# INTERPRETABLE DEEP STATE SPACE MODELS VIA SEMANTIC ALIGNMENT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Deep state-space models (DSSMs) are promising tools to control complex dynamical systems, but their latent states are typically unintelligible to humans and therefore difficult to inspect, steer, and debug. In response, we propose a general methodology for constructing expressive and *intrinsically interpretable* DSSMs that permit the alignment of both their state representations and dynamics with human semantics. Specifically, we decompose a DSSM's internal representation into an expert-defined dynamical concept state and a residual latent state, which is learned to compensate for the predictive information that is missing from the concepts. Notably, the resulting formulation supports interventions that propagate through the learned dynamics and steer the subsequent concept trajectory. We instantiate the methodology in both deterministic and stochastic model variants and evaluate them across four dynamical systems, demonstrating free-running concept forecasting accuracy on par with opaque DSSMs while also enabling interventions.

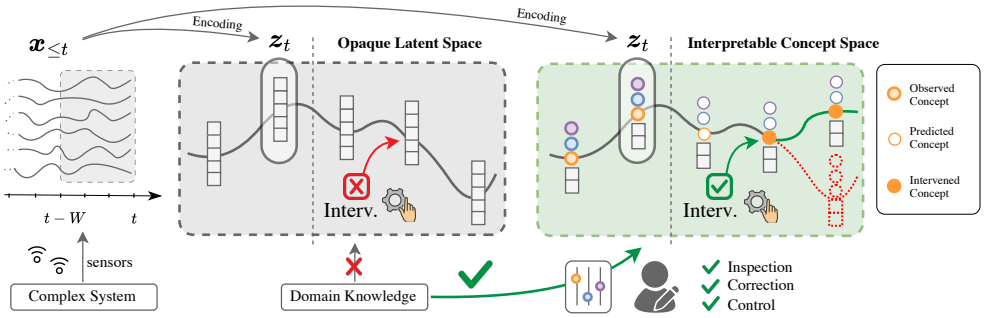


Figure 1: A standard DSSM encodes sensor measurements $\boldsymbol{x}_{\leq t}$ into unintelligible latent states $\boldsymbol{z}_t$ (**left, grey box**). An interpretable DSSM decomposes $\boldsymbol{z}_t$ into concepts (coloured circles) and a residual state (white squares), aligning both representations and dynamics with human semantics (**right, green box**). An expert can therefore inspect predicted concepts and steer the trajectory.

## 1 INTRODUCTION

Forecasting and controlling adaptive complex systems is becoming an urgent need (Kolt et al., 2025). Learned models are increasingly deployed across safety-critical domains and, above all, AI agents now execute long, multi-step tasks, with the first incidents already reported (Greenblatt et al., 2026). In such systems, safety failures often emerge from a trajectory of actions rather than from any single step (Kutasov et al., 2025; Leveson, 2016; Yueh-Han et al., 2025), so users must be able to inspect the state a model carries forward and intervene on it before damage accumulates (Rudin et al., 2022). Deep State-space models (DSSMs) are an effective formalism for modeling complex dynamical systems (Gedon et al., 2021; Ghadami & Epureanu, 2022; Hafner et al., 2025). Unlike predictors that map an observation window directly to future values, DSSMs explicitly learn an internal state that can be propagated in time under changing conditions or actions, making them useful for decision-making and feedback control (Girin et al., 2022; Nejatbakhsh & Wang, 2025; Pillonetto et al., 2025).

Yet, as in most deep time series models, nothing in a standard DSSM constrains its internal state to correspond to quantities that domain experts can inspect or act on, limiting deployment exactly where



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

they are more promising (Coussement et al., 2024; Ehrmann et al., 2023). A compelling solution to DSSMs' opacity could be grounded in the concept-based paradigm (Knab et al., 2026; Koh et al., 2020), which routes model predictions through human-understandable variables so that explanations are faithful by construction and thus auditable. However, existing instantiations are tailored to static data and cannot model how their interpretable state evolve over time (De Felice et al., 2026).

In this work, we first contribute an **extension of the SSM formulation to expose expert-defined concepts** as a process emitted from the latent state alongside the observations. This provides a formal basis for the task of concept forecasting, which we formalize under two interpretability desiderata on the model's state and on its transition law. Second, we bridge DSSMs and concept-based interpretability to **propose a general methodology for designing expressive and intrinsically interpretable DSSMs** (Figure 1). Specifically, the concept state is paired with a residual state designed to account for the predictive information missing from concepts. Concepts and residual are used jointly for accurate rollouts, while ablating the residual recovers the interpretable concept-only transition. Experts can therefore read the concepts as variables they understand at every step, inspect the mechanism through which those concepts evolve, and intervene on either. We instantiate the methodology in deterministic and stochastic DSSM variants. On a free-running concept forecasting task, both instantiations **match opaque alternatives** in accuracy while **supporting interventions** that propagate through the learned dynamics.

We view this contribution as a step toward deep time series models that are accurate and interpretable, and thus better suited to guide scientific reasoning, decision-making, and high-stakes deployment.

## 2 Problem Formalization

We first extend the SSM data-generating formulation to account for interpretable concepts (Section 2.1). We then formalize our objective of forecasting future concept trajectories under interpretability desiderata, and identify when these are compatible with accurate forecasting (Section 2.2).

### 2.1 State-space models with dynamical concepts

A classical SSM formulation models the *observation* process $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$ as generated from a *latent state* process $\boldsymbol{s}_t \in \mathbb{R}^{d_s}$ (Durbin & Koopman, 2012). Specifically, we consider the setting in which $\boldsymbol{x}_t$ are high-dimensional measurements of a complex system and are therefore not the variables through which domain experts describe and reason about the system (Gil et al., 2021; Marcinkevičs & Vogt, 2023). Instead, we assume reasoning relies on a small number of high-level variables that describe the system at a coarser scale. For example, an engineer monitoring a turbofan would not reason about raw sensor readings but about component efficiency, or a portfolio manager not about individual asset prices but about relative sector strength. To account for this distinction, we introduce a *concept* process, $\boldsymbol{c}_t \in \mathbb{R}^{d_c}$. We assume both observations and concepts are conditionally dependent on the underlying latent state $\boldsymbol{s}_t$, as shown in Figure 2:

$$\text{latent:} \quad \boldsymbol{s}_{t+1} \sim p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t) \tag{1a}$$

$$\text{observations:} \quad \boldsymbol{x}_t \sim p(\boldsymbol{x}_t \mid \boldsymbol{s}_t) \tag{1b}$$

$$\text{concepts:} \quad \boldsymbol{c}_t \sim p(\boldsymbol{c}_t \mid \boldsymbol{s}_t) \tag{1c}$$

with $\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)$, where $p$ denotes the true data-generating law and the conditionals are assumed time-invariant. The concept process in Equation (1c) is in general non-Markovian, meaning $\boldsymbol{c}_t$ alone may not suffice to predict $\boldsymbol{c}_{t+1}$. We instead assume *observational sufficiency*, that is, the concept history $\boldsymbol{c}_{\leq t}$ conveys no predictive information about $\boldsymbol{c}_{t+1}$ beyond what is already in the observation history $\boldsymbol{x}_{\leq t}$:

$$p(\boldsymbol{c}_{t+1} \mid \boldsymbol{x}_{\leq t}, \boldsymbol{c}_{\leq t}) = p(\boldsymbol{c}_{t+1} \mid \boldsymbol{x}_{\leq t}) \qquad \textit{for all } t. \tag{2}$$

This is automatic whenever the concepts are a function of the observation history, and restricts the data-generating process otherwise.

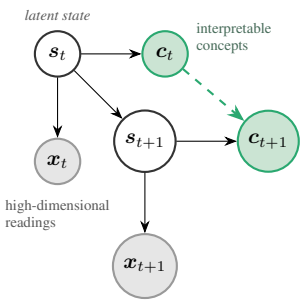


Figure 2: Data-generating process. The dashed arrow denotes the induced concepts' dynamics.

### 2.2 Problem formulation under interpretability desiderata

**Free-running concept forecasting.** We consider the problem of learning the joint predictive distribution over the future concept trajectory given the observations available up to time $t$:

$$p(\boldsymbol{c}_{t:t+H} \mid \boldsymbol{x}_{\leq t}) \qquad \forall\, H \in \mathbb{N}_+, \tag{3}$$

where $H$ is the forecast horizon. In the free-running regime considered here, observations $\boldsymbol{x}_{\leq t}$ are used to initialize the model, which is then propagated autonomously up to each forecast horizon $H$. To learn Equation (3), we assume access to a dataset $D = \{\boldsymbol{x}_{0:T-1}, \boldsymbol{c}_{0:T-1}\}$ of paired observation and concept sequences, where concepts may be given by ground-truth labels, by an expert-defined rule, or generated artificially using scalable concept discovery pipelines (Oikarinen et al., 2023).

We study the learning problem in Equation (3) under the following interpretability desiderata, collectively termed *semantic alignment* (De Felice et al., 2026):

**Desideratum D1** (Concept alignment)**.** *At each time $t$, the model must expose internal concept variables $\boldsymbol{c}_t$ whose conditional distribution, given the observation history $\boldsymbol{x}_{\leq t}$, equals that of the expert-defined concepts. Letting $p_\theta$ denote the law induced by the model, we require*

$$p_\theta(\boldsymbol{c}_t \mid \boldsymbol{x}_{\leq t}) = p(\boldsymbol{c}_t \mid \boldsymbol{x}_{\leq t}); \quad p_\theta(\boldsymbol{c}_{t+1} \mid \boldsymbol{x}_{\leq t}) = p(\boldsymbol{c}_{t+1} \mid \boldsymbol{x}_{\leq t}) \qquad \textit{for all } t.$$

D1 parallels the concept encoding constraint in concept-based models (Barbiero et al., 2026; Poeta et al., 2023), with the distinction that alignment must hold at every step over time.

**Desideratum D2** (Mechanism alignment)**.** *The model must expose a concept-level transition whose conditional distribution $p_\theta$ belongs to a user-specified admissible class. Specifically,*

$$p_\theta(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t) \in \mathcal{M} \qquad \textit{for all } t,$$

*where $\mathcal{M}$ denotes the class of admissible mechanisms derived from domain knowledge, such as physical laws, functional properties, or hard constraints.*

D2 mirrors the mechanism alignment condition of Barbiero et al. (2026); De Felice et al. (2026), applied here to the concept-level transition. Note that, if the concept process is *dynamically closed* (Bertschinger et al., 2006), i.e., $p(\boldsymbol{c}_{t+1} \mid \boldsymbol{x}_{\leq t}, \boldsymbol{c}_{\leq t}) = p(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t)$ for all $t$, then the concepts $\boldsymbol{c}_t$ carry all the information necessary for predicting the next concept state $\boldsymbol{c}_{t+1}$, and the rollout problem of Equation (3) decomposes as:

$$p(\boldsymbol{c}_{t:t+H} \mid \boldsymbol{x}_{\leq t}) = \prod_{h=0}^{H-1} \underbrace{p(\boldsymbol{c}_{t+h+1} \mid \boldsymbol{c}_{t+h})}_{\text{concept-only transition}} \cdot \underbrace{p(\boldsymbol{c}_t \mid \boldsymbol{x}_{\leq t})}_{\text{encoding}}. \tag{4}$$

Under dynamical closure, the transition of D2 is therefore sufficient on its own: a concept encoder and repeated application of a concept-only transition reproduce the target law.

Closure, however, is false in general. Concepts are a low-dimensional compression of the state and therefore generally omit the information necessary for predicting their own evolution (this is the temporal analogue of concept incompleteness (Yeh et al., 2020)). Under D1, requiring D2 to be the only mechanism contributing to the concept transition therefore compromises forecasting accuracy during rollout. In Section 3, we address this methodological problem by adding a residual pathway carrying the predictive information missing from the concepts. Switching it off recovers the transition of D2 exactly, so that each prediction remains attributable to the concepts that drive it.

## 3 Methodology

In the following, we detail a methodology for building DSSMs that satisfies the interpretability desiderata of Section 2 without assuming dynamical closure (Section 3.1). We then detail its training (Section 3.2) and concept interventions (Section 3.3). A graphical support is in Figure 3.

### 3.1 Concept-residual decomposition

**Concept closure gap.** Our goal is to construct a model that exposes the concepts as part of its state and preserves an explicit concept-only transition belonging to an admissible class $\mathcal{M}$ (D1–D2). Let

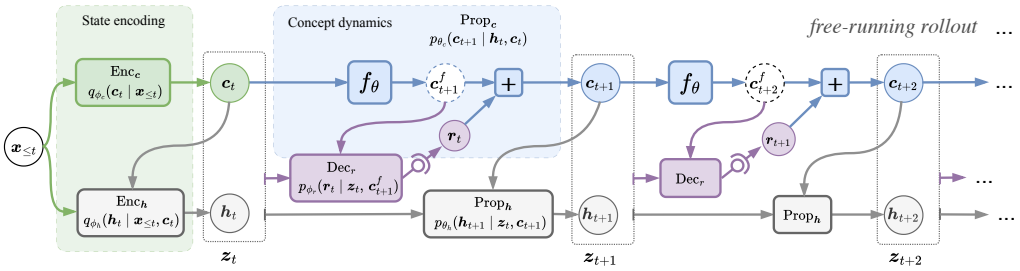


Figure 3: Computational graph of the proposed methodology. The observation history initializes $(\boldsymbol{c}_t, \boldsymbol{h}_t)$; thereafter the admissible transition $f_\vartheta$ proposes $\boldsymbol{c}^f_{t+1}$, the residual decoder corrects it to $\boldsymbol{c}_{t+1} = \boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t$, and $\textsc{Prop}_{\boldsymbol{h}}$ advances $\boldsymbol{h}_t$. Setting $\boldsymbol{r}_t = \boldsymbol{0}$ recovers $f_\vartheta$.

$f_\vartheta(\cdot \mid \boldsymbol{c}_t) \in \mathcal{M}$ denote the model concept-only transition. As noted in Section 2.2, propagating with such a map reproduces the target law of Equation (4) only if the concept process is dynamically closed, which is false in general. We measure the deviation from closure via the *concept closure gap*:

$$\Delta_c := I(\boldsymbol{c}_{t+1}; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t), \tag{5}$$

where $I(\cdot\,;\cdot \mid \cdot)$ is the conditional mutual information. $\Delta_c$ quantifies the information that $\boldsymbol{x}_{\leq t}$ carries about $\boldsymbol{c}_{t+1}$ beyond $\boldsymbol{c}_t$. Under observational sufficiency (Equation (2)), this quantity vanishes iff the concept process is dynamically closed. To isolate precisely what is missing from the admissible transition, we define the *residual*

$$\boldsymbol{r}_t := \boldsymbol{c}_{t+1} - \boldsymbol{c}^f_{t+1}, \tag{6}$$

where $\boldsymbol{c}^f_{t+1} \sim f_\vartheta(\cdot \mid \boldsymbol{c}_t)$. Given that $f_\vartheta(\cdot \mid \boldsymbol{c}_t)$ is conditioned on $\boldsymbol{c}_t$ only, the proposal $\boldsymbol{c}^f_{t+1}$ is therefore available at time $t$. Hence, $I(\boldsymbol{r}_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1}) = I(\boldsymbol{c}_{t+1}; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \Delta_c$, so the residual $\boldsymbol{r}_t$ carries the predictive information in $\boldsymbol{x}_{\leq t}$ missing from the concept state, which is exactly what we want to model.

