LEARNING TO TEST: LATENT REPRESENTATION LEARNING FOR DYNAMIC INSTABILITY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Stability assessment is central to safety-critical systems governed by differential–algebraic equations (DAEs), whose dynamics are constrained by physical laws and admissibility conditions. These systems operate under stochastically varying environmental inputs, so stability must be reassessed as the context distribution shifts, yet repeated large-scale DAE simulation is prohibitive in high-dimensional or real-time settings. We propose a test-oriented learning framework that tests a trajectory-level chance constraint from deployment contexts alone. Rather than re-estimating physical parameters or re-solving the DAE, we learn a latent representation from baseline context–trajectory data that retains stability-relevant information and is regularized toward a tractable reference distribution, and calibrate a latent discrepancy test over the composite safety null. We establish exact validity for oracle calibration and asymptotic validity for its empirical implementation under class-conditional latent invariance, and show that the guarantee degrades gracefully when this invariance holds only approximately. On simulated dynamical systems and a power-system real-world example, our approach strikes a balance between validity and power, whereas other baseline methods each fail in at least one regime. An ablation that injects controlled violations of the invariance assumption shows that decisions stay correct while the violation is below the safety margin, as the theory predicts.

1 INTRODUCTION

Many safety-critical scientific and engineering systems evolve according to physical laws while satisfying hard algebraic constraints, including power grids, structural systems, and biochemical networks (Kunkel & Mehrmann, 2006; Kundur et al., 2007; Ascher & Petzold, 1998; Rao & Arkin, 2003). Differential–algebraic equations (DAEs) provide a natural representation by coupling differential dynamics with algebraic constraints that define a feasible manifold (Brenan et al., 1995). In operation, however, exogenous contexts such as loads, environmental conditions, and external forces vary stochastically (Arnold, 2006; Khasminskii, 2011), shifting the operating point and altering the system's dynamics and stability properties (Lee & Chae, 2013; Milano, 2010; Roberts & Spanos, 2003; Sobczyk, 2013; Chiang et al., 1987). Stability must therefore be reassessed as the context distribution evolves (Liberzon, 2003).

Yet repeated stability assessment faces two fundamental challenges: (i) From a computational standpoint, each environmental change may require re-solving high-dimensional and often stiff DAE systems or recomputing local linearizations and spectral properties—tasks that quickly become prohibitive in large-scale or real-time settings. (ii) From a statistical standpoint, observations are typically available only from a single baseline domain, corresponding to a *singleton* within the safe regime. Such data capture system behavior under limited operating conditions and do not span the full set of stable scenarios. As a result, conventional end-to-end learning approaches that map inputs directly to stability labels can be systematically biased: they perform well near the training distribution but fail under distribution shift, while making limited use of the richer trajectory information available in the baseline data.

These challenges motivate a central question: how can we assess the stability and safety of constrained dynamical systems under stochastically varying inputs without incurring the cost of repeated full-scale analysis, while also ensuring reliable generalization beyond observed conditions? More broadly,

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.

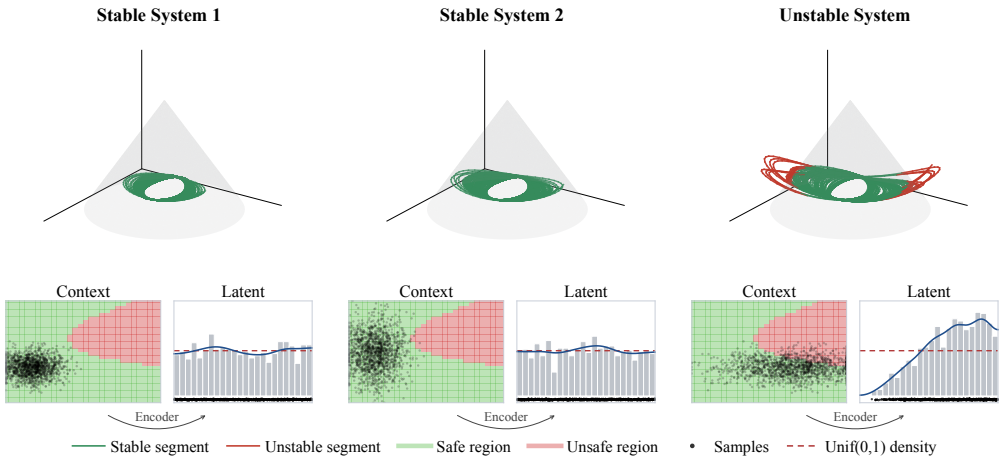


Figure 1: Illustration of Cartesian pendulum systems and the one-dimensional latent representations induced by our method. The system maps a context variable (*e.g.*, initial conditions or external disturbances) to a pendulum trajectory. A system is considered *stable* when the context follows the "safe" distribution. The top row shows representative trajectories under three exogenous context distributions: green segments satisfy the angular stability criterion, while red segments indicate violations. Under stable regimes, the learned latent representation is approximately uniform; under unstable regimes, it deviates noticeably from uniformity.

how can we efficiently detect, with uncertainty quantification, when such a system is approaching or has entered an unsafe regime in a changing environment? In this work, we propose a *test-oriented learning* framework, inspired by decision-focused learning (Mandi et al., 2024) and tailored to stability hypothesis testing. Using context–trajectory observations from a safe baseline domain, we learn a latent representation that preserves stability-relevant dynamics and enables calibrated distributional testing using only contextual variables at deployment. Rather than optimizing for trajectory prediction or reducing stability to pointwise labels, the representation is designed for the downstream inferential objective: detecting whether a shifted context distribution violates the prescribed stability requirement.

As illustrated by Figure 1, our central idea is to learn a latent representation of contextual variables that encodes stability-relevant features of the constrained dynamics, while being regularized toward a simple and tractable reference distribution. Trained on baseline data from a certified safe operating regime, we learn a structured latent geometry for subsequent distributional testing. At deployment, safety monitoring reduces to a latent space distributional hypothesis test. If contexts under the new regime remain close to the reference law, stability is inferred to persist; systematic deviations signal elevated instability risk. By learning the representation end-to-end with respect to this testing objective, we bypass repeated DAE simulation, extract stability-relevant structure from baseline trajectories, and enable scalable context-only monitoring under stochastic environmental change.

Our work makes three main contributions: (i) We introduce a *test-oriented learning* framework that directly targets stability verification under distribution shift, bypassing explicit system identification and repeated trajectory simulation at deployment. (ii) We provide a *latent reformulation* of stability hypothesis testing, showing that, under explicit conditions, a trajectory-level chance constraint in constrained dynamical systems can be equivalently characterized as a distributional test in latent space, and establish asymptotic validity and consistency for the resulting test. (iii) We conduct an *extensive empirical evaluation* on both synthetic dynamical systems and a real-world transient-stability assessment benchmark based on the IEEE 39-bus power system (Sokolović, 2024), demonstrating reliable and powerful detection of instability under context shift. Together, these contributions establish a scalable and statistically principled framework for context-only stability assessment in stochastic constrained dynamical systems.

Related Work. Stability analysis studies how dynamical systems respond to perturbations and whether trajectories remain near equilibria or reference motions. Classical theory uses linearization, Lyapunov functions, and invariance principles (Lyapunov, 1992; LaSalle, 1960), with broad applications in power systems (Anderson & Fouad, 2008; Kundur et al., 2007; 2004; Sauer et al.,

2017; Wu et al., 2023), control (Venkatesh et al., 2024; Walsh et al., 2002; Wang & Schirmer, 2010), transportation (Bastin et al., 2007; Kesting & Treiber, 2008; Lovisari et al., 2014; Sadri & Wu, 2013; Wang et al., 2020), chemical processes (Favache & Dochain, 2009; Wang et al., 2011), and epidemiology (Bonzi et al., 2011; Cangiotti et al., 2024; Earn & McCluskey, 2025; Fall et al., 2007; Korobeinikov & Wake, 2002). These methods commonly assess stability pointwise under fixed operating conditions or deterministic perturbations (Abbasi & Liu, 2024; Chiang et al., 1987; Lovisari et al., 2014), while stochastic extensions typically model process noise (Li et al., 2022). We instead treat exogenous contexts as random and test whether their distribution satisfies a trajectory-level chance constraint after a shift. Related probabilistic transient-stability methods sample operating scenarios (Papadopoulos & Milanović, 2016), but do not address context-only testing when deployment trajectories are unavailable.

Scientific machine learning learns dynamical surrogates using recurrent models (Elman, 1990; Rumelhart et al., 1986; Hochreiter & Schmidhuber, 1997; Cho et al., 2014; Chung et al., 2014; Yuchi et al., 2025), neural and latent ODEs (Biloš et al., 2021; Chen et al., 2018; 2025; Li et al., 2023; Rubanova et al., 2019), and physics-informed or DAE-aware architectures (Raissi et al., 2019; Rackauckas et al., 2020; Huang et al., 2024; Koch et al., 2025). These methods generally target prediction or system identification rather than testing under distribution shift. A direct alternative learns $X \mapsto y(X)$ and applies a binomial test. This is effective when label prediction transfers to P_1, but predictors trained on one baseline domain can be brittle under shift (Ovadia et al., 2019; Rosenfeld et al., 2022; Yu & Wang, 2024; Yuan et al., 2022; Zhang et al., 2024a), while a hard label discards trajectory severity and temporal structure (Rubanova et al., 2019; Xiao et al., 2022; Zhang et al., 2024b). Our method instead uses trajectory reconstruction and stability-aware supervision to learn a representation for stability testing (Biggs et al., 2023; Kim et al., 2021; Kirchler et al., 2020; Podkopaev & Ramdas, 2023).

Our test-oriented perspective is also related to learning-to-optimize and decision-focused learning (Zhu et al., 2022; Elmachtoub & Grigas, 2022; Li & Malik, 2016; Mandi et al., 2024; Wang et al., 2025), including differentiable optimization (Wilder et al., 2019) and ranking-based approaches (Mandi et al., 2022). It is also connected to learned statistical tests, including representation-based two-sample tests (Ginsberg et al., 2022; Kirchler et al., 2020; Wang & Tong, 2022) and sequential procedures (Wei et al., 2021; Zhang et al., 2025; Zhou et al., 2025; 2026). Unlike the former, our downstream object is a hypothesis test rather than an optimization decision; unlike generic learned testing, our representation is trained using observed trajectories and a stability-aware reconstruction loss, and the inferential target is a stability chance constraint.

2 Problem Setup

We consider a dynamical system, parameterized by deterministic but *unknown* physical parameters $\theta \in \Theta$, with state variables $\{s(t)\}_{t\geq 0}$ that evolves on the time horizon $t \in [0, +\infty]$ according to the following differential–algebraic equation (DAE):

$$\mathrm{d}s(t)/\mathrm{d}t = f_\theta\big(s(t), a(t); X\big), t \geq 0; \text{ and } s(0) = s_0(\theta, X), \tag{1a}$$

$$0 = g_\theta\big(s(t), a(t); X\big). \tag{1b}$$

Here $X \in \mathscr{X}$ is a *random* exogenous context (fixed once realized) drawn from a distribution, $a(\cdot) \in \mathscr{A}$ is an auxiliary algebraic variable enforcing instantaneous constraints, and $s_0(\cdot, \cdot)$ specifies the initial state of the system at time $t = 0$. For notational clarity, given a context realization $X = x$, we write the resulting state trajectory as $\{s_\theta(t; x)\}_{t\geq 0} \in \mathscr{S}$, where $\mathscr{S}$ denotes the space of admissible state trajectories, and the associated algebraic variables as $\{a_\theta(t; x)\}_{t\geq 0}$. We assume that f_θ and g_θ are unknown, continuously differentiable in (s, a) and that (1) is well-posed, so that for each (θ, x) there exists a unique solution $(s_\theta(\cdot; x), a_\theta(\cdot; x))$. When $g_\theta \equiv 0$ in (1b), the system in (1) reduces to an ordinary differential equation (ODE). It is worth emphasizing that the system trajectory $\{s_\theta(t; X)\}$ is stochastic, with randomness induced by the distribution of X. Concrete examples are given in Appendix A.

Let $\eta : \mathscr{S} \to \mathbb{R}$ be a user-specified trajectory functional that measures departure from an admissible operating region. For example, $\eta(s(\cdot)) = \max_{t\in[0,T]} |s(t)|$, records the maximum state magnitude over the time horizon $[0, T]$. Given a tolerance threshold $\tau > 0$, a context x is said to produce a trajectory violation if $\eta(s_\theta(\cdot; x)) > \tau$. We define the trajectory-level violation indicator as

$$y(x) \coloneqq \mathbb{1}\left\{\eta\big(s_\theta(\cdot; x)\big) > \tau\right\}.$$

In this work, we consider system stability at the distributional level by controlling the probability of this trajectory-level violation under $X \sim P$.

Definition 1 ((τ, α)-Stability)**.** The system is (τ, α)-*stable* under P if

$$\mathbb{E}_{X\sim P}[y(X)] = \mathbb{P}_{X\sim P}\big[\eta\big(s_\theta(\cdot; X)\big) > \tau\big] \leq \alpha, \tag{2}$$

i.e., the prescribed stability requirement $\eta(s_\theta(\cdot; X)) \leq \tau$ holds with probability at least $1 - \alpha$. Here "stability" means satisfaction of a trajectory-level chance constraint over the chosen observation horizon; it does not assert asymptotic stability of individual trajectories.

Definition 2 (Safe Regime)**.** Let $\mathcal{P}(\mathscr{X})$ denote the set of Borel probability measures on $\mathscr{X}$. For fixed (θ, τ, α), the *safe regime* is

$$\mathcal{P}_0 := \left\{P \in \mathcal{P}(\mathscr{X}) : \mathbb{P}_{X\sim P}\big[\eta\big(s_\theta(\cdot; X)\big) > \tau\big] \leq \alpha\right\}, \tag{3}$$

i.e., $\mathcal{P}_0$ is the set of (τ, α)-stable distributions in the sense of Definition 1.

The safe regime $\mathcal{P}_0$ is unknown because the physical parameters θ and governing mappings f_θ and g_θ are unknown. We assume access to observations collected from a baseline domain with distribution $P_0 \in \mathcal{P}_0$. Specifically, we observe i.i.d. samples $\{(X_i, \mathcal{S}_i)\}_{i=1}^n$, where $X_i \sim P_0$ and $\mathcal{S}_i = \{s_\theta(t; X_i)\}_{t\in[0,T]}$. Conditional on X_i, the trajectory $\mathcal{S}_i$ is deterministic and its randomness is induced by $X_i \sim P_0$. These data characterize a safe operating regime without revealing the underlying parameters or dynamics.

