The Dynamic-Probabilistic Consistency Gap in Chaotic Surrogate Modeling

Andre Herz[1,2] **Matthijs Pals**[3] **Daniel Durstewitz**[1,3,4] **Georgia Koppe**[1,2,5,6]

[1]Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany
[2]Faculty of Mathematics and Computer Science, Heidelberg University, Germany
[3]Dept. of Theoretical Neuroscience, Central Institute of Mental Health (CIMH), Mannheim, Germany
[4]Faculty of Physics and Astronomy, Heidelberg University, Germany
[5]Hector Institute for AI in Psychiatry and Dept. of Psychiatry and Psychotherapy, CIMH, Mannheim, Germany
[6]Hertie Institute for AI in Brain Health, University of Tübingen, Germany
Correspondence to: andre.herz@iwr.uni-heidelberg.de

Abstract

Dynamical systems reconstruction (DSR) aims to learn surrogate models that capture the dynamics underlying time-series data. Reliable scientific inference with these surrogates requires uncertainty estimates consistent with the learned dynamics. We expose a *dynamic-probabilistic consistency* (DPC) gap: optimizing finite-horizon probabilistic objectives can degrade the learned dynamics or decouple predictive uncertainty from the local tangent dynamics it should reflect. We isolate three mechanisms behind this gap: core collapse, noise masking, and blind uncertainty. Specifically, we show that open-loop rollout objectives can penalize covariance growth driven by the dynamics, encouraging optimization shortcuts that weaken local expansion. To mitigate this gap, we propose KAFFEE (Kalman-Aware Framework For Ergodic Emulation), a differentiable extended Kalman filter-based training framework that jointly learns dynamics and noise covariances by coupling filtered innovation likelihoods with covariance propagation through learned local Jacobians, while differentiating through the Kalman updates. On stochastic hyperchaotic Lorenz-96, KAFFEE mitigates the identified failure modes and improves both reconstruction of dynamical invariants and predictive scores relative to open-loop objectives. The DPC gap also appears when probabilistically adapting a DSR foundation model across 13 chaotic systems, where KAFFEE improves predictive likelihood over open-loop adaptation while preserving the pretrained model's long-term dynamics and enabling in-context Bayesian filtering.

1 Introduction

Dynamical systems reconstruction (DSR) aims to learn generative neural surrogates of latent dynamics from observed time series [1–4]. Short-horizon predictive accuracy is not sufficient here [5]; rather, the surrogate is asked to act as an autonomous scientific model. It should therefore generate open-loop rollouts that reproduce the long-run dynamical structure of the true system, including attractor geometry, invariant measures, and chaotic expansion rates as quantified by its Lyapunov exponents [6, 7]. These properties are central whenever the surrogate is used beyond forecasting, e.g., invariant-measure estimation [8, 9], bifurcation analysis [10], or perturbation-response studies [11, 12].

Scientific applications also require probabilistic state estimation and *uncertainty-aware* prediction from noisy observations [13, 14]. In settings such as data assimilation [15], extreme-event forecasting [16], or control under uncertainty [17], the model must not only forecast accurately on average, but

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

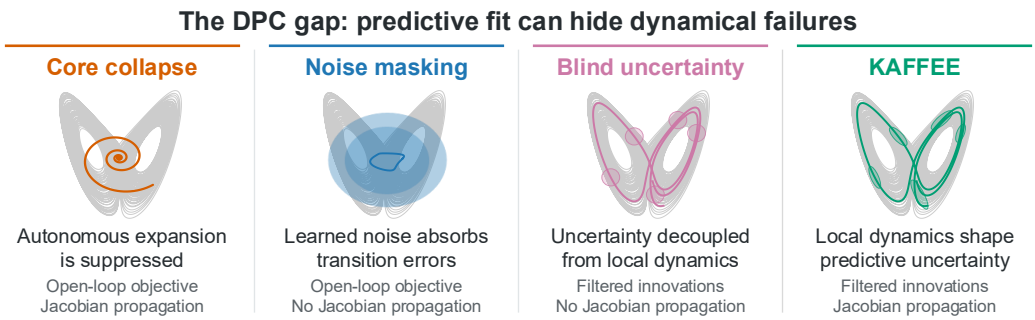


Figure 1: **The DPC gap in probabilistic DSR.** Schematic comparison of three DPC failure modes and KAFFEE. Finite-horizon probabilistic scores can improve while autonomous dynamics deteriorate or uncertainty remains blind to local dynamics. KAFFEE pairs filtered innovation scoring with Jacobian-based covariance propagation to mitigate the DPC gap. Grey curves illustrate ground-truth dynamics, colored curves learned autonomous dynamics, and shading learned predictive uncertainty.

also represent how uncertainty evolves under the dynamics, including the amplification of small state errors [18]. This motivates *probabilistic DSR*: a surrogate should couple autonomous physical fidelity with *dynamically grounded local uncertainty*, tied to the same structure that governs dynamical expansion and contraction.

Achieving this goal creates a structural optimization tension. Probabilistic objectives balance data fit against predictive uncertainty [19, 20]. In dynamical systems, uncertainty is propagated by the same local Jacobians that govern long-run dynamics [21], thereby coupling unstable directions with rapid covariance growth [18, 22]. This can create several optimization shortcuts: open-loop rollout objectives may contract learned Jacobian products to suppress covariance growth (*core collapse*); flexible noise parametrizations may absorb model error without correcting the learned autonomous dynamics (*noise masking*); and covariance parametrizations decoupled from local Jacobians may remain calibrated on average while being blind to locally unstable episodes (*blind uncertainty*).

We identify and examine these failure modes as instances of the *dynamic-probabilistic consistency* (DPC) gap, a mismatch between improving finite-horizon probabilistic scores and preserving the dynamical-probabilistic properties required of a scientific surrogate. The gap appears when the training objective can improve predictive scores by changing the model in ways that damage autonomous dynamics or the dynamical consistency of its local uncertainty. A model may thus achieve better probabilistic fit while misattributing observed variability to noise rather than deterministic instability inherent to the generating dynamics. For emerging dynamical-system foundation models (DSFMs) [23–25], this raises the practical challenge of adding probabilistic state estimation and predictive uncertainty tied to local dynamics, while preserving their zero-shot autonomous behavior.

Our analysis suggests two design principles for probabilistic DSR. First, evaluate likelihood through filtered innovations, scoring each observation before assimilating it to localize uncertainty for the next prediction. Second, local predictive uncertainty should remain dynamically grounded, so that covariance reflects the local Jacobian structure that governs the autonomous dynamics. We introduce KAFFEE (Kalman-Aware Framework For Ergodic Emulation) as a realization of these principles. KAFFEE uses a differentiable Extended Kalman Filter (EKF) and optimizes the negative log-likelihood (NLL) of the EKF innovation sequence [26–28].

Contributions. First, we *identify and formalize the DPC gap* and characterize three failure modes. We isolate open-loop covariance-volume pressure as a concrete mechanism for one failure mode, *core collapse*, in finite-horizon rollout objectives, where likelihood can penalize the Jacobian products that determine chaotic expansion. Second, we introduce a *three-axis evaluation framework* for probabilistic DSR that separates local probabilistic behavior, stochastic dynamical fidelity, and deterministic dynamical fidelity. On stochastic Lorenz-96 [29], this isolates *core collapse*, *noise masking*, and *blind uncertainty*. Third, we propose *KAFFEE*, a differentiable EKF-based framework that mitigates the DPC gap. Fourth, we show that the DPC gap arises when probabilistically adapting the pretrained DSFM DynaMix [23] across 13 held-out chaotic systems, and that KAFFEE enables in-context Bayesian filtering while largely preserving transferred autonomous dynamics.

2 Preliminaries

2.1 Probabilistic DSR: the core and the shell

We study the reconstruction of a chaotic dynamical system from time series using generative neural surrogates. Let $\boldsymbol{z}_{t+1} = F_\star(\boldsymbol{z}_t)$ denote an unknown, discrete-time autonomous system with latent states $\boldsymbol{z}_t \in \mathcal{Z} \subset \mathbb{R}^M$ and observations $\boldsymbol{x}_t \in \mathbb{R}^N$. We approximate the unknown dynamics $F_\star$ by a neural surrogate F_{θ_c} and embed it in a probabilistic state-space model (SSM). Specifically, we consider a Gaussian SSM with nonlinear transitions and linear observations:

$$p_\theta(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t) = \mathcal{N}\big(\boldsymbol{z}_{t+1}; F_{\theta_c}(\boldsymbol{z}_t), \boldsymbol{\Gamma}_{\theta_s}\big), \quad p_{\theta_s}(\boldsymbol{x}_t \mid \boldsymbol{z}_t) = \mathcal{N}\big(\boldsymbol{x}_t; \boldsymbol{B}\boldsymbol{z}_t, \boldsymbol{\Lambda}_{\theta_s}\big),$$

with $\boldsymbol{\Gamma}_{\theta_s} \in \mathbb{R}^{M\times M}$ and $\boldsymbol{\Lambda}_{\theta_s} \in \mathbb{R}^{N\times N}$ denoting the process and observation covariances, and $\boldsymbol{B} \in \mathbb{R}^{N\times M}$ denoting the observation mapping (held fixed in our experiments). The model parameters are partitioned into $\theta = (\theta_c, \theta_s)$. The *deterministic core* θ_c controls the autonomous transition mapping. It determines the local Jacobians $\boldsymbol{J}_{\theta_c}(\boldsymbol{z}) := \nabla_{\boldsymbol{z}} F_{\theta_c}(\boldsymbol{z})$, whose chain products govern local stretching and contraction, yielding the system's dynamical invariants and shaping the attractor geometry [21]. The *stochastic shell* θ_s parametrizes the uncertainty around this deterministic core, here through $\boldsymbol{\Gamma}_{\theta_s}$ and $\boldsymbol{\Lambda}_{\theta_s}$.

2.2 Marginal likelihood, filtering, and covariance propagation

Under the standard conditional independence and Markov assumptions of an SSM, the observed-data marginal likelihood (evidence) of a sequence $\boldsymbol{x}_{1:T}$ is given by

$$p_\theta(\boldsymbol{x}_{1:T}) = \int \left(\prod_{t=1}^{T} p_{\theta_s}(\boldsymbol{x}_t \mid \boldsymbol{z}_t)\right) p_\theta(\boldsymbol{z}_{1:T}) d\boldsymbol{z}_{1:T}, \tag{1}$$

where $p_\theta(\boldsymbol{z}_{1:T}) = p_\theta(\boldsymbol{z}_1)\prod_{t=1}^{T-1} p_\theta(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t)$ is the surrogate's *open-loop latent trajectory prior* distribution, generated by rolling out the autonomous core and stochastic shell without conditioning on intermediate observations. In the following, we will see how this prior interacts with the likelihood.

Evaluating Eq. 1 requires marginalizing the latent trajectory, typically via recursive filtering. This involves the predictive densities $p_\theta(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}) = \int p_\theta(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}) p_\theta(\boldsymbol{z}_{t-1} \mid \boldsymbol{x}_{1:t-1}) d\boldsymbol{z}_{t-1}$, which are intractable for nonlinear F_{θ_c}. A common approach is approximating the predictive distributions as Gaussian, $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{z}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$, as in the EKF, where the predict step analytically propagates the filtered covariance through the local Jacobians [28]:

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{J}_{\theta_c,t}\boldsymbol{\Sigma}_{t-1|t-1}\boldsymbol{J}_{\theta_c,t}^\top + \boldsymbol{\Gamma}_{\theta_s}, \qquad \boldsymbol{J}_{\theta_c,t} = \nabla_{\boldsymbol{z}} F_{\theta_c}(\boldsymbol{z}_{t-1|t-1}). \tag{2}$$

The likelihood is then given recursively via $p_\theta(\boldsymbol{x}_{1:T}) = p_\theta(\boldsymbol{x}_1)\prod_{t=2}^{T} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1})$. This evaluates the *innovations*: $\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1}$, which have covariance given by:

$$\boldsymbol{\Omega}_t = \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}. \tag{3}$$

Thus, the same local Jacobians that define the autonomous dynamics also determine how predictive uncertainty is propagated. This coupling implies that uncertainty cannot be treated as an independent calibration layer in probabilistic DSR; rather, predictive calibration must be evaluated together with the dynamics that propagate uncertainty.

3 Theory and Methods

3.1 Defining the DPC gap

Let $\mathcal{S}_H(\theta)$ be a finite-horizon probabilistic training objective (e.g., NLL) and let $\mathcal{D}(\theta)$ be a certification metric for the quality of the reconstructed dynamics; both defined such that lower values are better.

Definition 1 (Dynamic-Probabilistic Consistency gap)**.** A parameter update step $\theta \to \theta'$ exhibits a DPC gap with respect to $(\mathcal{S}_H, \mathcal{D})$ if

$$\mathcal{S}_H(\theta') < \mathcal{S}_H(\theta) \quad \text{but} \quad \mathcal{D}(\theta') > \mathcal{D}(\theta).$$

That is, the optimized probabilistic criterion can favor an update that improves finite-horizon predictive fit while worsening the certification loss, which encodes properties required for probabilistic DSR. In our experiments, we instantiate $\mathcal{D}$ along three axes: Local probabilistic behavior, shell-on stochastic dynamical fidelity, and shell-off deterministic dynamical fidelity (see Section 4.2).

3.2 How the DPC gap manifests

How the DPC gap manifests depends on two design choices: on how the training objective scores uncertainty, and on how the model parameterizes that uncertainty. These choices give rise to three failure modes we discuss here and isolate empirically in Section 5.

Core collapse from open-loop volume pressure. Learned surrogates are ultimately used as autonomous simulators. Training must address the exposure bias that can arise when one-step or teacher-forced objectives condition on noisy data, whereas autonomous prediction requires purely model-generated states [30–33]. Finite-window rollout losses are therefore common in system identification and DSR [34–37]. In probabilistic analogues of this setting, scoring multi-step predictive densities makes uncertainty propagation part of the objective [38, 39].

Suppose an h-step predictive density is approximated by a Gaussian, $p_\theta(\boldsymbol{x}_{t+h} \mid \boldsymbol{x}_{1:t}) = \mathcal{N}(\boldsymbol{x}_{t+h}; \boldsymbol{\mu}_{\theta,t+h}, \boldsymbol{\Omega}_{\theta,t+h})$. Its NLL then reads

$$-\log p_\theta(\boldsymbol{x}_{t+h} \mid \boldsymbol{x}_{1:t}) = \frac{1}{2}\log\det\boldsymbol{\Omega}_{\theta,t+h} + \frac{1}{2}\|\boldsymbol{x}_{t+h} - \boldsymbol{\mu}_{\theta,t+h}\|^2_{\boldsymbol{\Omega}^{-1}_{\theta,t+h}} + \text{const.}$$

The Mahalanobis-distance term rewards data fit, while the log-determinant term penalizes predictive volume. The latter term in the Bayesian marginal likelihood is referred to as an "automatic Occam's razor" [19, 20], favouring models that concentrate probability mass while penalizing unnecessarily large predictive volumes. In standard probabilistic forecasting this prevents the model from explaining errors by becoming arbitrarily diffuse.

In chaotic DSR, however, physically correct local expansion also increases predictive volume. This mechanism becomes apparent by considering the h-step Jacobian product $\boldsymbol{\Phi}_{\theta,t+h:t} := \boldsymbol{J}_{\theta_c,t+h}\cdots\boldsymbol{J}_{\theta_c,t+1}$. Crucially, the same products govern perturbation growth and their asymptotic singular-value growth rates determine the Lyapunov spectrum $\lambda_1 \geq \cdots \geq \lambda_M$, with leading Lyapunov exponent defined by $\lambda_1 = \lim_{h\to\infty}\frac{1}{h}\log\|\boldsymbol{\Phi}_{\theta,t+h:t}\|_2$ [21].

Under local linearization, the open-loop covariance $\boldsymbol{\Sigma}^{\text{open}}_{t+h} \in \mathbb{R}^{M\times M}$ of the surrogate's latent predictive density $p(\boldsymbol{z}_{t+h} \mid \boldsymbol{x}_{1:t})$ is given by repeatedly applying Eq. 2 without intermediate assimilation:

$$\boldsymbol{\Sigma}^{\text{open}}_{t+h} \approx \boldsymbol{\Phi}_{\theta,t+h:t}\boldsymbol{\Sigma}_t\boldsymbol{\Phi}^\top_{\theta,t+h:t} + \sum_{j=0}^{h-1}\boldsymbol{\Phi}_{\theta,t+h:t+j+1}\boldsymbol{\Gamma}_{\theta_s}\boldsymbol{\Phi}^\top_{\theta,t+h:t+j+1}.$$

The corresponding observation covariance is $\boldsymbol{\Omega}^{\text{open}}_{\theta,t+h} = \boldsymbol{B}\boldsymbol{\Sigma}^{\text{open}}_{t+h}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}$ (cf. Eq. 3). Thus, the log-determinant term directly penalizes volumes generated by products of the learned Jacobians. Chaotic systems inherently expand along at least one unstable direction ($\lambda_1 > 0$), causing these volumes to grow rapidly under autonomous propagation [18, 22]. The optimizer can alleviate this volume pressure by shrinking these products, thereby compromising the autonomous dynamics. Appx. A.1 formalizes this mechanism by showing that the open-loop log-determinant term increases monotonically along local Jacobian-expansion paths. This isolates the volume term; the Mahalanobis (data-fit) term in the NLL can oppose it. A similar volume term also appears for other exponential family objectives (given a Laplace approximation), see Appx. A.2.

Fig. 2 makes this tradeoff explicit on the tent map $f_s(z) = s\min\{z, 1-z\}$ with additive Gaussian process and observation noise. The open loop objective overestimates the spread of the true distribution. As a result the predictive volume term shifts the open-loop optimum toward weaker expansion. In contrast, filtering localizes uncertainty before each prediction, yielding calibrated predictive distributions and an objective that recovers the true slope. Further details in Appx. A.1.

Note that the argument here does not imply that open-loop likelihood is intrinsically biased. Under a correctly specified model and with the exact observed-data likelihood, maximum likelihood targets the true data-generating process under standard identifiability assumptions. DPC-type mechanisms arise from practical circumstances in probabilistic DSR: under open-loop propagation, predictive distributions can become strongly non-Gaussian and poorly represented by a locally linearized Gaussian approximation.

Noise masking through flexible stochastic shells. A further shortcut is possible when predictive uncertainty is parametrized too flexibly or independently of Jacobian propagation. In this case, the optimizer may inflate $\boldsymbol{\Gamma}_{\theta_s}$, $\boldsymbol{\Lambda}_{\theta_s}$ (or a learned neural variance parametrization) to absorb transition

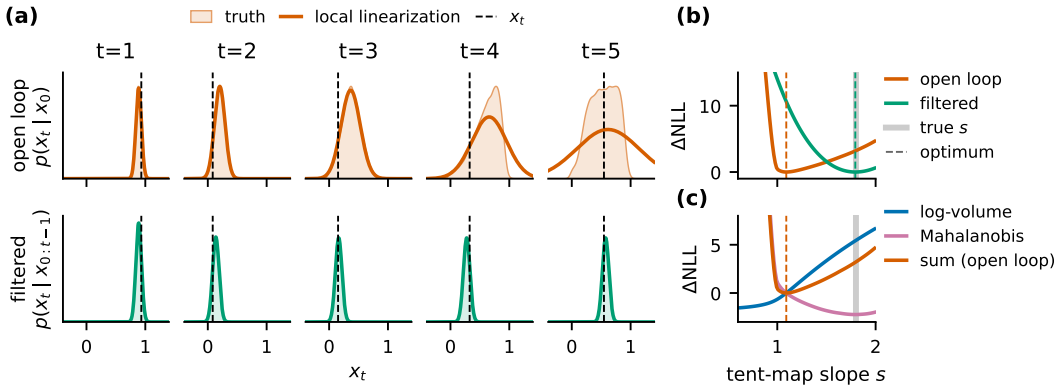


Figure 2: **Covariance-volume pressure in a noisy tent map. (a)** Linearized open-loop predictives $p(x_t \mid x_0)$ (top) overestimate the spread of the true density; filtered predictives $p(x_t \mid x_{0:t-1})$ (bottom) stay calibrated. **(b)** ΔNLL profiles vs. tent-map slope s: the open-loop optimum attenuates the slope, while the filtered optimum recovers it. **(c)** Open-loop log-volume and Mahalanobis terms, shifted to zero at the open-loop optimum. The growing log-volume term pulls the optimum down.

errors without improving the deterministic core [31]. This can improve predictive scores despite dynamically incorrect autonomous rollouts. Related effects can also arise in variational state-space models when overly flexible components reduce the penalty on transition-model mismatch, or when independence assumptions are too restrictive [40, 41]. Note that noise masking and core collapse can occur together, since a noise-inflated shell may conceal deterioration of the autonomous core.

Blind uncertainty from missing covariance propagation. A third failure mode concerns the local uncertainty itself. A model can preserve reasonable autonomous dynamics and obtain competitive average predictive scores, while its predictive covariance remains insensitive to state-dependent expansion and contraction. This happens when the covariance propagation (Eq. 2) is state-invariant. It may also happen when uncertainty is parametrized separately from the learned Jacobians. Such a model may be calibrated on average, but its local uncertainty is not dynamically grounded. We treat this as a DPC-gap instance where $\mathcal{D}$ assesses whether predictive uncertainty evolves in accordance with the true local tangent dynamics (measured in our experiments by $G_{\tan}$, see Section 4.2). Again, a related problem can arise in variational inference, with certain factorized variational families. If uncertainty is not propagated in the chosen factorization (such as in mean-field approximations, previously used for DSR [42]), learned dynamics can be biased [43, 44].

Relation to noisy-state attenuation. The open-loop volume mechanism is related to, but distinct from, classical noisy-state attenuation. In noisy-state attenuation, noisy observations are treated directly as true states, turning one-step Gaussian prediction into an errors-in-variables regression and biasing the learned transition coefficient toward zero [45–47]. This explains some failures of one-step training on noisy data, as illustrated by the scalar diagnostic in Appx. A.3 and the non-chaotic controls in Appx. D.4.

3.3 KAFFEE: Dynamically grounded local uncertainty

The preceding mechanisms suggest two desiderata for probabilistic DSR: likelihood should be evaluated through filtered innovations, and the covariance entering this local objective should remain coupled to the learned transition map. KAFFEE implements both using a differentiable EKF [26–28].

KAFFEE maintains a filtered Gaussian belief $p_\theta(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) \approx \mathcal{N}(\boldsymbol{z}_{t|t}, \boldsymbol{\Sigma}_{t|t})$. At each step, the deterministic core predicts the next-step mean $\boldsymbol{z}_{t|t-1} = F_{\theta_c}(\boldsymbol{z}_{t-1|t-1})$, and the covariance is propagated by the same local Jacobians that define the autonomous core (Eq. 2). After prediction, observations $\boldsymbol{x}_t$ are assimilated through the Kalman gain $\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top \boldsymbol{\Omega}_t^{-1}$, with the innovation covariance $\boldsymbol{\Omega}_t$ given by Eq. 3. The covariance update can be written as $\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}$. Thus, assimilation contracts uncertainty before the next prediction step (we have $\boldsymbol{\Sigma}_{t|t} \preceq \boldsymbol{\Sigma}_{t|t-1}$; see Appx. A.1).

This localizes the uncertainty scored by the likelihood before the next prediction. Appx. B.1 lists the complete EKF equations.

KAFFEE minimizes the resulting innovation NLL with stochastic gradient descent:

$$\mathcal{L}_{\text{EKF}} = -\sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{B}\boldsymbol{z}_{t|t-1}, \boldsymbol{\Omega}_t) = \sum_{t=1}^{T} \frac{1}{2} \log\det \boldsymbol{\Omega}_t + \frac{1}{2} \|\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1}\|^2_{\boldsymbol{\Omega}_t^{-1}} + \text{const.} \quad (4)$$

In the linear-Gaussian case, the Kalman filter gives the exact NLL; in nonlinear surrogates, the EKF provides a structured inductive bias for dynamically grounded local uncertainty. Appx. A.4 shows that the Mahalanobis term can outweigh the log-determinant once model covariance is underdispersed relative to the true residual spread.

While the underlying equations are those of a differentiable EKF, KAFFEE defines a dedicated training framework designed to translate deterministic DSR models into probabilistic surrogates with calibrated local uncertainty while reducing DPC-gap shortcuts. KAFFEE initializes the autonomous core F_{θ_c} from a pretrained deterministic surrogate (or zero-shot foundation model) and optimizes the full probabilistic model via windowed backpropagation through time (BPTT). This lets the model first capture the deterministic dynamical invariants, and then leverage the EKF's covariance propagation as a structural inductive bias for grounding local uncertainty. Furthermore, differentiating through the Kalman update makes KAFFEE a Bayesian-filtering analogue of forcing-based stabilization methods for chaotic recurrent training [34, 48–50], similar to Pals et al. [51]; see Section 6.

4 Experimental Setup

We examine the mechanisms behind the DPC gap and evaluate KAFFEE in two complementary settings: an interpretable mechanistic ablation with piecewise-affine recurrent neural networks (RNNs) [52, 53], and a scale-up using the pretrained DSFM DynaMix on a range of chaotic systems [23].

4.1 Deterministic priors and baselines

All probabilistic baselines are initialized from the same deterministic core: an almost-linear RNN (AL-RNN) [53] trained via BPTT with sparse teacher forcing (STF) [48]. AL-RNNs partition the state space into polyhedral regions with explicit local linear dynamics ($\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{W}\,\boldsymbol{\phi}^*(\boldsymbol{z}_t) + \boldsymbol{h}$), where $\boldsymbol{\phi}^*$ denotes ReLU activation, applied only to a subset of latent coordinates. This provides a powerful tangent space approximation of the underlying system with exact, closed-form regionwise Jacobians that KAFFEE exploits directly for covariance propagation. Their piecewise-affine structure also simplifies backpropagation (Appx. A.7). Details on AL-RNN pretraining are in Appx. C.1.

To isolate the mechanisms behind the DPC gap, all probabilistic models share the same deterministic core and adaptation budget. As illustrated in Fig. 1, we structure our baselines as a mechanistic taxonomy along two axes: *Objective* (Open-Loop vs. Filtered Innovation) and *Parameterization* (Jacobian-tied covariance propagation vs. no propagation). These controls separate the effects of objective and covariance propagation: open-loop training can cause both *core collapse* and *noise masking*, while no-propagation covariance yields *blind uncertainty*. For the "tied-propagation" baselines, the predictive covariance is propagated through the learned local Jacobian (Eq. 2), whereas in "no-propagation" this is replaced by a state-invariant recursion, $\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Gamma}$. The open-loop objectives score Gaussian rollouts while filtered objectives optimize innovation NLL (Eq. 4). We also include a *Frozen-Core* baseline to isolate the effect of post-hoc calibration. It uses the filtered covariance but keeps F_{θ_c} fixed.