**Extending concept state.** To retain the predictive information missing from the concept states, we consider an augmented state $\boldsymbol{z}_t = (\boldsymbol{c}_t, \boldsymbol{h}_t)$, where $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$ is a hidden state. The encoding distribution of the augmented state factorizes by the chain rule as

$$q_\phi(\boldsymbol{z}_t \mid \boldsymbol{x}_{\leq t}) = \underbrace{q_{\phi_c}(\boldsymbol{c}_t \mid \boldsymbol{x}_{\leq t})}_{\textsc{Enc}_{\boldsymbol{c}}} \underbrace{q_{\phi_h}(\boldsymbol{h}_t \mid \boldsymbol{x}_{\leq t}, \boldsymbol{c}_t)}_{\textsc{Enc}_{\boldsymbol{h}}}.$$

The first factor $\textsc{Enc}_{\boldsymbol{c}}$ encodes the semantic component of the state and, to satisfy concept alignment (D1), will be required to align with ground-truth concepts. The second factor $\textsc{Enc}_{\boldsymbol{h}}$ initializes the hidden state $\boldsymbol{h}_t$ that, unlike $\boldsymbol{c}_t$, is not required to have human-interpretable semantics.

**Decomposing the state dynamics.** To close the gap $\Delta_c$ during rollout, it is required to keep the information in $\boldsymbol{z}_t$ available at each time step. This motivates a map $p_\theta(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t)$ that propagates $\boldsymbol{h}_t$ and $\boldsymbol{c}_t$ jointly. This factorizes as

$$p_\theta(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t) = \underbrace{p_{\theta_c}(\boldsymbol{c}_{t+1} \mid \boldsymbol{h}_t, \boldsymbol{c}_t)}_{\textsc{Prop}_{\boldsymbol{c}}} \underbrace{p_{\theta_h}(\boldsymbol{h}_{t+1} \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}_{t+1})}_{\textsc{Prop}_{\boldsymbol{h}}}.$$

This factorization makes explicit where the information missing from the concept dynamics enters the rollout. Through $\textsc{Prop}_{\boldsymbol{c}}$, $\boldsymbol{h}_t$ can account for the predictive information about $\boldsymbol{c}_{t+1}$ not determined by $\boldsymbol{c}_t$ alone, while $\textsc{Prop}_{\boldsymbol{h}}$ retains such information for subsequent transitions.

**Residual decoder.** We now make explicit within $\textsc{Prop}_{\boldsymbol{c}}$ the contribution already accounted for by the admissible transition $f_\vartheta$. Using the reparameterization $\boldsymbol{r}_t = \boldsymbol{c}_{t+1} - \boldsymbol{c}^f_{t+1}$, we can derive the residual decoder $\textsc{Dec}_{\boldsymbol{r}}$ as the law of $\boldsymbol{r}_t$ induced by $\textsc{Prop}_{\boldsymbol{c}}$ as

$$p_{\theta_c}(\boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t \mid \boldsymbol{h}_t, \boldsymbol{c}_t) = \underbrace{p_{\theta_r}(\boldsymbol{r}_t \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1})}_{\textsc{Dec}_{\boldsymbol{r}}}.$$

We refer to Proposition B.1 for details. Together, $f_\vartheta$ (regardless of its choice) and $\textsc{Dec}_{\boldsymbol{r}}$ therefore provide an equivalent additive realization of $\textsc{Prop}_{\boldsymbol{c}}$ that does not compromise expressivity.

Crucially, expressing the residual as an additive correction in concept space preserves attribution by construction: $f_\vartheta$ is exactly recoverable by setting $\boldsymbol{r}_t = \mathbf{0}$, while the residual contribution to each concept update can be directly quantified through $\boldsymbol{r}_t$. Note how conditioning $\text{DEC}_{\boldsymbol{r}}$ explicitly on $(\boldsymbol{c}_t, \boldsymbol{c}^f_{t+1})$ provides a direct path through which concept interventions can affect the residual correction, although this is not guaranteed (Espinosa Zarlenga et al., 2023). We deepen this in Section 3.3. Finally, the construction is not tied to any particular DSSM architecture: for any $f_\vartheta \in \mathcal{M}$, any first-order DSSM exposing concept states as in Equation (1) admits an equivalent concept–residual decomposition of the form above (see Proposition B.1).

*Remark* 3.1. Requiring the residual to carry the predictive information missing from the concepts does not require it to carry *only* that information. Formally, $I(\boldsymbol{r}_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \Delta_c \implies I(\boldsymbol{r}_t; \boldsymbol{c}_t) = 0$ (see Proposition B.3), meaning nothing prevents the residual from also encoding concept information. On the contrary, such information is needed (i) to update the residual after a concept intervention, and (ii) in stochastic settings, to condition the correction on the sampled $\boldsymbol{c}_t$, so that it compensates for exactly what that realization misses. This distinguishes our approach from methods that impose orthogonality between concepts and residual (Ismail et al., 2023; Zabounidis et al., 2026).

### 3.2 Training strategy

If all maps were learned jointly, the concepts could encode information about $\boldsymbol{x}_{\leq t}$ beyond what they represent, trading concept alignment for forecasting accuracy. The ablation $\boldsymbol{r} = \mathbf{0}$ would then no longer recover a concept-only prediction, since $f_\vartheta$ would have been fitted to concepts carrying such extra information. This is an instance of *concept leakage*, a well-documented concern of concept-based models also known to reduce the effect of interventions (Debot & Marra, 2025; Espinosa Zarlenga, 2026; Havasi et al., 2022; Mahinpei et al., 2021). To prevent leakage, we train the concept pathway first and keep it frozen while learning the residual one.

**Stage 1: concept pathway.** First, we instantiate $\text{ENC}_{\boldsymbol{c}}$ and $f_\vartheta \in \mathcal{M}$ based on the available domain knowledge. Either may be given a priori, as when the concepts are a known function of the observations or their dynamics is a known (even approximate) law, in which case the learning is skipped. If unknown, either can be instantiated as a neural network, with parameters learned by supervision on the concept data. In particular, we fit the two maps independently (Koh et al., 2020):

$$\min_{\phi_c} \mathbb{E}[\ell(q_{\phi_c}(\cdot \mid \boldsymbol{x}_{\leq t}), \boldsymbol{c}_t)], \qquad \min_{\vartheta:\, f_\vartheta \in \mathcal{M}} \mathbb{E}\left[\sum_{h=0}^{H-1} \ell(f_\vartheta(\cdot \mid \boldsymbol{c}_{t+h}), \boldsymbol{c}_{t+h+1})\right],$$

this prevents leaked information from the observations $\boldsymbol{x}_{\leq t}$ from propagating into the training of the concept-only transition. The training of $f_\vartheta$ can be regularized by ground-truth concept interventions during training (Espinosa Zarlenga et al., 2022). We study sensitivity to this choice in Appendix F.4.2.

**Stage 2: residual pathway.** With the concept pathway fixed, we draw $\boldsymbol{c}^f_{t+1} \sim f_\vartheta(\cdot \mid \boldsymbol{c}_t)$ for each transition and obtain the target $\boldsymbol{r}_t$ from Equation (6). We then train $\text{ENC}_{\boldsymbol{h}}$, $\text{PROP}_{\boldsymbol{h}}$ and $\text{DEC}_{\boldsymbol{r}}$ so that the decoded residual predicts $\boldsymbol{r}_t$:

$$\min_{\phi_h, \theta_h, \theta_r} \mathbb{E}\left[\sum_{h=0}^{H-1} \ell\Big(p_{\theta_r}(\cdot \mid \boldsymbol{h}_{t+h}, \boldsymbol{c}_{t+h}, \boldsymbol{c}^f_{t+h+1}), \boldsymbol{r}_{t+h}\Big)\right], \quad \text{with } (\phi_c, \vartheta) \text{ fixed},$$

where the contribution at $h = 0$ trains $\text{ENC}_{\boldsymbol{h}}$ to extract the correction from the observation history, the $h > 0$ terms train $\text{PROP}_{\boldsymbol{h}}$ to carry it forward during the rollout, and every term trains $\text{DEC}_{\boldsymbol{r}}$ to decode it. The loss $\ell$ is instantiated as squared error in the deterministic variant and as negative log-likelihoods in the stochastic one; details are in Appendix E.

### 3.3 Interventions and model steerability

The presented methodology enables two distinct kinds of interventions: directly on the concept state and on the concept-only transition.

**Concept interventions.** At an intervention step $\bar{t}$, we replace one or more coordinates of the predicted concept state $\boldsymbol{c}_{\bar{t}}$ with corresponding ground-truth values, obtaining $\widetilde{\boldsymbol{c}}_{\bar{t}}$. We remark that, since both the admissible proposal and the residual are conditioned on the current concept state, the residual is recomputed, and the intervention propagates to $\widetilde{\boldsymbol{c}}_{\bar{t}+1}$ and, through the hidden state, to the entire

subsequent rollout (Figure 1). The effect of a concept intervention on the residual can be quantified as follows. For a rollout lag $\tau \geq 0$, let $\boldsymbol{r}_{\bar{t}+\tau}$ and $\widetilde{\boldsymbol{r}}_{\bar{t}+\tau}$ denote the residual corrections at step $\bar{t}+\tau$ without and with the intervention, and let $S^r_{\bar{t}+\tau} := W_2(\boldsymbol{r}_{\bar{t}+\tau}, \widetilde{\boldsymbol{r}}_{\bar{t}+\tau})$ be their 2-Wasserstein distance. For $\widetilde{\boldsymbol{c}}_{\bar{t}} \neq \boldsymbol{c}_{\bar{t}}$, we define the *residual intervention responsiveness* as

$$\mathcal{R}_{\bar{t}}(\tau) := \frac{S^r_{\bar{t}+\tau}}{\|\widetilde{\boldsymbol{c}}_{\bar{t}} - \boldsymbol{c}_{\bar{t}}\|_2}. \tag{7}$$

In Appendix B.2, we show that $S^r_{\bar{t}}$ bounds how far the complete one-step response can deviate from that of the admissible transition (Proposition B.4).

**Mechanism interventions.** The exposed admissible transition also provides a natural interface for intervening on the mechanism itself, by replacing $f_\vartheta \in \mathcal{M}$ with another admissible transition $\widetilde{f} \in \mathcal{M}$. Yet, the residual decoder was not trained under this new distribution $\widetilde{f}$, so mechanism interventions may induce a distribution shift in the residual pathway. We explore this briefly in Section 4.3 and leave a principled treatment of mechanism interventions to future work.

## 4 Experimental Evaluation

We investigate free-running forecasting accuracy (Section 4.1), semantic alignment of the state (Section 4.2), concept and mechanism interventions (Section 4.3), and robustness to the quality of prior knowledge (Section 4.4). Appendix F provides additional results.

**Datasets.** We consider four systems that vary how concepts are obtained and how the admissible mechanism is specified (Table 1): i) the Lorenz–96 (Lorenz, 1996) with slow variables as concepts and fast variables as observations; ii) an agent-based Lotka–Volterra model (Lotka, 1925), where the concepts are the total number of prey and predators; iii) Relative Rotation Graphs (RRG) use real daily prices of equity sectors, with the RS-Ratio and RS-Momentum of each sector as concepts; iv) N-CMAPSS (Arias Chao et al., 2021) simulates turbofan run-to-failure trajectories, with sensor and operating-condition channels as observations and component health parameters as concepts. Appendix C details each system.

| System | $d_x$ | $d_c$ | $\text{ENC}_{\mathbf{c}}$ (D1) | $\mathcal{M}$ (D2) |
|---|---|---|---|---|
| Lorenz–96 | 360 | 36 | learned | rule-based |
| Lotka–Volterra | 5000 | 2 | rule-based | rule-based |
| RRG (equities) | 47 | 18 | rule-based | rule-based |
| N-CMAPSS | 38 | 7 | learned | learned |

Table 1: Specifications for each dataset.

**Baselines.** We consider both deterministic and stochastic variants of DSSMs. As opaque baselines with unstructured latent states, we use a deep Koopman autoencoder (Lusch et al., 2018) as the deterministic variant and a Deep Markov Model (Krishnan et al., 2017) as the stochastic one. Instantiating the methodology of Section 3 with each baseline re-designs them into its interpretable counterpart (IntDSSMs). All models share data splits, context window length $W$, forecast horizons $H$, and checkpoint selection. All reported uncertainties represent 2 sample mean $\sigma$ across 5 independent runs. Model instantiations are detailed in Appendix D and training settings in Appendix E.

### 4.1 Concept forecasting

Our first experiment evaluates forecasting accuracy. Figure 4 reports the NRMSE between predicted and ground-truth concepts at each horizon $h \in [0, H]$, together with the ablation $\boldsymbol{r} = \mathbf{0}$ of IntDSSM.

**Free-running concept forecasting performance of IntDSSMs is on par with opaque DSSMs.** This is due to the residual, which makes the interpretable architecture as expressive as the opaque variant. Minor differences outside the reported uncertainties are attributable to the finite setting rather than to the expressiveness of the models, and could be reduced through a more extensive model search.

**The residual contribution is measurable and small where the prior is accurate.** Ablating the residual measures how much of each forecast the admissible mechanism explains on its own (D2). On RRG and N-CMAPSS, IntDSSM($\boldsymbol{r} = \mathbf{0}$) is close to the full model, so forecasts are almost entirely attributable to the expert mechanism. On Lotka–Volterra, the mean-field law cannot capture spatial structure and the ablation raises the error by up to an order of magnitude; on Lorenz–96, the gap grows with the horizon as the effect of the unresolved fast variables accumulates.