We now consider a new domain in which contexts $\{X_j\}_{j=n+1}^{n+m}$ are drawn i.i.d. from a potentially new distribution P_1, but the corresponding system trajectories are unobserved. The key question is whether the stability verified in the baseline domain persists under this distribution shift. Determining whether P_1 belongs to the safe regime naturally leads to the following hypothesis test

$$H_0 : \ P_1 \in \mathcal{P}_0 \qquad \text{vs.} \qquad H_1 : \ P_1 \notin \mathcal{P}_0. \tag{4}$$

Rejecting H_0 provides statistical evidence that the prescribed stability requirement may no longer hold under P_1, thereby triggering an alarm for further diagnosis or intervention.

3 Proposed Framework: Learning to Test

We propose an end-to-end framework, termed "learning to test" (`L2T`), that directly targets the hypothesis testing problem in (4), rather than first estimating the physical parameters and then simulating state trajectories under the new domain. We learn a latent representation of the contextual variable using baseline data so that it captures distributional factors governing stability of the DAE system. Uniformity regularization provides a bounded and interpretable reference geometry. At deployment, we measure the latent discrepancy from this reference and calibrate its rejection threshold over the composite safe regime. Thus, calibration supplies validity within the shift family specified below, while representation learning determines the power of the resulting test. Figure 2 summarizes the proposed `L2T` paradigm.

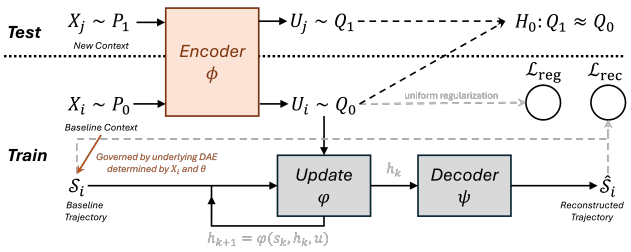


Figure 2: Overview of the "Learning-to-Test" paradigm.

This approach offers two advantages: (i) Representation learning can be carried out efficiently *offline* using baseline data, without requiring repeated simulation in the new domain. (ii) By operating directly at the level of the testing objective, the method bypasses the need to estimate physical parameters or repeatedly solve the DAE, focusing instead on detecting shifts relevant to stability.

3.1 Representation Learning for Stability Testing

We use a learned history-state surrogate to model the observed trajectories. A continuous-time formulation motivates the update below, while our implementation uses a discrete-time recurrent architecture. Specifically, for a continuous-time series $\{s(t), t \geq 0\}$, we define a compact *history*

embedding $h(t) \in \mathcal{H}$, where $\mathcal{H}$ denotes the embedding space. This history embedding encapsulates the key information about the past observations up to time t. Conceptually, its evolution can be described by

$$\mathrm{d}h(t)/\mathrm{d}t = \varphi(s(t), h(t); \theta, x), \tag{5}$$

where φ is a learned update function conditioned on the observed state and context; we do not impose the governing DAE equations as constraints on this surrogate. An Euler discretization of this conceptual model would take the form

$$h_{k+1} = h_k + \varphi(s_k, h_k; \theta, x)\Delta t_k, \quad k = 1, \ldots, K-1, \tag{6}$$

where h_k and h_{k+1} are the history embeddings at times t_k and t_{k+1}, respectively, and $\Delta t_k = t_{k+1} - t_k$ is time difference between them. Our experiments instead use the recurrent update described below.

Latent Representation. We seek to construct a latent representation of the contextual variable X such that: (i) it retains information about the observed trajectories relevant to stability; and (ii) it is regularized toward a uniform law. To this end, we introduce an *encoder*

$$\phi : \mathscr{X} \to [0,1]^d, \qquad u = \phi(x),$$

which maps each context x to a latent variable u. Accordingly, we reparameterize the history-embedding update and state reconstruction as

$$h_{k+1} = \varphi(s_k, h_k, u), \qquad \hat{s}_k = \psi(h_k). \tag{7}$$

Here φ denotes the history-update function using the current observation s_k, previous history h_k, and latent representation u, corresponding by a slight abuse of notation to the right-hand side of (6); ψ is a *decoder* that maps the history embedding h_k back to the state space. The history-update function φ can typically be implemented via a recurrent architecture (*e.g.*, recurrent neural network, long-short term memory (Yu et al., 2019), or Transformer (Vaswani et al., 2017)).

Training Objectives. The encoder ϕ is trained jointly with decoder ψ and update function φ to construct latent representations that preserve stability-relevant information while remaining distributionally uniform. Given the baseline data $\{(x_i, \mathcal{S}_i)\}_{i=1}^n$, where $\mathcal{S}_i = \{s_{i,k}\}_{k=1}^{K_i}$ consists of K_i observations and is associated with context x_i, the training objective consists of three components:

1. *Reconstruction loss*: To encourage the latent representation to retain information about trajectories observed in the baseline regime, we minimize the reconstruction loss:

$$\mathcal{L}_{\mathrm{rec}}(\phi, \psi, \varphi) \coloneqq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{K_i}\sum_{k=1}^{K_i}\|s_{i,k} - \hat{s}_{i,k}\|^2, \tag{8}$$

   where $\hat{s}_{i,k}$ are the reconstructed output in (7) with $u = \phi(x_i)$.
2. *Latent uniformity regularization*: To facilitate distributional testing in the latent space, we regularize the learned representations toward a uniform reference distribution by minimizing

$$\mathcal{L}_{\mathrm{reg}}(\phi) \coloneqq \mathrm{KL}\left[p_U \,\middle\|\, \mathrm{Unif}(0,1)^d\right] = -\mathcal{H}(p_U) + \text{const}, \tag{9}$$

   where $\mathrm{KL}[\cdot\|\cdot]$ represents Kullback–Leibler (KL) divergence, p_U denotes the empirical distribution of the latent variables $\{u_i\}_{i=1}^n$ with $u_i = \phi(x_i)$, and $\mathcal{H}(p) = -\int_{[0,1]^d} p(u)\log p(u)\,\mathrm{d}u$ is the differential entropy. Minimizing this objective is equivalent to maximizing the entropy of the latent representation, thereby encouraging uniformity. This regularization supplies the fixed reference geometry used by the calibrated test in Section 3.2.
3. *Stability-aware reweighting*: To preserve sensitivity to the stability condition, we augment the reconstruction loss in (8) with an asymmetric penalty based on the stability metric. For example, using $\eta(s(\cdot)) = \max_{t\in[0,T]}|s(t)|$, the sample-level weight can be defined as

$$\omega_{\mathrm{stb}}(s, \hat{s}) = \begin{cases} \omega, & |s| > \tau \text{ and } \mathrm{sign}(s)\hat{s} < \mathrm{sign}(s)s, \\ 1, & \text{otherwise}, \end{cases}$$

   for some $\omega > 1$, where $\mathrm{sign}(s) = 1$ if $s > 0$ and $\mathrm{sign}(s) = -1$ if $s < 0$. This weighting scheme distinguishes errors based on whether the trajectory lies in the unstable region and on the direction of the reconstruction error. When $|s| \le \tau$, the trajectory is considered stable and assigned a unit weight. When $s > \tau$ or $s < -\tau$, the weight depends on whether the reconstruction exaggerates or attenuates the instability. In particular, the model is encouraged to overestimate positive instability and underestimate negative instability. This asymmetric design intentionally exaggerates truly unstable trajectories, improving sensitivity near the stability boundary.

The overall objective is therefore

$$\mathcal{L}(\phi, \psi, \varphi) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \frac{1}{K_i} \sum_{k=1}^{K_i} \omega_{\mathrm{stb}}(s_{i,k}, \hat{s}_{i,k}) \cdot \|s_{i,k} - \hat{s}_{i,k}\|^2 + \lambda \cdot \mathcal{L}_{\mathrm{reg}}(\phi), \tag{10}$$

where $\lambda \geq 0$ controls the trade-off between reconstruction fidelity and latent uniformity regularization.

In practice, we represent ϕ, ψ, and φ using deep neural networks. We minimize the empirical objective $\mathcal{L}$ using stochastic gradient descent (SGD) or adaptive variants (*e.g.*, Adam (Kingma & Ba, 2014)). Each iteration samples a mini-batch from the baseline data, rolls out the latent dynamics via (7) to obtain reconstructions, evaluates the three loss components, and updates all parameters by backpropagation. We note that the latent uniformity term is implemented by estimating p_U from the mini-batch latent representations and differentiating a tractable surrogate of (9) (*e.g.*, via maximum mean discrepancy (MMD) (Gretton et al., 2012) or a flow matching loss (Lipman et al., 2022; Rezende & Mohamed, 2015)), so that gradients can be computed end-to-end. More implementation details are in Appendix B.

3.2 Latent Distributional Test for Dynamic Instability Detection

After training, we freeze the encoder ϕ and denote the latent representation variable as $U = \phi(X)$. For any context distribution P, denote the induced latent distribution by the encoder as

$$Q_P \coloneqq \phi_{\#} P = P \circ \phi^{-1}.$$

The training objective encourages U to retain stability-relevant dynamical information while organizing the latent space around a simple reference distribution. We therefore seek to reformulate the stability test in (4) as a distributional test on Q_{P_1}. To justify doing so, however, we must establish how the latent distribution reflects the probability of a stability violation.

We begin by separating the latent representations associated with stable and unstable trajectories. Because trajectories and their labels are available in the baseline domain under P_0, we define the baseline class-conditional latent distributions

$$R_0 \coloneqq \mathrm{Law}_{P_0}(U \mid y(X) = 0)\,, \qquad R_1 \coloneqq \mathrm{Law}_{P_0}(U \mid y(X) = 1)\,.$$

Here, R_0 and R_1 are the latent distributions associated with stable and unstable trajectories, respectively. Without further structure on the deployment shift, the marginal latent distribution Q_P need not identify the violation probability: changes in Q_P may arise from changes within either stability class rather than from a change in the prevalence of unstable trajectories. We therefore assume that the class-conditional latent distributions remain invariant from the baseline domain to deployment, while the relative prevalence of stable and unstable trajectories may change.

Assumption 1 (Class-Conditional Latent Invariance)**.** *For a deployment distribution P_1 under consideration and each class $a \in \{0, 1\}$ with $P_1(y(X) = a) > 0$, it holds that*

$$\mathrm{Law}_{P_1}(U \mid y(X) = a) = R_a. \tag{11}$$

Remark 1 (A Weaker Invariance Requirement)**.** *Assumption 1 is a representation-space analogue of the label-shift assumption in domain adaptation (Lipton et al., 2018; Garg et al., 2020) and is closely related to generalized label shift (Tachet des Combes et al., 2020). It is weaker than invariance of the class-conditional context distributions: the encoder can discard directions of X that do not affect stability, and contexts may move along them without changing their latent law.*

Assumption 1 is crucial for the validity of our approach. Without a structural link between the baseline and deployment domains, the violation probability is not identified from the observed data: in Figure 3(a,b), two systems that agree on the baseline support produce the same observed data but have violation probabilities on opposite sides of α (Proposition 1). Training does not enforce the assumption, but exact invariance is not needed: if the class-conditional latent laws drift by at most ε in D, inflating the calibrated threshold by ε retains asymptotic validity (Proposition 2; Figure 3c). Appendix E.1 checks the assumption empirically (Figure 8). For any distribution P satisfying Assumption 1, the latent distribution can be expressed as a mixture of the class-conditional latent distributions: $Q_P = \mathbb{P}_P(y(X) = 0)\, R_0 + \mathbb{P}_P(y(X) = 1)\, R_1$. The violation probability is therefore exactly the mixing weight assigned to R_1. Consequently, the original stability condition $\mathbb{P}_P(y(X) = 1) \leq \alpha$ becomes a constraint on Q_P, providing the desired bridge from the trajectory-level hypothesis in (4) to a distributional test in latent space. We thus obtain the following latent-space reformulation of the original test.

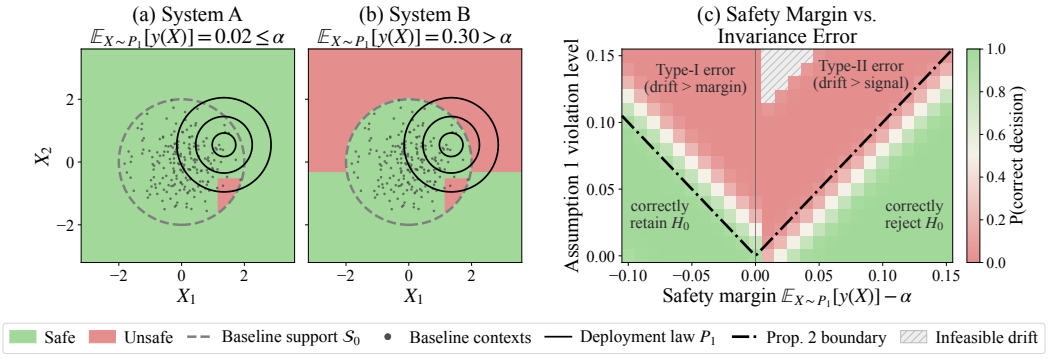


Figure 3: Why Assumption 1 is needed and what violating it costs. Panels (a, b) show two systems that agree on the baseline support $\mathcal{S}_0$ and produce identical observed data, yet have violation probabilities on opposite sides of $\alpha = 0.1$ (Proposition 1). Panel (c) shows the probability that the calibrated test decides correctly on the spring `L2T` latent under class-conditional drift toward the other class; the dash-dot boundary is predicted by Proposition 2, not fitted. Hatched cells mark infeasible drift.

Theorem 1 (Latent Reformulation)**.** *Suppose Assumption 1 holds and $R_0 \neq R_1$, and let $Q^* := \mathrm{Unif}([0,1]^d)$. Under a mild regularity condition (Appendix D.2), there exists a measurable transformation T such that, after replacing ϕ by $\widetilde{\phi} = T \circ \phi$, for any separating integral probability metric D there exists a threshold $r_\alpha > 0$ for which the original hypothesis test in* (4) *is equivalent to*

$$H_0 : D(Q_{P_1}, Q^*) \le r_\alpha \qquad \textit{vs.} \qquad H_1 : D(Q_{P_1}, Q^*) > r_\alpha, \tag{12}$$

where r_α is determined by α, D, and the transformed class-conditional latent distributions.