To demonstrate the pathology at scale, we probabilistically adapt DynaMix, a pretrained DSFM for zero-shot inference [23]. Here, the modular nature of KAFFEE allows us to embed the pretrained DSFM into an SSM framework. In particular, we attach a Gaussian shell to DynaMix and evaluate whether probabilistic fine-tuning allows for calibrated uncertainty while preserving its zero-shot transferred physics. Additional model and implementation details are provided in Appx. C.5.

4.2 Three-Axis Evaluation Framework

To empirically expose the DPC gap, we evaluate surrogates along three axes (details in Appx. C.4):

(1) Local probabilistic behavior assesses local predictive performance at a physically meaningful horizon ($H = \frac{1}{4}\tau_\lambda$, where $\tau_\lambda = \lambda_1^{-1}$ is the Lyapunov time). We quantify this using ensembles over evenly spaced forecast starts to compute held-out NLL and continuous ranked probability score (CRPS). Because these metrics are time-averaged, they can be dominated by easily predictable regions of the attractor and can obscure overconfidence during highly unstable episodes. To assess whether the covariance propagation of the model agrees with the local expansion of the dynamics, we introduce $G_{\tan}$, a Spearman correlation between process-noise-free log variance growth under the surrogate's covariance propagation and the corresponding growth under the true local Jacobians, pooled across observed coordinates.

(2) Stochastic dynamical fidelity evaluates long shell-on rollouts of the surrogate against the noisy ground truth via state-space divergence ($D_{\text{stsp}}^{\text{stoch}}$), which measures how well the state-space occupancy of generated trajectories agrees with that of the reference dynamics, and unstable-volume growth rate error ($|\Delta h_\lambda^{\text{stoch}}|$), quantified by the absolute reconstruction error in the sum of positive Lyapunov exponents. The sum of these exponents is a fundamental dynamical invariant related to the information production rate of a system [21].

(3) Deterministic dynamical fidelity isolates the deterministic core (shell removed) and evaluates it against the noise-free true system using state-space divergence ($D_{\text{stsp}}^{\text{core}}$), spectral overlap using Hellinger distance ($D_{\text{H}}^{\text{core}}$), and deterministic unstable-volume growth rate error ($|\Delta h_\lambda^{\text{core}}|$).

5 Results

5.1 Diagnosing the DPC Gap: Mechanistic ablation on stochastic Lorenz-96

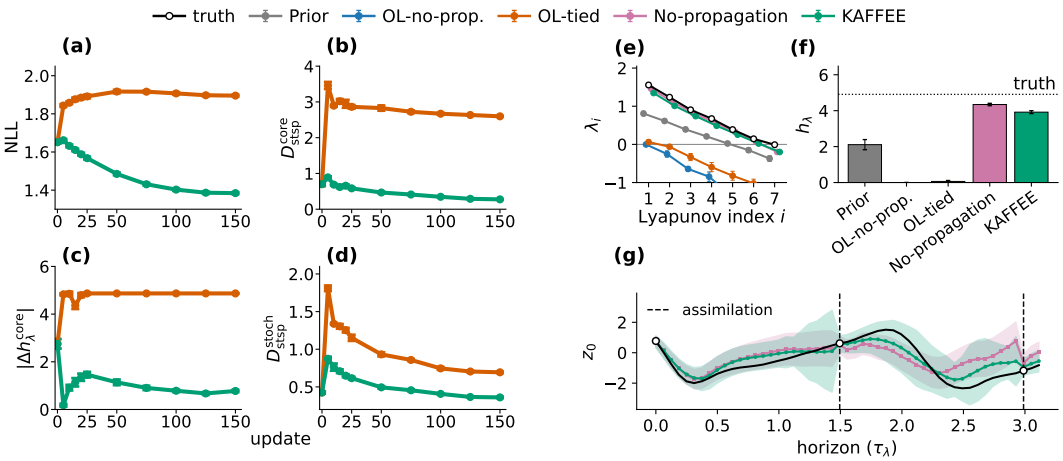


Figure 3: **Stochastic Lorenz-96 (20D) results. Left: DPC gap.** **(a)** Held-out NLL, **(b)** deterministic state-space divergence (D_{stsp}), **(c)** unstable-volume growth error, and **(d)** stochastic D_{stsp}. **Right: Invariant reconstruction and dynamically grounded local uncertainty.** **(e)** Median reconstruction of positive deterministic Lyapunov exponents λ_i. Only filtered innovations outperform the deterministic prior. **(f)** Corresponding unstable-volume growth rate h_λ^{core}. **(g)** Single Lorenz-96 latent; shaded bands = predictive $\pm 2\sigma$; assimilation after $1.5\,\tau_\lambda$. KAFFEE grounds local uncertainty through state-dependent Jacobians; no-propagation exhibits blind uncertainty. Errors = SEs over 20 seeds.

We examine the DPC gap on the Lorenz-96 system [29] (dimensionality $N = 20$, forcing $F = 8$) under substantial noise ($\sigma_{\text{proc}} = \sigma_{\text{obs}} = 0.1$ in standardized coordinates), see Appx. C.3. Lorenz-96 is hyperchaotic, i.e., it exhibits multiple expanding directions.

Fig. 3 shows the DPC failure modes in action: under matched covariance propagation and adaptation budgets, minimizing the open-loop NLL objective degrades the autonomous core, while filtered innovation objectives improve it (Panels a–d). Note that Panel (a) shows held-out ensemble NLL, not the optimized training loss. Panels (b) and (c) show autonomous core evolution during training in terms of state-space divergence $D_{\text{stsp}}^{\text{core}}$ and unstable-volume growth rate error ($|\Delta h_\lambda^{\text{core}}|$). Open-loop

Table 1: **Mechanistic ablation on stochastic Lorenz-96.** All probabilistic baselines share the *same* end-to-end training schedule and are initialized from the same deterministic AL-RNN cores. **Bold:** median whose ±SE interval overlaps the column's best; red: signature failure mode (bracketed label). Medians ± SEs across 20 seeds; see Appx. C.4.

	Local Probabilistic Behavior			Stochastic Dynamics (Shell+Core)		Deterministic Dynamics (Core)		
Model / Objective	NLL ($\frac{1}{4}\tau_\lambda$) ↓	CRPS ($\frac{1}{4}\tau_\lambda$) ↓	$G_{\tan}$ ↑	$D_{\text{stsp}}^{\text{stoch}}$ ↓	$\lvert\Delta h_\lambda^{\text{stoch}}\rvert$ ↓	$D_{\text{stsp}}^{\text{core}}$ ↓	$\lvert\Delta h_\lambda^{\text{core}}\rvert$ ↓	$D_{\text{H}}^{\text{core}}$ ↓
Deterministic prior								
AL-RNN	–	–	–	–	–	0.708±0.057	2.766±0.250	0.338±0.020
Frozen-Core Shell Calibration	1.428±0.016	**0.545±0.004**	0.283±0.011	**0.355±0.005**	1.982±0.058	0.708±0.057	2.766±0.250	0.338±0.020
Open-Loop objective								
No propagation [Noise Masking]	2.154±0.013	1.527±0.017	0.000±0.000	1.147±0.010	5.120±0.000	2.647±0.012	4.868±0.014	0.834±0.007
Tied propagation [Core Collapse]	1.898±0.008	0.956±0.019	0.322±0.012	0.685±0.013	5.120±0.000	2.645±0.016	4.868±0.000	0.845±0.003
Filtered objective								
No propagation [Blind Uncertainty]	1.437±0.005	0.569±0.003	0.000±0.000	0.492±0.009	**0.421±0.046**	0.310±0.009	**0.395±0.075**	**0.128±0.006**
Tied propagation (KAFFEE)	**1.388±0.006**	**0.551±0.003**	**0.528±0.003**	0.369±0.005	0.710±0.046	**0.291±0.008**	0.811±0.074	**0.130±0.003**

tied rapidly suppresses the core's expansion, while $D_{\text{stsp}}^{\text{stoch}}$ continues to improve late in adaptation (Panel d). Thus, shell-on rollouts can look more faithful in terms of probabilistic dynamical fidelity, even when the autonomous core has deteriorated. The remaining Panels e–g show the reconstruction of the dynamical invariant h_λ and the local dynamical grounding of uncertainty. Both filtered objectives (KAFFEE, no-propagation) retain the system's intrinsic hyperchaotic expansion, which open-loop objectives collapse and the deterministic prior struggles to capture (Panels e–f). KAFFEE combines improved autonomous reconstruction with dynamically grounded local uncertainty. In contrast, the no-propagation baseline re-adds fixed process noise $\mathbf{\Gamma}$ at every step, rendering its predictive variance state-invariant and overconfident during unstable episodes (Panel g).

Table 1 maps our empirical results to the mechanistic taxonomy of Fig. 1 and the three-axis evaluation framework (Section 4.2). The failure modes manifest as predicted. *Core collapse* (open-loop + tied propagation) destroys long-run fidelity. *Noise masking* (open-loop + no propagation) inflates the learned observation shell (median $\text{tr}(\mathbf{\Lambda})/N$ is 0.29 vs. 0.10 for KAFFEE), while the core also deteriorates. $\lvert\Delta h_\lambda^{\text{core}}\rvert$ saturates for both open-loop objectives, since those cores are not producing expansive dynamics in any direction. *Blind uncertainty* (filtered + no propagation) retains much of the improvement in autonomous reconstruction, yet its state-invariant covariance recursion cannot reflect local physical expansion: its $G_{\tan}$ is zero by construction. KAFFEE is the only method that pairs a large positive $G_{\tan}$ with the best held-out NLL as well as substantially improved deterministic dynamics over the prior. Frozen-core shell calibration shows that calibrating a shell around a fixed prior can produce positive $G_{\tan}$, but it cannot improve the deterministic dynamics of the prior.

These comparisons also show why no single metric is sufficient alone. The advantage of filtered over open-loop objectives in NLL and CRPS persists across the tested forecast horizons up to one Lyapunov time (Appx. D.2). KAFFEE has the lowest NLL, frozen-core shell calibration has a slightly lower CRPS, and filtered no-propagation has the smallest $\lvert\Delta h_\lambda^{\text{core}}\rvert$ while $G_{\tan}$ stays at zero. Conversely, a positive $G_{\tan}$ can coexist with core collapse, as visible for open-loop tied. Predictive scores, tangent-propagation alignment and autonomous/stochastic dynamical fidelity therefore answer different questions and require separate evaluation.

Appx. D.1 shows that these DPC gap signatures are not Lorenz-96-specific. On stochastic (3D) Lorenz-63, open-loop objectives still collapse the autonomous dynamics, and no-propagation filtering remains locally blind. KAFFEE avoids that collapse while keeping local uncertainty dynamically grounded. However, in the tested setting, the deterministic prior was already faithful, illustrating a limit of joint adaptation: frozen-core calibration preserves the autonomous reconstruction better and attains better local predictive scores. Hence joint adaptation is not uniformly preferable, especially when the prior is already close to the true deterministic dynamics.

Appx. D.3 further tests the role of the Kalman update during training. Stopping gradients through its mean and covariance updates worsens deterministic reconstruction despite identical forward updates at fixed parameters. This supports an important contribution induced by backpropagation through assimilation, consistent with the stabilization mechanism discussed in Section 6 and Appx. A.5.

5.2 In-context probabilistic adaptation of DynaMix

We next test whether the DPC gap appears when probabilistically adapting a pretrained DSFM. We use DynaMix [23], which can take in observations of a dynamical system in its context window, and

then simulate the system's long-term evolution. Here, we adapt DynaMix to 13 held-out dynamical systems, contaminated by 10% observation noise. All target-system learning is restricted to the 2,000 observations in DynaMix's context window.

We compare three probabilistic adaptation modes: *KAFFEE-DynaMix* jointly updates the DynaMix core and a Gaussian noise shell using the EKF innovation objective. The *open-loop* control uses the same transferred core and shell initialization, but updates the core using a Gaussian open-loop NLL objective. Finally, the *frozen-core* control optimizes only the diagonal shell under the EKF objective while freezing the DynaMix core; its autonomous dynamics are therefore identical to the pretrained core by construction. Details are reported in Appx. Table 3.

Fig. 4a shows the resulting trade-off after 50 in-context optimization steps. Open-loop tied adaptation deteriorates the autonomous core ($D_{\text{stsp}}^{\text{core}}/D_{\text{stsp}}^{\text{base}} = 1.30 \pm 0.11$); KAFFEE stays near the zero-shot attractor geometries (1.01 ± 0.01). Over the 13 systems, KAFFEE has lower $D_{\text{stsp}}^{\text{core}}$ than open-loop no-propagation on $13/13$ systems and than open-loop tied on $12/13$. It also beats both open-loop methods on held-out NLL on $13/13$ systems. Held-out NLL is on par with frozen-core shell calibration, which leaves relative $D_{\text{stsp}}^{\text{core}}$ at 1 by construction. Fig. 4b illustrates the complementary filtering capability. After adaptation, DynaMix acts as the transition model in an EKF for Bayesian state estimation. Uncertainty is propagated via approximate local Jacobians (Appx. C.5), while sparse observations are assimilated online.

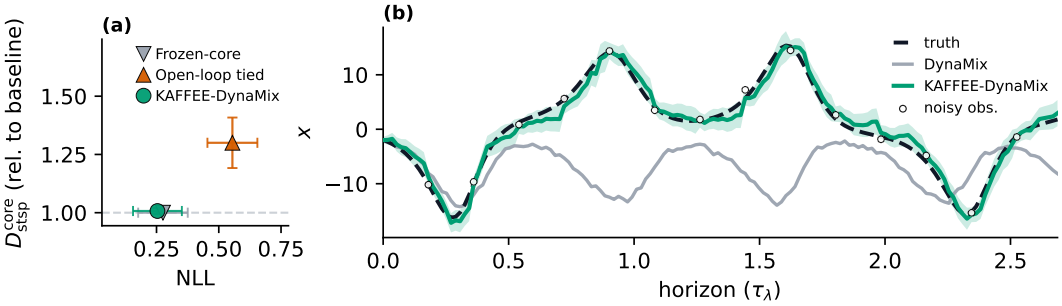


Figure 4: **DPC gap and online filtering in DynaMix. (a)** KAFFEE stays near the zero-shot attractor while open-loop tied inflates $D_{\text{stsp}}^{\text{core}}$. Markers are medians of the system-level seed-medians $\pm$ SEs across those $n = 13$ systems. **(b)** KAFFEE turns DynaMix into an online Bayesian filter that assimilates sparse noisy observations; illustrated here on one Lorenz-63 coordinate [54].

6 Related Work

Probabilistic DSR. Recovering chaotic attractors from data is a central challenge in DSR. Learned surrogates are asked to act as autonomous generative models that preserve long-run dynamical structure [1–3, 6]. Probabilistic neural SSMs and SDEs add latent uncertainty, typically via finite-horizon variational or marginal-likelihood objectives [42, 55]. Finite-horizon test error in learned dynamical systems need not imply recovery of moments, invariant measures, or Lyapunov structure [5]. We identify objective-level failure modes by which probabilistic objectives can degrade those invariants or mask error through inflated noise. The same issues arise when retrofitting DSFMs [23–25]. Concurrent work around scoring rule-based adaptation of deterministic simulators/foundation models for partial differential equations independently supports adapting strong deterministic cores rather than training probabilistic models from scratch [56]. Our results suggest that such proper scores should be complemented by diagnostics of autonomous invariants and locally grounded uncertainty when the scientific target goes beyond finite-horizon forecasting.

Data assimilation and differentiable filtering. Data assimilation (DA) is a field overlapping with DSR that combines a dynamical model, noisy observations, and uncertainty to estimate states and improve forecasts [15, 57]. Chaotic DA in particular emphasizes how filter covariances are shaped by tangent dynamics [18, 22]. Differentiable EKFs are a standard tool in SSMs [26–28, 58–60], and methods combining machine learning with DA use this machinery to learn or correct dynamics [61–63]. KAFFEE uses this established filtering machinery to implement a training principle for

probabilistic DSR: evaluate a filtered, localized objective while keeping local uncertainty tied to the learned dynamics. For a Gaussian innovation NLL objective and Jacobian-based covariance propagation, this results in the shown differentiable-EKF instantiation. We show how this pairing mitigates the DPC gap and scales to in-context adaptation and online Bayesian filtering with DynaMix.

Stabilizing BPTT for chaotic surrogates. Gradient-based optimization of recurrent surrogates in chaotic regimes is notoriously difficult due to exploding Jacobian product chains [48]. Methods like GTF [34] and Error Forcing [49] stabilize training by damping those products geometrically. However, determining forcing strengths or intervals remains difficult [64]. In Appx. A.5 we derive how a Kalman update induces a state-adaptive anisotropic damping of the Jacobian chain for piecewise-affine latent transitions, which can attenuate amplification analogously to forcing methods (as also noted in Pals et al. [51]). In Appx. D.3 we test training through this update against GTF-like fixed scalar forcing strength and against stopping gradients through the Kalman update. On Lorenz-96, KAFFEE achieves better autonomous reconstruction than either control, even when selected fixed gains attain comparable NLL.

7 Discussion

This work exposes the DPC gap: improving finite-horizon probabilistic training objectives can degrade properties required for DSR, and thus reliable scientific use. In Gaussian open-loop rollout objectives, covariance-volume pressure creates a shortcut toward suppressing physical expansion in the learned dynamics. Flexible observation and process noise parametrizations can also mask transition errors, while missing covariance propagation can leave uncertainty blind to what local dynamics dictate. Our three-axis evaluation separates local probabilistic behavior, stochastic dynamic fidelity, and deterministic dynamic fidelity. Across the AL-RNN and DynaMix experiments, filtered innovation objectives improve the balance between predictive fit and dynamical fidelity, mitigating the DPC gap. KAFFEE does so while also tying covariance propagation to local Jacobians.

In the appendix, we also examine practical extensions. A rank-reduced variant (rr-KAFFEE) achieves competitive performance with dense KAFFEE on Lorenz-96 at filter rank $r = 20$ (matching the rank of the true process noise), while its measured training time scales approximately quadratically with latent size at fixed rank [65] (Appx. D.5). Using an expressive class of piecewise-affine RNNs (shPLRNNs [34]), we also demonstrate probabilistic reconstruction on EEG recordings (Appx. D.6).

Limitations. KAFFEE mitigates the observed DPC gap failure modes, but does not guarantee preservation of autonomous dynamics. If the deterministic prior is already close to the truth, joint learning of noise and deterministic transition can worsen that prior (Appx. D.1). KAFFEE also inherits the EKF's local Gaussian approximation, which may not capture the full complexity of true predictive distributions in highly nonlinear regimes (e.g., for infrequent observations). KAFFEE-DynaMix uses approximate Jacobians, and its filtered no-propagation control remains competitive (Appx. C.5). Our certification diagnostic for dynamically grounded local uncertainty $G_{\tan}$ depends on knowledge of the true local Jacobians; on real-world data such as the studied EEG recordings, the unknown true dynamics limit conclusions about reconstruction and DPC gap mechanisms (Appx. D.6). The EKF's dense covariance propagation scales cubically with latent dimension (Appx. A.7). Rank reduction lowers its cost in our benchmark, but it requires choosing an appropriate rank (Appx. D.5).

Future work. To relax the EKF's local Gaussian assumption in strongly nonlinear or sparsely observed regimes, natural extensions include sigma-point, ensemble, particle, or structured variational approaches [41, 51, 66–69] (Appx. A.6). However, their scalability and ability to preserve KAFFEE's mitigation of the DPC gap remain to be established. The connection of KAFFEE to GTF in Appx. A.5 and ablations in Appx. D.3 also suggest using the same stabilization that KAFFEE provides in other training objectives for dynamically faithful probabilistic surrogates.

Taken together, our results show that predictive fit and dynamical reconstruction require separate evaluation, and that a filtered objective tying uncertainty to learned dynamics can reconcile both properties. We hope this work paves the way for surrogate models that handle uncertainty in a principled way without sacrificing dynamical consistency.

Code, data, and trained models for all experiments, together with documentation for reproducing the results, are available at `https://github.com/aherz/kaffee-dpc`.

Acknowledgments and Disclosure of Funding

This work was funded under the Excellence Strategy of the German Federal and State Governments; by the Federal Ministry of Research, Technology and Space (BMFTR) through the NAILIt project under the neuroAI initiative (grant numbers 01GQ2509A and 01GQ2509B); by the German Research Foundation (DFG) through Research Unit FOR 5159, subproject 11, the Collaborative Research Centre 265 subproject B08, and individual grant DU 354/15-1; and by the Hector II Foundation.

References

[1] J Pathak, Z Lu, B. R Hunt, M Girvan, and E Ott. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27 (12):121102, 2017. ISSN 1054-1500, 1089-7682.

[2] S Vyas, M. D Golub, D Sussillo, and K. V Shenoy. Computation Through Neural Population Dynamics. *Annual Review of Neuroscience*, 43(1):249–275, 2020. ISSN 0147-006X, 1545-4126.

[3] D Durstewitz, G Koppe, and M. I Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, 24(11):693–710, 2023.

[4] D Durstewitz, C. J Hemmer, F Hess, C. R Doll, and L Eisenmann. Position: A dynamical systems perspective is needed to advance time series modeling. In *Forty-third International Conference on Machine Learning Position Paper Track*, 2026.

[5] J Park, N. T Yang, and N Chandramoorthy. When are dynamical systems learned from time series data statistically accurate? In *Advances in Neural Information Processing Systems*, volume 37, pages 43975–44008, 2024.

[6] N Göring, F Hess, M Brenner, Z Monfared, and D Durstewitz. Out-of-Domain Generalization in Dynamical Systems Reconstruction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

[7] J Botvinick-Greenhouse, R Martin, and Y Yang. Invariant Measures in Time-Delay Coordinates for Unique Dynamical System Identification. *Phys. Rev. Lett.*, 135(16):167202, 2025.

[8] R Jiang, P. Y Lu, E Orlova, and R Willett. Training neural operators to preserve invariant measures of chaotic attractors. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, 2023.

[9] Y Schiff, Z. Y Wan, J. B Parker, S Hoyer, V Kuleshov, F Sha, and L Zepeda-Núñez. DySLIM: Dynamics stable learning by invariant measure for chaotic systems. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

[10] H Luo, Y Du, H Fan, X Wang, J Guo, and X Wang. Reconstructing bifurcation diagrams of chaotic circuits with reservoir computing. *Phys. Rev. E*, 109(2):024210, 2024.

[11] F Falasca. Probing forced responses and causality in data-driven climate emulators: Conceptual limitations and the role of reduced-order models. *Physical Review Research*, 7(4):043314, 2025. ISSN 2643-1564.

[12] A Vinograd, A Nair, J. H Kim, S. W Linderman, and D. J Anderson. Causal evidence of a line attractor encoding an affective state. *Nature*, 634(8035):910–918, 2024.

[13] G. A Gottwald and S Reich. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Physica D: Nonlinear Phenomena*, 423:132911, 2021. ISSN 01672789.

[14] A. F Psaros, X Meng, Z Zou, L Guo, and G. E Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023. ISSN 00219991.

[15] S Reich and C Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, 1 edition, 2015. ISBN 978-1-107-06939-8 978-1-107-66391-6 978-1-107-70680-4.

[16] I Price, A Sanchez-Gonzalez, F Alet, T. R Andersson, A El-Kadi, D Masters, T Ewalds, J Stott, S Mohamed, P Battaglia, R Lam, and M Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637 (8044):84–90, 2025. ISSN 0028-0836, 1476-4687.

[17] L Hewing, J Kabzan, and M. N Zeilinger. Cautious Model Predictive Control Using Gaussian Process Regression. *IEEE Transactions on Control Systems Technology*, 28(6):2736–2743, 2020. ISSN 1063-6536, 1558-0865, 2374-0159.

[18] A Trevisan and L Palatella. On the Kalman filter error covariance collapse into the unstable subspace. *Nonlinear Processes in Geophysics*, 18(2):243–250, 2011.

[19] D. J. C MacKay. Bayesian interpolation. *Neural Comput.*, 4(3):415–447, 1992. ISSN 0899-7667.

[20] C. E Rasmussen and C. K. I Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[21] J. P Eckmann and D Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57(3): 617–656, 1985.

[22] A Carrassi, M Bocquet, J Demaeyer, C Grudzien, P Raanes, and S Vannitsem. Data Assimilation for Chaotic Dynamics. In S. K Park and L Xu, editors, *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. IV)*, pages 1–42. Springer International Publishing, Cham, 2022. ISBN 978-3-030-77721-0 978-3-030-77722-7.

[23] C. J Hemmer and D Durstewitz. True Zero-Shot Inference of Dynamical Systems Preserving Long-Term Statistics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] J. B Lai, A Bao, and W Gilpin. Panda: A pretrained forecast model for chaotic dynamics. In *The Fourteenth International Conference on Learning Representations*, 2026.

[25] C Liu, B Zhao, J Ding, and Y Li. Chaosnexus: A foundation model for ODE-based chaotic system forecasting with hierarchical multi-scale awareness. In *Forty-third International Conference on Machine Learning*, 2026.

[26] N Gupta and R Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, 19(6):774–783, 1974. ISSN 0018-9286.

[27] A Gorad, Z Zhao, and S Särkkä. Parameter estimation in non-linear state-space models by automatic differentiation of non-linear kalman filters. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.

[28] S Särkkä and L Svensson. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, second edition, 2023.

[29] E. N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability, Reading, UK, ECMWF*, volume 1, pages 1–18, 1996.

[30] K Doya. Bifurcations in the Learning of Recurrent Neural Networks. In *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems*, 1992.

[31] M Levine and A Stuart. A framework for machine learning of model error in dynamical systems. *Communications of the American Mathematical Society*, 2(7):283–344, 2022. ISSN 2692-3688.

[32] S Bengio, O Vinyals, N Jaitly, and N Shazeer. Scheduled sampling for sequence prediction with recurrent Neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1171–1179, 2015.

[33] M Ranzato, S Chopra, M Auli, and W Zaremba. Sequence Level Training with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[34] F Hess, Z Monfared, M Brenner, and D Durstewitz. Generalized Teacher Forcing for Learning Chaotic Dynamics. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 13017–13049, 2023.

[35] M Forgione and D Piga. Continuous-time system identification with neural networks: Model structures and fitting criteria. *European Journal of Control*, 59:69–81, 2021.

[36] G Beintema, R Toth, and M Schoukens. Nonlinear state-space identification using deep encoder networks. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 241–250, 2021.

[37] G. I Beintema, M Schoukens, and R Tóth. Deep subspace encoders for nonlinear system identification. *Automatica*, 156(C), 2023. ISSN 0005-1098.