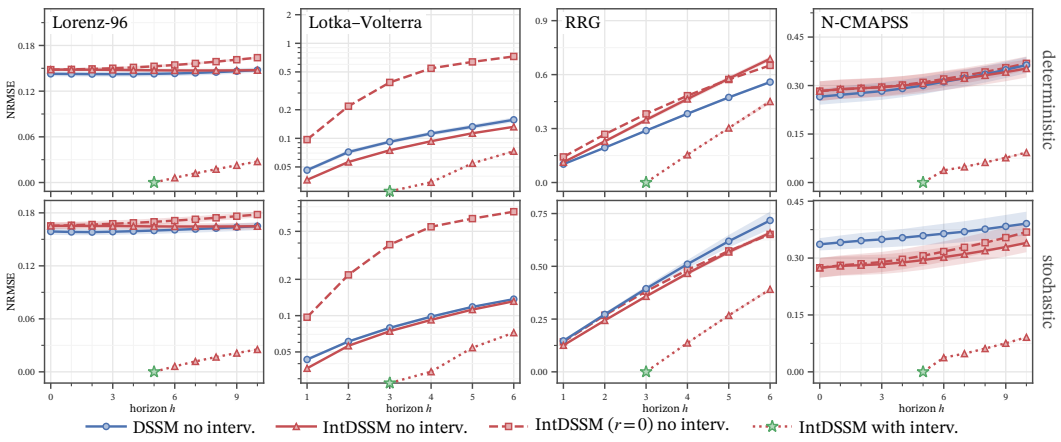


Figure 4: **Rollout error.** Concept NRMSE per horizon $h$, deterministic (top) and stochastic (bottom).

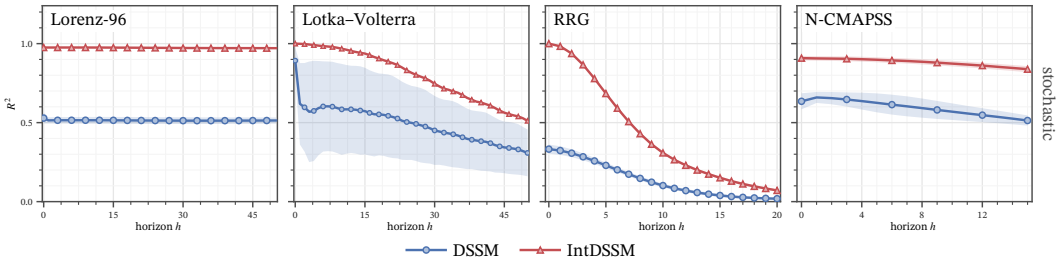


Figure 5: **Semantic alignment.** $R^2$ between ground-truth concepts and model state per horizon $h$.

### 4.2 Concept Alignment

We measure concept alignment (D1) during rollout as the $R^2$ between each ground-truth concept and its coordinate in the propagated state. For IntDSSM, this is the corresponding concept coordinate. For the opaque baseline, it is the most correlated latent coordinate, reselected at each step. We average over concepts and report the stochastic variant in Figure 5 (deterministic in Appendix F.1). **Concepts show strong semantic alignment that also persists during rollout.** On all systems, IntDSSMs achieve near-perfect semantic alignment ($R^2 \approx 1$) at the initial steps, and their alignment decreases during rollout as forecasting error accumulates. The opaque baselines consistently sit well below the interpretable counterparts, although forecasting accuracy is comparable. This implies accurate forecasting does not by itself produce latent coordinates that include concepts.

### 4.3 Concept and Mechanism Interventions

We now demonstrate that our methodology supports interventions on concept and mechanisms.

**Improvements from concept interventions propagate during rollout.** We replace the predicted concepts at an intermediate step $\bar{t}$ with their ground-truth values and let the model rollout autonomously (Figure 4, dotted red curve). As expected, the error vanishes at the intervention step. Remarkably, the effect of the intervention propagates, and the error remains below that of the unintervened curve. Opaque DSSMs admit no such intervention, since no coordinate of their state corresponds to a concept. Sensitivity to noisy interventions is measured in Appendix F.3.

**Conditioning the residual on the concept state increases responsiveness to concept interventions.** We compare IntDSSMs against an ablation where the residual decoder $\text{DEC}_{\boldsymbol{r}}$ *is not* conditioned on the concept state $\boldsymbol{c}_t$. For both, we perform concept intervention and measure the responsiveness of the residual using Equation (7). Results are in Table 2, while Appendix F.2.1 reports the response over longer lags. Without conditioning on the concept state, the residual cannot react at the intervention step ($\mathcal{R}_{\bar{t}}(0) = 0$) and responds only with a delay, often more weakly. This also compromises interventional accuracy, as shown in Appendix F.2.2.

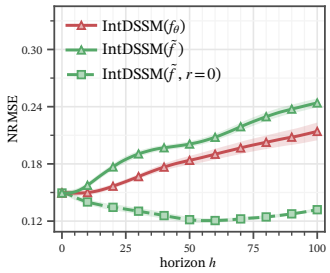


Figure 6: **Mechanism intervention on Lorenz–96.** NRMSE after replacing the fitted map $f_\vartheta$ with the true mechanism $\tilde{f}$.

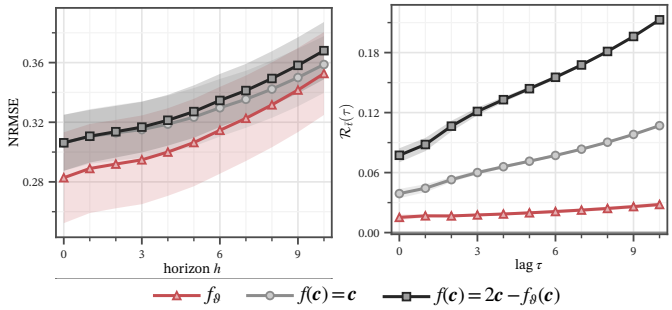


Figure 7: **Domain knowledge on N-CMAPSS.** NRMSE (left) and intervention responsiveness $\mathcal{R}_{\bar{\epsilon}}(\tau)$ of the residual (right) for the fitted mechanism (red) and two unrealistic priors (grey).

**The framework supports mechanism interventions.** Since the concept transition is exposed, an expert can intervene on the mechanism itself by replacing $f_\vartheta$ with another map at test time, without retraining. On Lorenz–96 the true mechanism is known, so we replace the fitted affine map with the solver $\tilde{f}$ (Figure 6). Avoiding retraining the residual path degrades performance w.r.t. the original model. The mechanism and residual are thus separable, but the residual is specific to the mechanism it was trained with, so a mechanism intervention must switch it off or retrain it.

### 4.4 Robustness to prior knowledge

On N-CMAPSS, we replace the fitted monotone decay $f_\vartheta$ with two different priors: persistence, $f(\boldsymbol{c}) = \boldsymbol{c}$, which ignores degradation, and $f(\boldsymbol{c}) = 2\boldsymbol{c} - f_\vartheta(\boldsymbol{c})$, which reverses its direction. The residual pathway is retrained for each prior (Figure 7).

**Forecasting accuracy is robust under different $f_\vartheta$.** Forecasting error after replacing the fitted mechanism with either unrealistic prior stays comparable with the original model within the reported uncertainty. This is, the residual can learn to compensate for the misfit of the mechanism, whatever its quality. This role can be highlighted by measuring the responsiveness of the residual to interventions. As in Figure 7 (right), the residual becomes markedly more responsive to interventions as the prior worsens. A poor prior is therefore visible in the residual, which is the quantity an expert should inspect before attributing a forecast to the mechanism.

| Dataset | Model | Variant | $\tau = 0$ | $\tau = 1$ | $\tau = 2$ |
|---|---|---|---|---|---|
| Lorenz-96 | Det | IntDSSM | **1.71**$_{\pm 0.15}$ | **1.71**$_{\pm 0.14}$ | **1.75**$_{\pm 0.12}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 0.22$_{\pm 0.03}$ | 0.39$_{\pm 0.04}$ |
| | Stoch. | IntDSSM | **1.73**$_{\pm 0.07}$ | **1.90**$_{\pm 0.08}$ | **2.12**$_{\pm 0.09}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 0.62$_{\pm 0.14}$ | 1.06$_{\pm 0.23}$ |
| Lotka–Volt. | Det | IntDSSM | **10.91**$_{\pm 0.83}$ | **10.69**$_{\pm 0.99}$ | **13.41**$_{\pm 0.77}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 4.7$_{\pm 1.8}$ | 8.7$_{\pm 2.1}$ |
| | Stoch. | IntDSSM | **12.34**$_{\pm 0.07}$ | **12.11**$_{\pm 0.15}$ | **14.59**$_{\pm 0.15}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 8.64$_{\pm 0.45}$ | 12.57$_{\pm 0.41}$ |
| RRG | Det | IntDSSM | **3.11**$_{\pm 0.28}$ | **8.96**$_{\pm 0.70}$ | **12.4**$_{\pm 1.2}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 7.78$_{\pm 0.31}$ | 11.84$_{\pm 0.56}$ |
| | Stoch. | IntDSSM | **7.84**$_{\pm 0.79}$ | **7.18**$_{\pm 0.53}$ | **7.21**$_{\pm 0.45}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 4.27$_{\pm 0.36}$ | 6.65$_{\pm 0.68}$ |
| N-CMAPSS | Det | IntDSSM | **1.52**$_{\pm 0.28}$ | **1.68**$_{\pm 0.16}$ | **1.67**$_{\pm 0.15}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 1.02$_{\pm 0.31}$ | 1.43$_{\pm 0.33}$ |
| | Stoch. | IntDSSM | **4.28**$_{\pm 0.68}$ | **4.02**$_{\pm 0.51}$ | **3.84**$_{\pm 0.38}$ |
| | | w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ | 0.0$_{\pm 0.0}$ | 1.10$_{\pm 0.25}$ | 1.12$_{\pm 0.22}$ |

Table 2: Responsiveness $\mathcal{R}_{\bar{\epsilon}}(\tau)$ ($\times 100$) of the residual correction to a concept intervention.

## 5 Related work

**Opaque DSSMs and post-hoc explanations.** DSSMs, from variational SSMs (Fraccaro et al., 2017; Girin et al., 2022; Krishnan et al., 2017) to structured and selective SSMs (Cai et al., 2026; Gu et al., 2022), are *semantically opaque*, meaning their latent states and transitions are not constrained to align with the variables domain experts use to reason about the system (D1). Their explanations are generally produced either post hoc, via input attributions (Bento et al., 2021; Crabbé & Van Der Schaar, 2021; Liu et al., 2024), counterfactual trajectories (Ates et al., 2021; Delaney et al., 2021), latent concept discovery (Gee et al., 2019; Kim et al., 2018; Obermair et al., 2023), or jointly with the forecast (Guo et al., 2024). In both cases, the explanation is not part of the propagated state. It is therefore not guaranteed to reflect the model's computation (Adebayo et al., 2018; Bilodeau et al.,

2024), and does not expose variables a user can inspect or intervene on. In time series, attribution methods have even been shown to disagree on the same predictor (Turbé et al., 2023).

**Intrinsic interpretability.** Intrinsically interpretable models (Marcinkevičs & Vogt, 2023) constrain the predictive process. A first class restricts the transition map to an admissible class: Koopman operators enforce linear latent dynamics (Lusch et al., 2018), SINDy selects sparse terms from a user-specified library (Brunton et al., 2016), and physics-informed networks impose known equations (Raissi et al., 2019). A second class restricts the state. SOM-VAE maps time series onto a discrete latent space with a predefined topology and a Markov model over its states (Fortuin et al., 2018); STEM-LTS decomposes the series into trend, seasonality, and residual, and encodes them into opaque embeddings (Zhao et al., 2025). Neither achieves semantic alignment of variables (D1), which prevents faithful attribution and interventions: constraining only the transitions can still leave the states they act upon with no agreed semantics (De Felice et al., 2026). Similarly, a topology or an independence penalty among states leaves the semantics of each coordinate unconstrained. Concept-based models instead explicitly route predictions through human-understandable variables (Poeta et al., 2023). Concept Bottleneck Models (CBMs) (Koh et al., 2020) have been extended with greater expressivity (Espinosa Zarlenga et al., 2022), leakage control (Zhang et al., 2024), concept generation (Oikarinen et al., 2023), scalable pipelines (Team et al., 2026), causal structure (De Felice et al., 2025), or by constraining the predictor to interpretable functional classes (De Santis et al., 2026; Debot et al., 2024), addressing mechanism alignment in the sense of D2. However, most target static settings, and temporal CBMs predict concepts from a window at each step without a transition that propagates them (Ferfoglia et al., 2024; Forest et al., 2025; Johnson et al., 2022; Knab et al., 2025; van Sprang et al., 2024), hence failing to model their dynamics. A concept intervention therefore impacts only the current prediction without being propagated forward. To our knowledge, our work constitutes the first general instantiation of concept-based interpretability with explicit modeling of dynamically evolving concepts.

## 6 Conclusion

We presented a methodology for building intrinsically interpretable DSSMs, whose state exposes expert-defined concepts and whose concept dynamics are governed by an admissible mechanism specified from domain knowledge. We showed that these two requirements are compatible with accurate forecasting only when the concept process is dynamically closed, and recovered accuracy through a residual that enters as an additive correction in concept space, so that ablating it recovers the admissible transition exactly.

**Applications and broader impact.** Our methodology is designed for decision-making in domains such as predictive maintenance, logistics, macroeconomics, and finance, where systems are measured through high-dimensional signals while experts reason about and control them through a few variables (Coussement et al., 2024; Ehrmann et al., 2023). In security and safety monitoring, the goal is often to detect early that a trajectory is heading toward a harmful state and to intervene before the system reaches it, a setting directly compatible with our methodology. Through intrinsically interpretable DSSMs, an expert can check the inferential process step by step and correct it when outside evidence contradicts it. Finally, we argue our formulation operates at the root level of the state-space formulation itself, it therefore applies to a broad class of models, and we expect it to carry over to recent architectures such as structured and selective SSMs (S4, S5, Mamba, and Mamba-3) (Gu & Dao, 2023; Gu et al., 2022; Lahoti et al., 2026; Smith et al., 2022) and to related lines of research, including world models (Ha & Schmidhuber, 2018; Mao et al., 2026; Schmidhuber, 1991) and joint-embedding predictive architectures (Assran et al., 2023; LeCun et al., 2022).

**Limitations and future work.** The methodology requires concept supervision, through labels or expert rules. Extending the methodology to noisy concept labels, e.g., from LLMs or simulators, would cover settings where manual annotation is infeasible. Moreover, while concept and mechanism alignment are grounded in interpretability principles (Barbiero et al., 2026; De Felice et al., 2026), the additive correction that closes the remaining gap is a design choice these principles do not dictate. Grounding it would require formalizing interpretability requirements on the residual pathway. Finally, instantiating the framework with modern structured and selective SSM architectures, as well as with state-space world models, would demonstrate broader applicability.

## AI USE STATEMENT

In this work, we used generative AI tools for tasks that fall under the required disclosure categories. Specifically, generative AI assisted in (i) brainstorming about the soundness of the proposed methodology; (ii) implementing the code infrastructure for performing the experimental evaluation; (iii) polishing the clarity of the preliminary version of the text, check pointers, and to debug LaTeX; (iv) polishing the presentation of the mathematical proofs. We have reviewed all AI-assisted work. We did not use generative AI tools to develop the conceptual framework, formulate the main theoretical results, propose hypotheses, or design the experiments. Mathematical proofs have been formulated and checked by the authors. The authors take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

## ETHICS STATEMENT

This paper presents work whose goal is to advance fundamental aspects of the field of Machine Learning. We believe this work does not involve ethical considerations beyond those associated with the publicly available datasets.