Remark 2. *The theorem reformulates the original hypothesis test as a latent-space test against the known reference distribution Q^*, using a fixed threshold r_α. For $D = \mathrm{TV}$ (total variation), the threshold is $r_\alpha = (\alpha/2)\,\mathrm{TV}(R_0, R_1)$; for $D = \mathrm{MMD}_k$ with a characteristic kernel k, it is $r_\alpha = (\alpha/2)C_k$, where $C_k := \mathrm{MMD}_k(R_0, R_1) > 0$. The transformation T is constructed by uniformizing an appropriately centered latent reference distribution while preserving the class-conditional structure. The construction and proof are provided in Appendix D.2.*

Theorem 1 is a population-level result. In practice, the sample is finite and the learned encoder realizes the uniformizing map T only approximately, so we introduce a calibration step that turns the reformulation into a practical test. Let $\mathcal{D}_{\mathrm{cal}}$ be a labeled sample from P_0 of size $n_{\mathrm{cal}} = |\mathcal{D}_{\mathrm{cal}}|$, disjoint from the split used to fit $\widehat{\phi}$, and let $\widehat{R}_0$ and $\widehat{R}_1$ be the empirical class-conditional latent distributions it induces. We form the reference $\widehat{Q}_{\alpha/2} := (1-\alpha/2)\widehat{R}_0 + (\alpha/2)\widehat{R}_1$, placing it at the midpoint of the safe-mixture range. For deployment contexts $X_{n+1}, \ldots, X_{n+m} \sim P_1$, let $\widehat{Q}_{P_1} := m^{-1}\sum_{j=n+1}^{n+m} \delta_{\widehat{\phi}(X_j)}$. The practical test statistic is $\widehat{T}_{\mathrm{test}} := D(\widehat{Q}_{P_1}, \widehat{Q}_{\alpha/2})$. For each $\lambda \in \mathcal{G}_\alpha \subset [0, \alpha]$ we draw B deployment-sized batches from $(1-\lambda)\widehat{R}_0 + \lambda\widehat{R}_1$, compare each with that same reference, and take $\widehat{r}_{\alpha,\beta}$ to be the largest resulting $(1-\beta)$-quantile, where β is the significance level of the test, distinct from the violation tolerance α; we reject when $\widehat{T}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}$. Full details are in Appendix C.

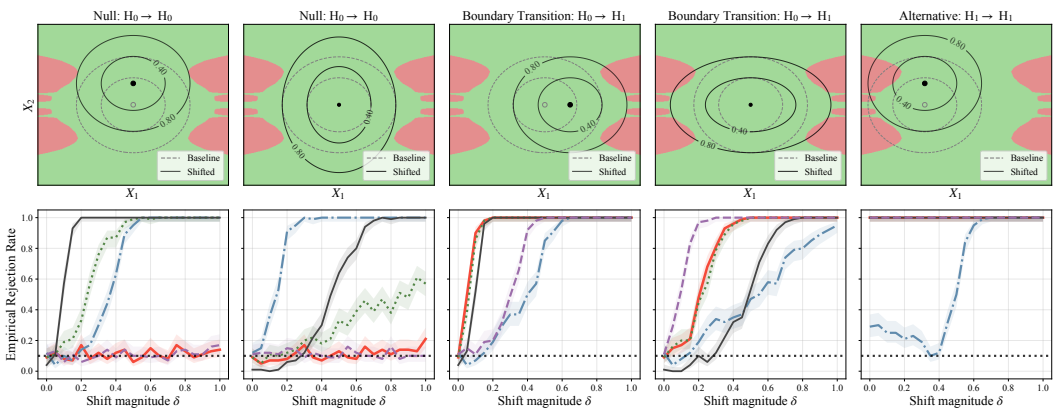
Theorem 2 (Calibrated Latent Test)**.** *Condition on the trained encoder $\widehat{\phi}$, and suppose the calibration conditions of Appendix C hold. Let $h(\mathcal{G}_\alpha) := \sup_{\lambda\in[0,\alpha]} \min_{\lambda'\in\mathcal{G}_\alpha} |\lambda - \lambda'|$ denote the calibration-grid resolution. For any P_1 satisfying Assumption 1, as $n_{\mathrm{cal}} \to \infty$, $B \to \infty$ and $h(\mathcal{G}_\alpha) \to 0$:*

- *If $P_1 \in \mathcal{P}_0$, for every fixed deployment size m,*

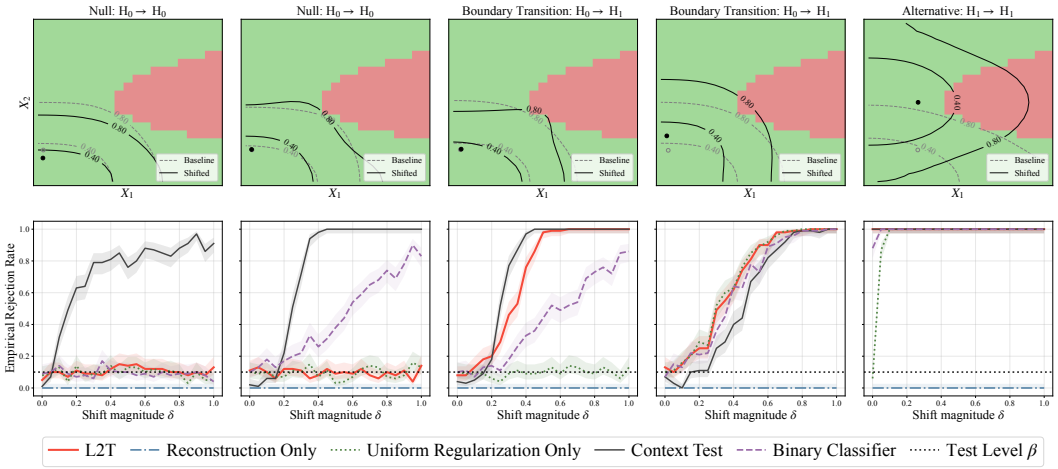
$$\limsup\, \mathbb{P}_{P_1}\left(\widehat{T}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}\right) \le \beta. \qquad \textit{(Type-I error control)}$$

- *If $P_1 \notin \mathcal{P}_0$,* i.e., *$p_1 := \mathbb{P}_{P_1}(y(X) = 1) > \alpha$, and $D(R_0, R_1) > 0$, then*

$$\mathbb{P}_{P_1}\left(\widehat{T}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}\right) \to 1 \quad \textit{as } m \to \infty. \qquad \textit{(Power)}$$



(a) Mass–spring–damper system.



(b) Cartesian pendulum system.

Figure 4: Comparison between L2T and other baselines under various distribution shifts. Green and red regions denote stable and unstable contexts, respectively, while dashed gray and solid black contours represent the baseline and shifted distributions. The top row of each panel shows baseline (P_0) and shifted (P_1) context distributions in the 2D space across representative regimes: null ($H_0 \to H_0$), boundary transition ($H_0 \to H_1$), and alternative ($H_1 \to H_1$). The bottom row shows empirical rejection rates over 100 runs against the shift magnitude, with 90% Wilson intervals shaded. The dotted line marks the test level $\beta = 0.1$.

4 Experiments

We evaluate L2T on two simulated systems, a mass–spring–damper and a Cartesian pendulum, and on a power-system benchmark generated from physics-based simulations. Stability is defined by trajectory-level functionals (maximum displacement or swing angle), and we apply controlled mean, covariance, marginal, and dependence shifts to the context distribution, covering null, boundary-transition, and alternative regimes.

For the power-system benchmark, we use the IEEE 39-bus transient stability assessment (TSA) dataset (Sokolović, 2024): 12,852 simulated trajectories of generator-level signals (50 features) under varying operating conditions and fault scenarios, each labeled stable or unstable. We train on a 95:5 stable-to-unstable mixture, reflecting the practical setting in which training data come from the safe regime and violations are rare. In Definition 1, we set the chance-constraint tolerance to $\alpha = 0.1$ for the synthetic systems and $\alpha = 0.05$ for TSA; full details are in Appendix E.

Baselines and Configurations. We compare L2T with two alternative detection strategies and two ablations of its training objective. (i) *Context Test*: a two-sample test on the raw contexts X that ignores system dynamics, using MMD (Gretton et al., 2012) in low dimensions and a projection-

Table 1: Comparison of empirical Type-I and Type-II errors at test level $\beta = 0.1$ (mean $\pm$ standard error).

Dataset	Metric	CONTEXT TEST	BINARY-CLF.	RECON-ONLY	UNIFORM-ONLY	L2T
Mass–spring–damper	Type-I ↓	1.00 ± 0.00	0.17 ± 0.04	1.00 ± 0.00	1.00 ± 0.00	0.14 ± 0.03
	Type-II ↓	0.47 ± 0.05	0.00 ± 0.00	0.54 ± 0.05	0.00 ± 0.00	0.00 ± 0.00
Cartesian pendulum	Type-I ↓	1.00 ± 0.00	0.83 ± 0.04	0.00 ± 0.00	0.13 ± 0.03	0.13 ± 0.03
	Type-II ↓	0.00 ± 0.00	0.56 ± 0.05	1.00 ± 0.00	0.90 ± 0.03	0.02 ± 0.01
IEEE 39-bus TSA	Type-I ↓	0.01 ± 0.01	1.00 ± 0.00	0.11 ± 0.02	0.12 ± 0.02	0.10 ± 0.02
	Type-II ↓	0.18 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.71 ± 0.03	0.01 ± 0.01

Type-I error is reported under the null, and Type-II error is reported at a representative shift magnitude in the boundary-transition regime. For Type-I error, values closer to level 0.1 indicate better control; for Type-II error, smaller values indicate higher detection power.

based test in high dimensions. (ii) ***Binary classifier***: a model that predicts trajectory stability from X, followed by a two-sample mean test on the predicted stability probabilities between baseline and deployment samples. (iii) ***Uniform-only*** and (iv) ***Reconstruction-only***: latent representations trained with only the uniformity regularization or only trajectory reconstruction, which isolate the contributions of latent geometry and of dynamics-aware modeling. Appendix E.1 further compares L2T with oracle simulation of the deployment trajectories and with a trajectory surrogate followed by a label count test (Figure 7). Unless otherwise specified, synthetic experiments use MMD-based uniformity regularization, all tests use significance level $\beta = 0.1$, and we report empirical rejection rates across shift magnitudes; implementation details are in Appendix E.

Results on Simulation Studies. Figure 4 shows the rejection curves for both systems, and Table 1 summarizes them. L2T keeps its rejection rate near the nominal level $\beta = 0.1$ under the null shifts while attaining low Type-II error, and hence high power, in the boundary-transition regime. In contrast, each competing method fails in at least one regime. The context test over-rejects benign shifts on both systems, since it reacts to any change in the context distribution. Reconstruction-only over-rejects on the spring and misses instability on the pendulum, indicating that accurate trajectory reconstruction alone does not yield a representation suited to stability testing. Uniform-only also over-rejects on the spring and loses power on the pendulum, reflecting the importance of dynamics-aware structure. The binary classifier is competitive on the spring but over-rejects and loses power on the pendulum. Together, these comparisons indicate that *effective stability testing requires both dynamics-aware representation learning and test-aligned latent geometry*. Appendix E.1 adds loss-weight sensitivity analyses and latent-density visualizations (Figures 5 and 6).

Results on TSA Benchmark. Because the TSA contexts cannot be shifted directly, we vary the unstable proportion of the test set from 0% to 30%, a label shift under which Assumption 1 holds by construction. Table 1 uses the $100{:}0$ stable-to-unstable mixture as the null and the $80{:}20$ mixture as the alternative, with L2T regularized by a normalizing flow on an 8-dimensional latent. L2T attains Type-I error at the nominal level and near-zero Type-II error, whereas the binary classifier severely over-rejects under the null and the context and uniform-only tests lose substantial power; reconstruction-only performs on par with L2T on this benchmark. Appendix E.2 reports continuous shift curves (Figure 10) and shows that MMD regularization with a one-dimensional latent is markedly less powerful (Table 3), suggesting that flow regularization with a richer latent suits the high-dimensional TSA dynamics better.

5 CONCLUSION AND FUTURE WORK

We proposed L2T, which learns a latent representation from baseline context–trajectory data and tests a trajectory-level chance constraint from deployment contexts alone; on two simulated systems and a power-system benchmark it balances Type-I error and power. It targets settings where deployment simulation is too costly; when simulation is affordable, a count test on simulated trajectories remains the reference (Appendix E.1). Its main limitation is Assumption 1, which holds by construction under shifts of the class proportions but only approximately under covariate shift: drift smaller than the safety margin is harmless whenever $p_1 \geq \alpha/2$ (Proposition 2), whereas near-boundary decisions under strong covariate shift remain sensitive to it. Calibration also needs labeled baseline data from both classes and is valid only asymptotically in their number, and power depends on how well the latent separates the classes. Future work includes bounding the drift from a small labeled deployment sample, learning representations whose class-conditional laws are invariant by design, and sequential monitoring, which updates the test as deployment contexts arrive.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: writing and polishing, literature search, code writing, and running experiments, as well as feedback on the research methodology and checking and revising the proofs in Appendix D. We did not use them to generate data sets; all trajectories come from numerical solvers or the public TSA dataset. The authors reviewed all AI-assisted work: they checked the proofs, tested AI-assisted code before using it, verified suggested references against the original sources, and made all methodological decisions. We take full responsibility for the content of this paper, including any text, code, or claims produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The problem formulation is given in Section 2 and the training objective of `L2T`, including the stability-aware weighting, in Section 3.1; Appendix B details the reconstruction loss and both uniformity regularizers (MMD and normalizing flow). The test statistic and its composite-null calibration are specified in Section 3.2, Appendix C, and Algorithm 1. All assumptions are stated in Section 3.2 and Appendices C and D; Appendix D also gives complete theorem statements and proofs. Appendix E.1 specifies the simulated systems (governing equations, parameters, initial conditions, and stability functionals), the baseline context distributions, the shift families (Table 2), sample sizes, training hyperparameters, the number of repetitions, and the hardware used; Appendix E.2 describes the TSA data source, data splits, normalization, context construction, and evaluation protocol. The TSA dataset is publicly available (Sokolović, 2024), and the synthetic data are generated by the simulators described in Appendix E.1. We will release the code for data generation, training, and testing upon publication.