[38] A Girard, C. E Rasmussen, and R Murray-Smith. Multiple-step ahead prediction for non linear dynamic systems–a Gaussian process treatment with propagation of the uncertainty. In *Sixteenth Annual Conference on Neural Information Processing Systems*, volume 15, pages 529–536, 2002.

[39] A Doerr, C Daniel, M Schiegg, N.-T Duy, S Schaal, M Toussaint, and T Sebastian. Probabilistic recurrent state-space models. In *International Conference on Machine Learning*, pages 1280–1289, 2018.

[40] A. D Ialongo, M Van Der Wilk, J Hensman, and C. E Rasmussen. Overcoming mean-field approximations in recurrent Gaussian process models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 2019.

[41] Y Zhao and S Linderman. Revisiting structured variational autoencoders. In *International Conference on Machine Learning*, pages 42046–42057, 2023.

[42] M Brenner, F Hess, J. M Mikhaeil, L. F Bereska, Z Monfared, P.-C Kuo, and D Durstewitz. Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 2022.

[43] R. E Turner and M Sahani. Two problems with variational expectation maximisation for time series models. In *Bayesian Time Series Models*, pages 104–124. Cambridge University Press, 2011. ISBN 978-0-511-98467-9.

[44] J Bayer, M Soelch, A Mirchev, B Kayalibay, and P van der Smagt. Mind the Gap when Conditioning Amortised Inference in Sequential Latent-Variable Models. In *International Conference on Learning Representations*, 2021.

[45] W. A Fuller. *Measurement Error Models*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 1987.

[46] J Staudenmayer and J. P Buonaccorsi. Measurement Error in Linear Autoregressive Models. *Journal of the American Statistical Association*, 100(471):841–852, 2005. ISSN 01621459.

[47] P. E McSharry and L. A Smith. Better Nonlinear Models from Noisy Data: Attractors with Maximum Likelihood. *Physical Review Letters*, 83(21):4285–4288, 1999. ISSN 0031-9007, 1079-7114.

[48] J. M Mikhaeil, Z Monfared, and D Durstewitz. On the difficulty of learning chaotic dynamics with RNNs. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, 2022.

[49] A. E Sağtekin, C Bredenberg, and C Savin. Error Forcing in Recurrent Neural Networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[50] A Herz, D Durstewitz, and G Koppe. Teacher Forcing as Generalized Bayes: Optimization Geometry Mismatch in Switching Surrogates for Chaotic Dynamics. *arXiv preprint arXiv:2604.25904*, 2026. Presented at the OPTIMAL Workshop at AISTATS 2026.

[51] M Pals, A. E Sağtekin, F Pei, M Gloeckler, and J Macke. Inferring stochastic low-rank recurrent neural networks from neural data. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pages 18225–18264, 2024.

[52] G Koppe, H Toutounji, P Kirsch, S Lis, and D Durstewitz. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. *PLOS Computational Biology*, 15(8):1–35, 2019.

[53] M Brenner, C. J Hemmer, Z Monfared, and D Durstewitz. Almost-Linear RNNs Yield Highly Interpretable Symbolic Codes in Dynamical Systems Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[54] E. N Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[55] X Li, T.-K. L Wong, R. T. Q Chen, and D Duvenaud. Scalable Gradients for Stochastic Differential Equations. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3870–3882, 2020.

[56] C Diaconu, M Cranmer, R. E Turner, T Marwah, and P Mukhopadhyay. Probabilistic retrofitting of learned simulators. In *Forty-third International Conference on Machine Learning*, 2026.

[57] A Carrassi, M Bocquet, L Bertino, and G Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *WIREs Climate Change*, 9(5):e535, 2018. ISSN 1757-7780, 1757-7799.

[58] J Kokkala, A Solin, and S Särkkä. Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems. *Journal of advances in information fusion*, 11(1):15, 2016.

[59] T Haarnoja, A Ajay, S Levine, and P Abbeel. Backprop KF: Learning discriminative deterministic state estimators. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4383–4391, 2016.

[60] I. C Tanoh, M Deistler, J. H Macke, and S Linderman. Identifying multi-compartment Hodgkin-Huxley models with high-density extracellular voltage recordings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[61] J Brajard, A Carrassi, M Bocquet, and L Bertino. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model. *Journal of Computational Science*, 44:101171, 2020. ISSN 18777503.

[62] M Bocquet, A Farchi, and Q Malartic. Online learning of both state and dynamics using ensemble Kalman filters. *Foundations of Data Science*, 3(3):305–330, 2021. ISSN 2639-8001.

[63] M Adrian, D Sanz-Alonso, and R Willett. Auto-differentiable data assimilation: Co-learning of states, dynamics, and filtering algorithms. *arXiv preprint arXiv:2603.20891*, 2026.

[64] V Sip, M Breyton, S Petkoski, and V Jirsa. Double projection for reconstructing dynamical systems: Between stochastic and deterministic regimes. *Machine Learning: Science and Technology*, 7(4):045047, 2026. ISSN 2632-2153.

[65] J Schmidt, P Hennig, J Nick, and F Tronarp. The Rank-Reduced Kalman Filter: Approximate Dynamical-Low-Rank Filtering In High Dimensions. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[66] A Doucet, N de Freitas, K Murphy, and S Russell. Rao–Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[67] E. A Wan and R Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158, 2000.

[68] G Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[69] M Brenner, F Hess, G Koppe, and D Durstewitz. Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[70] A. H Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Elsevier, 1970. ISBN 978-0-12-381550-7.

[71] K. T Alligood, T. D Sauer, and J. A Yorke. *Chaos: An Introduction to Dynamical Systems*. Textbooks in Mathematical Sciences. Springer New York, New York, NY, 1996.

[72] K. P Murphy. *Probabilistic machine learning: Advanced topics*. MIT press, 2023.

[73] J. H Macke, L Buesing, J Cunningham, B. M Yu, K. V Shenoy, and M Sahani. Empirical models of spiking in neural populations. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[74] M Dowling, Y Zhao, and I. M Park. eXponential FAmily dynamical systems (XFADS): Large-scale nonlinear gaussian state-space modeling. *Advances in Neural Information Processing Systems*, 37:13458–13488, 2024.

[75] C Naesseth, S Linderman, R Ranganath, and D Blei. Variational Sequential Monte Carlo. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 2018.

[76] T. A Le, M Igl, T Rainforth, T Jin, and F Wood. Auto-Encoding Sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

[77] C. J Maddison, J Lawson, G Tucker, N Heess, M Norouzi, A Mnih, A Doucet, and Y Teh. Filtering Variational Objectives. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[78] P. G Chang, K. P Murphy, and M Jones. On diagonal approximations to the extended Kalman filter for online training of Bayesian neural networks. In *Continual Lifelong Learning Workshop at ACML 2022*, 2022.

[79] P. G Chang, G Durán-Martín, A Shestopaloff, M Jones, and K. P Murphy. Low-rank extended Kalman filtering for online learning of neural networks from streaming data. In S Chandar, R Pascanu, H Sedghi, and D Precup, editors, *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 1025–1071, 2023.

[80] C Grudzien, A Carrassi, and M Bocquet. Chaotic dynamics and the role of covariance inflation for reduced rank Kalman filters with model error. *Nonlinear Processes in Geophysics*, 25(3):633–648, 2018. ISSN 1607-7946.

[81] K Reif, S Gunther, E Yaz, and R Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Transactions on Automatic Control*, 44(4):714–728, 1999. ISSN 00189286.

[82] D. P Kingma and J Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[83] M Brenner and G Koppe. Uncovering the computational roles of nonlinearity in sequence modeling using almost-linear RNNs. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

[84] S Hashemi, D Durstewitz, and G Koppe. Continual learning of dynamical systems in recurrent neural networks through recyclable unit gating. *arXiv preprint arXiv:2609.38356*, 2026.

[85] N Emonds and G Koppe. Learning transferable policies from action-free time series through dynamical embeddings. *arXiv preprint arXiv:2610.03065*, 2026.

[86] L Liu, H Jiang, P He, W Chen, X Liu, J Gao, and J Han. On the Variance of the Adaptive Learning Rate and Beyond. In *International Conference on Learning Representations*, 2020.

[87] J. R Hershey and P. A Olsen. Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages IV–317–IV–320, 2007.

[88] G Benettin, L Galgani, A Giorgilli, and J.-M Strelcyn. Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory. *Meccanica*, 15(1):9–20, 1980. ISSN 0025-6455, 1572-9648.

[89] K Geist, U Parlitz, and W Lauterborn. Comparison of Different Methods for Computing Lyapunov Exponents. *Progress of Theoretical Physics*, 83(5):875–893, 1990. ISSN 0033-068X.

[90] W Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[91] G. B Moody, R. G Mark, and A. L Goldberger. PhysioNet: A research resource for studies of complex physiologic and biomedical signals. *Computers in Cardiology*, 27:179–182, 2000. ISSN 0276-6574.

A Theory Details and Proofs

This section supplements the theory in the main text. We discuss the proof of volume pressure localization, a noisy-state attenuation diagnostic, and a condition under which covariance contraction increases the expected (i.e., population) Gaussian NLL. We also discuss and formalize KAFFEE's connection to intervention-based gradient stabilization training methods, in particular generalized teacher forcing (GTF) [34].

A.1 Alleviating core collapse via filtering localization

In Section 3.2, we discuss how Gaussian objectives applied to open-loop predictive distributions accumulate uncertainty under autonomous propagation. In contrast, filtered innovation likelihood objectives repeatedly pull forecasts back through data assimilation, such that each likelihood term scores uncertainty propagated from a local post-assimilation belief. Here, we demonstrate this contrast with a diagnostic that isolates the covariance-volume component of the Gaussian objective. As in the main text, we denote filtered estimates by writing $\tau \mid t$ for an estimate at time τ conditioned on observations up to time t. Under local linearization, the open-loop covariance analyzed below is known as the *free-forecast covariance* in Bayesian filtering/data assimilation [22]. In the linear time-varying case, its accumulated process-noise term is known as the *controllability matrix* [70].

The following Proposition 1 is a diagnostic result for the covariance-volume term in the Gaussian NLL. This does not mean that the full Gaussian objective always contracts the learned dynamics. The Mahalanobis (data-fit) term can oppose this pressure, especially when the predictive distribution is underdispersed. Instead, the proposition identifies a structural optimization shortcut available to open-loop likelihood training. With open-loop Jacobian covariance propagation, shrinking Jacobian products directly reduces the volume penalty. Removing propagation avoids this penalty, but also makes predictive uncertainty dynamically blind. Filtered innovation likelihood keeps Jacobian propagation, yet scores it after assimilation has reset the belief state, so the long-horizon volume pressure is localized.

Proposition 1 (Open-loop volume pressure and filtering localization)**.** *Fix time step t and horizon $h \geq 1$, and assume the observation matrix $\boldsymbol{B}$ and covariances $\boldsymbol{\Gamma}_{\theta_s} \succeq 0$, $\boldsymbol{\Sigma}_t \succeq 0$, $\boldsymbol{\Lambda}_{\theta_s} \succ 0$ to be fixed along each local scaling path $\boldsymbol{J}_{\theta_c,k}(s) = e^s \boldsymbol{J}_{\theta_c,k}$. Along those paths, the observed open-loop covariance can be written as*

$$\boldsymbol{\Omega}^{\text{open}}_{\theta,t+h}(s) = \boldsymbol{\Lambda}_{\theta_s} + \sum_{\ell=0}^{h} e^{2\ell s} \boldsymbol{A}_\ell, \qquad \boldsymbol{A}_\ell \succeq 0,$$

where $\boldsymbol{A}_\ell$ collects the observed-space covariance contributions that have undergone ℓ Jacobian propagations. Hence

$$\frac{\partial}{\partial s} \frac{1}{2} \log\det \boldsymbol{\Omega}^{\text{open}}_{\theta,t+h}(s) = \sum_{\ell=1}^{h} \ell e^{2\ell s} \operatorname{tr}\left[\boldsymbol{\Omega}^{\text{open}}_{\theta,t+h}(s)^{-1} \boldsymbol{A}_\ell\right] \geq 0,$$

with strict inequality whenever some propagated contribution $\boldsymbol{A}_\ell \neq 0$ for $\ell \geq 1$. Thus the open-loop log-determinant term penalizes covariance contributions in proportion to how many Jacobian propagations they have undergone. If the covariance recursion uses no Jacobian propagation, this derivative is zero.

For a filtered one-step score, let

$$\boldsymbol{\Sigma}_{t|t-1}(s) = e^{2s} \boldsymbol{J}_{\theta_c,t} \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{J}_{\theta_c,t}^\top + \boldsymbol{\Gamma}_{\theta_s}, \qquad \boldsymbol{\Omega}_t(s) = \boldsymbol{B} \boldsymbol{\Sigma}_{t|t-1}(s) \boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}.$$

Then

$$\frac{\partial}{\partial s} \frac{1}{2} \log\det \boldsymbol{\Omega}_t(s) = e^{2s} \operatorname{tr}\left[\boldsymbol{\Omega}_t(s)^{-1} \boldsymbol{B} \boldsymbol{J}_{\theta_c,t} \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{J}_{\theta_c,t}^\top \boldsymbol{B}^\top\right].$$

Moreover, with Kalman gain $\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{B}^\top \boldsymbol{\Omega}_t^{-1}$, the covariance update satisfies

$$\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{B}^\top \boldsymbol{\Omega}_t^{-1} \boldsymbol{B} \boldsymbol{\Sigma}_{t|t-1} \preceq \boldsymbol{\Sigma}_{t|t-1}.$$

Thus filtered innovation scoring keeps Jacobian-based covariance propagation, but evaluates it locally from a post-assimilation belief instead of on a long open-loop covariance product.

Proof. We write $\boldsymbol{J}_k := \boldsymbol{J}_{\theta_c,k}$ and define the ℓ-step product ending at $t+h$ by

$$\boldsymbol{\Phi}_{t+h}^{(\ell)} := \boldsymbol{J}_{t+h}\boldsymbol{J}_{t+h-1}\cdots\boldsymbol{J}_{t+h-\ell+1}, \qquad \boldsymbol{\Phi}_{t+h}^{(0)} := \boldsymbol{I}.$$

Along the stated path, $\boldsymbol{\Phi}_{t+h}^{(\ell)}(s) = e^{\ell s}\boldsymbol{\Phi}_{t+h}^{(\ell)}$. The initial covariance undergoes h propagations, whereas the successive process-noise contribution undergoes $0,\dots,h-1$ propagations. Hence

$$\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}}(s) = \boldsymbol{\Lambda}_{\theta_s} + \sum_{\ell=0}^{h} e^{2\ell s}\boldsymbol{A}_\ell,$$

where

$$\boldsymbol{A}_\ell = \begin{cases} \boldsymbol{B}\boldsymbol{\Phi}_{t+h}^{(h)}\boldsymbol{\Sigma}_t(\boldsymbol{\Phi}_{t+h}^{(h)})^\top\boldsymbol{B}^\top, & \text{for } \ell = h \\ \boldsymbol{B}\boldsymbol{\Phi}_{t+h}^{(\ell)}\boldsymbol{\Gamma}_{\theta_s}(\boldsymbol{\Phi}_{t+h}^{(\ell)})^\top\boldsymbol{B}^\top, & \text{for } 0 \le \ell < h. \end{cases}$$

Since $\boldsymbol{\Sigma}_t \succeq 0$ and $\boldsymbol{\Gamma}_{\theta_s} \succeq 0$, each $\boldsymbol{A}_\ell \succeq 0$, and since $\boldsymbol{\Lambda}_{\theta_s} \succ 0$, the observed covariance is positive definite with a positive definite inverse. Now we differentiate the log-determinant term. Using $\partial_s \log\det\boldsymbol{\Omega} = \mathrm{tr}(\boldsymbol{\Omega}^{-1}\partial_s\boldsymbol{\Omega})$ gives

$$\frac{\partial}{\partial s}\frac{1}{2}\log\det\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}}(s) = \sum_{\ell=1}^{h}\ell e^{2\ell s}\mathrm{tr}\left[\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}}(s)^{-1}\boldsymbol{A}_\ell\right].$$

Every trace is nonnegative, and is positive whenever $\boldsymbol{A}_\ell \neq 0$, because it equals the trace of the positive-semidefinite matrix $(\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}})^{-1/2}\boldsymbol{A}_\ell(\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}})^{-1/2}$:

$$\mathrm{tr}\left[(\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}})^{-1}\boldsymbol{A}_\ell\right] = \mathrm{tr}\left[(\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}})^{-1/2}\boldsymbol{A}_\ell(\boldsymbol{\Omega}_{\theta,t+h}^{\mathrm{open}})^{-1/2}\right].$$

With no Jacobian propagation, the covariance does not depend on the scaled Jacobians, so then this derivative is zero. This proves the open-loop part.

We next prove the filtered localization statement. The one-step filtered predictive covariance along the scaling path is

$$\boldsymbol{\Sigma}_{t|t-1}(s) = e^{2s}\boldsymbol{J}_{\theta_c,t}\boldsymbol{\Sigma}_{t-1|t-1}\boldsymbol{J}_{\theta_c,t}^\top + \boldsymbol{\Gamma}_{\theta_s},$$

with corresponding one-step innovation covariance

$$\boldsymbol{\Omega}_t(s) = \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}(s)\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}.$$

For the one-step innovation score, differentiating at a fixed $\boldsymbol{\Sigma}_{t-1|t-1}$ as above then gives the stated derivative

$$\frac{\partial}{\partial s}\frac{1}{2}\log\det\boldsymbol{\Omega}_t(s) = e^{2s}\mathrm{tr}\left[\boldsymbol{\Omega}_t(s)^{-1}\boldsymbol{B}\boldsymbol{J}_{\theta_c,t}\boldsymbol{\Sigma}_{t-1|t-1}\boldsymbol{J}_{\theta_c,t}^\top\boldsymbol{B}^\top\right].$$

Finally, write the Kalman gain

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top\boldsymbol{\Omega}_t^{-1}, \qquad \boldsymbol{\Omega}_t = \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}.$$

The covariance update can be written as

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{\Sigma}_{t|t-1}.$$

Substituting the expression for $\boldsymbol{K}_t$ gives

$$\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top\boldsymbol{\Omega}_t^{-1}\boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}.$$

The subtracted term is positive semidefinite, since

$$\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top\boldsymbol{\Omega}_t^{-1}\boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1} = \left(\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top\boldsymbol{\Omega}_t^{-1/2}\right)\left(\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top\boldsymbol{\Omega}_t^{-1/2}\right)^\top \succeq 0.$$

Hence

$$\boldsymbol{\Sigma}_{t|t} \preceq \boldsymbol{\Sigma}_{t|t-1}.$$

Thus the filtered objective still scores Jacobian-propagated covariance, but after each observation the covariance is contracted before the next prediction, whereas open-loop continues on the unassimilated product without contraction. □

Figure 2 details. Proposition 1 isolates the log-volume term of the Gaussian NLL; the Mahalanobis term can oppose this pressure. Fig. 2 illustrates this tradeoff on the tent map

$$f_s(z) = s \min\{z, 1-z\}, \qquad z \in [0,1],$$

with slope $s = 1.8$, putting it into the chaotic regime [71]. We generate data by clipping each noisy state $z_{t+1} = f_s(z_t) + \xi_t$ such that $z_{t+1} \in [0,1]$ (to ensure boundedness), with $\xi_t \sim \mathcal{N}(0, 0.04^2)$. Observations have i.i.d Gaussian noise of standard deviation 0.02. The reference predictive histograms in Panel (a) are obtained by Monte Carlo simulation for the open-loop case and a bootstrap particle filter with $4 \cdot 10^4$ particles for the filtered case. To profile the NLL objectives over s, we use $5 \cdot 10^4$ trajectories of 25 transitions, holding both noise scales fixed. The excess profiles in Panel (b) subtract each objective's own minimum, so their minima can be compared without equating the levels of different NLLs. Panel (c) shows how the log-volume term shifts the open-loop optimum toward weaker expansion.

A.2 Open-loop likelihood and volume pressure in exponential family models

Here, we show that the structural optimization shortcut identified in Proposition 1 extends beyond Gaussian likelihoods. By applying a Laplace approximation to the posterior in models with Gaussian latent dynamics and arbitrary exponential family observations, we obtain a similar covariance volume penalty under open-loop training.

We now assume the observation distribution is given by a conditional exponential family [72]:

$$p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) = h(\boldsymbol{x}_t) \exp\left(\boldsymbol{\eta}(\boldsymbol{z}_t)^\top \boldsymbol{T}(\boldsymbol{x}_t) - A(\boldsymbol{\eta}(\boldsymbol{z}_t))\right),$$

where $\boldsymbol{T}(\boldsymbol{x}_t) \in \mathbb{R}^N$ is the vector of sufficient statistics, $h(\boldsymbol{x}_t)$ is the base measure, and $A(\boldsymbol{\eta}(\boldsymbol{z}_t))$ is the log-partition function.

We consider a Generalized Linear Model (GLM) formulation where the natural parameter $\boldsymbol{\eta}(\boldsymbol{z}_t) \in \mathbb{R}^N$ is mapped from the latents $\boldsymbol{z}_t \in \mathbb{R}^M$ via emission matrix $\boldsymbol{B} \in \mathbb{R}^{N \times M}$ and bias $\boldsymbol{b} \in \mathbb{R}^N$:

$$\boldsymbol{\eta}(\boldsymbol{z}_t) = \boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{b}.$$

The prior $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1})$ is approximated as Gaussian $\mathcal{N}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$ via linearization / the EKF's predict step, as before. We now have to find new expressions for the update step:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) \propto p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}),$$

and the innovation likelihood:

$$\begin{aligned} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) &= \int p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1})\, d\boldsymbol{z}_t, \\ &= p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}) \frac{1}{p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})}. \end{aligned}$$

Thus concretely we need to find an approximation to $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})$.

Approximating the posterior with Laplace. The Laplace approximation gives a Gaussian approximation: $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) \approx \mathcal{N}(\boldsymbol{\mu}_t^{\mathrm{lp}}, \boldsymbol{\Sigma}_t^{\mathrm{lp}})$, where the mean is given by the posterior mode:

$$\boldsymbol{\mu}_t^{\mathrm{lp}} = \arg\max_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}),$$

and the covariance is the inverse negative Hessian of the log posterior:

$$\boldsymbol{\Sigma}_t^{\mathrm{lp}} = \left(-\nabla^2_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) \Big|_{\boldsymbol{z}_t = \boldsymbol{\mu}_t^{\mathrm{lp}}} \right)^{-1}.$$

The full log posterior is:

$$\begin{aligned} \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) = \log h(\boldsymbol{x}_t) &+ (\boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{b})^\top \boldsymbol{T}(\boldsymbol{x}_t) - A(\boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{b}) \\ &- \frac{1}{2} \log\det(2\pi \boldsymbol{\Sigma}_{t|t-1}) - \frac{1}{2} (\boldsymbol{z}_t - \boldsymbol{\mu}_{t|t-1})^\top \boldsymbol{\Sigma}_{t|t-1}^{-1} (\boldsymbol{z}_t - \boldsymbol{\mu}_{t|t-1}) \\ &- \log p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}). \end{aligned}$$

The covariance approximation is:

$$\begin{aligned}\boldsymbol{\Sigma}_t^{\mathrm{lp}} &= \left(-\nabla^2_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})\Big|_{\boldsymbol{z}_t=\boldsymbol{\mu}_t^{\mathrm{lp}}}\right)^{-1} \\ &= \left(-\nabla^2_{\boldsymbol{z}_t} \log p(\boldsymbol{x}_t \mid \boldsymbol{z}_t)\Big|_{\boldsymbol{z}_t=\boldsymbol{\mu}_t^{\mathrm{lp}}} + \boldsymbol{\Sigma}^{-1}_{t|t-1}\right)^{-1} \\ &= \left(\boldsymbol{B}^\top \boldsymbol{H} \boldsymbol{B} + \boldsymbol{\Sigma}^{-1}_{t|t-1}\right)^{-1},\end{aligned}$$

with $\boldsymbol{H} = \nabla^2_{\boldsymbol{\eta}} A(\boldsymbol{\eta}(\boldsymbol{\mu}_t^{\mathrm{lp}}))$, the curvature of the log partition function.

Note that the log posterior is globally concave for exponential-family observations with a Gaussian prior, and therefore the posterior has a unique mode. Starting from $\boldsymbol{z}_t^{(0)} = \boldsymbol{\mu}^{\mathrm{lp}}_{t|t-1}$, the mode can be found iteratively:

$$\begin{aligned}\boldsymbol{z}_t^{(\mathrm{new})} &= \boldsymbol{z}_t^{(\mathrm{old})} - \left[\nabla^2_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t^{(\mathrm{old})} \mid \boldsymbol{x}_{1:t})\right]^{-1} \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t^{(\mathrm{old})} \mid \boldsymbol{x}_{1:t}), \\ \boldsymbol{\mu}^{\mathrm{lp}}_{t|t} &= \boldsymbol{z}_t^{(\mathrm{final})},\end{aligned}$$

where the gradient is: $\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) = \boldsymbol{B}^\top \left(\boldsymbol{T}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{\eta}} A(\boldsymbol{\eta}(\boldsymbol{z}_t))\right) - \boldsymbol{\Sigma}^{-1}_{t|t-1}(\boldsymbol{z}_t - \boldsymbol{\mu}_{t|t-1})$.