## REPRODUCIBILITY STATEMENT

The methodology is specified in Section 3, with training in Section 3.2 and interventions in Section 3.3. Datasets are detailed in Appendix C. Architectural details and hyperparameters are reported in Appendix D. Training settings are reported in Appendix E. A repository with code to reproduce all experiments will be provided publicly upon acceptance.

## REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31, 2018.

Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1):5, 2021.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

Emre Ates, Burak Aksar, Vitus J. Leung, and Ayse K. Coskun. Counterfactual Explanations for Multivariate Time Series. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, pp. 1–8. IEEE, 2021. ISBN 978-1-7281-5934-8. doi: 10.1109/ICAPAI49758.2021.9462056.

Pietro Barbiero, Giovanni De Felice, Mateo Espinosa Zarlenga, Francesco Giannini, Filippo Bonchi, Mateja Jamnik, Giuseppe Marra, and Ruggero Noris. The standard interpretable model: A general theory of interpretable machine learning to deductively design interpretable methods using lagrangian mechanics. *arXiv preprint arXiv:2606.12289*, 2026.

Joao Bento, Pedro Saleiro, Andre F Cruz, Mario A T Figueiredo, and Pedro Bizarro. Timeshap: Explaining recurrent models through sequence perturbations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2565–2573, 2021.

Nils Bertschinger, Eckehard Olbrich, Nihat Ay, and Jürgen Jost. Information and closure in systems theory. In *Explorations in the Complexity of Possible Life. Proceedings of the 7th German Workshop of Artificial Life*, pp. 9–21. IOS Press: Amsterdam, The Netherlands, 2006.

Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120, 2024.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Xiuding Cai, Xueyao Wang, Yaoyao Zhu, and Yu Yao. Mambats: improved selective state space models for long-term time series forecasting. *Pattern Recognition*, pp. 114536, 2026.

Kristof Coussement, Mohammad Zoynul Abedin, Mathias Kraus, Sebastián Maldonado, and Kazim Topuz. Explainable ai for enhanced decision-making, 2024.

Jonathan Crabbé and Mihaela Van Der Schaar. Explaining time series predictions with dynamic masks. In *International conference on machine learning*, pp. 2166–2177. PMLR, 2021.

Giovanni De Felice, Arianna Casanova Flores, Francesco De Santis, Silvia Santini, Johannes Schneider, Pietro Barbiero, and Alberto Termine. Causally reliable concept bottleneck models. *Advances in Neural Information Processing Systems*, 2025.

Giovanni De Felice, Riccardo D'Elia, Alberto Termine, Pietro Barbiero, Giuseppe Marra, and Silvia Santini. Interpretability in deep time series models demands semantic alignment. In *International conference on machine learning*. PMLR, 2026.

Francesco De Santis, Gabriele Ciravegna, Giovanni De Felice, Arianna Casanova, Francesco Giannini, Michelangelo Diligenti, Mateo Espinosa Zarlenga, Pietro Barbiero, Johannes Schneider, and Danilo Giordano. Mixture of concept bottleneck experts. In *International conference on machine learning*. PMLR, 2026.

David Debot and Giuseppe Marra. Quantifying the accuracy-interpretability trade-off in concept-based sidechannel models. *arXiv preprint arXiv:2510.05670*, 2025.

David Debot, Pietro Barbiero, Francesco Giannini, Gabriele Ciravegna, Michelangelo Diligenti, and Giuseppe Marra. Interpretable concept-based memory reasoning. *Advances in Neural Information Processing Systems*, 37:19254–19287, 2024.

Eoin Delaney, Derek Greene, and Mark T. Keane. Instance-based Counterfactual Explanations for Time Series Classification, 2021.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford University Press (UK), 2012.

Daniel E Ehrmann, Shalmali Joshi, Sebastian D Goodfellow, Mjaye L Mazwi, and Danny Eytan. Making machine learning matter to clinicians: model actionability in medical decision-making. *NPJ Digital Medicine*, 6(1):7, 2023.

Mateo Espinosa Zarlenga. In defense of information leakage in concept-based models. In *International conference on machine learning*. PMLR, 2026.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in neural information processing systems*, 35:21400–21413, 2022.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. *Advances in Neural Information Processing Systems*, 36:37849–37875, 2023.

Irene Ferfoglia, Gaia Saveri, Laura Nenzi, and Luca Bortolussi. Ecats: Explainable-by-design concept-based anomaly detection for time series. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 175–191. Springer, 2024.

Florent Forest, Katharina Rombach, and Olga Fink. Interpretable prognostics with concept bottleneck models. *Information Fusion*, 124:103427, 2025.

Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Somvae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199*, 2018.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems*, 30, 2017.

Daniel Gedon, Niklas Wahlström, Thomas B Schön, and Lennart Ljung. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, 54(7):481–486, 2021.

Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop proceedings*, volume 2429, pp. 15, 2019.

Amin Ghadami and Bogdan I Epureanu. Data-driven prediction in dynamical systems: recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229), 2022.

Yolanda Gil, Daniel Garijo, Deborah Khider, Craig A Knoblock, Varun Ratnakar, Maximiliano Osorio, Hernán Vargas, Minh Pham, Jay Pujara, Basel Shbita, et al. Artificial intelligence for modeling complex systems: taming the complexity of expert models to improve decision making. *ACM Transactions on Interactive Intelligent Systems*, 11(2):1–49, 2021.

Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends in Machine Learning*, 15(1-2):1–175, 2022.

Ryan Greenblatt, Ajeya Cotra, and Hjalmar Wijk. Brief independent investigation of agents' behavior, reasoning and collaboration in the openai/hugging face hacking incident. *METR, Aug*, 26, 2026. URL `https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/#core-takeaways-about-this-incident`.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Xusen Guo, Qiming Zhang, Junyue Jiang, Mingxing Peng, Meixin Zhu, and Hao Frank Yang. Towards explainable traffic flow prediction with large language models. *Communications in Transportation Research*, 4:100150, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397, 2022.

Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *The twelfth international conference on learning representations*, 2023.

Nari Johnson, Sonali Parbhoo, Andrew S Ross, and Finale Doshi-Velez. Learning predictive and interpretable timeseries summaries from icu data. In *AMIA Annual Symposium Proceedings*, volume 2021, pp. 581, 2022.

Jackie Kazil, David Masad, and Andrew Crooks. Utilizing python for agent-based modeling: The mesa framework. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pp. 308–317. Springer, 2020.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Patrick Knab, Sascha Marton, Philipp J Schubert, Drago Guggiana, and Christian Bartelt. Concepts in motion: Temporal bottlenecks for interpretable video classification. *arXiv preprint arXiv:2509.20899*, 2025.

Patrick Knab, David Steinmann, Christian Bartelt, Kristian Kersting, Bernt Schiele, Thomas Seidl, Udo Schlegel, and Wolfgang Stammer. What's in the bottle? a survey and roadmap of concept bottleneck models. *Transactions on Machine Learning Research*, 2026.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, et al. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Noam Kolt, Michal Shur-Ofry, and Reuven Cohen. Lessons from complex systems science for ai governance. *Patterns*, 6(8), 2025.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Jonathan Kutasov, Yuqi Sun, Paul Colognese, Teun van der Weij, Linda Petrini, Chen Bo Calvin Zhang, John Hughes, Xiang Deng, Henry Sleight, Tyler Tracy, et al. Shade-arena: Evaluating sabotage and monitoring in llm agents. *arXiv preprint arXiv:2506.15740*, 2025.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, volume 2026, pp. 86173–86202, 2026.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. 2022.

Nancy G Leveson. *Engineering a safer world: Systems thinking applied to safety*. The MIT Press, 2016.

Zichuan Liu, Tianchun Wang, Jimeng Shi, Xu Zheng, Zhuomin Chen, Lei Song, Wenqian Dong, Jayantha Obeysekera, Farhad Shirani, and Dongsheng Luo. Timex++: learning time-series explanations with information bottleneck. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pp. 1–18. Reading, 1996.

Alfred James Lotka. *Elements of physical biology*. Williams & Wilkins, 1925.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *International Conference on Machine Learning*, 2021.

Zhenjiang Mao, Mrinall Eashaan Umasudhan, and Ivan Ruchkin. Physically interpretable world models via weakly supervised representation learning. In *2026 ACM/IEEE 17th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 177–188. IEEE, 2026.

Ričards Marcinkevičs and Julia E Vogt. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(3):e1493, 2023.

Amin Nejatbakhsh and Yixin Wang. Identifying neural dynamics using interventional state space models. In *Forty-second International Conference on Machine Learning*, 2025.

Christoph Obermair, Alexander Fuchs, Franz Pernkopf, Lukas Felsberger, Andrea Apollonio, and Daniel Wollmann. Example or prototype? learning concept-based explanations in time-series. In *Asian conference on machine learning*, pp. 816–831. PMLR, 2023.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *International Conference on Learning Representations*, 2023.

Gianluigi Pillonetto, Aleksandr Aravkin, Daniel Gedon, Lennart Ljung, Antônio H Ribeiro, and Thomas B Schön. Deep networks for system identification: a survey. *Automatica*, 171:111907, 2025.

Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *ACM Computing Surveys*, 2023.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Guide Labs Team, Andreas Madsen, Aya Abdelsalam Ismail, Giang Nguyen, Isaac Plant, Muawiz Chaudhary, Nathaniel Monson, Saqib Azim, Zhichen Guo, and Julius Adebayo. Scaling inherently interpretable language models. *arXiv preprint arXiv:2608.07594*, 2026.

Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 5(3):250–260, 2023.

Angela van Sprang, Erman Acar, and Willem Zuidema. Interpretability for time series transformers using a concept bottleneck framework. *arXiv preprint arXiv:2410.06070*, 2024.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33:20554–20565, 2020.

Chen Yueh-Han, Nitish Joshi, Yulin Chen, Maksym Andriushchenko, Rico Angell, and He He. Monitoring decomposition attacks in llms with lightweight sequential monitors. *arXiv preprint arXiv:2506.10949*, 2025.

Renos Zabounidis, Ini Oguntola, Konghao Zhao, Joseph Campbell, Woojun Kim, Simon Stepputtis, and Katia P Sycara. Disentangled concept-residual models: Bridging the interpretability–performance gap for incomplete concept sets. *Transactions on Machine Learning Research*, 2026.

Rui Zhang, Xingbo Du, Junchi Yan, and Shihua Zhang. The decoupling concept bottleneck model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1250–1265, 2024.

Zhe Zhao, Pengkun Wang, Haibin Wen, Shuang Wang, Liheng Yu, and Yang Wang. Stem-lts: Integrating semantic-temporal dynamics in llm-driven time series analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22858–22866, 2025.

# Appendix

## Table of Contents


**A Notation** . . . . . **16**

**B Proofs and derivations** . . . . . **16**

B.1 Concept-residual dependence . . . . . 17

B.2 Intervention responsiveness . . . . . 17

**C Dataset details** . . . . . **18**

C.1 Lorenz–96 . . . . . 18

C.2 Lotka–Volterra agent-based model . . . . . 19

C.3 Relative Rotation Graphs . . . . . 20

C.4 N-CMAPSS . . . . . 20

**D Implementation details** . . . . . **22**

D.1 Shared backbones . . . . . 22

D.2 Deterministic instantiation . . . . . 22

D.3 Stochastic instantiation . . . . . 22

D.4 Ablations and variants . . . . . 23

**E Experimental details** . . . . . **23**

**F Additional experiments** . . . . . **24**

F.1 Concept alignment: deterministic variant . . . . . 24

F.2 Intervention responsiveness . . . . . 24

F.2.1 Complete results . . . . . 24

F.2.2 Interventional accuracy . . . . . 25

F.3 Sensitivity to noisy interventions . . . . . 26

F.4 Sensitivity to design choices . . . . . 26

F.4.1 Size of the hidden state. . . . . 26

F.4.2 Probability of concept intervention during training. . . . . 27

F.5 Switching the residual on and off during a rollout . . . . . 27

# A NOTATION

For $t_1 \le t_2$, we write $\boldsymbol{x}_{t_1:t_2} = (\boldsymbol{x}_{t_1}, \dots, \boldsymbol{x}_{t_2})$ and $\boldsymbol{x}_{\le t} = \boldsymbol{x}_{0:t}$; the same applies to $\boldsymbol{c}$ and $\boldsymbol{r}$.

The symbol $p$ without a parameter subscript denotes the joint law obtained by composing the data-generating process of Equation (1) with the encoders $q_{\phi_c}$ and $q_{\phi_h}$, that is, the law of $(\boldsymbol{s}_{0:T}, \boldsymbol{x}_{0:T}, \boldsymbol{c}_{0:T}, \boldsymbol{h}_{0:T})$. Every marginal and conditional of that law is written $p$ as well and is identified by its arguments. Unless stated otherwise, expectations, and conditional mutual informations $I(\cdot\,;\cdot \mid \cdot)$ are taken under $p$.

# B PROOFS AND DERIVATIONS

**Proposition B.1** (Additive concept–residual reparameterization of first-order DSSMs)**.** *Let $\boldsymbol{z}_t = (\boldsymbol{c}_t, \boldsymbol{h}_t) \in \mathbb{R}^{d_c} \times \mathbb{R}^{d_h}$ be the state of a first-order DSSM whose first block is the concept state, with initialization kernel $q^\star(\boldsymbol{c}_t, \boldsymbol{h}_t \mid \boldsymbol{x}_{\le t})$ and transition kernel $p^\star(\boldsymbol{c}_{t+1}, \boldsymbol{h}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$. Let $f$ be a Markov kernel on $\mathbb{R}^{d_c}$, e.g. any $f \in \mathcal{M}$, and write $\boldsymbol{c}^f_{t+1} \sim f(\cdot \mid \boldsymbol{c}_t)$.*

*Then there exist kernels $q_c$, $q_h$, $p_r$ and $p_h$ such that the generative procedure*

i) $\boldsymbol{c}_t \sim q_c(\cdot \mid \boldsymbol{x}_{\le t})$, $\boldsymbol{h}_t \sim q_h(\cdot \mid \boldsymbol{x}_{\le t}, \boldsymbol{c}_t)$ *(state initialization)*

ii) $\boldsymbol{c}^f_{t+1} \sim f(\cdot \mid \boldsymbol{c}_t)$, $\boldsymbol{r}_t \sim p_r(\cdot \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1})$, $\boldsymbol{c}_{t+1} := \boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t$ *(concept propagation)*

iii) $\boldsymbol{h}_{t+1} \sim p_h(\cdot \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}_{t+1})$ *(hidden-state propagation)*

*generates, for every $t$, $(\boldsymbol{c}_t, \boldsymbol{h}_t)$ with conditional law $q^\star(\cdot \mid \boldsymbol{x}_{\le t})$, and $(\boldsymbol{c}_{t+1}, \boldsymbol{h}_{t+1})$ with conditional law $p^\star(\cdot \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$. In particular, the two models induce the same law of the rollout $(\boldsymbol{c}_{t:t+H}, \boldsymbol{h}_{t:t+H})$ given $\boldsymbol{x}_{\le t}$, for every $H$.*

*Moreover, $p_r$ is determined by the concept marginal $p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$ of $p^\star$: for every fixed $\boldsymbol{c}^f_{t+1}$,*

$$\boldsymbol{r}_t \sim p_r(\cdot \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1}) \quad \text{if and only if} \quad \boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t \sim p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t). \tag{8}$$

*Hence, for any choice of $f$, the additive concept–residual form entails no loss of generality at the level of first-order DSSM representations.*

*Proof.* Fix $t$.

