0.09
0.18	82.6	0.52	17.8	0.13
0.30	83.1	0.50	25.7	0.24
0.50	79.5	0.54	41.2	0.38

The budget sweep shows the expected knee: one or two microsteps repair many local errors, while increasing beyond $K_{\max} = 3$ gives small quality gains for additional latency. The trust-radius sweep shows the complementary effect of the stability envelope. Very small radii under-correct, moderate radii improve shape and success, and large radii increase violations and rejection because the local model is no longer reliable.