Innovation likelihood given Laplace approximation to the posterior. We can now obtain an expression for the innovation log likelihood:

$$\log p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) = \log p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) + \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}) - \log p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}).$$

Using the Laplace approximation for the posterior:

$$\begin{aligned}\log p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) \approx \log p(\boldsymbol{x}_t \mid \boldsymbol{z}_t) &- \frac{1}{2}\log\det(2\pi\boldsymbol{\Sigma}_{t|t-1}) \\ &- \frac{1}{2}(\boldsymbol{z}_t - \boldsymbol{\mu}_{t|t-1})^\top \boldsymbol{\Sigma}^{-1}_{t|t-1}(\boldsymbol{z}_t - \boldsymbol{\mu}_{t|t-1}) \\ &- \left(-\frac{1}{2}\log\det(2\pi\boldsymbol{\Sigma}_t^{\mathrm{lp}}) - \frac{1}{2}(\boldsymbol{z}_t - \boldsymbol{\mu}_t^{\mathrm{lp}})^\top (\boldsymbol{\Sigma}_t^{\mathrm{lp}})^{-1}(\boldsymbol{z}_t - \boldsymbol{\mu}_t^{\mathrm{lp}})\right).\end{aligned}$$

Evaluating at the Laplace mode $\boldsymbol{z}_t = \boldsymbol{\mu}_t^{\mathrm{lp}}$:

$$\begin{aligned}\log p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) \approx \log p(\boldsymbol{x}_t \mid \boldsymbol{\mu}_t^{\mathrm{lp}}) &- \frac{1}{2}(\boldsymbol{\mu}_t^{\mathrm{lp}} - \boldsymbol{\mu}_{t|t-1})^\top \boldsymbol{\Sigma}^{-1}_{t|t-1}(\boldsymbol{\mu}_t^{\mathrm{lp}} - \boldsymbol{\mu}_{t|t-1}) \\ &- \frac{1}{2}\log\frac{\det\boldsymbol{\Sigma}_{t|t-1}}{\det\boldsymbol{\Sigma}_t^{\mathrm{lp}}}.\end{aligned}$$

Plugging in the definition of the precision, $(\boldsymbol{\Sigma}_t^{\mathrm{lp}})^{-1} = \boldsymbol{B}^\top \boldsymbol{H} \boldsymbol{B} + \boldsymbol{\Sigma}^{-1}_{t|t-1}$, we simplify the ratio of determinants using the Matrix Determinant Lemma: $\det(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^\top) = \det(\boldsymbol{W}^{-1} + \boldsymbol{V}^\top \boldsymbol{A}^{-1}\boldsymbol{U})\det(\boldsymbol{W})\det(\boldsymbol{A})$:

$$\frac{\det\boldsymbol{\Sigma}_{t|t-1}}{\det\boldsymbol{\Sigma}_t^{\mathrm{lp}}} = \det(\boldsymbol{H})\det(\boldsymbol{H}^{-1} + \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top).$$

Therefore, the final expression becomes:

$$\begin{aligned}\log p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) \approx \log p(\boldsymbol{x}_t \mid \boldsymbol{\mu}_t^{\mathrm{lp}}) &- \frac{1}{2}(\boldsymbol{\mu}_t^{\mathrm{lp}} - \boldsymbol{\mu}_{t|t-1})^\top \boldsymbol{\Sigma}^{-1}_{t|t-1}(\boldsymbol{\mu}_t^{\mathrm{lp}} - \boldsymbol{\mu}_{t|t-1}) \\ &- \frac{1}{2}\log\det\boldsymbol{H} \\ &- \frac{1}{2}\log\det(\boldsymbol{H}^{-1} + \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top).\end{aligned}$$

Open loop volume pressure for exponential family. The open-loop innovation likelihood has the following expression:

$$\log p(\boldsymbol{x}_{t+h} \mid \boldsymbol{x}_{1:t}) \approx \log p(\boldsymbol{x}_{t+h} \mid \boldsymbol{\mu}^{\mathrm{lp}}_{t+h}) - \frac{1}{2}(\boldsymbol{\mu}^{\mathrm{lp}}_{t+h} - \boldsymbol{\mu}_{t+h|t})^\top \boldsymbol{\Sigma}^{-1}_{t+h|t}(\boldsymbol{\mu}^{\mathrm{lp}}_{t+h} - \boldsymbol{\mu}_{t+h|t}) - \frac{1}{2}\log\det \boldsymbol{H} - \frac{1}{2}\log\det(\boldsymbol{H}^{-1} + \boldsymbol{B}\boldsymbol{\Sigma}_{t+h|t}\boldsymbol{B}^\top).$$

This indicates that we have the same volume pressure term as in the Gaussian case. Note that the two crucial assumptions made here are that the prior $p(\boldsymbol{z}_{t+h} \mid \boldsymbol{x}_{1:t})$ is approximated as Gaussian (by applying the EKF's predict step h times), and that the local posterior $p(\boldsymbol{z}_{t+h} \mid \boldsymbol{x}_{1:t,t+h})$ is approximated as Gaussian (via the Laplace approximation). The first assumption is also made for the Gaussian open-loop case, and seems more likely to cause errors as nonlinear dynamics over h steps without data corrections can distort the original Gaussian distribution at time t (the true distribution at $t+h$ could be highly multimodal, see Fig. 2a), whereas under this first assumption, the log-posterior is guaranteed to remain unimodal.

Poisson observations. As a concrete example, we consider Poisson objectives as commonly used in neuroscience models [51, 73, 74]. Assuming $x_{t,i} \sim \mathrm{Poisson}(\lambda_{t,i})$ with a log-link function $\eta_{t,i} = \log \lambda_{t,i} = [\boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{b}]_i$, the Hessian is a diagonal matrix of the predicted rates evaluated at the mode: $\boldsymbol{H} = \mathrm{diag}\left(\exp(\boldsymbol{B}\boldsymbol{\mu}^{\mathrm{lp}}_t + \boldsymbol{b})\right) \equiv \boldsymbol{\Lambda}_t$. The determinant contribution then becomes:

$$-\frac{1}{2}\log\det \boldsymbol{\Lambda}_t - \frac{1}{2}\log\det(\boldsymbol{\Lambda}_t^{-1} + \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top).$$

Here, $\boldsymbol{\Lambda}_t^{-1}$ can be interpreted as an effective, state-dependent observation noise covariance.

A.3 Noisy-state attenuation and innovation identifiability

Here we discuss a scalar diagnostic of classical noisy-state attenuation [45, 46] and its relationship to the filtered innovation likelihood. Treating noisy observations directly as states yields an attenuated observation-to-observation regression coefficient. In contrast, the exact Kalman innovation likelihood identifies the latent multiplier in the linear-Gaussian setting. We emphasize that the result here is local and does not imply global consistency of EKF training for nonlinear chaotic systems.

Let the true data-generating process be

$$z_1 \sim \mathcal{N}(0, p_\star), \qquad z_{t+1} = a_\star z_t + \xi_t, \qquad x_t = z_t + \varepsilon_t,$$

where

$$\xi_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, q_\star), \qquad \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, r_\star),$$

with z_1, $\{\xi_t\}_{t\geq 1}$, and $\{\varepsilon_t\}_{t\geq 1}$ mutually independent, $p_\star, q_\star, r_\star > 0$, and $a_\star \neq 0$. We assume zero mean only for notational convenience, the same calculations hold with affine offsets after centering. We write

$$P_t^\star := \mathrm{Var}(z_t).$$

Noisy-state predictive attenuation. Consider

$$q_{a,v}(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1}; ax_t, v), \qquad v > 0.$$

The corresponding expected NLL at a fixed time t is

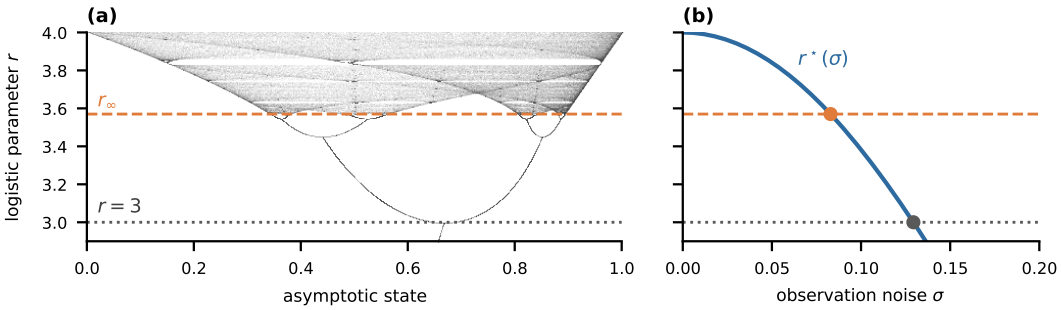
$$\mathcal{L}_{\mathrm{pred}}(a, v) = \frac{1}{2}\log(2\pi v) + \frac{1}{2v}\mathbb{E}_\star\left[(x_{t+1} - ax_t)^2\right],$$

where $\mathbb{E}_\star$ denotes expectation under the true data-generating process. For fixed a, the optimal variance is

$$v^\dagger(a) = \mathbb{E}_\star\left[(x_{t+1} - ax_t)^2\right].$$

Therefore minimizing the profiled (i.e., variance-optimized) NLL over a is equivalent to minimizing the mean squared prediction error. Hence the optimal multiplier is the classical population least-squares coefficient under errors-in-variables [45, 46]:

$$a^\dagger_{\mathrm{pred}} = \frac{\mathbb{E}_\star[x_{t+1}x_t]}{\mathbb{E}_\star[x_t^2]}.$$



Appendix Figure 1: **Errors-in-variables attenuation on the logistic map. (a)** Bifurcation diagram of the noise-free logistic map as a function of the parameter r. **(b)** Population noisy-state optimum $r^\star(\sigma)$ vs. observation-noise standard deviation. Increasing σ attenuates $r^\star$ through the onset of chaos ($r_\infty \approx 3.6$), the period-2 bifurcation ($r = 3$), and into the stable fixed-point regime ($r < 3$).

Using

$$x_t = z_t + \varepsilon_t, \qquad x_{t+1} = a_\star z_t + \xi_t + \varepsilon_{t+1},$$

and independence of $z_t, \xi_t, \varepsilon_t, \varepsilon_{t+1}$, we obtain

$$\mathbb{E}_\star[x_{t+1}x_t] = a_\star \operatorname{Var}(z_t) = a_\star P_t^\star,$$

and

$$\mathbb{E}_\star[x_t^2] = \operatorname{Var}(z_t) + \operatorname{Var}(\varepsilon_t) = P_t^\star + r_\star.$$

Thus

$$a^\dagger_{\text{pred}} = a_\star \frac{P_t^\star}{P_t^\star + r_\star}.$$

Since $P_t^\star > 0$ and $r_\star > 0$, the attenuation factor lies strictly in $(0, 1)$. Consequently,

$$|a^\dagger_{\text{pred}}| < |a_\star|.$$

In particular, an expanding latent direction is estimated as stable if

$$|a_\star| \frac{P_t^\star}{P_t^\star + r_\star} < 1.$$

If a single coefficient and shared variance are fitted over times $t = 1, \ldots, T-1$, the corresponding optimum is

$$a^\dagger_{\text{pred},T} = a_\star \frac{\sum_{t=1}^{T-1} P_t^\star}{\sum_{t=1}^{T-1}(P_t^\star + r_\star)},$$

again a strict attenuation whenever $r_\star > 0$.

Appx. Fig. 1 shows the same attenuation on the logistic map $z_{t+1} = r z_t(1 - z_t)$ [71]: treating noisy observations as states shifts the population optimum $r^\star(\sigma)$ through different dynamical regimes, crossing period-doubling and chaos onsets, so the fitted map can be non-chaotic even when the latent map is chaotic.

Innovation identifiability in the correctly specified family. Now consider the same linear-Gaussian process

$$z_1 \sim \mathcal{N}(0, p), \qquad z_{t+1} = a z_t + \xi_t, \qquad x_t = z_t + \varepsilon_t,$$

with

$$\xi_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, q), \qquad \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, r),$$

independent of z_1 and of each other with $p, q, r > 0$. Let $\theta = (a, q, r, p)$, and let $\mathbb{P}_\theta^{1:T}$ denote the joint distribution of $x_{1:T}$, with density $p_\theta(x_{1:T})$. Since the model is linear-Gaussian, the Kalman innovation factorization is the exact likelihood:

$$p_\theta(x_{1:T}) = p_\theta(x_1) \prod_{t=2}^{T} p_\theta(x_t \mid x_{1:t-1}) = p_\theta(x_1) \prod_{t=2}^{T} \mathcal{N}(x_t; \hat{x}^\theta_{t|t-1}, \Omega^\theta_t),$$

where $\hat{x}^{\theta}_{t|t-1}$ are the predicted observation means. The expected innovation NLL is therefore

$$\mathcal{L}_{\mathrm{KF}}(\theta) := \mathbb{E}_{\theta_\star}\left[-\log p_\theta(x_{1:T})\right],$$

Equivalently,

$$\mathcal{L}_{\mathrm{KF}}(\theta) = H(\mathbb{P}^{1:T}_{\theta_\star}) + \mathrm{KL}\left(\mathbb{P}^{1:T}_{\theta_\star} \,\|\, \mathbb{P}^{1:T}_{\theta}\right).$$

Since the KL term is nonnegative, the minimum value is attained at $\theta_\star$. Therefore every population minimizer must have the same observation law

$$\mathbb{P}^{1:T}_{\theta} = \mathbb{P}^{1:T}_{\theta_\star},$$

so it remains only to show identifiability.

Because the distributions are zero-mean Gaussian, equality of $\mathbb{P}^{1:T}_{\theta}$ and $\mathbb{P}^{1:T}_{\theta_\star}$ follows from the equality of their covariance matrices. For $T \geq 3$, the covariance matrix entries are:

$$\sigma_{11} := \mathrm{Cov}_\theta(x_1, x_1) = p + r,$$
$$\sigma_{21} := \mathrm{Cov}_\theta(x_2, x_1) = ap,$$
$$\sigma_{31} := \mathrm{Cov}_\theta(x_3, x_1) = a^2 p,$$

and

$$\sigma_{22} := \mathrm{Cov}_\theta(x_2, x_2) = a^2 p + q + r.$$

Since $a_\star \neq 0$ and $p_\star > 0$,

$$\sigma^\star_{21} = a_\star p_\star \neq 0.$$

Hence equality of the observation laws implies

$$a = \frac{\sigma_{31}}{\sigma_{21}} = \frac{\sigma^\star_{31}}{\sigma^\star_{21}} = a_\star.$$

Then

$$p = \frac{\sigma_{21}}{a} = \frac{\sigma^\star_{21}}{a_\star} = p_\star,$$

and

$$r = \sigma_{11} - p = \sigma^\star_{11} - p_\star = r_\star.$$

Finally,

$$q = \sigma_{22} - a^2 p - r = \sigma^\star_{22} - a^2_\star p_\star - r_\star = q_\star.$$

Therefore equality of the observation distributions implies

$$(a, q, r, p) = (a_\star, q_\star, r_\star, p_\star),$$

and the expected Kalman innovation NLL has the unique population-risk minimizer

$$(a^\dagger_{\mathrm{KF}}, q^\dagger_{\mathrm{KF}}, r^\dagger_{\mathrm{KF}}, p^\dagger_{\mathrm{KF}}) = (a_\star, q_\star, r_\star, p_\star).$$

A.4 When covariance contraction increases the expected NLL

The log-determinant term of the NLL favors smaller predictive covariances, but the Mahalanobis (data-fit) term can oppose this tendency. The following local calculation shows that, when the covariance is already too small relative to the prediction errors, further contraction increases the expected score.

To isolate a single contraction direction, fix a reference pair $\theta = (\theta_c, \theta_s)$. Consider Jacobians of the form

$$\boldsymbol{J}_{\theta_c,t}(s) = s\,\bar{\boldsymbol{J}}_{\theta_c,t}$$

along a chosen covariance direction, and write

$$\boldsymbol{A}_{\theta,t} := \boldsymbol{B}\bar{\boldsymbol{J}}_{\theta_c,t}\boldsymbol{\Sigma}_{t-1|t-1}\bar{\boldsymbol{J}}^\top_{\theta_c,t}\boldsymbol{B}^\top, \qquad \boldsymbol{C}_{\theta_s} := \boldsymbol{B}\boldsymbol{\Gamma}_{\theta_s}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}.$$

Assume $\boldsymbol{C}_{\theta_s} \succ 0$. Define

$$\boldsymbol{\Omega}_{\theta,t}(s) := s^2\boldsymbol{A}_{\theta,t} + \boldsymbol{C}_{\theta_s}, \qquad \boldsymbol{\Omega}^\star_t := \mathrm{Cov}(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}),$$

and

$$\mathcal{L}(s) := \frac{1}{2}\sum_{t=1}^{T}\left[\log\det\boldsymbol{\Omega}_{\theta,t}(s) + \mathrm{tr}\left(\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{\Omega}^\star_t\right)\right].$$

Lemma 1 (Anti-contraction along a covariance-rescaling path)**.** *Consider a local covariance-scaling path at fixed filtered covariances $\{\boldsymbol{\Sigma}_{t-1|t-1}\}_{t=1}^{T}$ and fixed innovation means. We characterize the partial derivative of the centered-covariance contribution to the expected innovation NLL with respect to a local Jacobian-rescaling direction (ignoring the recursion through all earlier EKF covariances). If there exists $s_0 > 0$ such that*

$$\boldsymbol{\Omega}_{\theta,t}(s_0) \preceq \boldsymbol{\Omega}_t^\star \qquad \forall t,$$

with

$$\boldsymbol{\Omega}_{\theta,t_\star}(s_0) \prec \boldsymbol{\Omega}_{t_\star}^\star$$

for at least one $t_\star$ satisfying $\boldsymbol{A}_{\theta,t_\star} \neq 0$, then

$$\frac{\partial}{\partial s}\mathcal{L}(s) < 0 \qquad \textit{for all } 0 < s \leq s_0.$$

Thus no local minimizer of $\mathcal{L}$ lies in $(0, s_0]$ of this contraction path.

Proof of Lemma 1. Since $\boldsymbol{\Sigma}_{t-1|t-1} \succeq 0$, we have $\boldsymbol{A}_{\theta,t} \succeq 0$. The assumption $\boldsymbol{C}_{\theta_s} \succ 0$ implies $\boldsymbol{\Omega}_{\theta,t}(s) \succ 0$ for all $s \geq 0$, so the inverse square roots below are well-defined.

Write $\theta = (\theta_c, \theta_s)$ and $\boldsymbol{\Omega}_t^\star := \mathrm{Cov}(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1})$. For each t,

$$\boldsymbol{\Omega}_{\theta,t}(s) = s^2 \boldsymbol{A}_{\theta,t} + \boldsymbol{C}_{\theta_s}, \qquad \frac{\partial}{\partial s}\boldsymbol{\Omega}_{\theta,t}(s) = 2s\boldsymbol{A}_{\theta,t}.$$

Differentiating $\mathcal{L}$ gives

$$\frac{\partial}{\partial s}\mathcal{L}(s) = \frac{1}{2}\sum_{t=1}^{T}\left(\mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{\Omega}'_{\theta,t}(s)\big) - \mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{\Omega}'_{\theta,t}(s)\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{\Omega}_t^\star\big)\right).$$

Substituting $\boldsymbol{\Omega}'_{\theta,t}(s) = 2s\boldsymbol{A}_{\theta,t}$ yields

$$\frac{\partial}{\partial s}\mathcal{L}(s) = s\sum_{t=1}^{T}\left(\mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{A}_{\theta,t}\big) - \mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{A}_{\theta,t}\boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{\Omega}_t^\star\big)\right).$$

Define

$$\boldsymbol{M}_t(s) := \boldsymbol{\Omega}_{\theta,t}(s)^{-1/2}\boldsymbol{A}_{\theta,t}\boldsymbol{\Omega}_{\theta,t}(s)^{-1/2} \succeq 0, \qquad \boldsymbol{Z}_t(s) := \boldsymbol{\Omega}_{\theta,t}(s)^{-1/2}\boldsymbol{\Omega}_t^\star\boldsymbol{\Omega}_{\theta,t}(s)^{-1/2}.$$

Using cyclicity of the trace,

$$\mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}^{-1}\boldsymbol{A}_{\theta,t}\big) = \mathrm{tr}\big(\boldsymbol{M}_t\big), \qquad \mathrm{tr}\big(\boldsymbol{\Omega}_{\theta,t}^{-1}\boldsymbol{A}_{\theta,t}\boldsymbol{\Omega}_{\theta,t}^{-1}\boldsymbol{\Omega}_t^\star\big) = \mathrm{tr}\big(\boldsymbol{M}_t\boldsymbol{Z}_t\big),$$

so

$$\frac{\partial}{\partial s}\mathcal{L}(s) = s\sum_{t=1}^{T}\mathrm{tr}\big(\boldsymbol{M}_t(s)(\boldsymbol{I} - \boldsymbol{Z}_t(s))\big).$$

Assume now that $0 < s \leq s_0$. Since $\boldsymbol{\Omega}_{\theta,t}(s) = s^2\boldsymbol{A}_{\theta,t} + \boldsymbol{C}_{\theta_s}$ is monotone increasing in s,

$$\boldsymbol{\Omega}_{\theta,t}(s) \preceq \boldsymbol{\Omega}_{\theta,t}(s_0) \preceq \boldsymbol{\Omega}_t^\star \qquad \forall t.$$

Therefore

$$\boldsymbol{Z}_t(s) = \boldsymbol{\Omega}_{\theta,t}(s)^{-1/2}\boldsymbol{\Omega}_t^\star\boldsymbol{\Omega}_{\theta,t}(s)^{-1/2} \succeq \boldsymbol{I},$$

so $\boldsymbol{I} - \boldsymbol{Z}_t(s) \preceq 0$. Because $\boldsymbol{M}_t(s) \succeq 0$, each summand is non-positive. Indeed, writing $\boldsymbol{I} - \boldsymbol{Z}_t = -\boldsymbol{N}_t$ with $\boldsymbol{N}_t \succeq 0$,

$$\mathrm{tr}\big(\boldsymbol{M}_t(\boldsymbol{I} - \boldsymbol{Z}_t)\big) = -\mathrm{tr}\big(\boldsymbol{M}_t\boldsymbol{N}_t\big) \leq 0,$$

since $\mathrm{tr}(\boldsymbol{AC}) \geq 0$ for positive semidefinite $\boldsymbol{A}, \boldsymbol{C}$.

For index $t_\star$, the assumption $\boldsymbol{\Omega}_{\theta,t_\star}(s_0) \prec \boldsymbol{\Omega}_{t_\star}^\star$ implies

$$\boldsymbol{\Omega}_{\theta,t_\star}(s) \preceq \boldsymbol{\Omega}_{\theta,t_\star}(s_0) \prec \boldsymbol{\Omega}_{t_\star}^\star,$$

hence $\boldsymbol{Z}_{t_\star}(s) \succ \boldsymbol{I}$. Since $\boldsymbol{A}_{\theta,t_\star} \neq 0$, we also have $\boldsymbol{M}_{t_\star}(s) \neq 0$. Therefore

$$\mathrm{tr}\big(\boldsymbol{M}_{t_\star}(s)(\boldsymbol{I} - \boldsymbol{Z}_{t_\star}(s))\big) < 0.$$

Summing over t yields

$$\frac{\partial}{\partial s}\mathcal{L}(s) < 0 \qquad \text{for all } 0 < s \leq s_0.$$

Hence no local minimizer can lie in $(0, s_0]$. □

For completeness, the conclusion also holds for biased predictions. With conditioning histories and predictions fixed, let $\boldsymbol{b}_t := \mathbb{E}[\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}] - \boldsymbol{B}\boldsymbol{z}_{t|t-1}$. The full expected Gaussian innovation NLL is

$$\mathcal{L}_{\text{full}}(s) = \mathcal{L}(s) + \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{b}_t^\top \boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{b}_t + \text{const.}$$

Increasing s increases or leaves unchanged the predictive variance in every direction, so the same fixed prediction bias receives a smaller or unchanged Mahalanobis penalty $\boldsymbol{b}_t^\top \boldsymbol{\Omega}_{\theta,t}(s)^{-1}\boldsymbol{b}_t$. The additional bias term is therefore non-increasing in s. Thus $\mathcal{L}'_{\text{full}}(s) \leq \mathcal{L}'(s) < 0$ for $0 < s \leq s_0$, i.e., further covariance contraction increases the expected NLL even when the fixed predictions are biased.

A.5 Kalman updates as adaptive anisotropic forcing

Isolating the optimization mechanism by which filtering alters chaotic Jacobian product chains connects the KAFFEE objective to work on exploding BPTT gradients and generalized teacher forcing (GTF) [34, 48]. Recent work in this direction extended GTF to project the network orthogonally toward the zero-error manifold [49], and interpreted identity teacher forcing as an intervention-based generalized-Bayes objective [50]. While GTF and error forcing modify the state recursion using a forcing-strength hyperparameter, KAFFEE instead determines the correction from the predicted covariance and observation-noise model through a Kalman update, without introducing a separate forcing-strength hyperparameter.

Consider GTF, where model predictions are combined with a data-inferred state using a scalar forcing strength. The Kalman update has a related form:

$$\boldsymbol{z}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})F_{\theta_c}(\boldsymbol{z}_{t-1|t-1}) + \boldsymbol{K}_t\boldsymbol{x}_t.$$

Here, the correction is determined by the predicted covariance and the observation-noise model. It can act differently in different latent directions, and incoming data can correct unobserved coordinates through their covariance with observed coordinates. This parallels the perspective in Pals et al. [51], where uncertainty-adaptive interpolation between model-predicted and data-inferred states is related to GTF. For the piecewise-affine transition models used in our experiments (AL-RNN and shPLRNN), this connection has a particularly simple local expression.

Proposition 2 (Filtering factor for piecewise-affine transitions)**.** *Consider the EKF recursion (Appx. B.1) with a piecewise-affine transition map, linear observations, and state-independent noise covariances with $\boldsymbol{\Lambda} \succ 0$. Fix the parameters, observations, and starting covariance $\boldsymbol{\Sigma}_{r|r}$ at step r. Suppose the filtered trajectory remains away from the ReLU-activation boundaries over the interval $r, \ldots, t$, such that sufficiently small perturbations of $\boldsymbol{z}_{r|r}$ preserve its sequence of affine regions $\{\mathbf{c}_k\}_{r:t}$ (Appx. C.1). Then*

$$\left.\frac{\partial \boldsymbol{z}_{t|t}}{\partial \boldsymbol{z}_{r|r}}\right|_{\boldsymbol{\Sigma}_{r|r}} = \boldsymbol{T}_t\boldsymbol{T}_{t-1}\cdots\boldsymbol{T}_{r+1}, \qquad \boldsymbol{T}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{B})\boldsymbol{J}_k,$$

where $\boldsymbol{J}_k = \nabla_{\boldsymbol{z}} F_{\theta_c}(\boldsymbol{z}_{k-1|k-1})$.

Proof. Within a fixed sequence of affine regions, each $\boldsymbol{J}_k$ is locally constant with respect to the state. Starting from the same covariance, the prediction

$$\boldsymbol{\Sigma}_{k|k-1} = \boldsymbol{J}_k\boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{J}_k^\top + \boldsymbol{\Gamma}$$

and the covariance update therefore give the same covariance sequence under the assumed sufficiently small perturbations of the starting mean. Consequently, the Kalman gains are also locally unchanged. Differentiating the update of the mean then gives

$$\left.\frac{\partial \boldsymbol{z}_{k|k}}{\partial \boldsymbol{z}_{r|r}}\right|_{\boldsymbol{\Sigma}_{r|r}} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{B})\boldsymbol{J}_k \left.\frac{\partial \boldsymbol{z}_{k-1|k-1}}{\partial \boldsymbol{z}_{r|r}}\right|_{\boldsymbol{\Sigma}_{r|r}}.$$

Applying this identity iteratively then proves the stated ordered product. □

For $\boldsymbol{B} = \boldsymbol{I}$ and a fixed scalar gain $\boldsymbol{K}_t = \alpha \boldsymbol{I}$, the update then becomes $(1-\alpha)F_{\theta_c}(\boldsymbol{z}_{t-1|t-1}) + \alpha \boldsymbol{x}_t$, with derivative $(1-\alpha)\boldsymbol{J}_t$, recovering the GTF damping factor of Hess et al. [34]. KAFFEE instead obtains the gain from the propagated covariance and learned noise model, without introducing a chosen separate scalar forcing strength. For example, in the scalar case (and with $B = 1$), we have $K_t = \Sigma_{t|t-1}/(\Sigma_{t|t-1} + \Lambda)$; so at fixed observation noise, a less certain prediction receives a larger observational correction. The general matrix gain extends this to anisotropic uncertainty. Although it is locally constant within the fixed activation sequence considered in the proof, it can change across time and across sequences through the covariance recursion and the regions visited by the filtered trajectory. Thus, with the "full" gain, the factor $\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B}$ can reduce amplification of state perturbations, but it need not contract every direction. Chaotic dynamics exhibit direction-dependent perturbation growth [18, 71], which motivates adaptive, anisotropic damping rather than a uniform scalar factor. Appx. D.3 also examines the empirical benefit of this approach on Lorenz-96.