**1. Initialization.** Let $q_c$ and $q_h$ be the two factors of the chain rule

$$q^\star(\boldsymbol{c}_t, \boldsymbol{h}_t \mid \boldsymbol{x}_{\le t}) = \underbrace{q^\star(\boldsymbol{c}_t \mid \boldsymbol{x}_{\le t})}_{q_c}\ \underbrace{q^\star(\boldsymbol{h}_t \mid \boldsymbol{x}_{\le t}, \boldsymbol{c}_t)}_{q_h}.$$

Then step *i)* draws $(\boldsymbol{c}_t, \boldsymbol{h}_t) \sim q^\star(\cdot \mid \boldsymbol{x}_{\le t})$.

**2. Concept propagation.** Let $p_r(\cdot \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1})$ be the concept marginal $p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$ shifted by $-\boldsymbol{c}^f_{t+1}$, that is, the distribution of $\boldsymbol{c} - \boldsymbol{c}^f_{t+1}$ for $\boldsymbol{c} \sim p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$ and fixed $\boldsymbol{c}^f_{t+1}$. Shifting by a fixed vector is invertible, so $\boldsymbol{r}_t \sim p_r(\cdot \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1})$ if and only if $\boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t \sim p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$, which is Equation (8).

Now run step *ii)* from a given $(\boldsymbol{c}_t, \boldsymbol{h}_t)$. By Equation (8), for every value of the proposal $\boldsymbol{c}^f_{t+1}$ the output $\boldsymbol{c}_{t+1} = \boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t$ has distribution $p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$. This distribution does not depend on $\boldsymbol{c}^f_{t+1}$, so averaging over $\boldsymbol{c}^f_{t+1} \sim f(\cdot \mid \boldsymbol{c}_t)$ leaves it unchanged: $\boldsymbol{c}_{t+1} \sim p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$, whatever $f$ is.

**3. Hidden-state propagation.** Let $p_h$ be the second factor of the chain rule

$$p^\star(\boldsymbol{c}_{t+1}, \boldsymbol{h}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t) = p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)\ \underbrace{p^\star(\boldsymbol{h}_{t+1} \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}_{t+1})}_{p_h}.$$

By Step 2, $\boldsymbol{c}_{t+1}$ has the first factor as its distribution; step *iii)* then draws $\boldsymbol{h}_{t+1}$ from the second, given $\boldsymbol{c}_{t+1}$. Hence $(\boldsymbol{c}_{t+1}, \boldsymbol{h}_{t+1}) \sim p^\star(\cdot \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$.

**4. Rollout.** Given $\boldsymbol{x}_{\leq t}$, both models are Markov chains on $\boldsymbol{z}_s = (\boldsymbol{c}_s, \boldsymbol{h}_s)$ with the same initial distribution $q^\star(\cdot \mid \boldsymbol{x}_{\leq t})$ (Step 1) and the same transition $p^\star$ (Steps 2–3, applied at every step). The distribution of a Markov chain is determined by these two, so $\boldsymbol{z}_{t:t+H}$ has the same distribution given $\boldsymbol{x}_{\leq t}$ under both models, for every $H$. □

*Remark* B.2. The proof uses no property of $f$ beyond being a transition kernel: the admissible class $\mathcal{M}$ matters for D2, not for expressivity. If $f(\cdot \mid \boldsymbol{c}_t)$ is a point mass at $\bar{f}(\boldsymbol{c}_t)$, then $\boldsymbol{r}_t = \boldsymbol{c}_{t+1} - \bar{f}(\boldsymbol{c}_t)$; if the concept transition is a map $g(\boldsymbol{c}_t, \boldsymbol{h}_t)$, then $p_r$ is the point mass at $g(\boldsymbol{c}_t, \boldsymbol{h}_t) - \boldsymbol{c}^f_{t+1}$. When $p^\star(\boldsymbol{c}_{t+1} \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$ is a density, Equation (8) reads $p_r(\boldsymbol{r}_t \mid \boldsymbol{h}_t, \boldsymbol{c}_t, \boldsymbol{c}^f_{t+1}) = p^\star(\boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t \mid \boldsymbol{c}_t, \boldsymbol{h}_t)$, the form used in the main text.

### B.1 Concept-residual dependence

**Proposition B.3.** *If $\boldsymbol{c}_t$ is not almost surely constant, then $I(\boldsymbol{r}_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \Delta_c \implies I(\boldsymbol{r}_t; \boldsymbol{c}_t) = 0$.*

*Proof.* We prove the claim by counterexample. Let $\boldsymbol{r}'_t$ satisfy $I(\boldsymbol{r}'_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \Delta_c$ and be independent of the concepts, $I(\boldsymbol{r}'_t; \boldsymbol{c}_t) = 0$. Now define the residual $\boldsymbol{r}_t = (\boldsymbol{r}'_t, \boldsymbol{c}_t)$, obtained by appending the concepts to $\boldsymbol{r}'_t$. By the chain rule, $\boldsymbol{r}_t$ satisfies the condition:

$$I(\boldsymbol{r}_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \underbrace{I(\boldsymbol{c}_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t)}_{=0} + I(\boldsymbol{r}'_t; \boldsymbol{x}_{\leq t} \mid \boldsymbol{c}_t) = \Delta_c,$$

However, $\boldsymbol{r}_t$ clearly depends on the concepts, since $\boldsymbol{r}_t$ contains $\boldsymbol{c}_t$:

$$I(\boldsymbol{r}_t; \boldsymbol{c}_t) = I(\boldsymbol{c}_t; \boldsymbol{c}_t) + \underbrace{I(\boldsymbol{r}'_t; \boldsymbol{c}_t \mid \boldsymbol{c}_t)}_{=0} = I(\boldsymbol{c}_t; \boldsymbol{c}_t) > 0,$$

Hence $\boldsymbol{r}_t$ is a valid counterexample. □

### B.2 Intervention responsiveness

We characterize how the residual correction modifies the one-step response to a concept intervention $\boldsymbol{c}_t \mapsto \widetilde{\boldsymbol{c}}_t$. Let $\boldsymbol{c}^f_{t+1}$ and $\widetilde{\boldsymbol{c}}^f_{t+1}$ denote the outputs of the admissible transition, and $\boldsymbol{r}_t$ and $\widetilde{\boldsymbol{r}}_t$ the corresponding residual corrections. Let $W_2$ denote the 2-Wasserstein distance between probability laws, and define

$$S^f_t := \|\widetilde{\boldsymbol{c}}^f_{t+1} - \boldsymbol{c}^f_{t+1}\|_2, \qquad S^r_t := W_2(\boldsymbol{r}_t, \widetilde{\boldsymbol{r}}_t), \qquad S^c_t := W_2(\boldsymbol{c}^f_{t+1} + \boldsymbol{r}_t,\ \widetilde{\boldsymbol{c}}^f_{t+1} + \widetilde{\boldsymbol{r}}_t).$$

Thus, $S^f_t$, $S^r_t$, and $S^c_t$ quantify the response to the intervention of the admissible transition output, the residual distribution, and the resulting concept distribution, respectively.

**Proposition B.4** (One-step intervention response bound)**.** *The complete one-step response satisfies*

$$|S^f_t - S^r_t| \leq S^c_t \leq S^f_t + S^r_t. \tag{9}$$

In particular, Equation (9) implies $|S^c_t - S^f_t| \leq S^r_t$, so the residual response bounds how far the complete response can deviate from that of the admissible transition.

*Proof.* Fix $\boldsymbol{h}$ and an intervention $\boldsymbol{c} \mapsto \widetilde{\boldsymbol{c}}$, and let

$$\boldsymbol{c}^f \sim f_\vartheta(\cdot \mid \boldsymbol{c}), \qquad \widetilde{\boldsymbol{c}}^f \sim f_\vartheta(\cdot \mid \widetilde{\boldsymbol{c}})$$

be fixed realizations of the admissible transition. Conditionally on these realizations, let

$$\boldsymbol{r} \sim p_{\theta_r}(\cdot \mid \boldsymbol{h}, \boldsymbol{c}, \boldsymbol{c}^f), \qquad \widetilde{\boldsymbol{r}} \sim p_{\theta_r}(\cdot \mid \boldsymbol{h}, \widetilde{\boldsymbol{c}}, \widetilde{\boldsymbol{c}}^f).$$

The corresponding next-concept variables are

$$\boldsymbol{c}^+ = \boldsymbol{c}^f + \boldsymbol{r}, \qquad \widetilde{\boldsymbol{c}}^+ = \widetilde{\boldsymbol{c}}^f + \widetilde{\boldsymbol{r}}.$$

Using the triangle inequality and translation invariance of the 2-Wasserstein distance,

$$W_2(\boldsymbol{c}^+, \widetilde{\boldsymbol{c}}^+) = W_2(\boldsymbol{c}^f + \boldsymbol{r}, \widetilde{\boldsymbol{c}}^f + \widetilde{\boldsymbol{r}}) \tag{10}$$

$$\leq W_2(\boldsymbol{c}^f + \boldsymbol{r}, \boldsymbol{c}^f + \widetilde{\boldsymbol{r}}) + W_2(\boldsymbol{c}^f + \widetilde{\boldsymbol{r}}, \widetilde{\boldsymbol{c}}^f + \widetilde{\boldsymbol{r}}) \tag{11}$$

$$= W_2(\boldsymbol{r}, \widetilde{\boldsymbol{r}}) + \|\widetilde{\boldsymbol{c}}^f - \boldsymbol{c}^f\|_2. \tag{12}$$

This gives the upper bound.

For the lower bound, the reverse triangle inequality gives

$$W_2(\boldsymbol{c}^+, \widetilde{\boldsymbol{c}}^+) \geq \left|W_2(\boldsymbol{c}^f + \boldsymbol{r}, \boldsymbol{c}^f + \widetilde{\boldsymbol{r}}) - W_2(\boldsymbol{c}^f + \widetilde{\boldsymbol{r}}, \widetilde{\boldsymbol{c}}^f + \widetilde{\boldsymbol{r}})\right| \tag{13}$$

$$= \left|W_2(\boldsymbol{r}, \widetilde{\boldsymbol{r}}) - \|\widetilde{\boldsymbol{c}}^f - \boldsymbol{c}^f\|_2\right|. \tag{14}$$

Therefore,

$$\left|\|\widetilde{\boldsymbol{c}}^f - \boldsymbol{c}^f\|_2 - W_2(\boldsymbol{r}, \widetilde{\boldsymbol{r}})\right| \leq W_2(\boldsymbol{c}^+, \widetilde{\boldsymbol{c}}^+) \leq \|\widetilde{\boldsymbol{c}}^f - \boldsymbol{c}^f\|_2 + W_2(\boldsymbol{r}, \widetilde{\boldsymbol{r}}).$$

Equivalently,

$$|S_t^f - S_t^r| \leq S_t^c \leq S_t^f + S_t^r.$$

□

# C Dataset details

This appendix specifies the four systems of Section 4: the simulator or data source, the observation $\boldsymbol{x}_t$ and the concept $\boldsymbol{c}_t$, the admissible class $\mathcal{M}$ with the fitted parameters, and the split. Table 3 summarizes the four settings; $H$ is the training horizon. Figures 8–11 show, for each system, the observation, the concept trajectory, and rollouts of the admissible map started from points on it. The training protocol common to all models is in Appendix E.

| System | $\textsc{Enc}_{\boldsymbol{c}}$ | $\mathcal{M}$ | Train/val/test | $T$ | $W$ | $H$ |
|---|---|---|---|---|---|---|
| Lorenz–96 | learned | affine | 49/10/11 trajectories | 2800 | 20 | 10 |
| Lotka–Volterra | rule (sum) | mean field of the rules | 32/8/8 trajectories | 75 | 1 | 10 |
| RRG | rule (SMA) | quadrant rotation | 70/15/15% of one series | 5999 | 40 | 3 |
| N-CMAPSS | learned | monotone decay | 20/4/16 trajectories | 65–100 | 20 | 15 |

Table 3: Overview of datasets details.

## C.1 Lorenz–96

The two-scale Lorenz–96 system (Lorenz, 1996) couples $K$ slow variables on a ring to $J$ fast variables per slow variable. We take the slow variables as concepts and the fast ones as observations. We write $c_k$ and $x_{j,k}$ for the components of $\boldsymbol{c}_t$ and $\boldsymbol{x}_t$ and drop the time index in the dynamics:

$$\dot{c}_k = -c_{k-1}(c_{k-2} - c_{k+1}) - c_k + F - \kappa \sum_{j=1}^{J} x_{j,k}, \tag{15a}$$

$$\dot{x}_{j,k} = -\tau^2\, x_{j+1,k}(x_{j+2,k} - x_{j-1,k}) - \tau\, x_{j,k} + \kappa\, c_k, \tag{15b}$$

with cyclic indices $c_{k+K} = c_k$ and $x_{j+J,k} = x_{j,k}$, so the $J$ fast variables of each slow variable form a separate ring. Here $\kappa$ is the coupling and $\tau$ the time-scale separation; in the usual parametrization, $h = \kappa$ and $b = c = \tau$. The system is a standard toy model of a resolved large scale driven by unresolved small-scale activity. In this setting, a forecaster reasons about the slow variables only. We use $K = 36$, $J = 10$, $\kappa = 2$, $\tau = 10$, and $F = 6$. We integrate with RK4 at step $10^{-3}$ and keep one sample every $0.01$ time units.

**Observations and concepts.** The observation is the sampled fast state, $\boldsymbol{x}_t \in \mathbb{R}^{360}$. The expert concept $\boldsymbol{c}_t \in \mathbb{R}^{36}$ is the sampled slow state, read from the simulator and used as the label for $\textsc{Enc}_{\boldsymbol{c}}$. Both are standardized per dimension with training statistics.