REFERENCES

Zohreh Abbasi and Xinzhi Liu. Switched epidemic models: A contact rate dependent analysis. *Applied Mathematical Modelling*, 128:168–194, 2024.

Paul M Anderson and Aziz A Fouad. *Power system control and stability*. John Wiley & Sons, 2008.

Ludwig Arnold. Random dynamical systems. In *Dynamical Systems: Lectures Given at the 2nd Session of the Centro Internazionale Matematico Estivo (CIME) held in Montecatini Terme, Italy, June 13–22, 1994*, pp. 1–43. Springer, 2006.

Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

Georges Bastin, Bertrand Haut, Jean-Michel Coron, and Brigitte d Andrea-Novel. Lyapunov stability analysis of networks of scalar conservation laws. *Networks and Heterogeneous media*, 2(4):749, 2007.

Felix Biggs, Antonin Schrab, and Arthur Gretton. Mmd-fuse: Learning and combining kernels for two-sample testing without data splitting. *Advances in Neural Information Processing Systems*, 36: 75151–75188, 2023.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

Bernard Bonzi, AA Fall, Abderrahman Iggidr, and Gauthier Sallet. Stability of differential susceptibility and infectivity epidemic models. *Journal of mathematical biology*, 62(1):39–64, 2011.

Kathryn Eleda Brenan, Stephen L Campbell, and Linda Ruth Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, 1995.

Nicolò Cangiotti, Marco Capolli, Mattia Sensi, and Sara Sottile. A survey on lyapunov functions for epidemic compartmental models. *Bollettino dell'Unione Matematica Italiana*, 17(2):241–257, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Shuyi Chen, Ferdinando Fioretto, Feng Qiu, and Shixiang Zhu. Global-decision-focused neural odes for proactive grid resilience management. *IEEE Transactions on Smart Grid*, pp. 1–1, 2025. doi: 10.1109/TSG.2025.3642407.

Hsiao-Dong Chiang, Felix Wu, and Peon Varaiya. Foundations of direct methods for power system transient stability analysis. *IEEE Transactions on Circuits and systems*, 34(2):160–173, 1987.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation*, pp. 103–111, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

DIgSILENT GmbH. Digsilent powerfactory, n.d. URL `https://www.digsilent.de/en/powerfactory.html`.

David JD Earn and C Connell McCluskey. Global stability of epidemic models with uniform susceptibility. *Proceedings of the National Academy of Sciences*, 122(49):e2510156122, 2025.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

A Fall, Abderrahman Iggidr, Gauthier Sallet, and Jean-Jules Tewa. Epidemiological models and lyapunov functions. *Mathematical Modelling of Natural Phenomena*, 2(1):62–83, 2007.

Audrey Favache and Denis Dochain. Thermodynamics and chemical systems stability: The cstr case study revisited. *Journal of Process Control*, 19(3):371–379, 2009.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C. Lipton. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Tom Ginsberg, Zhongyuan Liang, and Rahul G Krishnan. A learning based hypothesis test for harmful covariate shift. *arXiv preprint arXiv:2212.02742*, 2022.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `http://jmlr.org/papers/v13/gretton12a.html`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Yicun Huang, Changfu Zou, Yang Li, and Torsten Wik. Minn: Learning the dynamics of differential-algebraic equations and application to battery modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11331–11344, 2024.

Josh Kacher, Yao Xie, Sven P. Voigt, Shixiang Zhu, Henry Yuchi, Jordan Key, and Surya R. Kalidindi. In situ transmission electron microscopy: Signal processing challenges and examples. *IEEE Signal Processing Magazine*, 39(1):89–103, 2022. doi: 10.1109/MSP.2021.3119284.

Arne Kesting and Martin Treiber. How reaction time, update time, and adaptation time influence the stability of traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(2):125–137, 2008.

Rafail Khasminskii. *Stochastic stability of differential equations*, volume 66. Springer Science & Business Media, 2011.

Ilmun Kim, Aaditya Ramdas, Aarti Singh, and Larry Wasserman. Classification accuracy as a proxy for two-sample testing. *The Annals of Statistics*, 49(1):411–434, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Matthias Kirchler, Shahryar Khorasani, Marius Kloft, and Christoph Lippert. Two-sample testing using deep learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1398. PMLR, 2020.

James Koch, Madelyn Shapiro, Himanshu Sharma, Draguna Vrabie, and Ján Drgoňa. Learning neural differential algebraic equations via operator splitting. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 2077–2084. IEEE, 2025.

Andrei Korobeinikov and Graeme C Wake. Lyapunov functions and global stability for sir, sirs, and sis epidemiological models. *Applied mathematics letters*, 15(8):955–960, 2002.

Prabha Kundur. *Power System Stability and Control*. McGraw-Hill, New York, 1994.

Prabha Kundur, John Paserba, Venkat Ajjarapu, Göran Andersson, Anjan Bose, Claudio Canizares, Nikos Hatziargyriou, David Hill, Alex Stankovic, Carson Taylor, et al. Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions. *IEEE transactions on Power Systems*, 19(3):1387–1401, 2004.

Prabha Kundur et al. Power system stability. *Power system stability and control*, 10(1):7–1, 2007.

Peter Kunkel and Volker Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Zürich, 2006.

Dieter Landolt. *Corrosion and surface chemistry of metals*. EPFL press, 2007.

Joseph LaSalle. Some extensions of liapunov's second method. *IRE Transactions on circuit theory*, 7 (4):520–527, 1960.

Jun-Young Lee and Hyung-Jun Chae. 6.6-kw onboard charger design using dcm pfc converter with harmonic modulation technique and two-stage dc/dc converter. *IEEE Transactions on Industrial Electronics*, 61(3):1243–1252, 2013.

Ke Li and Jitendra Malik. Learning to optimize. *arXiv preprint arXiv:1606.01885*, 2016.

Lijuan Li, Yuange Li, Yongdong Chen, Hongliang Liu, and Qingsong Zhang. Analysis of power system angle stability based on heun algorithm under gauss and jump excitation. In *Journal of Physics: Conference Series*, volume 2369, pp. 012036. IOP Publishing, 2022.

Ting Li, Jianguo Li, and Zhanxing Zhu. Neural lad: A neural latent dynamics framework for times series modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bISkJSa5Td.

Daniel Liberzon. *Switching in systems and control*, volume 190. Springer, 2003.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3122–3130, 2018.

Enrico Lovisari, Giacomo Como, Anders Rantzer, and Ketan Savla. Stability analysis and control synthesis for dynamical transportation networks. *arXiv preprint arXiv:1410.5956*, 2014.

Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

Digby D Macdonald. The point defect model for the passive state. *Journal of the Electrochemical Society*, 139(12):3434–3449, 1992.

Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *International conference on machine learning*, pp. 14935–14947. PMLR, 2022.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024.

Federico Milano. *Power system modelling and scripting*. Springer Science & Business Media, 2010.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Panagiotis N Papadopoulos and Jovica V Milanović. Probabilistic framework for transient stability assessment of power systems with high penetration of renewable generation. *IEEE Transactions on Power Systems*, 32(4):3078–3088, 2016.

Aleksandr Podkopaev and Aaditya Ramdas. Sequential predictive two-sample and independence testing. *Advances in neural information processing systems*, 36:53275–53307, 2023.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

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.

Christopher V Rao and Adam P Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *The Journal of chemical physics*, 118(11): 4999–5010, 2003.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

John Brian Roberts and Pol D Spanos. *Random vibration and statistical linearization*. Courier Corporation, 2003.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. An online learning approach to interpolation and extrapolation in domain generalization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2641–2657. PMLR, 2022.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Sobhan Sadri and Christine Wu. Stability analysis of a nonlinear vehicle model in plane motion using the concept of lyapunov exponents. *Vehicle System Dynamics*, 51(6):906–924, 2013.

Peter W Sauer, Mangalore A Pai, and Joe H Chow. *Power system dynamics and stability: with synchrophasor measurement and power system toolbox*. John Wiley & Sons, 2017.

Kazimierz Sobczyk. *Stochastic differential equations: with applications to physics and engineering*, volume 40. Springer Science & Business Media, 2013.

Živko Sokolović. Dataset for transient stability assessment of IEEE 39-bus system. Mendeley Data, Version 1, 2024. URL https://doi.org/10.17632/p992nhb8ss.1.

Subra Suresh. *Fatigue of materials*. Cambridge university press, 1998.

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

R Venkatesh, Deepak Dasaratha Rao, V Sangeetha, Ch Subbalakshmi, V Bala Dhandayuthapani, and R Mekala. Enhancing stability in autonomous control systems through fuzzy gain scheduling (fgs) and lyapunov function analysis. *International Journal of Applied and Computational Mathematics*, 10(4):130, 2024.

Gregory C Walsh, Hong Ye, and Linda G Bushnell. Stability analysis of networked control systems. *IEEE transactions on control systems technology*, 10(3):438–446, 2002.

Hangzhou Wang, Zhihong Yuan, Bingzhen Chen, Xiaorong He, Jinsong Zhao, and Tong Qiu. Analysis of the stability and controllability of chemical processes. *Computers & chemical engineering*, 35 (6):1101–1109, 2011.

Prince Zizhuang Wang, Shuyi Chen, Jinhao Liang, Ferdinando Fioretto, and Shixiang Zhu. Gen-dfl: Decision-focused generative learning for robust decision making. *arXiv preprint arXiv:2502.05468*, 2025.

Xiaoting Wang and Sophie G Schirmer. Analysis of lyapunov method for control of quantum states. *IEEE Transactions on Automatic control*, 55(10):2259–2270, 2010.

Xinyi Wang and Lang Tong. Innovations autoencoder and its application in one-class anomalous sequence detection. *Journal of Machine Learning Research*, 23(49):1–27, 2022. URL `http://jmlr.org/papers/v23/21-0735.html`.

Yu Wang, Xiaopeng Li, Junfang Tian, and Rui Jiang. Stability analysis of stochastic linear car-following models. *Transportation Science*, 54(1):274–297, 2020.

Song Wei, Shixiang Zhu, Minghe Zhang, and Yao Xie. Goodness-of-fit test for mismatched self-exciting processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1243–1251. PMLR, 2021.

Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. End to end learning and optimization on graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Qing-Hua Wu, Yuqing Lin, Chao Hong, Yinsheng Su, Tianhao Wen, and Yang Liu. Transient stability analysis of large-scale power systems: A survey. *CSEE Journal of Power and Energy Systems*, 9 (4):1284–1300, 2023.

Qiao Xiao, Boqian Wu, Yu Zhang, Shiwei Liu, Mykola Pechenizkiy, Elena Mocanu, and Decebal Constantin Mocanu. Dynamic sparse network for time series classification: Learning what to "see". *Advances in Neural Information Processing Systems*, 35:16849–16862, 2022.

Rose Yu and Rui Wang. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121(27):e2311808121, 2024.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

Lingxiao Yuan, Harold S Park, and Emma Lejeune. Towards out of distribution generalization for problems in mechanics. *Computer Methods in Applied Mechanics and Engineering*, 400:115569, 2022.

Henry Yuchi, Shixiang Zhu, Li Dong, Yigit M. Arisoy, and Matthew C. Spencer. New user event prediction through the lens of causal inference. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2953–2961. PMLR, 03–05 May 2025. URL `https://proceedings.mlr.press/v258/yuchi25a.html`.

Aoran Zhang, Wenbin Zhou, Liyan Xie, and Shixiang Zhu. Recurrent neural goodness-of-fit test for time series. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2917–2925. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/zhang25i.html.

Guowei Zhang, Xianguang Kong, Qibin Wang, Jingli Du, Jinrui Wang, and Hongbo Ma. Single domain generalization method based on anti-causal learning for rotating machinery fault diagnosis. *Reliability Engineering & System Safety*, 250:110252, 2024a.

Xi Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis L Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modelling. *Advances in Neural Information Processing Systems*, 37:107198–107224, 2024b.

Wenbin Zhou, Liyan Xie, Zhigang Peng, and Shixiang Zhu. Sequential change point detection via denoising score matching. *arXiv preprint arXiv:2501.12667*, 2025.

Wenbin Zhou, Liyan Xie, and Shixiang Zhu. Score-based change-point detection and region localization for spatio-temporal point processes. *arXiv preprint arXiv:2602.04798*, 2026.

Shixiang Zhu, Liyan Xie, Minghe Zhang, Rui Gao, and Yao Xie. Distributionally robust weighted k-nearest neighbors. *Advances in Neural Information Processing Systems*, 35:29088–29100, 2022.

A Motivating Examples

We highlight two application domains in which the formulation in (4) arises naturally. In both, the dynamics follow physical laws with unknown parameters, the operating context shifts over time, and trajectories under a new regime are costly or risky to observe, so stability must be judged from baseline trajectories and deployment contexts alone.

Power systems with stochastic loads or generations. Electric power grids must maintain frequency, voltage, and thermal limits while accommodating stochastic demand and renewable generation. Their dynamics are described by the generator swing equations coupled with algebraic power-flow constraints (Anderson & Fouad, 2008; Kundur, 1994; Sauer et al., 2017). The state $s(t)$ includes generator rotor angles and frequencies, while algebraic variables $a(t)$ enforce network power balance and voltage constraints. The exogenous context X captures load profiles, renewable outputs, or network configurations. A baseline domain corresponds to normal operating conditions under which stability margins are well established. When the context distribution shifts—due to large data-center loads, renewable intermittency, or extreme weather—new operating conditions arise. Full dynamic trajectories under these regimes are typically unavailable, as large-scale disturbance experiments or exhaustive simulations are costly and potentially risky. Instead, operators often observe only contextual information such as load forecasts or topology changes. The proposed framework evaluates whether the probability of violating stability limits over a time horizon remains below a prescribed level under the shifted context distribution, enabling statistically grounded security assessment under emerging grid conditions.