The combined matrix $\boldsymbol{T}_t = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{J}_t$ can have a largest singular value below one even when $\boldsymbol{J}_t$ is expansive. If $\|\boldsymbol{T}_k\|_2 \leq \rho < 1$ throughout the interval, the proposition bounds the mean-to-mean sensitivity by ρ^{t-r}. However, while our proposition explains a connection to forcing methods, it does not guarantee bounded training gradients or preservation of autonomous dynamics. Moreover, the derivative here is with respect to the starting state at fixed parameters, while parameter changes still affect Jacobians, covariances, and the Kalman gains, and KAFFEE differentiates through these paths.

Beyond piecewise-affine transitions. Keeping the same observation and noise assumptions, for a sufficiently smooth transition model (for example, twice continuously differentiable), the one-step mean derivative at fixed incoming covariance additionally contains gain-variation terms:

$$\left.\frac{\partial \boldsymbol{z}_{t|t}}{\partial \boldsymbol{z}_{t-1|t-1}}\right|_{\boldsymbol{\Sigma}_{t-1|t-1}} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{J}_t + \sum_{j=1}^{N} (\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1})_j \frac{\partial \boldsymbol{k}_{t,j}}{\partial \boldsymbol{z}_{t-1|t-1}},$$

where $\boldsymbol{k}_{t,j}$ denotes column j of $\boldsymbol{K}_t$, and the derivatives on the right also hold the incoming covariance fixed. This follows from directly differentiating the mean update. Over multiple steps, the recurrent state comprises both the mean and covariance: an earlier mean perturbation may change a later covariance and hence a later mean, and the complete derivative includes these paths. The piecewise-affine case above avoids this complication because the covariance sequence does not change locally with the starting mean.

The ablations in Appx. D.3 test the contribution of differentiation through assimilation. On Lorenz-96, stopping gradients through the Kalman update worsens autonomous reconstruction despite identical forward updates at fixed parameters. Selected fixed scalar gains achieve comparable NLL but yield worse autonomous reconstruction than KAFFEE. These results support the practical value of learning through the Kalman update.

A.6 Particle-based inference and variational objectives

Our main-text volume-pressure argument is targeted at locally linearized Gaussian predictive densities. Sample/Particle-based inference and variational objectives [51, 75–77] can represent non-Gaussian state uncertainty without projecting the full distribution onto a single Gaussian. However, in high-dimensional chaotic settings, approximation error and computational cost incurred by particle degeneracy can still limit their applicability. Also, while this relaxes a specific approximation constraint, it does not by itself ensure that autonomous dynamics are preserved. Whether these alternatives mitigate the DPC gap at a practical computational cost thus remains to be established.

A.7 Computational cost and scaling

At a high level, KAFFEE inherits the scaling profile of EKF-style Gaussian filtering: it propagates one mean/covariance pair rather than a particle cloud or a large forecast ensemble. In our (dense) implementation, the learned shell covariances are diagonal, that is, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are parameterized by only $O(M+N)$ log-standard-deviation parameters rather than full $M \times M$ and $N \times N$ matrices. However, after those diagonal shells initialize the filter, the predictive and filtered state covariances produced by Jacobian-based propagation are transported as full dense matrices in latent space. Consequently, the dominant forward cost is the dense EKF recursion, not the shell parameterization itself: with latent dimension M and observation dimension N, the forward step is dominated by the Jacobian

propagation and Kalman update, with cost on the order of $O(M^3 + M^2N + N^3)$ per step when full latent covariance is retained. Note that for diagonal observation covariances, we can completely avoid the $O(N^3)$ inversion of the innovation covariance, via the Woodbury inversion lemma (at an additional $O(M^3)$ cost), which is practical for settings with high-D observations [20]. For a batched training window, the cost scales linearly in the batch size and loss horizon.

For smooth nonlinear transitions, the gain-variation terms discussed above matter mainly for the exact backward path rather than the forward EKF pass. In our implementation the differentiable EKF is optimized end-to-end, and automatic differentiation captures the dependence of these terms by differentiating through the Kalman gain and the Jacobian-based covariance propagation. The exact backward pass inherits second-order sensitivities of the transition map associated with the gain-variation, thus effectively requiring Hessian-vector product information (even when no dense Hessian tensor is formed explicitly). However, for the piecewise-linear (or piecewise-affine) latent transition models we use in our experiments (AL-RNN, shPLRNN), those second-order state derivatives vanish everywhere in state space, except on the boundaries of the affine subregions (whose configuration is expressed by a binary switching code depending on which ReLU units are active, see Appx. C.1 and Proposition 2). Thus, the main cost in that case is propagating and differentiating dense latent covariances.

Dense KAFFEE remains attractive in moderately high dimensions, but much larger latent states would likely require structured approximations and approximate Jacobian/backward passes, such as diagonal-, block-, or low-rank covariance [65, 78, 79]. Chaotic DA further motivates these reductions: in "perfect surrogate model" settings, Kalman error covariances concentrate on the unstable-neutral subspace [18, 22]. With additive model/process noise however, covariance in stable directions can be filled by the noise and remain bounded, which favors retaining extra directions rather than a purely unstable-subspace truncation [80]. We instantiate one such approximation for KAFFEE in Appx. D.5.

B KAFFEE algorithm and filtering equations

B.1 EKF Predict-update equations

For completeness, we provide the full recursive equations for the EKF utilized in the KAFFEE architecture. Let F_{θ_c} denote the autonomous transition map parameterized by the deterministic core, and let $\boldsymbol{\Gamma}_{\theta_s}$ and $\boldsymbol{\Lambda}_{\theta_s}$ denote the learned process and observation noise covariances parameterizing the stochastic shell. We assume a linear observation model $\boldsymbol{B}$.

Given a filtered state belief $\mathcal{N}(\boldsymbol{z}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1})$ at time $t-1$, the forward pass of KAFFEE performs the EKF's predict-update cycle at time t [28]:

1. Predict step (autonomous expansion). We first obtain the predictive density:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}) = \int p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}) p(\boldsymbol{z}_{t-1} \mid \boldsymbol{x}_{1:t-1}) d\boldsymbol{z}_{t-1}.$$

We apply the dynamics $p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$ and marginalize the previous latent $\boldsymbol{z}_{t-1}$. While the integral is intractable for nonlinear dynamics, the EKF proceeds by linearizing around $\boldsymbol{z}_{t-1|t-1}$. This results in a Gaussian approximation to $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1})$, where the deterministic core projects the state mean forward, while the local Jacobian $\boldsymbol{J}_{\theta_c,t} = \nabla_{\boldsymbol{z}} F_{\theta_c}(\boldsymbol{z}_{t-1|t-1})$ propagates the covariance:

$$\boldsymbol{z}_{t|t-1} = F_{\theta_c}(\boldsymbol{z}_{t-1|t-1}), \tag{5}$$

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{J}_{\theta_c,t}\, \boldsymbol{\Sigma}_{t-1|t-1}\, \boldsymbol{J}_{\theta_c,t}^\top + \boldsymbol{\Gamma}_{\theta_s}. \tag{6}$$

2. Update step (data assimilation). Next, we correct the state and covariance using the observations [15, 57]. This step (typically) contracts the predictive variance (Proposition 1).

Using $p(\boldsymbol{x}_t \mid \boldsymbol{z}_t, \boldsymbol{x}_{1:t-1}) = p(\boldsymbol{x}_t \mid \boldsymbol{z}_t)$ and $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1})$ obtained in the last step, we have the following expression for the joint predictive distribution $p(\boldsymbol{x}_t, \boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1})$:

$$\begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{x}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{z}_{t|t-1} \\ \boldsymbol{B}\boldsymbol{z}_{t|t-1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{t|t-1} & \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top \\ \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1} & \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s} \end{bmatrix} \right),$$

where we can read off the innovation covariance

$$\boldsymbol{\Omega}_t = \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}. \tag{7}$$

Given this joint distribution, we can apply standard Gaussian conditioning identities to obtain the filtered distribution $p(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})$, with mean and covariance given by:

$$\boldsymbol{z}_{t|t} = \boldsymbol{z}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1}), \tag{8}$$

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{\Sigma}_{t|t-1}. \tag{9}$$

Where $\boldsymbol{K}_t$, is the Kalman Gain, which determines the optimal blending weight between the prediction and the incoming noisy observation $\boldsymbol{x}_t$:

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top \boldsymbol{\Omega}_t^{-1}. \tag{10}$$

Equivalently, for numerical stability, one may use the symmetric Joseph form for the covariance:

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{\Sigma}_{t|t-1}(\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})^\top + \boldsymbol{K}_t\boldsymbol{\Lambda}_{\theta_s}\boldsymbol{K}_t^\top. \tag{11}$$

During training, the innovation $(\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1})$ and their covariance $\boldsymbol{\Omega}_t$ are passed directly to the innovation NLL objective (Eq. 4) to compute the (localized) loss before proceeding to $t+1$.

Interleaving prediction and assimilation prevents the innovation objective from scoring a single long open-loop covariance product. Under standard linear-Gaussian regularity conditions, this recursion approaches a stable equilibrium [81]. In the nonlinear setting, we use the same local mechanism as a stabilizing inductive bias, see also Proposition 1.

B.2 KAFFEE optimization details

We now proceed to describe additional training details of KAFFEE. The differentiable EKF described in Appendix B.1 is optimized end-to-end. Algorithm 1 then states the generic KAFFEE optimization loop: each step samples one or more truncated windows, propagates the predict–propagate–update recursion through every step, accumulates the innovation NLL, and backpropagates jointly through the core parameters θ and the shell parameters $(\boldsymbol{\Gamma}, \boldsymbol{\Lambda})$.

AL-RNN Jacobian. The closed-form piecewise-linear Jacobian of the AL-RNN is (cf. Eq. 12),

$$\frac{\partial F_{\boldsymbol{\theta}}}{\partial \boldsymbol{z}}(\boldsymbol{z}) = \boldsymbol{A} + \boldsymbol{W}\boldsymbol{D}(\boldsymbol{c}(\boldsymbol{z})), \qquad c_j(\boldsymbol{z}) = \mathbf{1}\{z_{M-P+j} > 0\}.$$

This Jacobian is used for covariance propagation, $G_{\tan}$, and Lyapunov spectra. For DynaMix experts we use the analogous per-expert Jacobian, with mixture weights detached as described in Appx. C.5.

Shell initialization and optimization details. KAFFEE jointly optimizes the AL-RNN core and diagonal shell from the AL-RNN checkpoint with the initialization embedding $\boldsymbol{E}$ fixed after deterministic training. The diagonal noise shell is initialized from the same-seed AL-RNN's teacher-forcing residual scale (in standardized observation coordinates). Writing $\hat{\boldsymbol{r}}_0 \in \mathbb{R}_+^N$ for the coordinate-wise standard deviation of the teacher-forcing residuals on the noisy training observations, we use a fixed $90/10$ variance split ($\alpha = 0.9$ in Algorithm 1)

$$\boldsymbol{\Lambda}_0 = \operatorname{diag}\left((\sqrt{0.9}\,\hat{\boldsymbol{r}}_0)^2\right), \qquad \boldsymbol{\Gamma}_0 = \bar{q}_0^2\boldsymbol{I}_M, \qquad \bar{q}_0 := \frac{1}{N}\sum_{j=1}^{N}\sqrt{0.1}\,\hat{r}_{0,j}.$$

At each gradient step we sample 16 contiguous windows of total length 40 transitions (i.e., 41 observations) from the standardized noisy training trajectory, propagate the EKF through the full window, and include only the final 32 innovation terms in the loss after an 8-step burn-in. Optimization uses Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [82] with learning rate 5×10^{-3} decayed to 10^{-5} via cosine annealing, gradient clipping 2.0, and 150 joint iterations. Among tracked optimization iterations, we select the model with lowest $D_{\text{stsp}}^{\text{stoch}}$ against a 10^4-step noisy validation rollout.

Windowed truncation and covariance burn-in. During training, KAFFEE is optimized using BPTT over contiguous truncated sequences. Consequently, at the start of each window, the filter mean is anchored to the observed state and the predictive covariance is reinitialized from the "static" shell parameters. If those first few innovation terms enter the loss immediately, the optimizer can partly improve the objective by exploiting transient covariance that exists only because the window was truncated there. The goal is to couple predictive uncertainty to the learned transition map through

Jacobian-propagated covariance, so the loss-contributing portion of each window should reflect covariance propagation that has already had a short opportunity to align with the local dynamics.

For that reason, KAFFEE runs use a short covariance burn-in. We sample windows of total length $L_{\text{burn}}+L+1$ observations with burn-in phase L_{burn} and loss horizon L, propagate the filter recursion across all steps, but exclude the first L_{burn} innovation terms from the loss. Writing the per-step innovation contribution as

$$\ell_t = \frac{1}{2}\Big[\log\det \boldsymbol{\Omega}_t + \boldsymbol{v}_t^\top \boldsymbol{\Omega}_t^{-1}\boldsymbol{v}_t + N\log(2\pi)\Big],$$

the loss on one sampled window becomes

$$\mathcal{L}_{\text{EKF}}^{\text{score}} = \frac{1}{L}\sum_{t=L_{\text{burn}}+1}^{L_{\text{burn}}+L} \ell_t.$$

The covariance burn-in phase alleviates the optimizer's incentive to fit a covariance transient. To match schedules, the open-loop NLL control mirrors the same sampled window length. In practice, L_{burn} should be long enough that the predictive covariance is no longer dominated by its initialization. In our experiments, we set $L_{\text{burn}} = 8$ throughout.

C Experimental Details

This section consolidates model, dataset, metric, and optimization protocol details required to reproduce Tables 1 and 3.

C.1 AL-RNN architecture and deterministic training

All baselines in Table 1 share the same AL-RNN core and deterministic pretraining protocol. AL-RNNs were introduced for interpretable DSR through sparse use of nonlinear units, enabling parsimonious piecewise linear representations and compact symbolic descriptions of the learned dynamics [53]. They have also been used for analyzing the role of nonlinearity in sequence modeling [83], continual DSR [84], and learning control policies from reconstructed dynamics [85].

The AL-RNN's deterministic transition for $\boldsymbol{z}_t \in \mathbb{R}^M$ is

$$\boldsymbol{z}_{t+1} = F_{\boldsymbol{\theta}}(\boldsymbol{z}_t) = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{W}\,\boldsymbol{\phi}^*(\boldsymbol{z}_t) + \boldsymbol{h}, \tag{12}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times M}$, $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ are diagonal and full matrices, respectively. $\boldsymbol{\phi}^*$ applies a ReLU gating to the last P coordinates only, and the first $M-P$ coordinates are linear. Equivalently, writing the binary switching code $\boldsymbol{c}_t \in \{0,1\}^P$ and diagonal gate matrix

$$\boldsymbol{D}(\boldsymbol{c}_t) = \operatorname{diag}\big(\underbrace{1,\ldots,1}_{M-P}, c_{t,1},\ldots,c_{t,P}\big),$$

the transition can be written as

$$\boldsymbol{z}_{t+1} = \big(\boldsymbol{A} + \boldsymbol{W}\boldsymbol{D}(\boldsymbol{c}_t)\big)\boldsymbol{z}_t + \boldsymbol{h}. \tag{13}$$

We use latent dimension $M = 100$, $N = 20$, and $P = 50$ ReLU-gates. Observations are always the first N latent coordinates, so the observation operator is fixed to

$$\boldsymbol{B} = [\boldsymbol{I}_N\ \boldsymbol{0}] \in \mathbb{R}^{N\times M}, \qquad \boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{z}_t. \tag{14}$$

To initialize a rollout from the first observation $\boldsymbol{x}_1 \in \mathbb{R}^N$, a learned linear embedding $\boldsymbol{E} \in \mathbb{R}^{M\times N}$ is applied to overwrite the observed coordinates [53],

$$\boldsymbol{z}_1 \leftarrow \boldsymbol{E}\boldsymbol{x}_1, \qquad \boldsymbol{z}_1 \leftarrow \boldsymbol{z}_1 + \boldsymbol{B}^\top(\boldsymbol{x}_1 - \boldsymbol{B}\boldsymbol{z}_1).$$

Deterministic prior pretraining. Each deterministic AL-RNN core is pretrained on the standardized noisy Lorenz-96 training observations described in Appx. C.3. Training uses BPTT with sparse identity teacher forcing and forcing interval $\tau = 20$ [48, 53]. Because the observation map is the

Algorithm 1 KAFFEE optimization protocol

Input: observations $\boldsymbol{x}_{1:T}$, transition map F_θ, observation matrix $\boldsymbol{B}$, deterministic prior model $\theta_c^{(0)}$, covariance burn-in length L_{burn}, scored length L, variance split $\alpha \in (0,1)$ (here, $\alpha = 0.9$ throughout), AL-RNN embedding matrix $\boldsymbol{E}$.

Initialization. Compute one-step teacher-forcing residual scales $\hat{\boldsymbol{r}}_0 \in \mathbb{R}_+^N$ on noisy training trajectory under $\theta_c^{(0)}$. Set

$$\boldsymbol{\Lambda}_0 = \operatorname{diag}\left(\alpha\, \hat{\boldsymbol{r}}_0^{\,2}\right), \qquad \boldsymbol{\Gamma}_0 = \bar{q}_0^2 \boldsymbol{I}_M, \qquad \bar{q}_0 = \frac{1}{N}\sum_{j=1}^{N} \sqrt{1-\alpha}\, \hat{r}_{0,j}.$$

Set $\theta_c \leftarrow \theta_c^{(0)}$, $\theta_s \leftarrow (\boldsymbol{\Gamma}_0, \boldsymbol{\Lambda}_0)$, and initialize $\theta \leftarrow (\theta_c, \theta_s)$.

for each optimization step **do**

  Sample a mini-batch $\mathcal{B}$ of windows $\boldsymbol{x}_{t_0:t_0+L_{\text{burn}}+L}$.

  Form the current diagonal shell covariances $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$.

  Set $\mathcal{L}_{\text{batch}} \leftarrow 0$.

  for each window in $\mathcal{B}$ **do**

    Initialize filtered belief: In case of using the AL-RNN as deterministic prior, set $\bar{\boldsymbol{z}}_{t_0} = \boldsymbol{E}\boldsymbol{x}_{t_0}$ using the fixed embedding inherited from deterministic training (Appx. C.1). Initialize the filter through the observation matrix and initialize the covariance from the current learned process covariance:

$$\boldsymbol{z}_{t_0|t_0} = \bar{\boldsymbol{z}}_{t_0} + \boldsymbol{B}^\top(\boldsymbol{x}_{t_0} - \boldsymbol{B}\bar{\boldsymbol{z}}_{t_0}), \qquad \boldsymbol{\Sigma}_{t_0|t_0} = \boldsymbol{\Gamma}.$$

    for $t = t_0+1, \ldots, t_0 + L_{\text{burn}} + L$ **do**

      Predict:

$$\boldsymbol{z}_{t|t-1} = F_\theta(\boldsymbol{z}_{t-1|t-1}), \qquad \boldsymbol{J}_t = \frac{\partial F_\theta}{\partial \boldsymbol{z}}(\boldsymbol{z}_{t-1|t-1}).$$

      Propagate covariance:

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{J}_t \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{J}_t^\top + \boldsymbol{\Gamma}.$$

      Form innovation:

$$\boldsymbol{v}_t = \boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_{t|t-1}, \qquad \boldsymbol{\Omega}_t = \boldsymbol{B}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top + \boldsymbol{\Lambda}.$$

      if $t > t_0 + L_{\text{burn}}$ **then**

        Accumulate innovation NLL

$$\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \frac{1}{2|\mathcal{B}|L}\left[\log\det \boldsymbol{\Omega}_t + \boldsymbol{v}_t^\top \boldsymbol{\Omega}_t^{-1} \boldsymbol{v}_t + N\log(2\pi)\right].$$

      end if

      Update:

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{B}^\top \boldsymbol{\Omega}_t^{-1},$$

$$\boldsymbol{z}_{t|t} = \boldsymbol{z}_{t|t-1} + \boldsymbol{K}_t \boldsymbol{v}_t, \qquad \boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})\boldsymbol{\Sigma}_{t|t-1}(\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{B})^\top + \boldsymbol{K}_t\boldsymbol{\Lambda}\boldsymbol{K}_t^\top.$$

    end for

  end for

  Backpropagate $\mathcal{L}_{\text{batch}}$ through the predict–propagate–update computation and update θ.

end for

identity projection onto the first N coordinates, this is identity teacher forcing (ITF): at forcing times $\mathcal{T}_\tau = \{t : t \equiv 0 \pmod{\tau},\ t > 0\}$ we apply

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t + \boldsymbol{B}^\top(\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_t),$$

which overwrites the observed coordinates and leaves the remaining coordinates untouched. Between forcing times the model is free-running. Training minimizes next-step MSE on the observed coordinates along these interleaved rollouts,

$$\mathcal{L}_{\mathrm{ITF}} = \frac{1}{L_{\mathrm{BPTT}} N} \sum_{t=1}^{L_{\mathrm{BPTT}}} \|\boldsymbol{B}\boldsymbol{z}_{t+1} - \boldsymbol{x}_{t+1}\|_2^2, \qquad L_{\mathrm{BPTT}} = 50,$$

using mini-batches of 16 contiguous windows, 50 batches per epoch, and $2 \cdot 10^3$ epochs on the $T_{\mathrm{train}} = 8 \cdot 10^4$ time steps long training trajectory. Optimization uses RAdam with an exponential learning-rate decay from 10^{-3} to 10^{-5} [86]. Initialization follows [53]: $\boldsymbol{h} = \boldsymbol{0}$, the entries of $\boldsymbol{W}$ are i.i.d. samples drawn from $\mathcal{N}(0, 0.01)$, the diagonal of $\boldsymbol{A}$ is taken from a normalized random positive-definite matrix, and the embedding $\boldsymbol{E}$ is initialized entry-wise from $\mathcal{U}(-1/\sqrt{N}, 1/\sqrt{N})$. Table 1 reports 20 independent AL-RNNs trained with this same protocol. Every probabilistic method in this table inherits the corresponding deterministic core and initializer $\boldsymbol{E}$.

C.2 Probabilistic baselines

For Lorenz-96, the matched controls use the same AL-RNN initializations and the same diagonal noise parametrization and initialization as KAFFEE. For shell training via BPTT, KAFFEE and all controls use Adam with learning rate 5×10^{-3} cosine-annealed to 10^{-5}, gradient clipping 2.0, and 150 joint iterations (Appx. B.2). Each sampled window contains 40 transitions (and 41 observations): 8 transitions are propagated for covariance burn-in and the final 32 steps are scored. The EKF covariance update uses the Joseph form (Eq. 11).

Open-loop objectives. For a training sequence with origin t_0, the two open-loop baselines in Table 1 propagate $\boldsymbol{z}_{t|t_0} = F_{\theta_c}(\boldsymbol{z}_{t-1|t_0})$ and $\boldsymbol{\Sigma}_{t|t_0}$ without intermediate assimilation. They then score the Gaussian open-loop NLL $-\log \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{B}\boldsymbol{z}_{t|t_0}, \boldsymbol{B}\boldsymbol{\Sigma}_{t|t_0}\boldsymbol{B}^\top + \boldsymbol{\Lambda}_{\theta_s}\right)$. The two open-loop methods differ only in their covariance recursion: *tied propagation* (core collapse) uses the Jacobian recursion (Eq. 2); *no propagation* (noise masking) replaces this by the state-invariant recursion $\boldsymbol{\Sigma}_{t+1|t_0} = \boldsymbol{\Sigma}_{t|t_0} + \boldsymbol{\Gamma}_{\theta_s}$, so the only path through which the core enters the loss is the predicted mean.

Filtered no covariance propagation. This control instantiates the filtered *no-propagation* (blind uncertainty) control. The objective and optimization schedule are identical to KAFFEE. The only difference is the covariance recursion: $\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Gamma}_{\theta_s}$ replaces the Jacobian-propagated recursion of Eq. 2, so the core no longer enters the predictive covariance. The mean is still predicted by the nonlinear core transition and corrected using the Kalman update. We can see this as running the regular EKF but using an identity approximation for the local Jacobians.

Frozen-core shell calibration. This row in Table 1 quantifies how much of the improvement is attainable by post-hoc noise calibration alone. We freeze the AL-RNN prior and optimize only the diagonal stochastic shell $(\boldsymbol{\Gamma}_{\theta_s}, \boldsymbol{\Lambda}_{\theta_s})$ under the EKF innovation NLL. By construction the shell-off core metrics are identical to the deterministic AL-RNN prior. This control therefore shows how much filtered probabilistic and stochastic dynamical performance is achievable without ever updating the deterministic core.

C.3 Lorenz-96 dataset

For Table 1 we use the $N = 20$-dimensional Lorenz-96 system [29]:

$$\frac{\mathrm{d}z_i}{\mathrm{d}t} = (z_{i+1} - z_{i-2})z_{i-1} - z_i + F, \qquad i = 1, \ldots, 20,$$

with forcing $F = 8$ and discretization interval $\Delta t = 0.04$. We denote the corresponding discrete-time latent state by $\boldsymbol{z}_t \in \mathbb{R}^{20}$. Let $\boldsymbol{\mu}_{\mathrm{train}} \in \mathbb{R}^{20}$ and $\boldsymbol{s}_{\mathrm{train}} \in \mathbb{R}^{20}_+$ denote the coordinate-wise mean and standard deviation of the noisy training trajectory in raw coordinates. The models are trained on trajectories generated in raw coordinates and subsequently standardized,

$$\tilde{\boldsymbol{z}}_t = \mathrm{diag}(\boldsymbol{s}_{\mathrm{train}})^{-1}(\boldsymbol{z}_t^{\mathrm{raw}} - \boldsymbol{\mu}_{\mathrm{train}}), \qquad \tilde{\boldsymbol{x}}_t = \mathrm{diag}(\boldsymbol{s}_{\mathrm{train}})^{-1}(\boldsymbol{x}_t^{\mathrm{raw}} - \boldsymbol{\mu}_{\mathrm{train}}).$$

The process noise σ_{proc} is defined on a standardized scale. Concretely, we first fix reference coordinate scales $\boldsymbol{s}_{\text{ref}} \in \mathbb{R}^{20}_{+}$ from a long clean run and then simulate the raw stochastic dynamics as a Runge–Kutta 4-th order drift step plus Euler–Maruyama diffusion,

$$\boldsymbol{z}^{\text{raw}}_{t+1} = \text{RK4}_{\Delta t}(\boldsymbol{z}^{\text{raw}}_t) + \sigma_{\text{proc}} \sqrt{\Delta t}\ \text{diag}(\boldsymbol{s}_{\text{ref}})\boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{20}).$$

Thus σ_{proc} specifies the diffusion magnitude relative to a fixed standardized scale, while the numerical integration itself is carried out in raw coordinates. After standardization by $\boldsymbol{s}_{\text{train}}$, the resulting per-step diffusion in standardized coordinates is therefore approximately $\sigma_{\text{proc}}\sqrt{\Delta t}$ per coordinate.