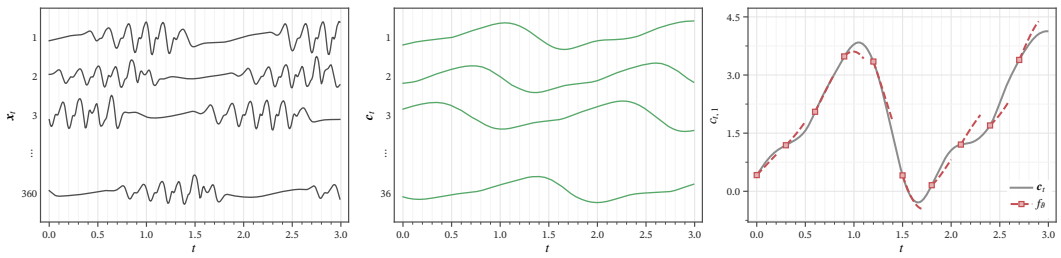

Figure 8: Lorenz–96, first 3 time units of a training segment. Left: components 1, 2, 3, and 360 of the observation $\boldsymbol{x}_t$ (fast variables). Middle: components 1, 2, 3, and 36 of the concepts $\boldsymbol{c}_t$ (slow variables). Each channel is scaled to its own range. Right: $c_{t,1}$ with 20-step rollouts of $f_\vartheta$ Equation (16) started from points on it.

**Admissible class.** The admissible map is affine in the slow variables,

$$f(\boldsymbol{c}) = A\boldsymbol{c} + \boldsymbol{b}, \qquad A \in \mathbb{R}^{36\times 36}, \tag{16}$$

which can be fitted from past transitions without knowing Equation (15). We fit $(A, \boldsymbol{b})$ once by ridge regression (penalty $10^{-3}$) on the one-step pairs $(\boldsymbol{c}_t, \boldsymbol{c}_{t+1})$ of the normalized training concepts and keep it fixed.

**Split and windows.** We generate the 70 segments with $F = 6$ and drop the first 2 time units after each switch, which leaves 2800 samples per segment. The segments are assigned in temporal order to training (49), validation (10), and test (11). $W = 20$, $H = 10$.

### C.2 Lotka–Volterra agent-based model

Population ecologists reason about total abundances and the predator–prey laws that relate them (Lotka, 1925). The observed data are spatial. We simulate an agent-based predator–prey model on a $50 \times 50$ torus (Mesa, Kazil et al., 2020). At each step, agents move to random cells and predators eat prey in their cell. Prey reproduce with a probability that falls with local density. Predators reproduce only if they eat, and die more often when they do not. We run 48 trajectories starting from about 3500 prey and 800 predators, with randomly perturbed initial counts. Each run is logged for 300 steps after a burn-in of 100 steps, and none goes extinct.

**Observations and concepts.** The observation is the microstate, that is, the per-cell prey and predator counts $\boldsymbol{x}_t \in \mathbb{Z}_{\geq 0}^{50\times 50\times 2}$, flattened to $d_x = 5000$. The concept is the pair of total abundances, $\boldsymbol{c}_t = (n_t^{\text{prey}}, n_t^{\text{pred}})$, obtained by summing over cells. Concepts are divided by their training root mean square.

**Admissible class.** The admissible map is the mean-field closure of the agent rules, applied to the raw counts $(p, q)$ of prey and predators. Under uniform mixing, the prey and predator counts of a cell are approximately independent, with $n \sim \text{Poisson}(p/N)$ and $m \sim \text{Poisson}(q/N)$, where $N = 2500$ is the number of cells. In expectation, one simulator step then gives

$$p \leftarrow \max\big(p + B - E,\ 0\big), \qquad q \leftarrow \max\big(q + \gamma\, E - \mu\,(q - E),\ 0\big), \tag{17}$$

with prey births $B = \beta N\, \mathbb{E}[n \max(0, 1 - n/n_{\text{cap}})]$ and prey eaten $E = N\, \mathbb{E}[m\,(1 - e^{-\eta n})]$. One model step applies Equation (17) four times. In the agent rules, $(\beta, n_{\text{cap}}, \eta, \gamma, \mu) = (0.35, 10, 0.55, 0.31, 0.55)$, where $\gamma = 0.34 - 0.03$ is the net growth of a predator that eats. We fit the five constants once by least squares on the one-step differences of the training runs and keep them fixed at $(0.313, 10.05, 0.338, 0.379, 0.532)$.

**Split and windows.** The 48 runs are split into 32 for training, 8 for validation, and 8 for test. One model step is four simulator steps, so a run has $T = 75$ model steps. The population cycle lasts about 26 simulator steps. The microstate is Markov, so $W = 1$. We set $H = 10$ model steps, about 1.5 cycles.

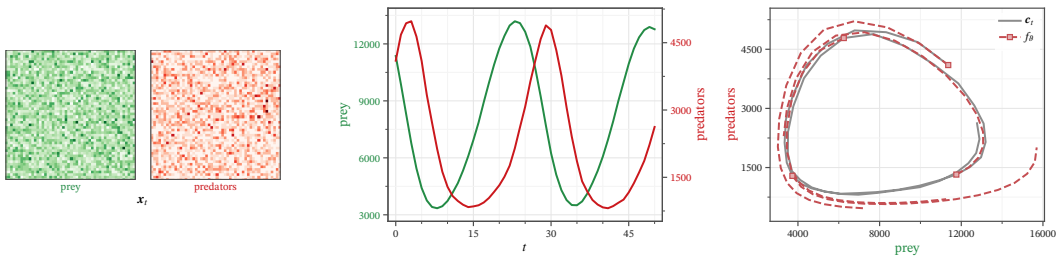


Figure 9: Lotka–Volterra ABM, one training run. Left: the observation $\boldsymbol{x}_t$ at $t = 25$, i.e., prey (green) and predator (red) counts per cell. Middle: the concepts $\boldsymbol{c}_t$, i.e., total prey and predators, over the first 50 simulator steps (two cycles). Right: the same 50 steps in the phase plane, with 15-step rollouts of $f_\vartheta$ Equation (17) started at four phases of the cycle.

### C.3 Relative Rotation Graphs

Portfolio managers track sector rotation, that is, which sectors gain or lose strength relative to the market. They follow it on a Relative Rotation Graph (RRG), a plane of relative strength against its momentum. On this plane, sectors move clockwise through the quadrants Leading, Weakening, Lagging, and Improving. We use CRSP daily prices (open, high, low, close, and volume; split-adjusted) for the nine SPDR sector ETFs (XLK, XLF, XLV, XLE, XLI, XLY, XLP, XLU, and XLB) and for the SPY benchmark. After the warm-up of the indicators below, the series covers 5999 trading days, from 2001-02-27 to 2024-12-31.

**Observations and concepts.** The observation $\boldsymbol{x}_t \in \mathbb{R}^{47}$ stacks the per-sector log-returns of open, high, low, close, and volume(45); the cross-sectional standard deviation of the nine close returns (1); and the SPY close log-return (1). The concepts $\boldsymbol{c}_t \in \mathbb{R}^{18}$ are the two RRG coordinates of each sector. With $s_t = P_t^i / P_t^{\mathrm{SPY}}$ the relative price of sector $i$ and $\mathrm{SMA}_n$ the $n$-day simple moving average,

$$\text{RS-Ratio}_t = 100\,\frac{\mathrm{SMA}_{10}(s)_t}{\mathrm{SMA}_{30}(s)_t}, \qquad \text{RS-Momentum}_t = 100\,\frac{\text{RS-Ratio}_t}{\mathrm{SMA}_{10}(\text{RS-Ratio})_t}. \tag{18}$$

$\textsc{Enc}_{\boldsymbol{c}}$ evaluates Equation (18) on the window. It rebuilds $s$ up to a constant factor from the cumulative sum of sector-minus-SPY close log-returns; the ratios in Equation (18) do not depend on that factor. The chain of moving averages needs 39 days, hence $W = 40$. Concepts are centered at 100 and divided by one root mean square per sector, computed on the whole series.

**Admissible class.** Practitioners describe RRG trails as a clockwise rotation about the center $(100, 100)$ whose speed depends on the quadrant. For sector $i$ with centered coordinates $\boldsymbol{\delta} = (\text{RS-Ratio} - 100,\ \text{RS-Momentum} - 100)$ in quadrant $q = q(\boldsymbol{\delta})$, the admissible map is the rotation–contraction

$$\boldsymbol{\delta}_{t+1} = \lambda_{q,i}\, R(\varphi_{q,i})\, \boldsymbol{\delta}_t, \qquad R(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}, \tag{19}$$

that is, a map $\boldsymbol{c}_{t+1} = A(\boldsymbol{c}_t)\,\boldsymbol{c}_t$ on the centered concepts, with $A$ block diagonal over sectors and piecewise constant over quadrants. We fit the 36 pairs $(\lambda_{q,i}, \varphi_{q,i})$ once by least squares (L-BFGS-B, $\lambda \in [0.5, 2]$, $\varphi \in [-\pi, 0]$) and keep them fixed. The fit uses all one-step transitions of the training series, grouped by sector and by the quadrant of $\boldsymbol{\delta}_t$.

**Split and windows.** The series is split chronologically into 70/15/15% by forecast origin, the last step of the context window. Training origins run from 2001-04-24 to 2017-10-31, validation origins to 2021-05-18, and test origins to 2024-12-02. Each split also holds the 39 days before its first origin as context. $W = 40$, $H = 3$.

### C.4 N-CMAPSS

In prognostics, an engineer reasons about component health: how far the fan, the compressors, and the turbines have degraded. Sensors reflect this only indirectly and depend strongly on the flight condition. N-CMAPSS (Arias Chao et al., 2021) simulates turbofan run-to-failure trajectories under

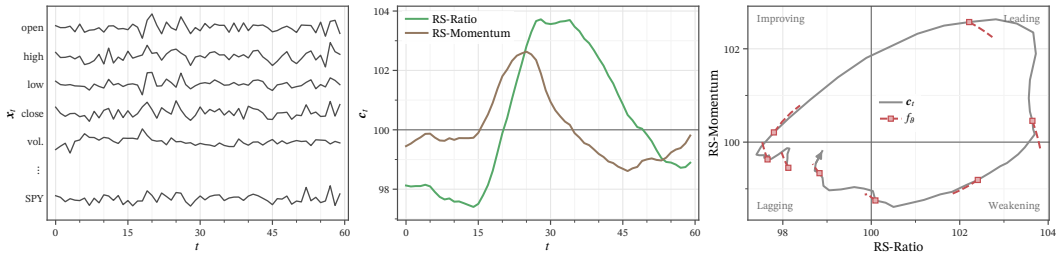


Figure 10: RRG, sector XLY over 60 trading days from 2007-12-24. Left: its open, high, low, and close log-returns, its detrended log-volume, and the SPY return, each scaled to its own range; the dots stand for the other 41 channels of $\boldsymbol{x}_t$. Middle: its RS-Ratio and RS-Momentum (concepts). Right: its trail on the RRG plane with 3-step rollouts of $f_\vartheta$ Equation (19).

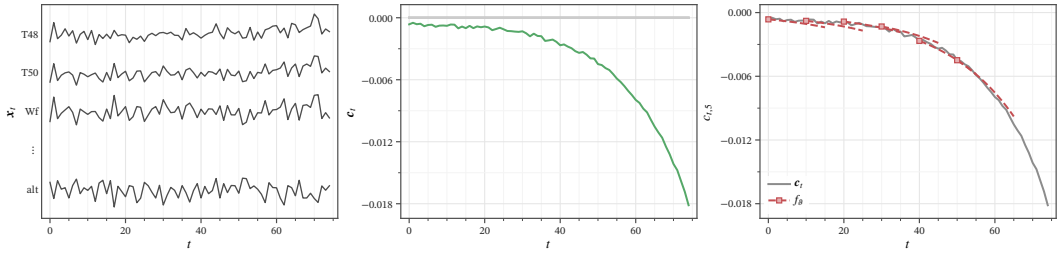


Figure 11: N-CMAPSS, engine 2 of DS01, whose run to failure takes 75 cycles. Left: per-cycle means of the turbine outlet temperatures T48 and T50, the fuel flow Wf, and the altitude, each scaled to its own range; the dots stand for the other 34 channels of $\boldsymbol{x}_t$. Middle: the seven health modifiers (concepts). DS01 degrades the HPT only, so only its efficiency moves. Right: the HPT efficiency $c_{t,5}$ with 15-cycle rollouts of $f_\vartheta$ Equation (20).

real flight profiles. Together with the sensors, it records the health parameters of the simulator; these are the concept labels. We use the subsets DS01, DS04, DS05, and DS07, whose failure modes affect the HPT, the fan, the HPC, and the LPT, respectively. Each subset has 10 engines; we pool its development and test partitions and re-split them by engine. Each trajectory is one engine, from healthy to failure, over 65–100 flight cycles.

**Observations and concepts.** One time step is one flight cycle. The observation $\boldsymbol{x}_t \in \mathbb{R}^{38}$ summarizes the cycle. It contains the mean and standard deviation of the 14 sensor channels (28) and of the four operating conditions (8), the log cycle length (1), and the flight class (1). The operating conditions are altitude, Mach number, throttle-resolver angle, and inlet temperature. The concept $\boldsymbol{c}_t \in \mathbb{R}^7$ collects the health modifiers that degrade in these subsets: efficiency and flow of the fan, HPC, and LPT, and efficiency of the HPT. A value of 0 is healthy, and degradation is negative. The LPC modifiers never degrade and the HPT flow modifier is constant, so we drop them. Observations are ZCA-whitened and then standardized, both with training statistics. Concepts are divided by their training root mean square without centering, so that a healthy component stays at 0.

**Admissible class.** Degradation is irreversible: a health modifier never recovers. An admissible map must therefore satisfy $f(\boldsymbol{c}) \leq \boldsymbol{c}$ componentwise. We use the diagonal parametrization

$$f(\boldsymbol{c}) = \boldsymbol{c} - \boldsymbol{a} \odot \mathrm{relu}(-\boldsymbol{c}) - \boldsymbol{b}, \qquad \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^7_{\geq 0}, \tag{20}$$

a decay whose rate grows with the accumulated degradation. We fit $(\boldsymbol{a}, \boldsymbol{b})$ once by non-negative least squares on the 1686 one-step transitions of the training engines and keep them fixed. The map acts on the normalized concepts.

**Split and windows.** Engines 1–5 of each subset are used for training (20 trajectories), engine 6 for validation (4), and engines 7–10 for test (16). $W = 20$, $H = 15$.

# D IMPLEMENTATION DETAILS

We write DSSM-D and IntDSSM-D for the deterministic instantiations, the deep Koopman autoencoder of Section 4 and its interpretable counterpart, and DSSM-S and IntDSSM-S for the stochastic ones, the deep Markov model and its counterpart. Within a dataset the four models share the observation backbone, the context window $W$, the training horizon $H$ and the selection criterion (Appendix E); the two interpretable models share the concept encoder and the admissible mechanism $f_\vartheta$ of Appendix C. Table 4 gives the architecture on each dataset.