Material degradation and structural integrity. Material degradation processes such as corrosion and fatigue crack growth are governed by physical laws that couple time-evolving damage dynamics with thermodynamic and mechanical constraints. These processes are commonly modeled by differential equations for damage evolution together with algebraic constitutive or equilibrium relations (Kacher et al., 2022; Landolt, 2007; Macdonald, 1992; Suresh, 1998). The state $s(t)$ may represent corrosion depth or crack length, while the exogenous context X encodes environmental and usage conditions such as humidity, temperature, salinity, or load cycles. A baseline regime corresponds to operating conditions where degradation behavior and safety margins are well understood. When the environment changes, the distribution of X shifts, yet full degradation trajectories under the new regime are typically unavailable without long-term monitoring or destructive testing. Our framework asks whether the probability that degradation exceeds a critical threshold over a given horizon remains below a target level under the new context distribution, enabling uncertainty-aware maintenance and inspection decisions.

B Implementation Details: Representation Learning for Stability Testing

This section describes practical implementations of the three components in the loss function (10). Throughout, we train the parameters of the encoder ϕ, decoder ψ, and update function φ using mini-batch stochastic gradient methods. Each iteration samples a mini-batch $\mathcal{B} = \{(x_i, \mathcal{S}_i)\}_{i\in\mathcal{I}}$ from the baseline data, computes latent representations $u_i = \phi(x_i)$, rolls out the latent dynamics in (7) to obtain reconstructions, evaluates the losses on the mini-batch, and updates parameters by backpropagation through the unrolled computation graph. We use the MMD regularizer in the simulation experiments, where the latent is one-dimensional, and the normalizing-flow regularizer on the TSA benchmark, where the latent has eight dimensions.

Reconstruction loss in (8): The sample size K_i may vary across trajectories (irregular sampling). We either (i) pad shorter sequences and apply a mask, or (ii) subsample a fixed-length window per trajectory per iteration. Gradients are computed by differentiating through $\hat{s}_{i,k} = \psi(h_{i,k})$ and the recurrent updates $h_{i,k+1} = \varphi(s_{i,k}, h_{i,k}, u_i)$.

Uniformity regularization in (9): We encourage the latent representations $u = \phi(x) \in [0,1]^d$ to follow $\mathrm{Unif}(0,1)^d$, so that downstream stability testing reduces to detecting deviations from uniformity in the new domain. We present two practical implementations.

Maximum Mean Discrepancy (MMD) to Uniform. Let $\{u_i\}_{i=1}^B$ be latent representations from a mini-batch with B being the sample size of the mini-batch, and let $\{v_j\}_{j=1}^B$ be i.i.d. samples from $v_j \sim \mathrm{Unif}(0,1)^d$. The squared maximum mean discrepancy (MMD) (Gretton et al., 2012) between p_U and the uniform distribution is

$$\mathrm{MMD}^2(p_U, \mathrm{Unif}) = \Big\| \mathbb{E}_{u\sim p_U}[\kappa(u,\cdot)] - \mathbb{E}_{v\sim \mathrm{Unif}}[\kappa(v,\cdot)] \Big\|_{\mathscr{H}_\kappa}^2,$$

where κ is a positive definite kernel and $\mathscr{H}_\kappa$ is the associated reproducing kernel Hilbert space. Using unbiased (or biased) V-statistic estimators, we implement

$$\begin{aligned}\mathcal{L}_{\mathrm{reg}}^{\mathrm{MMD}} &\coloneqq \widehat{\mathrm{MMD}}^2(\{u_i\}_{i=1}^B, \{v_j\}_{j=1}^B) \\ &= \frac{1}{B^2}\sum_{i=1}^B\sum_{i'=1}^B \kappa(u_i,u_{i'}) + \frac{1}{B^2}\sum_{j=1}^B\sum_{j'=1}^B \kappa(v_j,v_{j'}) - \frac{2}{B^2}\sum_{i=1}^B\sum_{j=1}^B \kappa(u_i,v_j).\end{aligned}$$

A common choice is the Gaussian kernel $\kappa(u,u') = \exp(-\|u-u'\|^2/(2\sigma^2))$ with bandwidth σ chosen by a median heuristic or a small grid. This loss is fully differentiable in $\{u_i\}$, hence differentiable in ϕ via $u_i = \phi(x_i)$.

At each iteration, we proceed as follows: (i) compute $u_i = \phi(x_i)$ for $i = 1,\dots,B$; (ii) sample $v_j \sim \mathrm{Unif}(0,1)^d$ for $j = 1,\dots,B$; (iii) evaluate the kernel matrices $K_{uu} = [\kappa(u_i,u_{i'})]$, $K_{vv} = [\kappa(v_j,v_{j'})]$, $K_{uv} = [\kappa(u_i,v_j)]$; (iv) compute $\mathcal{L}_{\mathrm{reg}}^{\mathrm{MMD}}$ via the above V-statistic and backpropagate. This option is typically stable and requires no density estimation when d is small.

Direct Entropy Maximization via a Normalizing Flow. An alternative is to model the latent distribution explicitly using a bijective normalizing flow (Lipman et al., 2022; Rezende & Mohamed, 2015). Let $v \in \mathbb{R}^d$ be a base random variable with known density p_V (*e.g.*, uniform). Let $\mathcal{F}_\beta : \mathbb{R}^d \to (0,1)^d$ be an invertible map parameterized by β, and define the latent code as

$$u = \mathcal{F}_\beta(v), \qquad v \sim p_V.$$

The induced density of u is given by the change-of-variables formula:

$$p_U(u) = p_V\big(\mathcal{F}_\beta^{-1}(u)\big) \cdot \Big|\det \nabla_u \mathcal{F}_\beta^{-1}(u)\Big|. \tag{13}$$

To couple this with the context encoder, we parameterize ϕ itself as a flow mapping from a simple base distribution conditioned on x. A simple and effective approach is to fit a flow to the empirical latent representations and use the flow log-density to approximate differential entropy.

The differential entropy satisfies $\mathcal{H}(p_U) = -\mathbb{E}_{u\sim p_U}[\log p_U(u)]$. Using (13), we obtain

$$\begin{aligned}\mathcal{H}(p_U) &= -\mathbb{E}_{u\sim p_U}\big[\log p_U(u)\big] \\ &= -\mathbb{E}_{u\sim p_U}\Big[\log p_V\big(\mathcal{F}_\beta^{-1}(u)\big) + \log\big|\det \nabla_u \mathcal{F}_\beta^{-1}(u)\big|\Big].\end{aligned} \tag{14}$$

Thus, maximizing entropy is equivalent to minimizing the negative log-likelihood under the flow model. In practice, given a mini-batch $\{u_i\}_{i=1}^B$, we implement

$$\mathcal{L}_{\mathrm{reg}}^{\mathrm{flow}} \coloneqq \frac{1}{B}\sum_{i=1}^B \Big[-\log p_V(v_i) - \log\big|\det \nabla_u \mathcal{F}_\beta^{-1}(u_i)\big|\Big], \qquad v_i = \mathcal{F}_\beta^{-1}(u_i),$$

and differentiate through both ϕ (via $u_i = \phi(x_i)$) and the flow parameters β.

To explicitly enforce $\mathrm{Unif}(0,1)^d$, we can choose the base density p_V to be uniform and select $\mathcal{F}_\beta$ as an invertible transformation on $(0,1)^d$. In this case, minimizing $\mathcal{L}_{\mathrm{reg}}^{\mathrm{flow}}$ encourages $\{u_i\}$ to match the target law by making them high-likelihood under a flow whose output is constrained to $[0,1]^d$.

At each iteration, we proceed as follows: (i) compute $u_i = \phi(x_i)$ for $i = 1,\dots,B$; (ii) evaluate $v_i = \mathcal{F}_\beta^{-1}(u_i)$ and $\log|\det \nabla_u \mathcal{F}_\beta^{-1}(u_i)|$ via the flow; (iii) compute $\mathcal{L}_{\mathrm{reg}}^{\mathrm{flow}}$ and backpropagate through (ϕ,β). Compared with the MMD-based approach, the flow-based formulation provides an explicit parametric density model for p_U and a direct estimate of differential entropy. It scales more naturally to higher-dimensional latent spaces, albeit at the cost of increased computational overhead and additional modeling complexity.

Algorithm 1 Learning-to-Test (L2T) with composite-null calibration

Require: Baseline train/calibration data; new contexts $\{x'_j\}_{j=1}^m$; D, α, β; grid $\mathcal{G}_\alpha \subset [0, \alpha]$; repetitions B.

Train $\widehat{\phi}$ on the training split and freeze it.
Form $\widehat{\mathscr{R}}_0$ and $\widehat{\mathscr{R}}_1$ from the independent labeled calibration split.
$\widehat{Q}^* \leftarrow (1-\alpha/2)\widehat{R}_0 + (\alpha/2)\widehat{R}_1$; keep this reference fixed.
for $\lambda \in \mathcal{G}_\alpha$ **do**
  for $b = 1, \dots, B$ **do**
    Draw a batch of size m from $(1-\lambda)\widehat{R}_0 + \lambda\widehat{R}_1$ and form its empirical law $\widehat{Q}_\lambda^{(b)}$.
    $\widehat{\mathcal{T}}_\lambda^{(b)} \leftarrow D(\widehat{Q}_\lambda^{(b)}, \widehat{Q}^*)$.
  end for
  $\widehat{r}_\beta(\lambda) \leftarrow \mathrm{Quantile}_{1-\beta}\{\widehat{\mathcal{T}}_\lambda^{(b)}\}_{b=1}^B$.
end for
$\widehat{r}_{\alpha,\beta} \leftarrow \max_{\lambda \in \mathcal{G}_\alpha} \widehat{r}_\beta(\lambda)$.
$\widehat{Q}_{P_1} \leftarrow m^{-1}\sum_{j=1}^m \delta_{\widehat{\phi}(x'_j)}$ and $\widehat{\mathcal{T}}_{\mathrm{test}} \leftarrow D(\widehat{Q}_{P_1}, \widehat{Q}^*)$.
Reject $H_0 : P_1 \in \mathcal{P}_0$ if $\widehat{\mathcal{T}}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}$.
return $\mathbb{1}\{\widehat{\mathcal{T}}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}\}$.

C Details of Composite-Null Calibration

The null hypothesis $H_0 : P_1 \in \mathcal{P}_0$ is composite: it contains every deployment law whose violation probability is at most α, not only the baseline law. A two-sample test against the baseline therefore rejects safe deployments that merely differ from P_0, for instance those with fewer violations. Under Assumption 1, every null law induces a latent mixture $Q_\lambda = (1-\lambda)R_0 + \lambda R_1$ with $\lambda \in [0, \alpha]$, so it suffices to simulate these mixtures from labeled baseline data and to take the largest null quantile over λ. Centering the reference at $\lambda = \alpha/2$ places every alternative farther from the reference than any null law (Theorem 1).

Theorem 2 calibrates the statistic over the entire composite null $P \in \mathcal{P}_0$. To approximate this calibration for a learned encoder, reserve an independent labeled baseline split and form the empirical conditional latent pools

$$\widehat{\mathscr{R}}_0 = \{\widehat{\phi}(X_i) : y(X_i) = 0\}, \qquad \widehat{\mathscr{R}}_1 = \{\widehat{\phi}(X_i) : y(X_i) = 1\}.$$

Let $\widehat{R}_a$ denote the empirical distribution supported on $\widehat{\mathscr{R}}_a$. Construct the fixed empirical reference $\widehat{Q}^* = (1-\alpha/2)\widehat{R}_0 + (\alpha/2)\widehat{R}_1$. For each grid value $\lambda \in \mathcal{G}_\alpha \subset [0, \alpha]$, resample batches of size m from $\widehat{Q}_\lambda = (1-\lambda)\widehat{R}_0 + \lambda\widehat{R}_1$, compute their discrepancy to the same $\widehat{Q}^*$ used at deployment, and retain the largest empirical null quantile. The reference and calibration batches use the same labeled split; independence is required between this split and the deployment sample, not between the two calibration calculations.

The calibration result conditions on the trained encoder and assumes: (i) the calibration and deployment samples are independent, and both class-conditional calibration sample sizes diverge; (ii) $\widehat{R}_a$ consistently estimates R_a and $\widehat{Q}^*$ consistently estimates the centered mixture $Q_{\alpha/2} = (1-\alpha/2)R_0 + (\alpha/2)R_1$ in the topology induced by D; (iii) $B \to \infty$ and $h(\mathcal{G}_\alpha) \to 0$; and (iv) for each fixed deployment size m, the quantile of the two-sample statistic is continuous in λ and jointly consistent when the class-conditional laws and the reference are replaced by their empirical estimates, uniformly over the grid. For consistency, we additionally assume that $D(\widehat{Q}_{P_1}, \widehat{Q}^*) \to D(Q_{P_1}, Q_{\alpha/2})$ and that the null quantiles converge uniformly over $\lambda \in [0, \alpha]$ as $m \to \infty$. These conditions hold, for example, for MMD with a bounded continuous kernel under the usual quantile-continuity condition; a characteristic kernel additionally ensures distributional separation.

Remark 3. *The oracle supremum in* (23)*, which uses the population class-conditional laws and centered reference, has exact finite-sample validity for each fixed deployment size. The implemented statistic estimates both the reference and the calibration mixtures from the same labeled split; its stated validity is asymptotic. Finite calibration, Monte Carlo, and grid errors can be reduced*

with more calibration observations, more repetitions, a finer grid, or an additional conservative correction, and, for small calibration samples, by drawing the reference and the calibration batches from independent halves of the labeled split. The interpretation also requires the conditional latent laws to remain representative under the deployment shifts under consideration. Here α denotes the allowed violation probability, whereas β denotes the test significance level.

Remark 4 (Reference choice and power)**.** *For $Q_\lambda = (1-\lambda)R_0 + \lambda R_1$, the centered reference $Q_{\alpha/2}$ separates every $P_1 \notin \mathcal{P}_0$ from the composite null under a separating IPM. Using the unweighted baseline law Q_{P_0} as the reference would still permit composite-null calibration, but if $\mathbb{P}_{P_0}(y(X) = 1) > \alpha/2$, some alternatives need not be separated from the null. This is why the empirical two-sample test constructs $\widehat{Q}^*$ with mixing weight $\alpha/2$ rather than using the raw baseline mixture.*

D Theoretical Results and Proofs

This section collects the theory. Appendix D.1 shows that some assumption linking the baseline and deployment domains is necessary, and Appendix D.2 proves Theorem 1. Appendix D.3 quantifies what happens when Assumption 1 holds only approximately and illustrates the result, and Appendix D.4 proves Theorem 2.