Observation noise is added separately, after latent simulation, in raw coordinates:

$$\boldsymbol{x}^{\text{raw}}_t = \boldsymbol{z}^{\text{raw}}_t + \sigma_{\text{obs}}\ \text{diag}(\boldsymbol{s}_{\text{latent}})\boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{20}).$$

Because $\boldsymbol{s}_{\text{train}}$ is the coordinate-wise standard deviation of these noisy observations, it includes the observation-noise variance. After z-scoring, the observation-noise standard deviation is therefore slightly below the generation coefficient $\sigma_{\text{obs}} = 0.1$ (in our generated data, 0.0995). s_{latent} is the coordinate-wise standard deviation of the simulated latent trajectory (before adding observation noise). Table 1 quotes these generation coefficients $\sigma_{\text{obs}} = \sigma_{\text{proc}} = 0.1$. All models (AL-RNN, KAFFEE, probabilistic baselines) train on these standardized noisy observations.

C.4 Metric computation and aggregation

For the predictive metrics in Table 1, the forecast horizon used is a quarter of a Lyapunov time, i.e. $\frac{1}{4}\tau_\lambda = 0.25\lambda^{core\,-1}_{1,\text{ref}}$:

$$H = \left\lfloor \frac{0.25}{\lambda^{\text{core}}_{1,\text{ref}}\,\Delta t} \right\rfloor = 4 \text{ steps},$$

where $\lambda^{\text{core}}_{1,\text{ref}}$ is the leading Lyapunov exponent of the ground truth deterministic dynamics. To evaluate the local predictive metrics, each seed/ensemble cohort first averages its predictive metrics over evenly spaced forecast starts, and the table medians and robust SE values are then formed across those independent runs. Throughout, we use $K = 200$ ensemble members and an observation-noise floor of 10^{-3} for the standard deviation inside the Gaussian likelihood.

NLL. For each forecast start, all ensemble members share the latent state initialized from the noisy observation at that origin. We autoregressively propagate this ensemble forward H steps without assimilating intermediate observations. At every $h = 1, \ldots, H$ we compute the ensemble mean $\boldsymbol{\mu}_h$ and standard deviation $\boldsymbol{\sigma}_h$ over the ensemble members and evaluate the per-step Gaussian NLL against the noisy reference $\boldsymbol{y}_h$ by

$$\text{NLL}(\boldsymbol{y}_h; \boldsymbol{\mu}_h, \boldsymbol{\sigma}_h) = \frac{1}{N}\sum_{j=1}^{N} \frac{1}{2}\left[\log(2\pi\sigma^2_{h,j}) + \frac{(y_{h,j} - \mu_{h,j})^2}{\sigma^2_{h,j}}\right].$$

The reported NLL is the path mean over horizon H, averaged over 100 forecast starts.

CRPS. The CRPS is evaluated at the endpoint of the bounded horizon H, using 10 forecast starts. For the reference observation $\boldsymbol{y}$ at the endpoint and the corresponding ensemble $\{\boldsymbol{x}^{(k)}\}_{k=1}^{K}$, we compute the marginal ensemble CRPS

$$\text{CRPS}(\boldsymbol{y}, \{\boldsymbol{x}^{(k)}\}) = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{K}\sum_{k=1}^{K}|x^{(k)}_j - y_j| - \frac{1}{2K^2}\sum_{k=1}^{K}\sum_{\ell=1}^{K}|x^{(k)}_j - x^{(\ell)}_j|\right].$$

The reported entry is the average of this endpoint CRPS over the 10 starts.

Dynamical grounding correlation $G_{\tan}$. To measure whether a model's local uncertainty is dynamically grounded, this metric correlates the coordinate-wise one-step predictive variance growth against the local tangent expansion of the true system. We use $G_{\tan}$ as a controlled probabilistic DSR certification metric: it requires access to the true local Jacobians and is therefore not directly available on real-world datasets. In such settings, analogous diagnostics would need to rely on proxy tangent estimates, withheld-observation spread growth, or innovation residual checks rather than oracle Jacobians.

Let $\boldsymbol{\Sigma}^m_{t|t}$ be the filtered/posterior covariance of method/model m, and let

$$\boldsymbol{\Omega}^m_{t|t} := \boldsymbol{B}\boldsymbol{\Sigma}^m_{t|t}\boldsymbol{B}^\top$$

be its observed-space posterior covariance. Let $\boldsymbol{J}^m_t \in \mathbb{R}^{M\times M}$ be the Jacobian used for that model's covariance propagation, evaluated at the filtered mean, and let $\boldsymbol{J}^\star_t \in \mathbb{R}^{N\times N}$ be the reference one-step Jacobian at the corresponding true state. The filter's predict step still adds process noise $\boldsymbol{\Gamma}$ (Eq. 2), but for $G_{\tan}$ we only use the noise-free pushforward of the posterior. For the model this (observation-space) pushforward is $\boldsymbol{B}\boldsymbol{J}^m_t\boldsymbol{\Sigma}^m_{t|t}\boldsymbol{J}^{m\top}_t\boldsymbol{B}^\top$, and for the reference, the observed posterior is pushed forward by the reference Jacobians, $\boldsymbol{J}^\star_t\boldsymbol{\Omega}^m_{t|t}\boldsymbol{J}^{\star\top}_t$. We define the model's predictive log covariance growth of observed coordinate j at time t as

$$\Delta u^m_{j,t} := \log\left[\boldsymbol{B}\boldsymbol{J}^m_t\boldsymbol{\Sigma}^m_{t|t}\boldsymbol{J}^{m\top}_t\boldsymbol{B}^\top\right]_{jj} - \log\left[\boldsymbol{\Omega}^m_{t|t}\right]_{jj},$$

and the reference tangent log-growth as

$$\Delta u^{\star,m}_{j,t} := \log\left[\boldsymbol{J}^\star_t\boldsymbol{\Omega}^m_{t|t}\boldsymbol{J}^{\star\top}_t\right]_{jj} - \log\left[\boldsymbol{\Omega}^m_{t|t}\right]_{jj}.$$

The first quantity is the variance growth assigned by the model; the second one is the variance growth that the true local tangent dynamics would assign to the same posterior. We then report the rank correlation of these two quantities:

$$G_{\tan}(m) := \mathrm{Spearman}_{(j,t)}\left(\Delta u^m_{j,t}, \Delta u^{\star,m}_{j,t}\right), \tag{15}$$

pooled over observed coordinates and time. Since we subtract the posterior log variance baseline $\log(\boldsymbol{\Omega}^m_{t|t})_{jj}$ on both sides, $G_{\tan}$ focuses on local variance growth, not on absolute magnitude. If either growth is constant (in our implementation, if either pooled series has standard deviation below 10^{-5}), the correlation is defined to be 0. This corresponds to the "no-propagation" case: there, the covariance propagation map is the identity (not the core's transition Jacobian), so $\Delta u^m_{j,t} = 0$ at every coordinate and time.

Unless otherwise stated, $G_{\tan}$ denotes the one-step propagated variance growth correlation. It is computed after a 200-step warm-up and then pooled over the following $4 \cdot 10^3$ prediction/update steps.

For the forecast-skill spectrum in Appx. D.2, we also use an H-step version, denoted $G^{(H)}_{\tan}$. Here the filter assimilates only every H steps. From the filtered covariance at assimilation, the model covariance is pushed forward by its covariance propagation map (i.e., Jacobian for tied propagation, identity for no-propagation models) at each forecast state, and the reference covariance by the true tangent map at the true state, for every lead time $h = 1, \ldots, H$. We do not add process noise $\boldsymbol{\Gamma}$ between the assimilations (this is for computing $G^{(H)}_{\tan}$; the filter continues to add process noise when propagating the predictive covariance). The subtracted baseline is the filtered variance from the last assimilation, and the correlation pools these lead times over observed coordinates and assimilation cycles.

Stochastic dynamical fidelity. For shell-on stochastic evaluation, we initialize the model rollout on a noisy state from held-out data, then score a total length of $T = L = 5 \cdot 10^4$ time steps after discarding the first 10^3 steps to account for a potential transient phase. The metrics compare those rollouts with stored reference trajectories of the same length. For the three-dimensional systems of the DynaMix experiments (and the additional Lorenz-63 ablation, Appx. D.1), the state-space divergence is the histogram KL divergence

$$D^{\mathrm{stoch}}_{\mathrm{stsp}} = \sum_b p_{\mathrm{ref}}(b) \log \frac{p_{\mathrm{ref}}(b)}{p_{\mathrm{gen}}(b)},$$

using 30 bins per coordinate. For Lorenz-96, the exponential scaling of bins with data dimension renders this histogram approach infeasible, so density estimation is executed instead by Gaussian mixture models [34, 52, 53]. The reference and generated distributions are approximated by isotropic Gaussian mixtures centered on the respective trajectories in standardized observation coordinates, denoted by $\boldsymbol{x}^{\mathrm{ref}}$ and $\boldsymbol{x}^{\mathrm{gen}}$:

$$p_{\mathrm{ref}}(\boldsymbol{x}) \approx \frac{1}{T}\sum_{t=1}^{T}\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}^{\mathrm{ref}}_t, \boldsymbol{\Sigma}), \qquad p_{\mathrm{gen}}(\boldsymbol{x}) \approx \frac{1}{L}\sum_{\ell=1}^{L}\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}^{\mathrm{gen}}_\ell, \boldsymbol{\Sigma}), \tag{16}$$

and the KL divergence between them is estimated via:

$$\tilde{D}_{\text{stsp}}\big(p_{\text{ref}}(\boldsymbol{x}),\, p_{\text{gen}}(\boldsymbol{x})\big) \approx \frac{1}{n}\sum_{i=1}^{n} \log \frac{\frac{1}{T}\sum_{t=1}^{T} \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{x}_t^{\text{ref}}, \boldsymbol{\Sigma})}{\frac{1}{L}\sum_{\ell=1}^{L} \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{x}_\ell^{\text{gen}}, \boldsymbol{\Sigma})}, \tag{17}$$

using n Monte Carlo samples $\boldsymbol{x}_i \sim p_{\text{ref}}$ [87]. The covariance $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$ is a scaled identity, and for compatibility with Hess et al. [34], Brenner et al. [53] we set $\sigma^2 = 1$. In our implementation $n = 10^3$ MC draws.

The unstable-volume growth rate of the stochastic dynamics and its error are given by the sum of positive Lyapunov exponents λ_i^{stoch}:

$$h_\lambda^{\text{stoch}} := \sum_i \max(\lambda_i^{\text{stoch}}, 0), \qquad |\Delta h_\lambda^{\text{stoch}}| := |\hat{h}_\lambda^{\text{stoch}} - h_{\lambda,\text{ref}}^{\text{stoch}}|,$$

using QR re-orthonormalization on the (shell-on) one-step Jacobian recursion after a 10^3-step burn-in and $3 \cdot 10^4$ scored transitions to estimate Lyapunov spectra [88, 89]. Observation noise is excluded from this tangent map, so we only evaluate Jacobians at process-noise-perturbed states. The reference spectra are obtained from the process-noisy true maps along the same latent reference.

Deterministic dynamical fidelity. To evaluate the autonomous core we initialize the deterministic rollout from a clean (noise-free) state, and apply the same rules (rollout lengths, transient burn-in) as for the stochastic dynamical fidelity, using the noise-free ground truth. The autonomous deterministic versions of the state space divergence $D_{\text{stsp}}^{\text{core}}$ and unstable-volume growth rate h_λ^{core} are evaluated analogously, by stripping the shell, i.e. generating rollouts only using the autonomous transition map.

To evaluate spectral overlap, we use the mean Hellinger distance between the per-coordinate power spectra, D_H^{core}, after each signal is centered, standardized, transformed with a real Fourier transform, smoothed by a Gaussian kernel of width 20 (chosen according to prior literature [34, 53]), and renormalized to unit mass. Writing the smoothed normalized spectra as $p_j(\omega)$ and $q_j(\omega)$, we report

$$D_H^{\text{core}} = \frac{1}{N}\sum_{j=1}^{N} \sqrt{1 - \sum_{\omega} \sqrt{p_j(\omega) q_j(\omega)}}.$$

Aggregation. We first compute every metric per seed or per disjoint ensemble cohort and then report the median together with the robust standard error of the median,

$$\text{SE}_{\text{median}} \approx 0.929\, \frac{\text{IQR}}{\sqrt{n}}.$$

For the Lorenz-96 experiments (Table 1) we use 20 independent seeds. For DynaMix (Table 3), we use 20 seeds as well on each of the 13 systems except the Dadras system. Here, since open-loop tied diverged for 3 seeds, we dropped those seeds for every method.

C.5 DynaMix architecture and probabilistic adaptation protocol

This section records the complete numerical results and additional details of the DynaMix benchmark experiment in Section 5.2. We do not re-state the full DynaMix architecture; for that we refer to Hemmer and Durstewitz [23]. The purpose here is to state the state-space interpretation of DynaMix used in the benchmark, the shell parameterization, the Jacobian construction, the KAFFEE protocol, and the definitions of the applied baselines.

State-space model form of DynaMix. DynaMix is a DSR foundation model, in the sense that its zero-shot dynamics are not supposed to (only) constitute accurate forecasts, but also capture the long-term structure underlying the (chaotic) data-generating system [4, 23]. While DynaMix is aligned with deterministic DSR, our present work emphasizes the need for uncertainty-aware, probabilistic DSR (foundation) models. To pursue this with DynaMix, we apply KAFFEE to embed it into a Bayesian filter/state-space model framework. In particular, we treat DynaMix as a pretrained latent transition model with fixed context.

Let $\boldsymbol{z}_t \in \mathbb{R}^M$ denote the latent state, $\boldsymbol{x}_t \in \mathbb{R}^N$ the observed state, and ctx the context supplied to the pretrained model. This context consists of a reference sequence of observations from the target

system, from which DynaMix is asked to zero-shot the underlying dynamics. The transition is written as

$$\boldsymbol{z}_{t+1} = F_{\text{DynaMix}}(\boldsymbol{z}_t, \text{ctx}, \boldsymbol{\varepsilon}_t^{\text{gate}}), \qquad \boldsymbol{\varepsilon}_t^{\text{gate}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\text{gate}}),$$

with DynaMix's mixture-of-experts form

$$F_{\text{DynaMix}}(\boldsymbol{z}_t, \text{ctx}) = \sum_{k=1}^{E} w_{t,k} f_k(\boldsymbol{z}_t), \qquad \boldsymbol{w}_t = g_\psi(\text{ctx}, \boldsymbol{z}_t, \boldsymbol{\varepsilon}_t^{\text{gate}}),$$

where $f_k : \mathbb{R}^M \to \mathbb{R}^M$ denotes the k-th (AL-RNN) expert's transition map, and $w_{t,k}$ is its state- and context-dependent mixture weight. The gating network g_ψ combines a convolutional neural network encoding of the context trajectory with the current latent state via a state attention mechanism to produce these expert-weights for the E experts on a simplex $\boldsymbol{w}_t \in \Delta^{E-1}$. This state attention mechanism contains an internal diagonal Gaussian perturbation ("exploration noise") $\boldsymbol{\varepsilon}_t^{\text{gate}}$.

Gate stochasticity. The pretrained DynaMix model carries a small internal standardized per-coordinate noise in its attention computation: before context-attention weights are formed, the current latent state is projected to observation space and perturbed by a learnable diagonal Gaussian noise term. In the model used here, $\boldsymbol{\Sigma}_{\text{gate}} \approx \text{diag}(0.003, 0.003, 0.008)$ in standardized coordinates. We retain this mechanism during adaptation rather than modifying the pretrained foundation model, see Eq. 18. For the EKF approximation, each gate evaluation uses one realization of this internal noise, we do not explicitly marginalize its contribution to the predictive covariance. We interpret the learned diagonal process covariance $\boldsymbol{\Gamma}$ as an effective process noise term that absorbs residual error from this approximation. We refer to F_{DynaMix} with the external Gaussian shell removed as the shell-off core transition; this transition retains the internal gate noise.

DynaMix specifications. The model specifications were directly adopted from [23]. Specifically, the model has $M = 30$ latent dimensions, $N = 3$ observed dimensions, $E = 10$ AL-RNN experts with $P = 2$ ReLU-gated coordinates, respectively. Each expert is therefore piecewise affine,

$$f_k(\boldsymbol{z}) = \boldsymbol{A}_k \boldsymbol{z} + \boldsymbol{W}_k \boldsymbol{\phi}_k^*(\boldsymbol{z}) + \boldsymbol{h}_k,$$

with AL-RNN parametrization as described in Appx. C.1.

Throughout the DynaMix experiments, the first N latent coordinates are identified with the observable state. In other words, we apply the same observation operator $\boldsymbol{B}$ that was used for the single AL-RNN experiments on Lorenz-96 (Eq. 14).

Shell parameterization. All probabilistic DynaMix baselines embed the model into a state-space model framework using an external Gaussian shell around the DynaMix core. We use the same diagonal parameterization for every DynaMix variant:

$$\boldsymbol{z}_{t+1} = F_{\text{DynaMix}}(\boldsymbol{z}_t, \text{ctx}, \boldsymbol{\varepsilon}_t^{\text{gate}}) + \boldsymbol{\Gamma}^{1/2} \boldsymbol{\xi}_t, \qquad \boldsymbol{x}_t = \boldsymbol{B} \boldsymbol{z}_t + \boldsymbol{\Lambda}^{1/2} \boldsymbol{\eta}_t, \tag{18}$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_M)$ and $\boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_N)$, and

$$\boldsymbol{\Gamma} = \text{diag}(q_1^2, \ldots, q_M^2), \qquad \boldsymbol{\Lambda} = \text{diag}(r_1^2, \ldots, r_N^2).$$

These initial scales are chosen analogously to the Lorenz-96 AL-RNN experiments, via teacher-forced residuals and variance split $\alpha = 0.9$ (see Appx. B.2).

In-context KAFFEE protocol. We use the released model `dynamix-3d-alrnn-v1.0`, distributed under CC BY 4.0, published alongside [23] in https://github.com/DurstewitzLab/DynaMix-python. This model was pretrained on a benchmark of three-dimensional chaotic attractors [90]. In our experiments, we adapt it to previously unseen systems. Table 2 reports the 13 held-out chaotic ODEs we evaluated and Table 3 shows the corresponding aggregated results.

For each system, the first $2 \cdot 10^3$ noisy observations form the fixed context. Our probabilistic KAFFEE-adaptation only trains on windows drawn inside this context. The scored window length W_{scored} is chosen per system according to its Lyapunov time:

$$W_{\text{scored}} = \text{round}\left(\frac{\tau_\lambda}{\Delta t}\right).$$

We clip W_{scored} to a range between 24 and 128 steps. The covariance burn-in stays at $L_{\text{burn}} = 8$ and the shell initialization at $\alpha = 0.9$ for all systems (see Appx. B.2). The DynaMix batch loss also averages the innovation NLL over the N observation coordinates.

Table 2: **Held-out DynaMix benchmark systems.** The table lists the 13 3-dimensional ODE systems we used for the DynaMix in-context KAFFEE adaptation, together with the system parameters, numerical integration step size Δt, transient discarding (burn-in) length, and the reference leading Lyapunov exponent λ_1^{ref}. The exponents listed here are computed using the ground-truth Jacobians along 10^4 steps of reference rollouts using QR re-orthonormalization [88, 89]. For a complete description, visualization and sources of these systems we refer to Gilpin [90].

System	**Parameters**	Δt	**Burn-in**	λ_1^{ref}
Aizawa	a=0.95, b=0.7, c=0.6, d=3.5, e=0.25, f=0.1	0.02	1000	0.088
Burke–Shaw	n=10, e=13	0.005	1000	2.314
Chen	a=35, b=3, c=28	0.005	2000	2.042
Chua	α=15.6, β=28, m_0=−1.143, m_1=−0.714	0.02	500	0.407
Dadras	p=3, q=2.7, r=1.7, s=2, e=9	0.02	500	0.367
Genesio–Tesi	a=0.44, b=1.1, c=1	0.02	500	0.108
Hadley	a=0.2, b=4, f=9, g=1	0.02	500	0.215
Halvorsen	a=1.4, b=4	0.01	1000	0.685
Lorenz–63	σ=10, ρ=28, β=8/3	0.02	500	0.902
Nose–Hoover	a=1.5	0.02	500	0.005
Qi–Chen	a=38, b=8/3, c=80	0.005	2000	3.677
Rabinovich–Fabrikant	γ=0.87, α=1.1	0.01	1000	0.215
Sprott B	—	0.05	500	0.207

Jacobian construction. To apply KAFFEE on the shell-off DynaMix core, we require a local transition Jacobian. Here, we approximate those Jacobians while freezing the mixture weights for the experts, which yields an analytical Jacobian again. Concretely, we evaluate the gating network at the current latent state under one realization of the gate noise $\varepsilon_t^{\text{gate}}$, and the resulting mixture weights $\boldsymbol{w}_t \in \Delta^{E-1}$ are detached (i.e., treated as locally constant in $\boldsymbol{z}$). The local Jacobian is assembled as the convex combination of the experts' closed-form AL-RNN Jacobians,

$$\boldsymbol{J}_t \approx \sum_{k=1}^{E} w_{t,k}\, \boldsymbol{J}_k(\boldsymbol{z}_{t|t}) \in \mathbb{R}^{M\times M}, \qquad \boldsymbol{J}_k(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} f_k(\boldsymbol{z}) = \boldsymbol{A}_k + \boldsymbol{W}_k\, \boldsymbol{D}_k(\boldsymbol{c}_t),$$

using the AL-RNN expert transition maps f_k, where $\boldsymbol{A}_k$ is diagonal, $\boldsymbol{W}_k$ is full, and $\boldsymbol{D}_k(\boldsymbol{c}_t) = \operatorname{diag}(1,\dots,1,c_{t,1},\dots,c_{t,P})$ is the binary switching matrix evaluated at the current latent state (i.e., $c_{t,j} = \mathbf{1}\{\boldsymbol{z}_{t,M-P+j} > 0\}$), see Eq. 13. The mixture weights are held fixed only for covariance propagation, while the gate remains trainable through the predicted mean. The Jacobian approximation thus omits the term

$$\sum_{k=1}^{E} f_k(\boldsymbol{z}) \nabla_{\boldsymbol{z}} w_{t,k}(\boldsymbol{z})^{\top}.$$

We use this Jacobian approximation for KAFFEE-DynaMix, the open-loop tied-propagation control, and frozen-core shell calibration. The no-propagation controls instead use the identity to propagate the covariance.

KAFFEE-DynaMix training. The DynaMix ablations shown in Table 3 use the same general procedure as the AL-RNN study (see Algorithm 1): starting from a teacher-forced-residual shell initialization ($\alpha = 0.9$), KAFFEE jointly optimizes the DynaMix core and the process/observation noise under the innovation NLL. Optimization uses Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [82], learning rate 5×10^{-5}, batch size 16, gradient clipping 1.0, and proceeds for 50 iterations on the 2000-step context. The context observations are held fixed for all windows of a given target system.

C.5.1 DynaMix evaluation protocol

While the training objectives differ, we evaluate all probabilistic DynaMix baselines using the same protocol with 256 ensemble members. Forecasts start from an encoding of the noisy observation at each origin. As in Appx. C.4, NLL averages diagonal Gaussian scores over leads $1,\dots,H$, forecast starts, and observed coordinates. Here, the noisy targets are standardized using the context's mean and standard deviation, and $H = \lfloor 0.25\tau_\lambda/\Delta t \rfloor$ is clipped to $[24, 128]$ steps.

In this experiment we also report the ten-step forecast root-mean-square error

$$\mathrm{RMSE}_{10} = \mathbb{E}_t \left[\sqrt{\frac{1}{N} \left\| \bar{\boldsymbol{x}}_{t+10|t} - \boldsymbol{x}^{\star}_{t+10} \right\|_2^2} \right], \tag{19}$$

where $\bar{\boldsymbol{x}}_{t+10|t}$ is the ensemble-mean ten-step forecast initialized by encoding the noisy observation at the held-out origin, and $\boldsymbol{x}^{\star}_{t+10}$ is the corresponding clean reference state; the expectation is taken over different starts on held-out windows. We report RMSE_{10} in observation coordinates standardized using the noisy context's mean and standard deviation. Shell-off dynamical fidelity metrics ($D^{\mathrm{core}}_{\mathrm{stsp}}$, D_H) are computed from rollouts of the adapted core with the external stochastic shell removed, since our evaluation question here is whether the transferred core dynamics can be retained under probabilistic adaptation. Since absolute D_{stsp} is not comparable across the 13 systems, we compute both the adapted core's and the base model's divergence from the same clean reference trajectory, and form $D^{\mathrm{core}}_{\mathrm{stsp}}/D^{\mathrm{base}}_{\mathrm{stsp}}$ for each system and seed before aggregation.

C.5.2 DynaMix baseline definitions

In the main text we report a compact comparison between KAFFEE-DynaMix, frozen-core shell calibration ("frozen-core"), and the open-loop tied-propagation control ("open-loop"). Table 3 expands this comparison to the mechanistic two-axis taxonomy of Fig. 1: *Objective* $\in \{\text{open-loop, filtered}\} \times$ *Covariance propagation* $\in \{\text{tied (Jacobian), no propagation (identity)}\}$. Thus, as for the Lorenz-96 ablation (Table 1), all four probabilistic baselines share one common training schedule and the only varying factor is the per-step objective and the covariance recursion. Open-loop adaptation inflates core occupancy relative to the DynaMix base model. Among methods that update the core, the two filtered objectives have similar aggregate held-out NLL, RMSE_{10}, and occupancy; frozen-core remains competitive on NLL while leaving core dynamics untouched by construction.

DynaMix base. The pretrained base model with no adaptation (and no shell). This is the core transition model for the probabilistic adaptation methods.