## D.1 SHARED BACKBONES

**Temporal backbone (Lorenz–96, RRG, N-CMAPSS).** A window of $W$ steps and $D$ input channels passes through two residual blocks of causal dilated convolutions (kernel 3, dilations 1 and 2, two convolutions per block, GELU, dropout) and one layer of causal multi-head self-attention (2 heads, pre-LayerNorm, residual), which yields one causal feature $\boldsymbol{\varphi}_s \in \mathbb{R}^d$ per window step $s$. A linear head on the last step gives the encoder's output; the per-step features are the context of the stochastic filters. The width $d$ is 64, 32 and 128 on the three datasets.

**Spatial backbone (Lotka–Volterra).** Given the markovianity of the simulation, we used only one timestep. Each $50 \times 50 \times 2$ occupancy grid passes through three stages of $3 \times 3$ convolution, GroupNorm, GELU, dropout and $2 \times 2$ max-pooling (widths 32, 64, 128), then a global average pooling; any non-grid input channel is added by a linear layer.

## D.2 DETERMINISTIC INSTANTIATION

**DSSM-D.** The backbone maps the window to a latent state $\boldsymbol{z}_t \in \mathbb{R}^{64}$, which is propagated by a constant affine map and read out affinely,

$$\boldsymbol{z}_t = \mathrm{Enc}_\psi(\boldsymbol{x}_{t-W+1:t}), \qquad \boldsymbol{z}_{t+1} = \boldsymbol{K}\boldsymbol{z}_t + \boldsymbol{b}, \qquad \hat{\boldsymbol{c}}_t = \boldsymbol{W}_c\boldsymbol{z}_t + \boldsymbol{b}_c, \tag{21}$$

with $\boldsymbol{K}$ initialised to the identity.

**IntDSSM-D.** The state is $\boldsymbol{z}_t = (\boldsymbol{c}_t, \boldsymbol{h}_t)$ with $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$, $d_h = 32$. $\textsc{Enc}_{\boldsymbol{c}}$ is learned on Lorenz–96 and N-CMAPSS (the temporal backbone with a $d_c$-dimensional head) and is the analytical rule of the dataset otherwise (the sum over cells on Lotka–Volterra, the RS-Ratio/RS-Momentum formula on RRG). $\textsc{Enc}_{\boldsymbol{h}}$ applies the backbone to the window with the concepts appended as extra channels, $\boldsymbol{c}_t$ broadcast along the window (on RRG the window's own concept sequence, which is also the black box's input), and reads $\boldsymbol{h}_t$ off the last step through a linear head. The transition is

$$\boldsymbol{c}_{t+1} = f_\vartheta(\boldsymbol{c}_t) + \boldsymbol{r}_t, \qquad \boldsymbol{r}_t = G([\boldsymbol{h}_t; \boldsymbol{c}_t]) - G([\boldsymbol{0}; \boldsymbol{c}_t]), \tag{22}$$

$$\boldsymbol{h}_{t+1} = \boldsymbol{A}\boldsymbol{h}_t + \boldsymbol{B}_0\boldsymbol{c}_t + \boldsymbol{B}_1\boldsymbol{c}_{t+1} + \boldsymbol{b}, \tag{23}$$

where $G$ is an MLP with GELU activations and the hidden state is propagated by an affine map, as DSSM-D's latent is. Subtracting $G$ at $\boldsymbol{h}_t = \boldsymbol{0}$ makes the correction vanish there whatever $G$ has learned, so the ablation $\boldsymbol{r} = \boldsymbol{0}$ of Section 4, which sets $\boldsymbol{h}_t = \boldsymbol{0}$, recovers $f_\vartheta$ exactly. $\boldsymbol{A}$ is initialised to $a\boldsymbol{I}$ (Table 4) and $\boldsymbol{B}_0$, $\boldsymbol{B}_1$, $\boldsymbol{b}$ to zero, so the hidden state starts autonomous. One implementation choice departs from the notation of Section 3.1: $\textsc{Dec}_{\boldsymbol{r}}$ is conditioned on $(\boldsymbol{h}_t, \boldsymbol{c}_t)$ and not on the proposal $\boldsymbol{c}^f_{t+1}$; every $f_\vartheta$ used here is deterministic, so $\boldsymbol{c}^f_{t+1}$ is a function of $\boldsymbol{c}_t$ (Remark B.2). The read-out is the identity on the concept block.

## D.3 STOCHASTIC INSTANTIATION

**DSSM-S.** The deep Markov model with a concept emission, on a latent $\boldsymbol{z}_s \in \mathbb{R}^{64}$:

$$\boldsymbol{z}_1 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2), \qquad \boldsymbol{z}_{s+1} \mid \boldsymbol{z}_s \sim \mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{z}_s), \boldsymbol{\sigma}_\theta^2(\boldsymbol{z}_s)\big), \qquad \boldsymbol{c}_s \mid \boldsymbol{z}_s \sim \mathcal{N}\big(m_e(\boldsymbol{z}_s), \boldsymbol{\sigma}_e^2\big), \tag{24}$$

with the gated transition of that model,

$$\begin{aligned} \boldsymbol{g} &= \sigma\big(\mathrm{MLP}_g(\boldsymbol{u})\big), & \boldsymbol{n} &= \mathrm{ReLU}\big(\mathrm{MLP}_n(\boldsymbol{u})\big), \\ \boldsymbol{\mu}_\theta &= (1-\boldsymbol{g}) \odot (\boldsymbol{W}_\mu\boldsymbol{u} + \boldsymbol{b}_\mu) + \boldsymbol{g} \odot \boldsymbol{n}, & \boldsymbol{\sigma}_\theta^2 &= \mathrm{softplus}(\boldsymbol{W}_\sigma\boldsymbol{n} + \boldsymbol{b}_\sigma), \end{aligned} \tag{25}$$

Table 4: Architecture per dataset.

| | Lorenz–96 | Lotka–Volterra | RRG | N-CMAPSS |
|---|---|---|---|---|
| Backbone | TCN 64 | CNN 32/64/128 | TCN 32 | TCN 128 |
| Dropout | 0.1 | 0.1 | 0.2 | 0.1 |
| $\text{ENC}_{\boldsymbol{c}}$ | TCN 64 | sum over cells | RS formula | TCN 32 (no attn., dropout 0.3) |
| DSSM latent $d_z$ / hidden state $d_h$ | 64 / 32 | 64 / 32 | 64 / 32 | 64 / 32 |
| Correction $G$ | $2 \times 64$ | $2 \times 64$ | $1 \times 16$ | $2 \times 64$ |
| $\boldsymbol{A}$ initialisation $a$ (-D / -S) | 1 / 1 | 1 / 1 | 0.2 / 0.2 | 0.2 / 1 |
| Gated transition ($\text{PROP}_{\boldsymbol{h}}$, DSSM-S) | $1 \times 64$ | $1 \times 16$ | $2 \times 16$ | $1 \times 32$ |
| Emission $m_e$ (DSSM-S) | $2 \times 64$ | $2 \times 16$ | affine | $2 \times 32$ |
| Residual noise $\boldsymbol{\sigma}_r$ (IntDSSM-S) | constant | constant | state-dependent | constant |

read on $\boldsymbol{u} = \boldsymbol{z}_s$, an MLP mean $m_e$ and a learned per-concept constant variance $\boldsymbol{\sigma}_e^2$. Inference is the structured (filtering) network of the same model: the backbone supplies the causal context $\boldsymbol{\varphi}_s$ at every window step and the shallow combiner gives $q(\boldsymbol{z}_s \mid \boldsymbol{z}_{s-1}, \boldsymbol{x}_{\leq s}) = \mathcal{N}\big(\boldsymbol{W}_q \boldsymbol{v} + \boldsymbol{b}_q,\ \text{softplus}(\boldsymbol{W}_q' \boldsymbol{v} + \boldsymbol{b}_q')\big)$ with $\boldsymbol{v} = \frac{1}{2}\big(\tanh(\boldsymbol{W}_z \boldsymbol{z}_{s-1} + \boldsymbol{b}_z) + \boldsymbol{\varphi}_s\big)$. The point forecast rolls the transition mean from the posterior mean at the window end and reads it out with $m_e$.

**IntDSSM-S.** The state, the mechanism and the read-out are IntDSSM-D's. Where $\text{ENC}_{\boldsymbol{c}}$ is learned, $q_{\phi_c}(\boldsymbol{c}_t \mid \boldsymbol{x}_{\leq t}) = \mathcal{N}\big(\boldsymbol{\mu}_c(\boldsymbol{x}_{t-W+1:t}), \boldsymbol{\sigma}_c^2\big)$ with a per-concept constant variance (the analytical rules are exact). The residual decoder is

$$\text{DEC}_{\boldsymbol{r}}: \qquad p_{\theta_r}(\boldsymbol{r}_t \mid \boldsymbol{h}_t, \boldsymbol{c}_t) = \mathcal{N}\big(G([\boldsymbol{h}_t; \boldsymbol{c}_t]) - G([\boldsymbol{0}; \boldsymbol{c}_t]),\ \boldsymbol{\sigma}_r^2\big), \tag{26}$$

with $\boldsymbol{\sigma}_r^2$ a learned per-concept constant, initialised to the variance of $\boldsymbol{c}_{t+1} - f_\vartheta(\boldsymbol{c}_t)$ over the training labels. $\text{PROP}_{\boldsymbol{h}}$ is the gated transition of Equation (25) read on $\boldsymbol{u}_t = [\boldsymbol{c}_t; \boldsymbol{c}_{t+1}; \boldsymbol{h}_t]$, so that $\boldsymbol{h}_{t+1} \sim \mathcal{N}\big(\boldsymbol{\mu}_h(\boldsymbol{u}_t), \boldsymbol{\sigma}_h^2(\boldsymbol{u}_t)\big)$, initialised as Equation (23) ($\boldsymbol{W}_\mu = [\boldsymbol{0}, \boldsymbol{0}, a\boldsymbol{I}]$, the concept columns of the gate and proposal networks zeroed). $\text{ENC}_{\boldsymbol{h}}$ is DSSM-S's filter applied to the hidden state: the backbone's context on the window with $\boldsymbol{c}_t$ appended, the same shallow combiner $q_{\phi_h}(\boldsymbol{h}_s \mid \boldsymbol{h}_{s-1}, \boldsymbol{x}_{\leq s}, \boldsymbol{c}_t)$ at every window step and a learned prior at the first; $\boldsymbol{h}_t$ is the filtered mean at the window end.

### D.4 Ablations and variants

*IntDSSM w/o* $\boldsymbol{r}_t \,|\, \boldsymbol{c}_t$ removes $\boldsymbol{c}_t$ from $G$, $\boldsymbol{r}_t = G(\boldsymbol{h}_t) - G(\boldsymbol{0})$, so an intervention reaches the residual only through $\text{PROP}_{\boldsymbol{h}}$, one step later. The $p_{\text{int}}$ variants of Appendix F.4.2 change both Stage-1 and Stage-2 objectives by performing ground-truth interventions during training of all propagation maps. The alternative mechanisms of Figure 7: persistence $f(\boldsymbol{c}) = \boldsymbol{c}$ and the reversed rate $f(\boldsymbol{c}) = 2\boldsymbol{c} - f_\vartheta(\boldsymbol{c})$, are trained from scratch under the same protocol, whereas the true mechanism of Figure 6 replaces $f_\vartheta$ at test time only, without retraining.

## E Experimental details

**Data.** Splits, context window $W$, training horizon $H$ and normalisation are those of Appendix C. A training sample is a window $\boldsymbol{x}_{t-W+1:t}$ with the labels $\boldsymbol{c}_{t-W+1:t+H}$.

**Stage 1.** Where $\text{ENC}_{\boldsymbol{c}}$ is learned (Lorenz–96, N-CMAPSS) it is fitted alone: by the mean squared error on $\boldsymbol{c}_t$ for IntDSSM-D, and by the Gaussian likelihood of the labels under $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$ for IntDSSM-S (on Lorenz–96 with a KL regulariser, of weight 0.1, towards the concept-only transition's one-step prediction of the read-out). The mechanism $f_\vartheta$ is fitted once, from the labels alone (Appendix C). Stage 1 is selected on the validation mean squared error of the read-out.

**Stage 2.** With $\text{ENC}_{\boldsymbol{c}}$ and $f_\vartheta$ frozen, $\text{ENC}_{\boldsymbol{h}}$, $\text{PROP}_{\boldsymbol{h}}$ and $\text{DEC}_{\boldsymbol{r}}$ are trained on the labels at every step. The target is the residual evaluated on the labels, $\boldsymbol{r}_{t+k} = \boldsymbol{c}_{t+k+1} - f_\vartheta(\boldsymbol{c}_{t+k})$, and the prediction is the decoded residual $g_{\theta_r}(\boldsymbol{h}, \boldsymbol{c}) = G([\boldsymbol{h}; \boldsymbol{c}]) - G([\boldsymbol{0}; \boldsymbol{c}])$, with $\boldsymbol{h}_t = \text{ENC}_{\boldsymbol{h}}(\boldsymbol{x}_{\leq t}, \boldsymbol{c}_t)$ and $\boldsymbol{h}_{t+k+1} = \text{PROP}_{\boldsymbol{h}}(\boldsymbol{h}_{t+k}, \boldsymbol{c}_{t+k}, \boldsymbol{c}_{t+k+1})$. IntDSSM-D minimises the squared-error

$$\mathcal{L}_{\text{IntDSSM-D}} = \big\|g_{\theta_r}(\boldsymbol{h}_t, \boldsymbol{c}_t) - \boldsymbol{r}_t\big\|^2 + \frac{1}{H-1} \sum_{k=1}^{H-1} \big\|g_{\theta_r}(\boldsymbol{h}_{t+k}, \boldsymbol{c}_{t+k}) - \boldsymbol{r}_{t+k}\big\|^2, \tag{27}$$

IntDSSM-S minimises the negative ELBO of the same labelled path: the Gaussian log-likelihood of each $\boldsymbol{r}_{t+k}$ under $\text{DEC}_{\boldsymbol{r}}$ replaces the squared error, the hidden state is inferred by the filter along the

| | Lorenz–96 | Lotka–Volterra | RRG | N-CMAPSS |
|---|---|---|---|---|
| $W$ / $H$ | 20 / 10 | 1 / 10 | 40 / 3 | 20 / 15 |
| $H_{\mathrm{eval}}$ / $h_{\max}$ | 20 / 17 | 20 / 10 | 3 / 3 | 15 / 15 |
| Training origins | every 8 steps | every 2 steps | every step | every step |
| Batch size | 64 | 64 | 16 | 8 |
| Input of the black boxes | $\boldsymbol{x}$ | $[\boldsymbol{x}; \boldsymbol{c}]$ | $[\boldsymbol{x}; \boldsymbol{c}]$ | $\boldsymbol{x}$ |
| $\lambda_{\mathrm{lc}}$ / $H_{\mathrm{lc}}$ (DSSM-D) | 0.5 / 10 | 0.5 / 10 | 0.5 / 5 | 25 / 15 |
| $\beta_{\max}$ (DSSM-S) | 0.1 | 1 | 0.1 | 1 |
| Free bits (DSSM-S) | – | – | – | 0.2 |
| Weight decay | 0 | 0 | 0.1 (DSSM-S encoder, $\mathrm{ENC}_{\boldsymbol{h}}$) | 0 |

Table 5: Training and selection settings that differ across datasets.

window and sampled from $\mathrm{PROP}_{\boldsymbol{h}}$ over the forecast steps, and the KL between the filter and $\mathrm{PROP}_{\boldsymbol{h}}$ enters with a weight annealed from 0 over the first 20 epochs.