D.1 Why a Link Between the Domains Is Necessary

The observed data consist of baseline context–trajectory pairs, whose law is determined by P_0 and by the trajectories $s_\theta(\cdot; x)$ for contexts x in the support $\mathcal{S}_0$ of P_0, and of deployment contexts, whose law is P_1. The next proposition shows that these data cannot settle (4) once P_1 places enough mass where the baseline domain provides no trajectories, so some assumption linking the two domains is unavoidable.

Proposition 1 (Non-identifiability without a link between the domains)**.** *Let $\mathcal{S}_0$ be the support of P_0 and suppose that $P_1(\mathscr{X} \setminus \mathcal{S}_0) > \alpha$. Then there exist two systems whose trajectories coincide for every context in $\mathcal{S}_0$, such that $P_1 \in \mathcal{P}_0$ for the first and $P_1 \notin \mathcal{P}_0$ for the second. The observed data have the same law under both systems, so any test of* (4) *based on these data rejects with the same probability under both; a test with level β under the first has power at most β under the second.*

Proof. Let the first system never violate the tolerance, so that $\mathbb{P}_{P_1}(y(X) = 1) = 0 \le \alpha$. Let the second system have the same trajectories on $\mathcal{S}_0$ and violate the tolerance for every context outside $\mathcal{S}_0$, so that $\mathbb{P}_{P_1}(y(X) = 1) \ge P_1(\mathscr{X} \setminus \mathcal{S}_0) > \alpha$. Such a pair exists within model (1): because f_θ may depend on the context, adding to the first system's f_θ a forcing term that is active only for $x \notin \mathcal{S}_0$ and drives the trajectory beyond the tolerance yields the second system; the added term is constant in (s, a), so smoothness and well-posedness are preserved. The baseline data depend on the trajectories only through contexts in $\mathcal{S}_0$, and the deployment data contain no trajectories, so the observed data have the same law under both systems. A test is a function of these data, so its rejection probability is the same under both systems. □

Figure 3(a, b) illustrates the construction. Support overlap together with a model for the density ratio $\mathrm{d}P_1/\mathrm{d}P_0$ is one way to link the domains; Assumption 1 is another, imposed on the learned representation, and Proposition 2 quantifies how its approximate violation affects the test.

D.2 Proof of Theorem 1

Full statement. Theorem 1 holds under the following conditions, some of which the main text leaves implicit: (i) $0 < \alpha < 1$; (ii) the deployment law P_1 satisfies Assumption 1; (iii) $R_0 \neq R_1$; (iv) the centered mixture $Q^\dagger := (1-\alpha/2)R_0 + (\alpha/2)R_1$ is non-atomic, which holds, for example, when R_0 and R_1 have densities on $[0,1]^d$; and (v) $D(\mu, \nu) = \sup_{f \in \mathcal{F}} |\int f \,\mathrm{d}(\mu - \nu)|$ is an integral probability metric that separates Borel probability measures on $[0,1]^d$. Condition (iv) is the regularity condition of the main text and is used only to construct T; conditions (iii) and (v) make the threshold r_α positive.

Intuition. Under Assumption 1, every deployment law induces a latent law on the segment between R_0 and R_1, at the position given by its violation probability. The null is the part of the segment with violation probability at most α. Centering the reference at the midpoint $\alpha/2$ of this part turns the null into a ball of radius $(\alpha/2)D(R_0, R_1)$ around the reference, and a transformation of the latent makes the reference uniform.

Proof. We consider the nondegenerate case $0 < \alpha < 1$. For a deployment law P that satisfies Assumption 1, write $\pi_P \coloneqq \mathbb{P}_{X\sim P}(y(X) = 1)$. The assumption implies

$$Q_P = (1 - \pi_P)R_0 + \pi_P R_1. \tag{15}$$

If a class has probability zero, its conditional law is not needed because its mixture weight is zero. Thus, all admissible deployment laws induce distributions on the line segment joining R_0 and R_1, with the violation probability π_P as the mixing weight.

Define the centered mixture

$$Q^\dagger \coloneqq \left(1 - \frac{\alpha}{2}\right) R_0 + \frac{\alpha}{2} R_1.$$

By condition (iv), $Q^\dagger$ is non-atomic. Because $[0, 1]^d$ is a standard Borel space, the isomorphism theorem for non-atomic Borel probability measures yields a measurable map $T : [0, 1]^d \to [0, 1]^d$ that is injective outside a $Q^\dagger$-null set and satisfies

$$T_\# Q^\dagger = Q^* \coloneqq \mathrm{Unif}([0, 1]^d).$$

For $d = 1$, the probability integral transform gives such a map, up to changes on a $Q^\dagger$-null set.

Let

$$\widetilde{R}_a \coloneqq T_\# R_a, \qquad \widetilde{Q}_P \coloneqq T_\# Q_P, \qquad a \in \{0, 1\}.$$

Applying the same transformation to both classes preserves Assumption 1, and (15) becomes

$$\widetilde{Q}_P = (1 - \pi_P)\widetilde{R}_0 + \pi_P \widetilde{R}_1, \qquad Q^* = \left(1 - \frac{\alpha}{2}\right) \widetilde{R}_0 + \frac{\alpha}{2} \widetilde{R}_1.$$

Moreover, $R_a \ll Q^\dagger$ for $a \in \{0, 1\}$, because both mixture weights of $Q^\dagger$ are positive. Let E be a Borel set of full $Q^\dagger$-measure on which T is injective; it also has full R_a-measure. By the Lusin–Souslin theorem, T maps Borel subsets of E to Borel sets, and injectivity gives $T^{-1}(T(B)) \cap E = B$, so $R_a(B) = \widetilde{R}_a(T(B))$ for every Borel $B \subseteq E$. Hence equality $\widetilde{R}_0 = \widetilde{R}_1$ would imply $R_0 = R_1$, and therefore $\widetilde{R}_0 \neq \widetilde{R}_1$.

Subtracting the two mixture representations gives

$$\widetilde{Q}_P - Q^* = \left(\pi_P - \frac{\alpha}{2}\right) (\widetilde{R}_1 - \widetilde{R}_0).$$

For an integral probability metric $D(\mu, \nu) = \sup_{f\in\mathcal{F}} |\int f \,\mathrm{d}(\mu - \nu)|$, absolute homogeneity in the signed measure $\mu - \nu$ therefore yields

$$D(\widetilde{Q}_P, Q^*) = \left|\pi_P - \frac{\alpha}{2}\right| D(\widetilde{R}_0, \widetilde{R}_1). \tag{16}$$

Because D is separating and $\widetilde{R}_0 \neq \widetilde{R}_1$, the final factor is strictly positive. Since $\pi_P \in [0, 1]$,

$$\pi_P \leq \alpha \quad \Longleftrightarrow \quad \left|\pi_P - \frac{\alpha}{2}\right| \leq \frac{\alpha}{2} \quad \Longleftrightarrow \quad D(\widetilde{Q}_P, Q^*) \leq r_\alpha,$$

where

$$r_\alpha \coloneqq \frac{\alpha}{2} D(\widetilde{R}_0, \widetilde{R}_1) > 0.$$

Taking $P = P_1$ proves (12). After replacing ϕ by $\widetilde{\phi} = T \circ \phi$, dropping the tildes recovers the notation used in the theorem. In particular, $r_\alpha = (\alpha/2)\,\mathrm{TV}(R_0, R_1)$ for total variation and $r_\alpha = (\alpha/2)\,\mathrm{MMD}_k(R_0, R_1)$ for MMD with a characteristic kernel. If squared MMD is used as the statistic, the corresponding threshold is r_α^2. □

D.3 Robustness to Approximation

Assumption 1 rarely holds exactly for a learned encoder. We consider two kinds of approximation: imperfect recovery of the label by the latent, and drift of the class-conditional latent laws. In both cases the guarantees degrade gracefully: the error enters additively and is at most the size of the approximation.

D.3.1 Approximate Latent Sufficiency

The main reformulation (Theorem 1) rests on class-conditional invariance and does not require the latent to recover individual labels. We give here a complementary route through approximate label recovery, which applies without class-conditional invariance, and quantify how it degrades when the learned representation only approximately recovers $y(X)$. This result does not assume that approximate sufficiency alone produces uniformity or conditional reference alignment.

Assumption 2 (Approximate latent sufficiency)**.** *There is a measurable decision rule $\rho : [0,1]^d \to \{0,1\}$ such that, for $e \in \{0,1\}$, with $X \sim P_e$ and $U = \phi(X)$,*

$$\mathbb{P}_{P_e}\big(y(X) \neq \rho(U)\big) \leq \varepsilon_e.$$

Theorem 3 (Robust latent reformulation)**.** *Under Assumption 2, let $A_1 = \{u : \rho(u) = 1\}$, $p_e = \mathbb{E}_{P_e}[y(X)]$, and $Q_e = P_e \circ \phi^{-1}$. Then*

$$|p_e - Q_e(A_1)| \leq \varepsilon_e, \qquad e \in \{0,1\}. \tag{17}$$

Consequently,

$$\mathrm{TV}(Q_1, Q_0) \geq |p_1 - p_0| - \varepsilon_0 - \varepsilon_1. \tag{18}$$

If the ideal reference-centering condition $Q^*(A_1) = \alpha/2$ also holds, then

$$\mathrm{TV}(Q_1, Q^*) \geq \left|p_1 - \frac{\alpha}{2}\right| - \varepsilon_1. \tag{19}$$

In particular, $\mathrm{TV}(Q_1, Q^*) \leq \alpha/2 - \varepsilon_1$ certifies $p_1 \leq \alpha$. Moreover, whenever $|p_1 - p_0| > \varepsilon_0 + \varepsilon_1$, we have $Q_1 \neq Q_0$; a characteristic-kernel MMD can therefore detect this fixed alternative consistently.

Proof. Set $\widehat{y} = \rho(U)$. For binary variables, the difference between the probabilities of either event is bounded by their disagreement probability, so

$$|p_e - Q_e(A_1)| = |\mathbb{P}_{P_e}(y(X) = 1) - \mathbb{P}_{P_e}(\widehat{y} = 1)| \leq \mathbb{P}_{P_e}(y(X) \neq \widehat{y}) \leq \varepsilon_e.$$

The variational definition of total variation and the triangle inequality give

$$\begin{aligned}\mathrm{TV}(Q_1, Q_0) &\geq |Q_1(A_1) - Q_0(A_1)| \\ &\geq |p_1 - p_0| - |p_1 - Q_1(A_1)| - |p_0 - Q_0(A_1)|,\end{aligned}$$

which proves (18). Similarly, if $Q^*(A_1) = \alpha/2$,

$$\mathrm{TV}(Q_1, Q^*) \geq |Q_1(A_1) - Q^*(A_1)| \geq \left|p_1 - \frac{\alpha}{2}\right| - \varepsilon_1.$$

If the left-hand side is at most $\alpha/2 - \varepsilon_1$, then $|p_1 - \alpha/2| \leq \alpha/2$ and hence $p_1 \leq \alpha$. The final claim follows directly from (18) and the defining property of a characteristic kernel. □

D.3.2 Class-Conditional Latent Drift

Assumption 1 asks the class-conditional latent laws to be identical in the two domains. For the trained encoder they move slightly even under benign shifts (Figure 8). The next proposition shows that such drift shifts the statistic by at most its own size, so small drift costs little.

Let $R_a \coloneqq \mathrm{Law}_{P_0}(U \mid y(X) = a)$ and, for a distribution P, let $p \coloneqq \mathbb{P}_P(y(X) = 1)$, $Q_P \coloneqq P \circ \phi^{-1}$, and $S_{P,a} \coloneqq \mathrm{Law}_P(U \mid y(X) = a)$. If a class has probability zero, its conditional law may be chosen arbitrarily because its mixture weight is zero. Define

$$\overline{Q}_p \coloneqq (1-p)R_0 + pR_1, \qquad \delta_D(P) \coloneqq (1-p)D(S_{P,0}, R_0) + pD(S_{P,1}, R_1).$$

Proposition 2 (Robustness to class-conditional latent drift)**.** *For any integral probability metric D,*

$$\left|D(Q_P, Q^*) - D(\overline{Q}_p, Q^*)\right| \le \delta_D(P). \tag{20}$$

Consequently, if $\delta_D(P) \le \varepsilon_{\text{inv}}$ for every $P \in \mathcal{P}_0$, then

$$\sup_{P\in\mathcal{P}_0} D(Q_P, Q^*) \le \sup_{0\le p\le\alpha} D(\overline{Q}_p, Q^*) + \varepsilon_{\text{inv}}. \tag{21}$$

Moreover, for any fixed $P_1 \notin \mathcal{P}_0$, let $p_1 \coloneqq \mathbb{P}_{P_1}(y(X) = 1)$. Then

$$\begin{aligned} &D(Q_{P_1}, Q^*) - \sup_{P\in\mathcal{P}_0} D(Q_P, Q^*) \\ &\quad\ge D(\overline{Q}_{p_1}, Q^*) - \sup_{0\le p\le\alpha} D(\overline{Q}_p, Q^*) - \delta_D(P_1) - \varepsilon_{\text{inv}}. \end{aligned} \tag{22}$$

Thus the population null boundary grows by at most ε_{inv}. If Q^* is the reference of the calibrated test and its threshold is inflated by ε_{inv}, then, under the consistency conditions of Appendix C and as m, n_{cal}, and B diverge and $h(\mathcal{G}_\alpha) \to 0$, the rejection probability tends to zero at every null law with $\delta_D(P) \le \varepsilon_{\text{inv}}$ that lies strictly inside the inflated boundary, and to one at every alternative for which the right-hand side of (22) *is positive. Assumption 1 makes all these drift radii zero.*

Proof. Since $Q_P = (1-p)S_{P,0} + pS_{P,1}$, convexity of an integral probability metric gives

$$D(Q_P, \overline{Q}_p) \le (1-p)D(S_{P,0}, R_0) + pD(S_{P,1}, R_1) = \delta_D(P).$$

The reverse triangle inequality proves (20). If $P \in \mathcal{P}_0$, then $p \le \alpha$, and (21) follows by taking the supremum. Applying the lower side of (20) to P_1 and subtracting (21) yields (22). □

Per-law guarantee with the centered reference. For the centered reference $Q^* = Q_{\alpha/2} \coloneqq (1-\alpha/2)R_0 + (\alpha/2)R_1$ of the calibrated test, the homogeneity argument behind (16), which holds for any integral probability metric, gives $D(\overline{Q}_p, Q^*) = |p - \alpha/2|\, D(R_0, R_1)$, and the population null radius is $r_\alpha = (\alpha/2)D(R_0, R_1)$. By (20), the population decision for a law P, which rejects when $D(Q_P, Q^*) > r_\alpha$, is therefore correct whenever

$$\delta_D(P) < \min\{p,\ \alpha - p\}\, D(R_0, R_1) \ \text{ if } p \le \alpha, \qquad \delta_D(P) < (p-\alpha)\, D(R_0, R_1) \ \text{ if } p > \alpha.$$

Drift that moves latent codes toward the other class, along the segment between R_0 and R_1, attains these bounds for alternatives and for null laws with $p \ge \alpha/2$. For null laws with $p < \alpha/2$, such drift first moves the latent law toward the reference; there the binding case is drift away from the segment, which can raise a false alarm once $\delta_D(P) > p\, D(R_0, R_1)$.