Frozen-core shell calibration. The diagonal shell is optimized under the EKF innovation NLL (Eq. 4), while the pretrained DynaMix weights are held fixed. This baseline exists to isolate the contribution of post-hoc noise calibration from any learned adaptation of the transition map. The shell-off rollouts for $D^{\mathrm{core}}_{\mathrm{stsp}}$ and D^{core}_{H} for this control hence match the baseline by construction.

Open-loop, tied propagation. This is the direct DynaMix analogue of the Lorenz-96 "Open-loop, tied propagation" control: DynaMix core and shell are updated jointly under a Gaussian open-loop NLL, where at each scored step the predictive mean and covariance are propagated by the EKF predict step (Eq. 2).

Open-loop, no propagation. Identical to the open-loop tied baseline, except that the predictive covariance recursion is replaced by $\boldsymbol{\Sigma}_{t|t_0} = \boldsymbol{\Sigma}_{t-1|t_0} + \boldsymbol{\Gamma}_{\theta_s}$, where t_0 denotes the fixed conditioning origin (see also the Lorenz-96 analogue, Appx. C.2).

Filtered, tied propagation (KAFFEE-DynaMix). Shell and core are updated jointly under the EKF innovation NLL (Eq. 4) with predictive covariance propagated using the Jacobian approximation above, according to Algorithm 1. This is the direct analogue of the Lorenz-96 KAFFEE method.

Filtered, no propagation. Identical to KAFFEE-DynaMix except that the predict-step covariance recursion is replaced by the state-invariant version; the EKF update is still applied at every step.

Table 3 shows that open-loop adaptation inflates core occupancy and is worse as a (local) forecast. Open-loop tied has a shell-off spectral distance similar to the zero-shot core, while no propagation increases that distance. Note that the scored training window is one Lyapunov time only when that length falls inside $[24, 128]$ steps; the evaluation horizon instead corresponds to a clipped quarter Lyapunov time. Thus, in contrast to the Lorenz-96 experiment, the volume penalty here may distort the attractor without harming its spectral frequencies. The two filtered objectives have similar aggregate NLL, RMSE_{10}, and state-space divergence, and broadly keep the zero-shot core's shell-off quality. Red highlighting in the table marks an open-loop core metric that is worse than the baseline/frozen-core autonomous dynamics.

Table 3: **In-context probabilistic adaptation of DynaMix on 13 held-out ODE systems.** All probabilistic methods share the same in-context adaptation schedule and diagonal noise parametrization, and are initialized from the same pretrained DynaMix core. NLL is the diagonal ensemble-Gaussian score averaged over leads $1, \ldots, H$, where H corresponds to $0.25\tau_\lambda$ clipped to $[24, 128]$ steps. **Bold**: median whose $\pm$ SE interval overlaps the column's best among methods that update the core; red: open-loop's degradation of autonomous dynamics. Medians $\pm$ robust SEs over $n = 13$ systems (per-system seed-medians); see Appx. C.5.1–C.5.2.

	Local Probabilistic Behavior		**Shell-off Dynamics (Core)**	
Model / Objective	NLL $(0.25\,\tau_\lambda)\downarrow$	$\mathrm{RMSE}_{10}\downarrow$	$D_{\mathrm{stsp}}^{\mathrm{core}}/D_{\mathrm{stsp}}^{\mathrm{base}}\downarrow$	$D_H^{\mathrm{core}}\downarrow$
Zero-shot transferred core				
DynaMix base (no adaptation)	–	–	1.000±0.000	0.399±0.066
Frozen-Core Shell Calibration	0.276±0.100	0.178±0.005	1.000±0.000	0.399±0.066
Open-Loop objective				
No propagation	0.700±0.145	0.242±0.015	1.427±0.228	0.471±0.033
Tied propagation	0.555±0.100	0.225±0.012	1.300±0.108	**0.398±0.032**
Filtered objective				
No propagation	**0.255±0.097**	**0.168±0.005**	**1.000±0.014**	**0.349±0.035**
Tied propagation (KAFFEE-DynaMix)	**0.254±0.098**	**0.168±0.004**	**1.007±0.013**	**0.382±0.038**

C.6 Compute resources

The following numbers are based on runs using NVIDIA RTX PRO 6000 Blackwell GPUs and are reported as median $\pm$ robust SE. On Lorenz-96 Table 1, one 150-step shell run (batch size 16, window 40), including the per-iteration 10^4-step validation $D_{\mathrm{stsp}}^{\mathrm{stoch}}$ used for model selection, takes 121 ± 2 s for KAFFEE and 114–119 s for the four matched controls. Peak allocated CUDA memory for KAFFEE is 0.22 GiB. The training without selection is 9.4 ± 0.2 s for KAFFEE ($n = 20$); one selection score takes 0.682 ± 0.001 s. Sequentially, the table grid is 5 methods $\times$ 20 seeds $\approx$ 3.3 single-GPU hours (shared AL-RNN core pretraining not included). For DynaMix, each recorded time is the 50-step adaptation only (batch size 16), not the subsequent held-out scoring or $5 \cdot 10^4$-step core evaluation. Aggregating as in Table 3 (median over $n = 13$ systems of per-system seed-medians), time is 47 ± 3 s for KAFFEE-DynaMix, 44 ± 2 s for open-loop tied, 36 ± 2 s for filtered no-propagation, 32 ± 2 s for open-loop no-propagation, and 31 ± 2 s for frozen-core shell calibration. Peak allocated CUDA memory for KAFFEE-DynaMix is 0.25 ± 0.01 GiB. Sequentially, that adaptation grid is 5 methods $\times$ 13 systems $\times$ 20 seeds $\approx$ 14 hours when using a single GPU.

D Additional Experiments

This section contains six supporting experiments. First, the stochastic Lorenz-96 ablation is repeated on the 3-dimensional stochastic Lorenz-63 system [54] (Appx. D.1). Second, we verify that the Lorenz-96 ablation is not tied to the particular forecasting horizon used in Table 1 (Appx. D.2). Third, we isolate whether adaptive/anisotropic Kalman gains are offering deterministic-core reconstruction benefits that a constant forcing strength (or a stop-gradient through the Kalman update) does not (Appx. D.3). Fourth, we use two non-chaotic systems to isolate the classical noisy-state attenuation mechanism discussed in Appx. A.3 (Appx. D.4). Fifth, Appx. D.5 reports performance and empirical scaling exponents for time/memory of dense vs. rank-reduced KAFFEE (rr-KAFFEE) on Lorenz-96. Sixth, we evaluate probabilistic adaptation on an EEG recording to assess real-world data feasibility (Appx. D.6).

D.1 Ablation on stochastic Lorenz-63

Table 1 showed a hyperchaotic system (multiple unstable directions), whose deterministic prior was miscalibrated. The Lorenz-63 system has a single positive (i.e., unstable) Lyapunov exponent ($\lambda_1 = 0.905$), and an already faithful deterministic prior (see Table 4). This ablation therefore tests not only whether the DPC gap signatures persist on an additional system, but also whether they appear when the deterministic prior does not need to be retuned. For this, we keep the taxonomy of Fig. 1 and once more use the three-axis evaluation of Section 4.2.

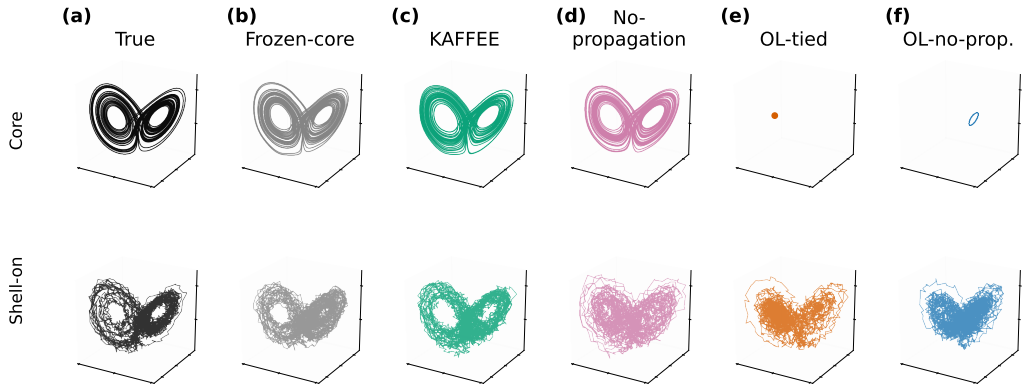
Table 4: **Mechanistic ablation on stochastic Lorenz-63.** All probabilistic baselines share the same training schedule and are initialized from the same AL-RNN cores. Open-loop objectives collapse the attractor; no-propagation remains locally blind to the dynamics-induced uncertainty. **Bold**: median whose ±SE interval overlaps the column's best; red: signature failure mode (bracketed label). Medians ± SEs across 20 seeds.

	Local Probabilistic Behavior			Stochastic Dynamics (Shell+Core)		Deterministic Dynamics (Core)		
Model / Objective	NLL ($\frac{1}{4}\tau_\lambda$) ↓	CRPS ($\frac{1}{4}\tau_\lambda$) ↓	$G_{\tan}$ ↑	$D_{\text{stsp}}^{\text{stoch}}$ ↓	$\lvert\Delta h_\lambda^{\text{stoch}}\rvert$ ↓	$D_{\text{stsp}}^{\text{core}}$ ↓	$\lvert\Delta h_\lambda^{\text{core}}\rvert$ ↓	$D_{\text{H}}^{\text{core}}$ ↓
Deterministic prior								
AL-RNN (no shell)	–	–	–	–	–	**0.362±0.033**	**0.040±0.008**	**0.069±0.005**
Frozen-Core Shell Calibration	**1.771±0.003**	**0.818±0.011**	0.740±0.020	0.302±0.014	0.117±0.007	**0.362±0.033**	**0.040±0.008**	**0.069±0.005**
Open-Loop objective								
No propagation [Core Collapse]	2.227±0.066	1.794±0.065	0.000±0.000	2.968±0.616	0.380±0.099	15.080±1.183	0.916±0.054	0.683±0.080
Tied propagation [Core Collapse]	**2.026±0.361**	1.367±0.360	0.683±0.025	1.972±0.617	0.455±0.132	15.365±0.277	0.916±0.002	0.630±0.060
Filtered objective								
No propagation [Blind Uncertainty]	1.779±0.002	**0.828±0.009**	0.000±0.000	0.282±0.015	**0.073±0.023**	0.449±0.030	**0.042±0.005**	0.090±0.004
Tied propagation (KAFFEE)	1.934±0.025	0.944±0.027	**0.776±0.013**	**0.246±0.008**	**0.056±0.010**	1.351±0.134	0.074±0.012	0.148±0.017

Experimental details. Data generation follows Appx. C.3, using the 3-dimensional Lorenz-63 system [54]:

$$\begin{aligned}\frac{\mathrm{d}z_1}{\mathrm{d}t} &= \sigma(z_2 - z_1),\\ \frac{\mathrm{d}z_2}{\mathrm{d}t} &= z_1(\rho - z_3) - z_2,\\ \frac{\mathrm{d}z_3}{\mathrm{d}t} &= z_1 z_2 - \beta z_3,\end{aligned}$$

in the standard chaotic regime $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, discretization interval $\Delta t = 0.01$, and $\sigma_{\text{obs}} = \sigma_{\text{proc}} = 0.1$. As in Table 1, we use 20 AL-RNNs as deterministic priors; here with $M = 30$, $P = 10$, and teacher-forcing interval $\tau = 16$. Deterministic pretraining otherwise follows Appx. C.1. Shell training uses the same matched baselines and settings as Appx. C.2, except that the learning rate is cosine-annealed from 10^{-3} rather than 5×10^{-3}. Deterministic pretraining uses sequences of length 200. Shell training uses the same length with 8 covariance burn-in steps. This roughly matches the Lorenz-96 setting in Lyapunov time: 192 scored steps equals approximately $1.7\tau_\lambda$ (for Lorenz-96 we scored 32 steps, approximately $2\tau_\lambda$). Local NLL and CRPS use the same $\frac{1}{4}\tau_\lambda$ horizon as Appx. C.4, hence $H = 27$ here.



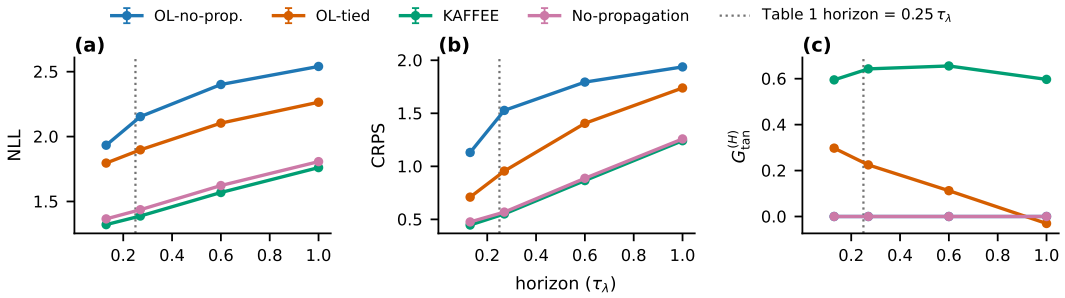
Appendix Figure 2: **Shell-off and shell-on rollouts for Lorenz-63.** Shown is one exemplary core seed. **Top:** Trained autonomous cores under the different objectives/parametrizations. **(a–d)** Frozen-core, KAFFEE and filtered no-propagation leave the faithful deterministic prior intact and stay close to the true attractor. **(e–f)** The open-loop objectives both collapse the core. Open-loop tied collapses to a fixed point; open-loop no-propagation collapses to a limit cycle. **Bottom:** the shell-on rollouts (with process and observation noise) of the same models. **(a–d)** The filtered objectives produce good overlap with the noisy ground truth, frozen-core noise calibration is closest in agreement. **(e–f)** Open-loop objectives use the DPC gap's failure modes to "mask" their collapsed core; producing a Lorenz-like attractor with their shell-on rollouts.

The DPC signatures appear here as well. Open-loop tied collapses the autonomous dynamics: median $D_{\mathrm{stsp}}^{\mathrm{core}} = 15.37$ and $h_{\lambda}^{\mathrm{core}} \approx 0$, see e.g. Appx. Fig. 2(e). With no covariance propagation, open-loop does the same ($15/20$ cores collapse the unstable dynamics) and remains locally blind. However, in this experiment, the open-loop no-propagation baseline did not inflate Λ, but collapsed the core through the Mahalanobis term driving the updates (hence we call both failures "core collapse" here). Appx. Fig. 2 shows that the noisy shell-on rollouts can indeed mask a collapsed core.

For the filtered innovation objectives, no-propagation retains the deterministic dynamics and exhibits competitive NLL/CRPS, but its state-invariant covariance recursion cannot follow the unstable direction. KAFFEE achieves the highest alignment of local dynamics with uncertainty, $G_{\tan} \approx 0.776$. On this system, the AL-RNN prior is already a strong deterministic core (median $D_{\mathrm{stsp}}^{\mathrm{core}} = 0.362$, $|\Delta h_{\lambda}^{\mathrm{core}}| = 0.040$), so frozen-core shell calibration already gives strong local probabilistic behavior and grounded uncertainty. This shows that post-hoc noise calibration can be preferential whenever the core must not change or is already a strong approximation to the deterministic dynamics.

D.2 Forecast-skill spectrum across horizons

Table 1 reports local predictive scores at $H = \frac{1}{4}\tau_{\lambda}$. To check that the conclusions do not depend on this single choice, Appx. Fig. 3 evaluates the same 20 Lorenz-96 seeds across $H \in \{0.13, 0.27, 0.60, 1.00\}\tau_{\lambda}$. The NLL is the autoregressive ensemble NLL, CRPS is scored at the endpoint $t + H$, and $G_{\tan}^{(H)}$ measures whether predictive variance growth over the assimilation cycle follows local tangent expansion.



Appendix Figure 3: **Lorenz-96 forecast-skill spectrum. (a, b)** Held-out NLL and CRPS worsen for all objectives as the horizon grows. Filtered objectives remain below open-loop ones, and filtered no-propagation stays close to KAFFEE. **(c)** Among the jointly adapted models, only KAFFEE keeps a large positive $G_{\tan}^{(H)}$. Both no-propagation methods stay at zero, and open-loop tied propagation falls toward zero for larger horizons. Markers are medians across 20 seeds; error bars are robust SEs.

The spectrum over horizons supports the same conclusion as Table 1. NLL and CRPS worsen for every objective as the horizon grows, as expected for chaotic dynamics. Filtered objectives remain below the open-loop ones at every horizon. However, KAFFEE is still close to filtered no-propagation in NLL and CRPS. The local grounding metric separates these cases. KAFFEE maintains a large positive $G_{\tan}^{(H)}$ at every horizon H, and both no-propagation methods stay at zero. Open-loop tied propagation falls toward zero as the horizon grows. The blind-uncertainty failure is thus not a horizon-specific artifact.

D.3 Fixed-gain forcing and stop-gradient through the Kalman update

The Kalman update in KAFFEE both assimilates observations in forward mode and can attenuate amplification in the backward Jacobian product [48], see Appx. A.5. Table 5 isolates whether KAFFEE's benefit in reconstructing the deterministic core comes from learning through that update in BPTT, rather than from forward assimilation or a fixed forcing strength, using the Lorenz-96 data and setting of Table 1. We implement two different controls: "Update-Stopgradient" (Update-SG) keeps the forward EKF identical to KAFFEE at fixed parameters but stops gradients through the

Kalman update. Denoting the stop-gradient function by $\mathrm{sg}(\cdot)$, we propagate

$$\widetilde{\boldsymbol{z}}_t = \mathrm{sg}(\boldsymbol{z}_{t|t}) + \boldsymbol{z}_{t|t-1} - \mathrm{sg}(\boldsymbol{z}_{t|t-1}), \qquad \widetilde{\boldsymbol{\Sigma}}_t = \mathrm{sg}(\boldsymbol{\Sigma}_{t|t}) + \boldsymbol{\Sigma}_{t|t-1} - \mathrm{sg}(\boldsymbol{\Sigma}_{t|t-1}).$$

So the filter still assimilates in the forward pass (and the loss is still the filtered innovation NLL), but backpropagation skips this update and gradients flow only through the filter's predictions. GTF-α replaces the learned Kalman gain by a constant/scalar forcing strength $\boldsymbol{K}_\alpha = \alpha \boldsymbol{B}^\top$, followed by the covariance update. This mimics the fixed forcing strength applied in GTF [34]. We refer to these controls as GTF-α and evaluate $\boldsymbol{K}_\alpha$ for a range of α values. The fixed gains are used during training only. At evaluation, all fitted models use ordinary Kalman assimilation to compute the filtered states and covariances entering $G_{\tan}$; NLL uses the common forecast-ensemble evaluation of Appx. C.4.

Table 5: **KAFFEE mechanism ablations on Lorenz-96.** Deterministic dynamics reconstruction performance. Medians ± robust SE over 20 seeds. Settings match Table 1. **Bold**: median whose ±SE interval overlaps the column's best.

Method	NLL ↓	$G_{\tan}$ ↑	$D_{\mathrm{stsp}}^{\mathrm{core}}$ ↓	$\lvert\Delta h_\lambda^{\mathrm{core}}\rvert$ ↓
KAFFEE (Table 1)	**1.388±0.006**	**0.528±0.003**	**0.291±0.008**	**0.811±0.074**
Update-SG	1.662±0.017	0.283±0.029	2.664±0.337	4.132±0.320
Fixed gain (GTF-α)				
$\alpha = 0$	1.893±0.008	0.314±0.015	2.621±0.011	4.868±0.000
$\alpha = 0.1$	1.431±0.004	0.402±0.006	1.284±0.414	4.137±0.139
$\alpha = 0.25$	**1.394±0.004**	0.484±0.006	0.533±0.040	2.830±0.086
$\alpha = 0.5$	1.405±0.005	**0.529±0.003**	0.529±0.017	2.968±0.053
$\alpha = 1$	2.078±0.013	0.492±0.006	0.821±0.104	3.692±0.166

Table 5 shows that models trained with constant scalar forcing can achieve competitive predictive scores and dynamical alignment at selected α values. $\alpha = 0.25$ matches KAFFEE on NLL, and $\alpha = 0.5$ matches KAFFEE on $G_{\tan}$. Nevertheless, none of the α values we tested beats its (paired-seed) KAFFEE model on $D_{\mathrm{stsp}}^{\mathrm{core}}$ or $\lvert\Delta h_\lambda^{\mathrm{core}}\rvert$ across the 20 seeds. Update-SG also delivers substantially worse core reconstruction under the matched schedule. $\alpha = 0$ matches the open-loop-tied (core collapse) failure mode we observed in the main Lorenz-96 experiment (Table 1), because $\boldsymbol{K}_\alpha = 0$ means no assimilation takes place.

These controls thus support an important contribution from differentiation through the Kalman update: if gradients through the Kalman update are stopped, autonomous dynamics reconstruction worsens or collapses (while NLL remains only moderately worse). Constant forcing can achieve competitive predictive scores for some α values, but it remains dependent on hyperparameter selection, whereas KAFFEE's use of probabilistically principled assimilation via the Kalman gain provides a data-dependent and anisotropic alternative, that does not require choosing an α.

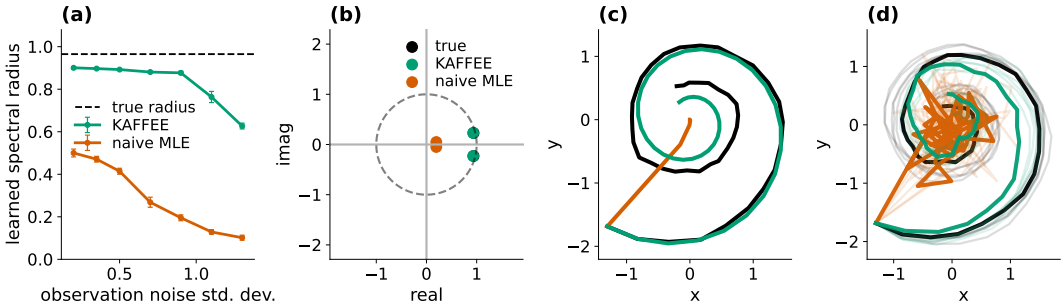
D.4 Non-chaotic controls: isolating noisy-state attenuation

This work examines DPC mechanisms in the context of chaotic DSR. A related but simpler mechanism is classical noisy-state attenuation: if noisy observations are treated directly as states, then one-step Gaussian prediction estimates an observation-to-observation map rather than the latent transition (i.e., it becomes errors-in-variables regression). Appx. A.3 shows this analytically in the scalar linear-Gaussian case.

The following controls isolate this simpler mechanism. In both experiments, the competing methods use the same transition family and initialization. The difference is only the objective. The *noisy-state predictive objective* ("naive maximum likelihood estimation (MLE)") scores Gaussian predictions directly on noisy observations, i.e. it treats $\boldsymbol{x}_t$ as if it were the latent state. KAFFEE instead fits the corresponding state-space model by filtering the latent state and scoring innovations. In the linear-Gaussian case, the EKF used by KAFFEE is the standard Kalman filter, so KAFFEE's training objective reduces to the exact Kalman innovation maximum likelihood. The deterministic core of KAFFEE and the noisy-state predictive objectives are not pretrained models here, but initialized randomly by drawing from Gaussian distributions.

Stable linear spiral. We first consider a two-dimensional stable linear spiral observed with additive Gaussian noise. The spectral radius of a transition matrix $\boldsymbol{A}$ is $\rho(\boldsymbol{A}) = \max_i |\operatorname{eig}(\boldsymbol{A})|$, the largest

absolute eigenvalue. The true spiral here has $\rho(\boldsymbol{A}) = 0.965 < 1$, so a smaller learned radius is a more strongly damped autonomous map. As shown in Fig. 4, the noisy-state predictive objective collapses this radius, while KAFFEE recovers the correct damping. Panel (a) shows the same effect across multiple observation-noise levels: the noisy-state objective becomes progressively more contractive as noise increases, while KAFFEE remains close to the true radius until the largest noise levels.



Appendix Figure 4: **Attenuation on a noisy linear spiral. (a)** Learned spectral radius $\rho(\hat{\boldsymbol{A}})$ against observation-noise level. Error bars are 95% confidence intervals over 10 seeds. **(b)** Eigenvalues of learned Jacobians, **(c)** deterministic autonomous rollouts from the common initialization, and **(d)** stochastic state rollouts from that same initialization. Both models share the same linear Gaussian state-space parametrization and initialization. The noisy state predictive objective over-damps the learned autonomous map, while KAFFEE recovers the correct spectral radius and rollout geometry.

Van der Pol limit cycle. We next use a nonlinear but non-chaotic system: a noisy Van der Pol limit cycle. The data are generated from

$$\dot{\boldsymbol{z}} = \begin{bmatrix} v \\ \mu(1 - x^2)v - x \end{bmatrix}, \qquad \mu = 1.6,$$

with RK4 step size $\Delta t = 0.05$, process noise $\sigma_{\text{proc}} = 0.035$, and observation noise $\sigma_{\text{obs}} = 0.25$. Both methods fit the same nonlinear Gaussian state-space model with diagonal process and observation covariances and a cubic polynomial transition function

$$f_\theta(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{z}) + \boldsymbol{h}, \qquad \boldsymbol{z} = [x, v]^\top,$$

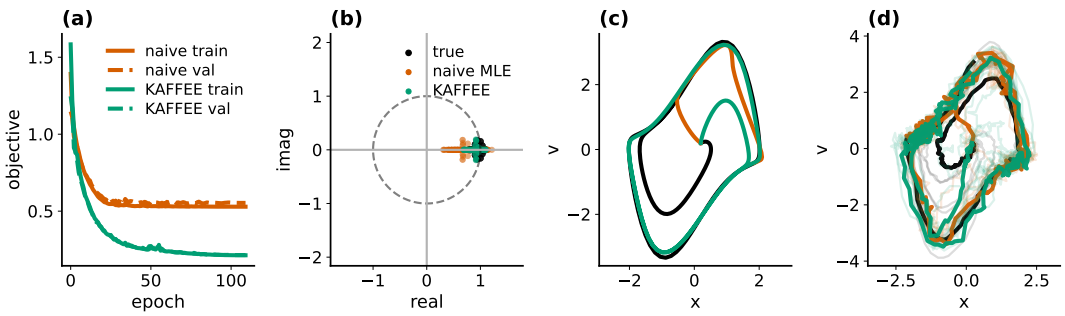
where

$$\boldsymbol{\phi}([x, v]^\top) = \left[x^2, xv, v^2, x^3, x^2 v, xv^2, v^3\right]^\top.$$

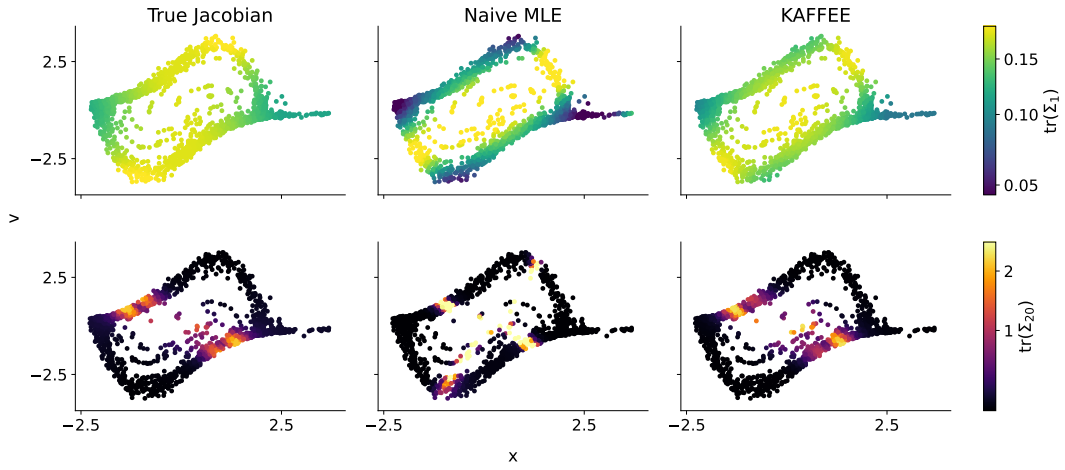
Appx. Fig. 5 shows the same qualitative failure in a nonlinear setting. The noisy-state predictive NLL learns an overcontractive autonomous map: its shell-off rollout does not return to the limit cycle and instead collapses to a spurious fixed point. The stochastic shell-on rollout can partly "mask" this deterministic failure, but it does not repair the learned core. KAFFEE preserves the transient and returns to the correct orbit, with local Jacobian spectra closer to those of the true flow map.