**Black boxes.** DSSM-D is trained on the squared error of its free-running forecast over $h = 0, \ldots, H$, weighted $\propto 1/(h+1)$, plus the latent-consistency (linearity) term of the deep Koopman autoencoder, of weight $\lambda_{\mathrm{lc}}$ over $H_{\mathrm{lc}}$ steps (Table 5). DSSM-S is trained on the negative ELBO of the window's labels, with the same horizon weights on the forecast likelihoods and the KL weight annealed over the first 20 epochs to $\beta_{\max}$ (Table 5).

**Training protocol.** All models on every dataset are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $10^{-3}$, no weight decay except on RRG, Table 5), gradient-norm clipping at 1, and a learning rate halved whenever the validation criterion has not improved for 10 epochs. Training runs for at most 1000 epochs and stops when the criterion has not improved by $0.2\%$ for 20 epochs. The reported checkpoint is the one with the best validation criterion from epoch 20 on. Every model is trained with 5 seeds, each seed re-running the whole chain (Stage 1, then Stage 2). Curves and tables report report the mean over the seeds, with shaded bands in figures and $\pm$ values in tables indicating 2 standard errors of the mean. All runs use a single NVIDIA GeForce RTX 2080 GPU in single precision.

**Model selection.** Every model of a dataset is ranked on the same quantity, the validation error of its free-running rollout (the means, for the stochastic models): the mean squared error averaged with equal weight over horizons $h = 1, \ldots, h_{\max}$ on validation windows that carry $H_{\mathrm{eval}}$ future steps (Table 5).

**Evaluation.** Metrics are computed on 600 test windows, drawn once at random among those carrying $H_{\mathrm{pop}}$ future steps (20, 20, 6 and 15 on the four systems) and shared by every model of the dataset.

# F Additional experiments

We provide here additional experiments complementing Section 4.

## F.1 Concept alignment: deterministic variant

Figure 12 repeats the alignment analysis of Section 4.2 for the deterministic instantiation, with the same protocol and the same refit strategy for the baseline. **The deterministic variant confirms the stochastic results.** The opaque baselines sit well below the interpretable counterparts, although forecasting accuracy is comparable.

## F.2 Intervention responsiveness

We replace the predicted concepts at step $\bar{t}$ with their ground-truth values and let the model roll out autonomously. In the following, We compare IntDSSM with the ablation IntDSSM w/o $\boldsymbol{r}_t \mid \boldsymbol{c}_t$ in terms of responsiveness of the residual and interventional accuracy.

### F.2.1 Complete results

First, in Figure 13, we extend the responsiveness analysis of Table 2 to longer lags.

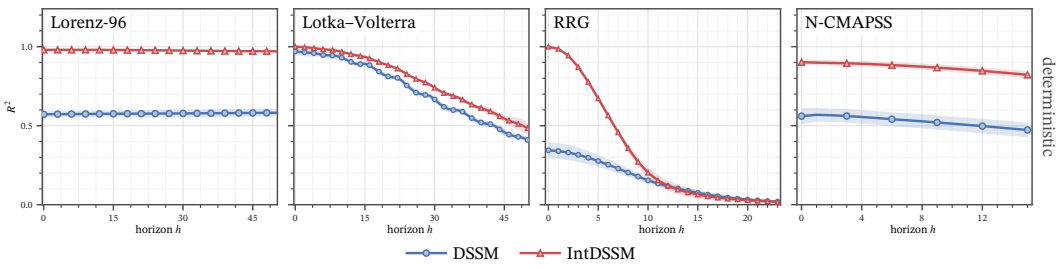


Figure 12: **Semantic alignment, deterministic variant.** $R^2$ between ground-truth concepts and model state per horizon h. The stochastic variant is in Figure 5.

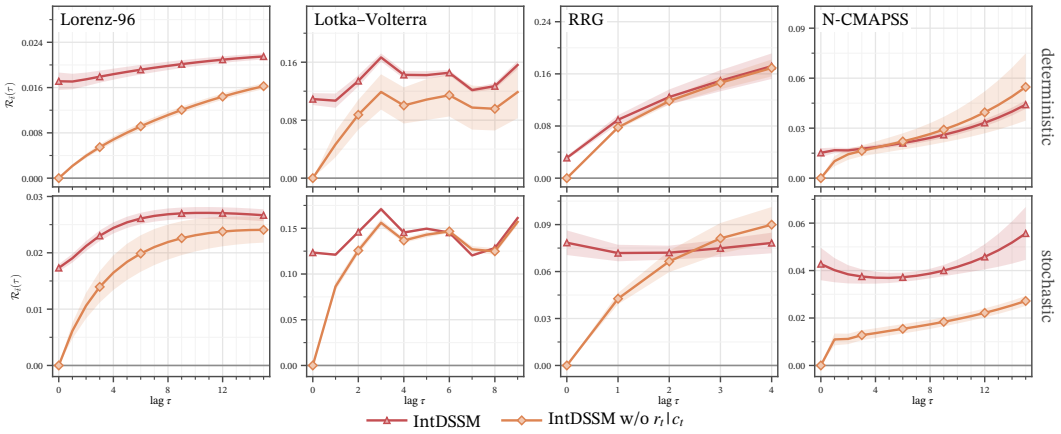


Figure 13: **Residual responsiveness over longer lags.** Responsiveness $\mathcal{R}_{\bar{t}}(\tau)$ of the residual correction after a concept intervention at step $\bar{t}$, as a function of the lag $\tau$.

**Conditioning on the concepts mainly affects the short-lag response.** Without conditioning, the residual cannot react at $\tau = 0$ and responds only as the intervention propagates through the hidden state. The gap then generally narrows with the rollout lag.

### F.2.2 Interventional accuracy

Within the same ablation setting, we also report in Figure 14 the forecasting error after the intervention.

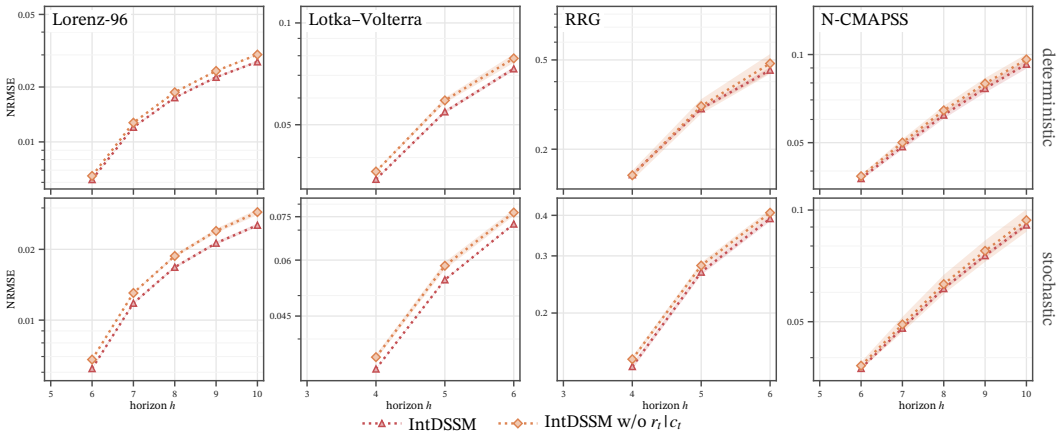


Figure 14: **Forecasting error after a concept intervention.** NRMSE of the rollout from the intervention step $\bar{t}$ onwards.

**Concept conditioning slightly but consistently improves interventional accuracy.** After the intervention, IntDSSM attains lower NRMSE than the ablation on all systems, for both variants and at every horizon. The gain is small, and largest on Lorenz-96 and Lotka–Volterra. We speculate this is attributable to the original model being possessing a more responsive residual to concept interventions w.r.t. the ablation.

### F.3 Sensitivity to noisy interventions

We study the sensitivity to noisy interventions by adding Gaussian noise to the ground-truth applied at the intervention step. The noise standard deviation is $\alpha$ times the model's own error at $\bar{t}$. Hence, $\alpha = 0$ is the exact intervention of Section 3.3, and $\alpha = 1$ corresponds to an expert as inaccurate as the model. We use $\alpha \in \{0, 0.5, 1, 2\}$. Results are in Figure 15.

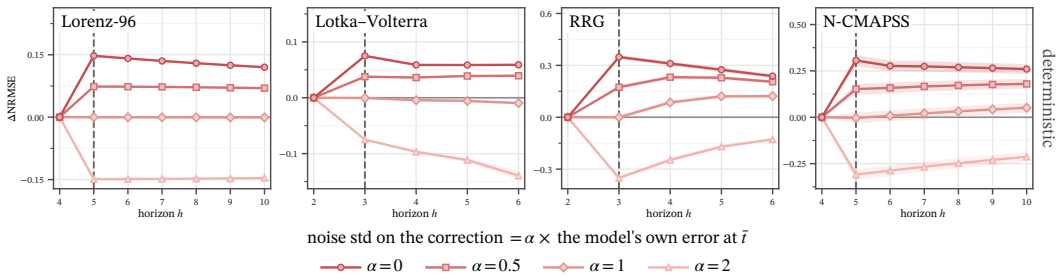


Figure 15: **Noisy concept interventions.** ΔNRMSE during rollout after noisy concept intervention. The used ground-truth value carries Gaussian noise of standard deviation $\alpha$ times the model's error at $\bar{t}$.

As expected, **interventional accuracy decreases with noise**.

### F.4 Sensitivity to design choices

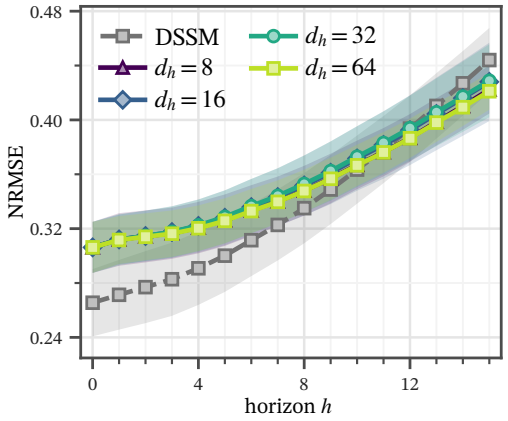


Figure 16: **Sensitivity of deterministic IntDSSM to the hidden-state size $d_h$, on N-CMAPSS.** NRMSE per horizon, with DSSM for reference.

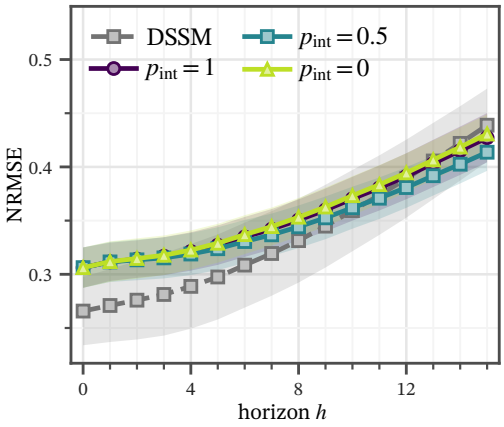


Figure 17: **Sensitivity of the deterministic IntDSSM to different rates $p_{int}$ on N-CMAPSS.** NRMSE per horizon, with DSSM for reference.

#### F.4.1 Size of the hidden state.

In Figure 16, we study the sensitivity to the dimension $d_h \in \{8, 16, 32, 64\}$ of the hidden state of the IntDSSM on N-CMAPSS, bounding how much information the residual pathway can retain during the rollout. We observe **forecasting accuracy is robust to the size of the hidden state**. In fact, the four configurations overlap with the default $d_h = 32$ within their uncertainty at every horizon. This agrees with Figure 4, where the admissible mechanism alone explains most of the forecast on N-CMAPSS.

### F.4.2 PROBABILITY OF CONCEPT INTERVENTION DURING TRAINING.

In Figure 17, on N-CMAPSS, we study the sensitivity of the model to different rates $p_{int}$ of ground-truth interventions on the concepts during training of the concept-only transition map $f_\vartheta$, a regularization technique introduced in Espinosa Zarlenga et al. (2022). At test time, **the response to exact concept interventions is similar across different values of** $p_{int}$. We keep $p_{\text{int}} = 1$ in all other experiments.

## F.5 SWITCHING THE RESIDUAL ON AND OFF DURING A ROLLOUT

Ablating the residual over the whole rollout measures how much of the forecast the concept-only mechanism explains on its own (Figure 4). We now switch it off and back on within a single rollout. On Lorenz–96, we roll out IntDSSM for 100 steps and set $\boldsymbol{r}_t = \mathbf{0}$ for $h \in [25, 50)$ and $h \in [75, 100)$. The hidden state keeps being propagated while the residual is off. The resulting forecasting accuracy is in Figure 18.

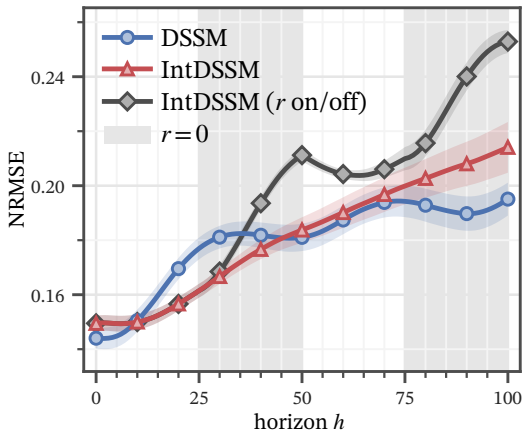


Figure 18: **Residual switched on and off on Lorenz–96.** NRMSE per horizon of DSSM, IntDSSM, and IntDSSM with the residual switched off ($\boldsymbol{r}_t = \mathbf{0}$) in the shaded intervals $h \in [25, 50)$ and $h \in [75, 100)$, and on elsewhere.

**The residual can be switched off and restored at later steps.** While the residual is off, the error grows faster than for IntDSSM and exceeds that of DSSM. Once the residual is restored at $h = 50$, the error stays roughly constant until the next switch.