Illustration. Figure 3(c) illustrates Proposition 2 on the trained `L2T` latent of the mass–spring–damper system. Its horizontal axis is the safety margin $p_1 - \alpha$, with $p_1 = \mathbb{E}_{X\sim P_1}[y(X)]$, and its vertical axis, the violation level of Assumption 1, is the relative drift $\delta_D(P_1)/D(R_0, R_1)$. We inject class-conditional drift of known size toward the other class: for $p_1 \le \alpha$, a fraction of the stable class receives violation-like latent codes, and for $p_1 > \alpha$, a fraction of the violating class receives stable-like codes. Under this drift the population decision is correct exactly when $|p_1 - \alpha|\, D(R_0, R_1) > \delta_D(P_1)$, and by the per-law guarantee above this drift is the least favorable one for alternatives and for null laws with $p_1 \ge \alpha/2$. With $m = 2{,}000$ deployment contexts, and with the reference and the calibration batches of Algorithm 1 taken from the two halves of a labeled split of 20,000 contexts, the test decides correctly with high probability below this boundary and fails above it; the width of the transition reflects the finite deployment sample. Each cell of Figure 3(c) averages 200 deployment batches for each of five independent calibration splits, and hatched cells would require more drift than the violating class can carry. In words, a class-conditional drift of size $\delta_D(P_1)$ shifts the violation probability seen by the test by at most $\delta_D(P_1)/D(R_0, R_1)$. Under drift toward the other class, deployments far from the tolerance are therefore robust to sizeable violations of Assumption 1 and only near-boundary deployments are sensitive to it; in every case, a representation that separates the two classes more widely tolerates more drift, because all the bounds above scale with $D(R_0, R_1)$.

D.4 Proof of Theorem 2

Full statement. Theorem 2 conditions on the trained encoder $\widehat{\phi}$, whose class-conditional latent laws are R_0 and R_1, and applies to a deployment law P_1 that satisfies Assumption 1, with $p_1 \coloneqq \mathbb{P}_{P_1}(y(X) = 1)$. Type-I error control holds for every fixed deployment size m as $n_{\mathrm{cal}}, B \to \infty$ and $h(\mathcal{G}_\alpha) \to 0$, under conditions (i)–(iv) of Appendix C and continuity of the distribution function of the oracle statistic $\widehat{T}_m(Q_{p_1})$, defined below, at the oracle threshold $r^{(m)}_{\alpha,\beta}$. Power holds as m, n_{cal}, and B diverge and $h(\mathcal{G}_\alpha) \to 0$, under the consistency conditions of Appendix C and $D(R_0, R_1) > 0$.

Intuition. Validity holds because the oracle threshold is the largest null quantile over the whole segment $\lambda \in [0, \alpha]$, so it controls the Type-I error at every null law, and the empirical threshold converges to it. Power holds because every alternative lies a distance $(p_1 - \alpha)D(R_0, R_1)$ beyond the null radius, so the statistic eventually exceeds the threshold.

Proof. For $\lambda \in [0, \alpha]$, write $Q_\lambda \coloneqq (1-\lambda)R_0 + \lambda R_1$, so the population centered reference is $Q_{\alpha/2}$. For a fixed deployment sample size m, let $\widehat{Q}_{\lambda,m}$ be the empirical law of m i.i.d. draws from Q_λ and set $\widehat{T}_m(Q_\lambda) \coloneqq D(\widehat{Q}_{\lambda,m}, Q_{\alpha/2})$. Define

$$r^{(m)}_\beta(\lambda) \coloneqq \inf\left\{t : \mathbb{P}_{Q_\lambda}\left(\widehat{T}_m(Q_\lambda) \le t\right) \ge 1 - \beta\right\}.$$

The oracle composite-null threshold is

$$r^{(m)}_{\alpha,\beta} \coloneqq \sup_{0 \le \lambda \le \alpha} r^{(m)}_\beta(\lambda). \tag{23}$$

By the definition of the quantile, for every $\lambda \in [0, \alpha]$,

$$\mathbb{P}_{Q_\lambda}\left(\widehat{T}_m(Q_\lambda) > r^{(m)}_\beta(\lambda)\right) \le \beta.$$

Since $r^{(m)}_{\alpha,\beta}$ is at least each of these pointwise quantiles,

$$\mathbb{P}_{Q_\lambda}\left(\widehat{T}_m(Q_\lambda) > r^{(m)}_{\alpha,\beta}\right) \le \beta$$

for every $\lambda \in [0, \alpha]$. Under Assumption 1, any $P_1 \in \mathcal{P}_0$ has $p_1 \in [0, \alpha]$ and induces $Q_{P_1} = Q_{p_1}$, proving the oracle Type-I bound.

We next relate the empirical threshold to its oracle counterpart. Let $\widehat{r}^{(m)}_\beta(\lambda)$ be the Monte Carlo quantile of $D(\widehat{Q}^{(b)}_\lambda, \widehat{Q}^*)$, where both $\widehat{Q}^{(b)}_\lambda$ and the fixed reference $\widehat{Q}^*$ are constructed from the same empirical class-conditional laws $\widehat{R}_0, \widehat{R}_1$. Then

$$\begin{aligned}\left|\widehat{r}_{\alpha,\beta} - r^{(m)}_{\alpha,\beta}\right| \le \max_{\lambda \in \mathcal{G}_\alpha} \left|\widehat{r}^{(m)}_\beta(\lambda) - r^{(m)}_\beta(\lambda)\right| \\ + \left|\max_{\lambda \in \mathcal{G}_\alpha} r^{(m)}_\beta(\lambda) - \sup_{\lambda \in [0,\alpha]} r^{(m)}_\beta(\lambda)\right|.\end{aligned}$$

The first term converges to zero in probability as the class-conditional calibration sample sizes and B diverge: the assumptions in Appendix C cover estimation of the reference as well as estimation of the simulated deployment laws and their quantiles. The second term converges to zero as $h(\mathcal{G}_\alpha) \to 0$ by continuity in λ. Hence

$$\widehat{r}_{\alpha,\beta} - r^{(m)}_{\alpha,\beta} \xrightarrow{p} 0.$$

For every fixed m and $P_1 \in \mathcal{P}_0$, consistency of the shared calibration estimates and independence of calibration and deployment data imply that $\widehat{T}_{\mathrm{test}}$ converges in distribution to $\widehat{T}_m(Q_{p_1})$. Together with continuity of the distribution function of $\widehat{T}_m(Q_{p_1})$ at $r^{(m)}_{\alpha,\beta}$, this gives

$$\limsup_{\substack{n_{\mathrm{cal}}, B \to \infty \\ h(\mathcal{G}_\alpha) \to 0}} \mathbb{P}_{P_1}\left(\widehat{T}_{\mathrm{test}} > \widehat{r}_{\alpha,\beta}\right) \le \beta.$$

For consistency, fix $P_1 \notin \mathcal{P}_0$ and define the population separation

$$\Delta \coloneqq D(Q_{P_1}, Q_{\alpha/2}) - \sup_{0\le\lambda\le\alpha} D(Q_\lambda, Q_{\alpha/2}),$$

which is positive, as shown at the end of the proof. The consistency assumptions in Appendix C imply

$$\widehat{T}_{\text{test}} \xrightarrow{p} D(Q_{P_1}, Q_{\alpha/2}), \qquad \widehat{r}_{\alpha,\beta} \xrightarrow{p} \sup_{0\le\lambda\le\alpha} D(Q_\lambda, Q_{\alpha/2}).$$

Consequently,

$$\widehat{T}_{\text{test}} - \widehat{r}_{\alpha,\beta} \xrightarrow{p} \Delta > 0,$$

which proves that the rejection probability converges to one.

Finally, let $C_D \coloneqq D(R_0, R_1)$, which is positive by assumption, for example when D is a separating IPM and $R_0 \neq R_1$. The same mixture identity used in Theorem 1 gives

$$D(Q_\lambda, Q_{\alpha/2}) = \left|\lambda - \frac{\alpha}{2}\right| C_D.$$

Therefore,

$$\sup_{0\le\lambda\le\alpha} D(Q_\lambda, Q_{\alpha/2}) = \frac{\alpha}{2} C_D,$$

whereas every alternative has $p_1 > \alpha$ and hence

$$D(Q_{P_1}, Q_{\alpha/2}) - \sup_{0\le\lambda\le\alpha} D(Q_\lambda, Q_{\alpha/2}) = (p_1 - \alpha) C_D > 0.$$

Thus every fixed alternative is positively separated from the composite null by the centered population reference whenever $C_D > 0$. The same ordering holds if squared MMD is used as the statistic. □

E Details of Experimental Configuration and Additional Results

This section gives the configurations of the simulation experiments (Appendix E.1) and of the TSA benchmark (Appendix E.2), together with additional results. For the simulation experiments, the additional analyses ask, in turn, how the weights of the training objective shape the representation, how `L2T` compares with simulating the deployment trajectories or learning a trajectory surrogate, and whether Assumption 1 holds for the trained encoder.

E.1 Simulation Experiments

This section describes the synthetic experimental setups used to evaluate our method. We consider two representative systems: an unconstrained mass–spring–damper system and a constrained Cartesian pendulum governed by DAEs. The spring is an ordinary differential equation with a two-dimensional context, whose stability map can be evaluated by simulation over the whole context space; the pendulum adds an algebraic constraint and tests the method on a DAE.

Mass–Spring–Damper System. We consider a one-dimensional mass–spring–damper system with state $s(t) \in \mathbb{R}$ evolving as

$$m\ddot{s}(t) + c\dot{s}(t) + ks(t) = u(t; X), \quad t \in [0, T],$$

where $\theta = (m, c, k) = (1, 1, 2)$ are fixed physical parameters. Unless otherwise specified, we initialize the system at rest with $s(0) = 0$ and $\dot{s}(0) = 0$. The external forcing is parameterized by a random context $X = (X_1, X_2) \in \mathscr{X}$ via

$$u(t; X) = X_1 \sin(X_2 t),$$

where X_1 controls the forcing magnitude and X_2 the oscillation frequency. For a trajectory $s_\theta(\cdot; X)$, we define the stability functional

$$\eta\big(s_\theta(\cdot; X)\big) = \max_{t\in[t_\ell, t_u]} |s_\theta(t; X)|,$$

over the window $[t_\ell, t_u] = [6, 10]$, and declare the trajectory stable if $\eta \le \tau$ for a given tolerance $\tau > 0$.

The baseline context distribution is $P_0 = \mathcal{N}(\mathbf{0}, I_2)$ over (X_1, X_2). We consider two types of distribution shifts: (i) *mean shifts*, $P_1(\delta) = \mathcal{N}(\mu_0 + \delta v, I_2)$ with $\|v\|_2 = 1$, and (ii) *covariance shifts*, $P_1(\delta) = \mathcal{N}(\mathbf{0}, \Sigma_\delta)$ with

$$\Sigma_\delta = \begin{pmatrix} 1+\delta_1 & 0 \\ 0 & 1+\delta_2 \end{pmatrix}.$$

These perturbations induce regimes corresponding to null, boundary-transition, and alternative settings (Table 2).

Cartesian Pendulum System. We consider a constrained dynamical system governed by DAEs. Let $s(t) \in \mathbb{R}^3$ denote the bob position and $u(t; X) \in \mathbb{R}^3$ the pivot trajectory. The dynamics are

$$m\ddot{s}(t) = mg_0 + (s(t) - u(t; X))\, a(t), \quad g_0 = (0, 0, -g)^\top, \tag{24a}$$

$$0 = \tfrac{1}{2}\big(\|s(t) - u(t; X)\|_2^2 - \ell^2\big), \tag{24b}$$

where $\theta = (m, \ell, g) = (1, 1, 9.81)$ and $a(t)$ enforces the rigid rod constraint. Unless otherwise specified, we initialize the system with bob position

$$s(0) = (0.15,\, 0,\, -0.9887)^\top,$$

initial velocity

$$\dot{s}(0) = (0,\, 0.8,\, 0)^\top,$$

and initial pivot position

$$u(0; X) = (0,\, 0,\, 0)^\top.$$

The pivot trajectory is parameterized by a random context $X = (X_1, X_2, X_3) \in \mathscr{X}$ via

$$u(t; X) = X_1 \exp\!\left(-\frac{(t - t_0)^2}{2X_3^2}\right) \sin\big(X_2(t - t_0)\big)\, q,$$

where $q = (1, 0, 0)^\top$ and the excitation is centered at $t_0 = T/2 = 5$. Here X_1, X_2, and X_3 control the excitation magnitude, oscillation frequency, and temporal spread, respectively.

We define stability via the maximum swing angle

$$\eta(s_\theta(\cdot; X)) = \max_{t \in [t_0, T]} \varphi_\theta(t; X), \quad \varphi_\theta(t; X) = \arccos\!\left(-\frac{(s_\theta(t; X) - u(t; X))^\top e_z}{\ell}\right),$$

with $e_z = (0, 0, 1)^\top$. A trajectory is stable if $\eta \le \tau$ for $\tau \in (0, \pi/2)$. Geometrically, this restricts motion on the sphere of radius ℓ to a downward cone, equivalently a bounded region in the horizontal projection.