Appx. Fig. 6 shows a covariance propagation diagnostic for this example. At each state we initialize the same isotropic reference covariance and propagate it for one or twenty steps under either the true flow Jacobian or the learned Jacobian products. Additive process noise is omitted in this diagnostic, so darker colors correspond to stronger contraction of the propagated covariance. The noisy-state objective is systematically more contractive than the reference, whereas KAFFEE better matches the reference covariance-propagation pattern.

Together, these controls isolate the simple failure mode known as attenuation bias [45, 46], and relate it to the DPC perspective: a finite-horizon predictive objective can improve local fit by contracting the learned transition when noisy observations are used as states. Scoring innovations removes this shortcut in the correctly specified linear-Gaussian case and remains a useful inductive bias for the nonlinear example.



Appendix Figure 5: **Noisy Van der Pol.** **(a)** Optimization traces, **(b)** Jacobian eigenvalues along a deterministic reference rollout, **(c)** deterministic autonomous rollouts from the common initialization, and **(d)** stochastic state rollouts from that same initialization. Both methods use the same cubic polynomial transition family. The noisy-state predictive objective learns an overcontractive autonomous map that collapses to a spurious fixed point, while the stochastic rollout masks this failure. KAFFEE preserves the autonomous core by fitting the latent state-space model through scoring assimilated innovations.



Appendix Figure 6: **Van der Pol covariance propagation map. Top row:** One-step propagated reference-covariance trace over the same sampled state cloud: left for the true flow Jacobian, center for the noisy-state predictive objective, right for KAFFEE. **Bottom row:** Analogous twenty-step covariance propagation map with the same ordering of panels. Overall, the noisy-state predictive objective yields damped/contracted uncertainty propagation, while KAFFEE recovers a dynamically grounded covariance propagation.

D.5 Rank-reduced KAFFEE

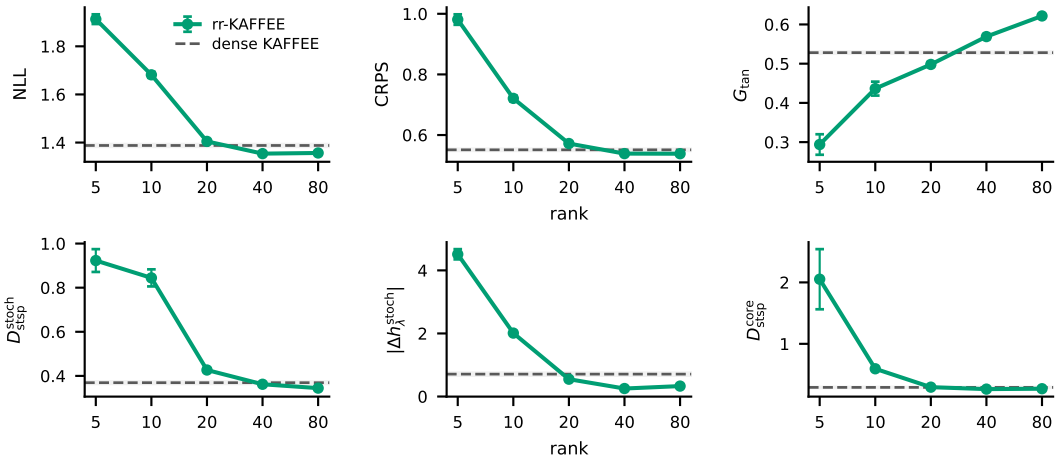
In Appx. A.7, we identify dense covariance propagation and its cubic scaling in latents as a computational bottleneck of the EKF training of KAFFEE. This subsection instantiates and evaluates a structured low-rank approximation on the Lorenz-96 experiment.

In Table 1, we use a dense $M \times M$ latent covariance with $M = 100$. rr-KAFFEE replaces that covariance by the rank-reduced Kalman filter (RRKF) of Schmidt et al. [65]: a rank-r square-root factor obtained by truncated SVD after Jacobian propagation. Concretely, rr-KAFFEE represents the filtering covariance as $\boldsymbol{\Sigma}_{t|t} = \boldsymbol{V}_t \boldsymbol{V}_t^\top$, with $\boldsymbol{V}_t \in \mathbb{R}^{M \times r}$. It additionally learns a low-rank process-noise factor $\boldsymbol{S} \in \mathbb{R}^{M \times r}$, such that $\boldsymbol{\Gamma} = \boldsymbol{S}\boldsymbol{S}^\top$, while the observation noise $\boldsymbol{\Lambda}$ remains diagonal. Prediction then truncates to keep only the leading r singular components of $[\boldsymbol{J}_t \boldsymbol{V}_{t-1}, \boldsymbol{S}]$. In this sense, rr-KAFFEE is a rank-reduced variant of KAFFEE, but it changes both the filtering representation and the process-noise parametrization (low-rank vs. "only" diagonal). For training, we differentiate through the RRKF's square-root operations and truncated SVDs; training and evaluation otherwise follow the same Lorenz-96 shell protocol (see Appx. C.2).

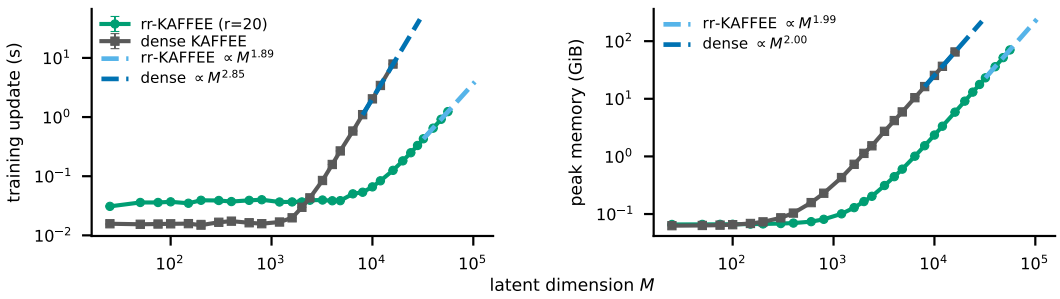
Rank $r = 20$ matches the observation dimension N and the true process-noise rank of the studied Lorenz-96 system. Appx. Fig. 7 shows that this rank already recovers dense-like quality; larger ranks remain competitive or slightly improve several local scores.

Next we study the scaling behavior of rr-KAFFEE with latent dimension. Appx. Fig. 8 reports training time (1 iteration) and peak allocated CUDA memory vs. latent dimension M (to isolate the asymptotic scaling past the computational-overhead regime, we used batch size 1, window 10, $r = 20$, and float32 precision). Across the four largest latent dimensions tested successfully before GPU memory was exhausted, wall-clock time scales as $M^{2.85}$ for dense KAFFEE vs. $M^{1.89}$ for rr-KAFFEE, so cubic vs. quadratic in M. Peak allocated memory remains approximately quadratic for both, consistent with the dense core weights and their gradients and optimizer state. Dense KAFFEE last completes at $M = 1.6 \cdot 10^4$ and runs out of memory at $M = 2.0 \cdot 10^4$; rr-KAFFEE last completes at $M = 5.6 \cdot 10^4$ and runs out of memory at $M = 6.4 \cdot 10^4$.

These results show that rr-KAFFEE achieves approximately quadratic scaling of training time with latent dimension, and substantially larger feasible latent dimensions before running out of memory. Additionally, for the Lorenz-96 experiment we studied, rr-KAFFEE retains competitive reconstruction and predictive performance when the rank is chosen appropriately. A principled choice of the rank (e.g., the rank of the process noise covariance) could be a direction for future work.



Appendix Figure 7: **rr-KAFFEE quality vs. filter rank on Lorenz-96.** Markers are seed medians; dashed horizontal lines are dense Table 1 KAFFEE. Rank $r = 20$ already recovers dense-like scores; $r = 40$ and $r = 80$ remain competitive.



Appendix Figure 8: **Dense KAFFEE vs. rr-KAFFEE compute scaling.** Training time (1 iteration) (left) and peak allocated CUDA memory (right) vs. latent dimension M on log-log scale. Dashed lines are power-law fits on the last four completed latent sizes, respectively. Dense KAFFEE gives $M^{2.85}$ (time) and M^2 (memory); rr-KAFFEE ($r = 20$) gives $M^{1.89}$ (time) and $M^{1.99}$ (memory).

D.6 Probabilistic adaptation on EEG data

To evaluate probabilistic adaptation beyond synthetic dynamical systems, we apply the models/objectives of Table 1 to an electroencephalogram (EEG) recording. We use an eyes-open, task-free (resting-state) recording from the PhysioNet EEG Motor Movement/Imagery dataset (subject/recording S001R01) [91], distributed under the Open Data Commons Attribution License v1.0, which was studied in multiple prior DSR works [34, 42, 48, 51]. Starting from the same pretrained deterministic cores for each objective, we compare joint adaptations of the core and stochastic shell with frozen-core shell calibration. This tests whether adapting the dynamics improves probabilistic fit and long-run reconstruction of the observed recording beyond post-hoc noise calibration and beyond deterministic core training alone. We then repeat the comparison on recordings of four additional subjects of the same dataset.

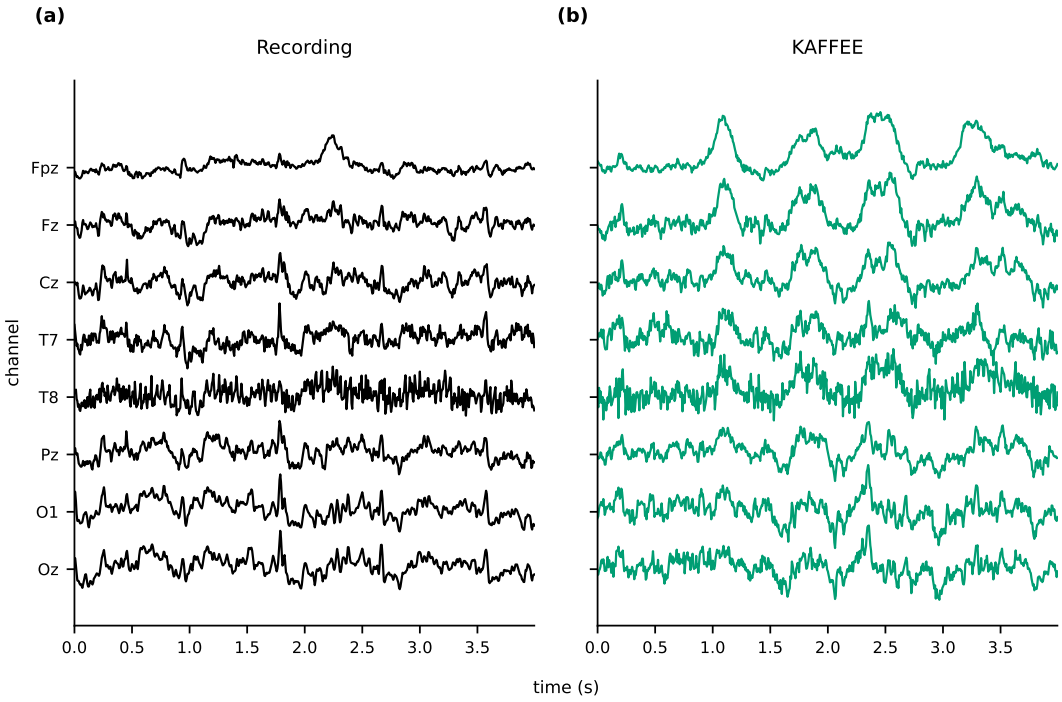
Experimental details. Here, the deterministic prior used is a clipped shallow piecewise-linear RNN (shPLRNN) [34]. It parametrizes the latent transition of $\boldsymbol{z}_t \in \mathbb{R}^M$ as

$$F_{\theta_c}(\boldsymbol{z}_t) = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{W}_1\big[\phi(\boldsymbol{W}_2\boldsymbol{z}_t + \boldsymbol{h}_2) - \phi(\boldsymbol{W}_2\boldsymbol{z}_t)\big] + \boldsymbol{h}_1, \tag{20}$$

where ϕ is the element-wise applied ReLU function (yielding piecewise-linear dynamics), $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ is diagonal, $\boldsymbol{W}_1 \in \mathbb{R}^{M\times L}$, $\boldsymbol{W}_2 \in \mathbb{R}^{L\times M}$, $\boldsymbol{h}_1 \in \mathbb{R}^M$, and $\boldsymbol{h}_2 \in \mathbb{R}^L$. Thus, unlike the ALRNN (Appx. C.1), the shPLRNN contains a hidden layer of width L, separate from the recurrent/latent state dimension M. This allows low-dimensional latent dynamics while keeping an expressive (wide) nonlinear transition map, hence the transition Jacobian and the propagated latent covariance are still $M \times M$, where $L \gg M$. Following the EEG settings of Hess et al. [34], we use $M = 16$ and $L = 512$ in our experiments. Further, we use a linear observation model $\boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{z}_t$, $\boldsymbol{B} \in \mathbb{R}^{N\times M}$ that is learned during the respective core's pretraining and held fixed during probabilistic adaptation.

Each EEG recording contains $N = 64$ channels sampled at 160 Hz. To match the length of the EEG time series studied in Hess et al. [34], we use the first $T = 9{,}640$ samples of each recording. Each channel is standardized using its mean and standard deviation over this entire time series. The same series is used for training and evaluation. We train 20 independent cores per recording using GTF with $\alpha = 0.1$ for $5 \cdot 10^3$ epochs, and select the model at the last epoch. Core training uses RAdam with an exponentially decaying learning rate from 10^{-3} to 10^{-6}, 50 batches per epoch, batch size 16, and 50-step windows. These settings are the same that were used in Hess et al. [34] on recording S001R01. However, in that work, the authors trained and evaluated the models on the standardized and smoothed recording (using a 15-bin Hann window). Our preprocessing and evaluation differs from that procedure: since our goal is probabilistic DSR, we retain those fluctuations in the data and train the deterministic cores on unsmoothed (but standardized) data, and the trained models are scored against the unsmoothed recording.

Each core initializes all five probabilistic objectives of Appx. C.2. Probabilistic adaptation then uses Adam for 900 updates with batch size 32 and 40-step windows (8 burn-in and 32 scored steps), with gradient clipping at 2, and learning rate 10^{-3}, which was cosine-annealed. Models are selected by the lowest state-space divergence of shell-on rollouts against the same recording, evaluated after update 1 and every 25th update.



Appendix Figure 9: **Example EEG reconstruction.** **(a)** Eight channels from the (standardized) PhysioNet EEG recording S001R01 [91]. **(b)** The corresponding channels from a freely generated stochastic (shell-on) KAFFEE rollout after burn-in. The two segments (recording/model-generated) are not aligned in time.

For local probabilistic behavior, we evaluate all probabilistic models on one-step predictions ($H = 1$) from the filtered history, using each model's covariance-propagation rule. After a burn-in phase of 14 conditioning observations, each new observation is scored before assimilation, yielding 9,626 scored predictions per recording. NLL then scores the joint Gaussian prediction with its full covariance $\mathbf{\Omega}_t$ (Eq. 3), averaged over time and normalized by the number of channels. CRPS is computed in closed form from the corresponding Gaussian marginals and averaged over time and channels. Note that this differs from the CRPS used e.g. for the Lorenz-96 experiments in Appx. C.4, where we instead compute CRPS directly from forecast ensembles.

For stochastic dynamical fidelity, we compare freely generated shell-on trajectories with the unsmoothed recording. Again following Hess et al. [34], we score $T = 9{,}640$ generated observations after letting the model run for $0.25T$ from the first data point. State-space divergence $D_{\text{stsp}}^{\text{stoch}}$ and the

Table 6: **Probabilistic adaptation on EEG.** Medians ± SEs across 20 paired core seeds for recording S001R01. **Bold:** median whose ±SE interval overlaps the column's best.

Model / Objective	NLL ↓	CRPS ↓	$D_{\text{stsp}}^{\text{stoch}}$ ↓	$D_{\text{H}}^{\text{stoch}}$ ↓
Deterministic prior				
shPLRNN	–	–	4.73±7.69	0.217±0.083
Frozen-Core Shell Calibration	−0.001±0.007	0.1911±0.0010	1.58±0.06	0.218±0.009
Open-Loop objective				
No propagation	0.097±0.006	0.2256±0.0032	3.41±0.27	0.220±0.009
Tied propagation	0.042±0.003	0.2327±0.0029	1.91±0.06	0.190±0.009
Filtered objective				
No propagation	−0.027±0.004	**0.1726±0.0008**	1.61±0.03	**0.108±0.004**
Tied propagation (KAFFEE)	**−0.042±0.004**	0.1781±0.0007	**1.39±0.03**	**0.114±0.005**

mean Hellinger distance between the per-channel power spectra, $D_{\mathrm{H}}^{\mathrm{stoch}}$, are computed as described in Appx. C.4. Fig. 9 illustrates eight channels from the recording alongside the corresponding channels of a freely generated stochastic KAFFEE rollout.

KAFFEE improves all four reported scores relative to frozen-core shell calibration, and both dynamic fidelity metrics relative to the deterministic prior (Table 6). Calibrating the shell alone reduces $D_{\mathrm{stsp}}^{\mathrm{stoch}}$ from 4.73 to 1.58, but leaves the spectral/temporal mismatch $D_{\mathrm{H}}^{\mathrm{stoch}}$ nearly unchanged. Joint adaptation under either filtered objective yields substantially better spectral agreement, together with improved NLL and CRPS. KAFFEE also improves state-space divergence relative to frozen-core calibration. KAFFEE achieves the lowest NLL and $D_{\mathrm{stsp}}^{\mathrm{stoch}}$, while filtered no-propagation has the lowest CRPS. Both filtered objectives outperform the open-loop controls on all four reported metrics.

Additional subjects. To assess whether these results depend on the single recording S001R01, we repeat the experiment on resting-state recordings for four additional subjects (recordings S002R01–S005R01). The preprocessing, training/model settings, and evaluation protocol remain the same. Table 7 reports the comparison with frozen-core shell calibration and filtered no-propagation.

Table 7: **Probabilistic adaptation on four additional EEG recordings.** Settings match Table 6. Shown are medians ± SEs across 20 paired core seeds per recording. **Bold:** median whose ±SE interval overlaps the column's best within each recording.

Subject	**Method**	NLL ↓	CRPS ↓	$D_{\mathrm{stsp}}^{\mathrm{stoch}}$ ↓	$D_{\mathrm{H}}^{\mathrm{stoch}}$ ↓
S002	Frozen-core shell	0.443 ± 0.004	0.3222 ± 0.0018	3.20 ± 0.05	$\mathbf{0.161\pm0.003}$
	Filtered no-propagation	0.400 ± 0.003	$\mathbf{0.2652\pm0.0006}$	2.77 ± 0.05	$\mathbf{0.159\pm0.003}$
	KAFFEE	$\mathbf{0.373\pm0.002}$	0.2757 ± 0.0004	$\mathbf{2.51\pm0.02}$	0.185 ± 0.005
S003	Frozen-core shell	0.123 ± 0.002	0.2080 ± 0.0013	1.97 ± 0.04	0.260 ± 0.004
	Filtered no-propagation	0.107 ± 0.003	$\mathbf{0.1772\pm0.0008}$	2.13 ± 0.04	$\mathbf{0.192\pm0.005}$
	KAFFEE	$\mathbf{0.087\pm0.002}$	0.1880 ± 0.0005	$\mathbf{1.80\pm0.03}$	$\mathbf{0.187\pm0.007}$
S004	Frozen-core shell	0.393 ± 0.003	0.2539 ± 0.0012	$\mathbf{4.54\pm0.04}$	0.246 ± 0.004
	Filtered no-propagation	$\mathbf{0.371\pm0.003}$	$\mathbf{0.2274\pm0.0006}$	4.81 ± 0.05	$\mathbf{0.217\pm0.006}$
	KAFFEE	$\mathbf{0.376\pm0.006}$	0.2386 ± 0.0005	$\mathbf{4.46\pm0.04}$	$\mathbf{0.212\pm0.006}$
S005	Frozen-core shell	0.212 ± 0.003	0.1978 ± 0.0001	2.33 ± 0.04	0.178 ± 0.002
	Filtered no-propagation	0.176 ± 0.003	$\mathbf{0.1907\pm0.0003}$	2.18 ± 0.02	0.171 ± 0.003
	KAFFEE	$\mathbf{0.167\pm0.002}$	0.1937 ± 0.0001	$\mathbf{2.05\pm0.02}$	$\mathbf{0.164\pm0.002}$

Across all five recordings, KAFFEE has the lowest median $D_{\mathrm{stsp}}^{\mathrm{stoch}}$ among the probabilistic objectives, whereas filtered no-propagation has the lowest median CRPS. On S004, their NLL intervals overlap. In terms of temporal/spectral reconstruction, filtered no-propagation has the lowest median D_{H} on S001 and S002, while KAFFEE has the lowest on S003–S005. On S002, frozen-core calibration also has a lower D_{H} than KAFFEE.

Together, these results support the feasibility of probabilistically adapting deterministic surrogates to real-world data. However, since the true latent dynamics are unknown, an improved fit to the observed recordings cannot establish recovery of the underlying deterministic dynamics, nor can we directly verify the DPC gap failure modes (core collapse, noise masking, or blind uncertainty). Proxies for metrics that can reliably assess these properties without oracle knowledge warrant further research. The mechanistic evidence for the DPC gap failure modes thus remains in our controlled/synthetic Lorenz-63, Lorenz-96, and DynaMix experiments.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our main claims in the abstract and the introduction. We validate them through the mechanistic Lorenz-96 ablation and DynaMix adaptation experiments, and through additional supporting results in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We summarize limitations in Section 7, and discuss each of the mentioned limitations in more detail in the respective appendix sections.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We state the assumptions underlying the DPC-gap definition and the covariance-volume mechanism in the main text, and provide formal derivations and proofs in the appendix, including for the open-loop volume-pressure argument, EKF equations, and anti-contraction analysis of the NLL. Informal intuition is given in the main text and detailed derivations alongside illustrations are given in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main text describes the experimental settings, baselines, evaluation axes, metrics, noise levels, and number of seeds. Full implementation details, optimization schedules/algorithm, metric definitions, hyperparameters, and dataset-generation procedures are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code that contains implementations of our experiments, the data that was used, and the trained models at https://github.com/aherz/kaffee-dpc.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all training and test details necessary to understand and reproduce the results. The code we provide contains configuration files for each experiment that also specifies the used settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main figures and tables report medians with robust standard errors over 20 random seeds. We specify the number of seeds and the type of uncertainty reported. These error bars are shown for the experiments supporting the main claims. The figure captions and metric section in the appendix specify how errors were aggregated and the sample counts: for the Lorenz-63/96 and the EEG experiments, we use individual training

seeds as statistical unit. For DynaMix, we aggregate individual per-system seeds over the 13 tested systems. The resulting error bars describe the corresponding dispersion, they are not formal hypothesis tests of statistical equivalence.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss scaling in Appx. A.7 and report measured GPU times and peak memory allocation for the main experiments in Appx. C.6. Those measurements were taken using NVIDIA RTX PRO 6000 Blackwell GPUs. They exclude preliminary and exploratory runs that did not enter the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is methodological and uses synthetic dynamical systems, previously published model assets, and secondary analysis of existing public EEG recordings (from the "PhysioNet" dataset [91]; see Appx. D.6). We reviewed the NeurIPS Code of Ethics and our work conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work is foundational/methodological research on probabilistic modeling of dynamical systems. We briefly discuss potential positive and negative impacts in the paper. In summary, potential positive impacts include improved uncertainty-aware modeling for scientific forecasting, control, active data collection/experimental design, and discovery. Potential negative impacts are indirect and dependent on the application: inaccurate probabilistic surrogates could lead to overconfident downstream decisions in scientific or engineering applications. The paper discusses the need for careful evaluation of dynamical systems models to mitigate such risks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release such high-risk models or other assets with a substantial risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original sources for existing assets that we use in the paper, including data (such as Lorenz-96/63, the benchmark dataset containing 13 chaotic systems, and the studied EEG recordings [29, 54, 90, 91]) and previously published model classes or algorithms (such as DynaMix, AL-RNNs, and shPLRNNs [23, 34, 53]), and specify their licenses in the corresponding experimental-details sections.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We do not introduce a new dataset or benchmark asset that would require separate documentation. The synthetic data-generation procedures and experimental configurations are described in the paper. The model assets alongside training and licenses are included in the provided code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve participant recruitment, crowdsourcing, or compensation. Our EEG experiment reuses existing public recordings [91], as described in Appx. D.6.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Experiments were performed using synthetic data except for the EEG recordings in Appx. D.6. However, this experiment was a secondary analysis of existing public recordings. We did not recruit participants or collect new human data.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Our core method development, experiments, and scientific contributions do not involve LLMs as an important, original, or non-standard component. LLMs were used to support writing, editing and formatting, and for implementing methods and editing code. All LLM-assisted work was reviewed and verified by the authors; we take responsibility for the final content of this work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.