For the baseline distribution, we fix $X_3 = 1$ and sample (X_1, X_2) independently using scaled Beta distributions:

$$X_1 \sim A_{\min} + (A_{\max} - A_{\min})\, \mathrm{Beta}(\alpha_1, \beta_1), \quad [A_{\min}, A_{\max}] = [0.01, 0.3],$$

$$X_2 \sim \omega_{\min} + (\omega_{\max} - \omega_{\min})\, \mathrm{Beta}(\alpha_2, \beta_2), \quad [\omega_{\min}, \omega_{\max}] = [1, 5],$$

with $\alpha_1, \beta_1 = 1, 2$ and $\alpha_2, \beta_2 = 1, 4$. We consider two classes of shifts: (i) *marginal shifts*, induced by changing the Beta parameters of (X_1, X_2), and (ii) *dependence shifts*, induced by introducing correlation ρ between X_1 and X_2. These shifts define null, boundary-transition, and alternative regimes (Table 2).

Experimental Configurations. Across both synthetic examples, trajectories are simulated over the time interval $[0, T]$ with $T = 10$ and discretized using a uniform step size $\Delta t = 0.05$, yielding 201 time points per trajectory. For each setting in Table 2, contexts are drawn from the baseline distribution P_0 for training and held-out evaluation, and from the corresponding shifted distribution P_1 for testing. We use 5,000 trajectories for training, 1,000 held-out baseline trajectories for evaluation, and 2,000 test contexts per run under the shifted regime. Unless otherwise specified, the training data

Table 2: Summary of the synthetic shift settings.

Example	Shift	Null	Boundary-transition	Alternative
Spring	Mean	$\mu_0 = (0,0)$, $v = (0,1)$	$\mu_0 = (0,0)$, $v = (1,0)$	$\mu_0 = (-1,0)$, $v = (0,1)$
	Covariance	$\Sigma_\delta = \begin{pmatrix} 1 & 0 \\ 0 & 1+\delta \end{pmatrix}$	$\Sigma_\delta = \begin{pmatrix} 1+\delta & 0 \\ 0 & 1 \end{pmatrix}$	N/A
Pendulum	Marginal	$(\alpha_1, \beta_1) = \left(1 + \frac{10}{3}\delta,\ 2 + \frac{20}{3}\delta\right)$ $(\alpha_2, \beta_2) = (1,4)$	$(\alpha_1, \beta_1) = (1,2)$ $\alpha_2 = 1 + 1.5\delta,\ \beta_2 = 4 - 1.5\delta$	$(\alpha_1, \beta_1) = (1.5, 1.5)$ $\alpha_2 = 1 + 1.5\delta,\ \beta_2 = 4 - 1.5\delta$
	Dependence	$\rho = -\delta$	$\rho = \delta$	N/A

are constructed to contain approximately 90% stable and 10% unstable trajectories. The stability threshold is set to $\tau = 1$ for the spring system and to $\tau = 35^\circ$ (equivalently, $\tau \approx 0.61$ radians) for the pendulum. Throughout the synthetic experiments, we use a one-dimensional latent representation, *i.e.*, $d = 1$. All rejection rates in Figures 4a and 4b and in Table 1 are computed over 100 independent runs.

Models are trained for 100 epochs using Adam with learning rate 3×10^{-3}, regularization weight $\lambda = 0.01$, and stability-aware weighting parameter $\omega = 2$. For the binary classifier, we use a multi-layer network trained with the same Adam configuration and logistic binary cross-entropy loss. All main neural-network experiments are run on a workstation with an NVIDIA GeForce RTX 5070 Ti GPU with 16GB memory. Lightweight preprocessing, plotting, and small-scale sanity checks are also performed on a Mac with an Apple M4 chip.

Sensitivity Analysis and Ablation Studies. We study how the weights of the three loss terms shape the learned representation. Composite-null calibration controls Type-I error when the class-conditional latent laws are invariant (Assumption 1 and Proposition 2), so the loss weights matter through the representation itself: whether the latent stays invariant under benign shifts and how strongly it responds to risk-increasing ones. We measure both with the uncalibrated two-sample test between the baseline and shifted latent samples. Its rejection rate under a null shift, where the violation rate is unchanged, measures how much the latent moves under a shift that should leave it invariant; its Type-II error under a boundary-transition shift measures how weakly the latent responds to a risk-increasing shift. Results are averaged over multiple random seeds.

We first study the contribution of each term in the training objective. Recall that the stability-aware reconstruction loss can be decomposed into a standard reconstruction term and an asymmetric boundary penalty. This allows us to isolate the effect of the asymmetric stability-aware component and write the total loss as

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{stb}}\mathcal{L}_{\text{stb}}, \tag{25}$$

where $\lambda_{\text{rec}} + \lambda_{\text{reg}} + \lambda_{\text{stb}} = 1$. Here, $\mathcal{L}_{\text{rec}}$ is the standard unweighted mean squared reconstruction loss, $\mathcal{L}_{\text{reg}}$ is the uniformity regularization term, and $\mathcal{L}_{\text{stb}}$ isolates the asymmetric penalty applied to underestimated unstable trajectories:

$$\mathcal{L}_{\text{stb}} \coloneqq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{K_i}\sum_{k=1}^{K_i}\big(\omega_{\text{stb}}(s_{i,k}, \hat{s}_{i,k}) - 1\big)\|s_{i,k} - \hat{s}_{i,k}\|^2. \tag{26}$$

Because $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{stb}}$ have different numerical scales from the reconstruction loss, sweeping the full simplex is not informative in practice: assigning scale-one weights to the auxiliary terms often causes poor testing performance. We therefore examine a local reconstruction-dominant simplex with vertices

$$(\lambda_{\text{rec}}, \lambda_{\text{reg}}, \lambda_{\text{stb}}) \in \{(1,0,0),\ (0.9, 0.1, 0),\ (0.9, 0, 0.1)\}.$$

This region varies λ_{reg} and λ_{stb} within a practically meaningful range while maintaining a dominant weight on trajectory reconstruction.

Figure 5 shows the results at significance level $\beta = 0.1$. Without a shift (left panel), the rejection rate stays between 0 and 0.15 for every weight setting, close to the nominal level. Under the null shift (middle panel), settings with a larger uniform-regularization weight yield a latent that moves with the

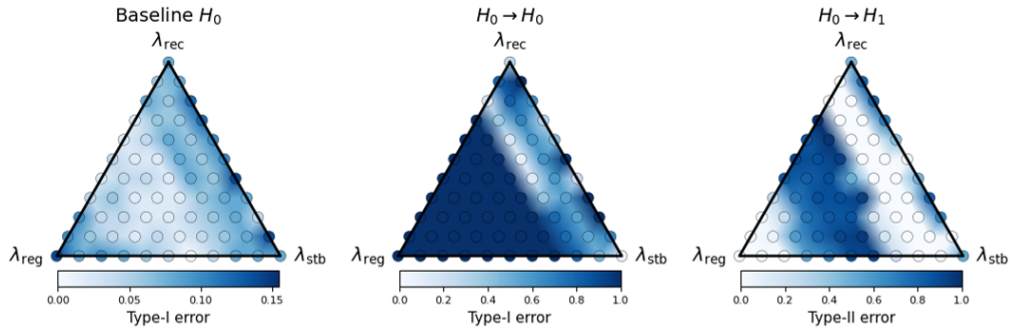


Figure 5: Sensitivity of the learned representation to the loss weights over a reconstruction-dominant simplex parameterized by $(\lambda_{\text{rec}}, \lambda_{\text{reg}}, \lambda_{\text{stb}})$, evaluated with the uncalibrated two-sample test between baseline and shifted latent samples. The three panels correspond to the baseline without shift, a null shift, and a boundary-transition shift. Under the null shift a lower rejection rate indicates a latent that stays invariant to the benign shift; under the boundary-transition shift a lower Type-II error indicates a latent that responds to the risk-increasing shift.

benign shift and is rejected almost always, whereas a band of settings that combine reconstruction with a small amount of uniform regularization and the stability-aware term leaves the latent nearly unchanged. The same band keeps a low Type-II error under the boundary-transition shift (right panel), while reconstruction alone or heavier uniform regularization does worse on at least one of the two criteria. This supports the design of `L2T`: reconstruction retains trajectory information, the stability-aware term keeps the latent responsive to risk-increasing shifts, and a small amount of uniform regularization spreads the latent without breaking its invariance under benign shifts.

To visualize the learned latent geometry, Figure 6 plots the empirical latent density relative to the uniform reference as the shift grows. Under the null shift, the `L2T` density stays close to uniform across all shift magnitudes, whereas the uniform-only latent tilts away from it, which explains the false alarms of the uniform-only variant. Under the boundary-transition and alternative shifts, both latents depart increasingly from uniformity as δ grows.

Additional Baselines: Oracle Simulation and Trajectory Surrogate. We further consider two baselines that label deployment trajectories and apply a one-sided binomial test of $H_0 : \mathbb{P}_{P_1}(y(X) = 1) \leq \alpha$ to the labels: *oracle simulation*, which simulates the trajectories of k deployment contexts, and *trajectory surrogate + label*, which predicts them with a network trained on the baseline trajectories. Figure 7 reports both on the mass–spring–damper system, in the settings of Figure 4a. Oracle simulation is valid for every simulation size k, and its power near the boundary grows with k, so the simulation budget matters for power. The trajectory surrogate is also valid but has low power in the boundary-transition setting, because it under-predicts violations. `L2T` needs no deployment simulation; near the boundary it rejects somewhat earlier than full simulation.

Empirical Check of Assumption 1. Assumption 1 cannot be checked at deployment, where trajectories are unobserved, but it can be checked in simulation, where the true labels are available. Figure 8 measures how far the class-conditional latent laws of the trained `L2T` encoder drift from their baseline values R_0 and R_1 along the two null shifts of Figure 4a, in units of the null radius r_α, the scale that matters in Proposition 2. Although the contexts within each class move, their latent laws barely change: the drift stays below $0.8\, r_\alpha$ on average, close to the sampling floor, so Proposition 2 covers these shifts. The assumption does not hold for every covariate shift, however: along the risk-increasing mean shift of X_1 (third column of Figure 4a), the drift of the violation class reaches 9–20 r_α, and the early rejections of `L2T` in that panel reflect this drift rather than a change in the violation rate.

E.2 Transient Stability Assessment on IEEE 39-Bus System

We evaluate our framework on a transient stability assessment (TSA) benchmark generated from the IEEE New England 39-bus power system (Sokolović, 2024). The dataset contains 12,852

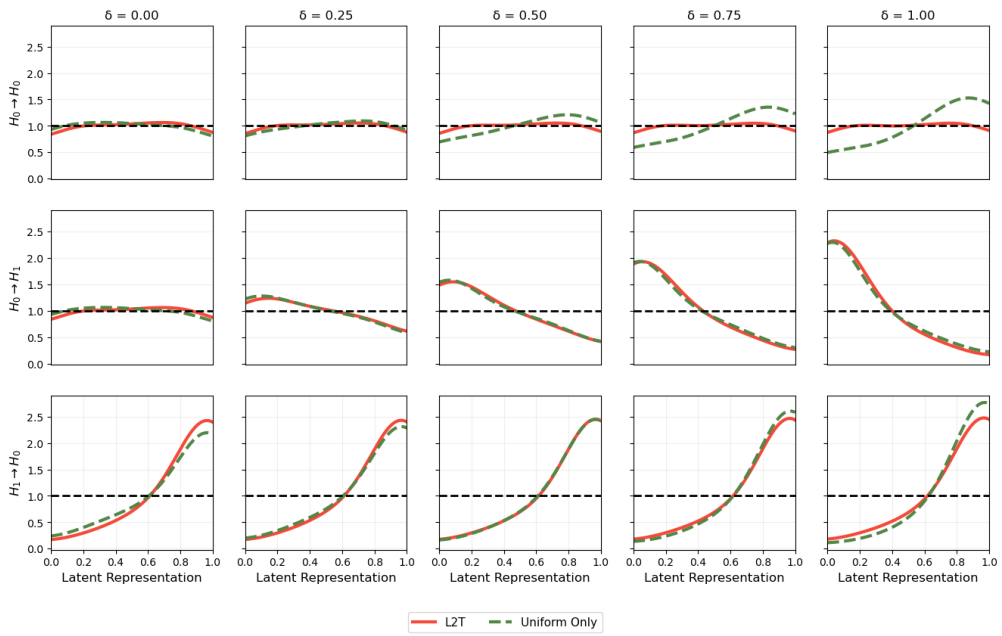


Figure 6: Latent-space under increasing shift magnitude δ. Columns correspond to values of δ, and rows show the latent density relative to the uniform reference under the null, boundary-transition, and alternative shifts. Under the null shift, the `L2T` latent stays near uniform across shift levels, whereas the uniform-only latent tilts away from it, which leads to false alarms. Under the boundary-transition and alternative shifts, both latents deviate increasingly from uniformity as δ grows.

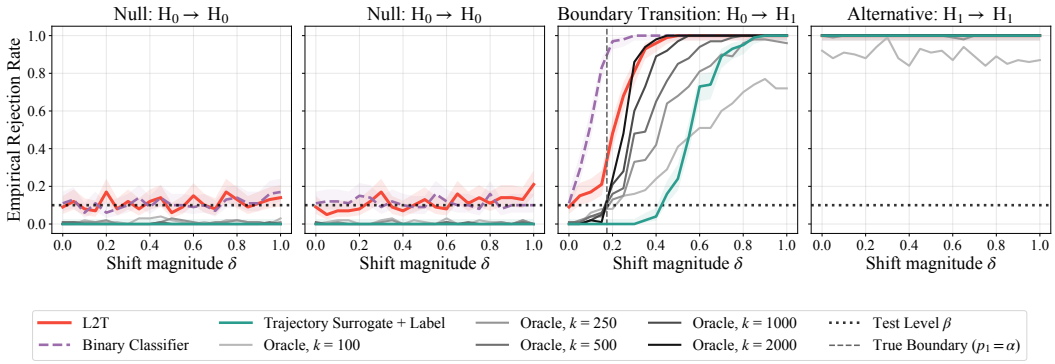


Figure 7: Additional baselines on the mass–spring–damper system; the panels are columns 1, 2, 4 and 5 of Figure 4a. *Oracle simulation* simulates the first k of the 2,000 deployment contexts and applies an exact binomial test at α; *trajectory surrogate + label* applies the same test to the labels of trajectories predicted by a 2×32 MLP. The dashed vertical line marks where the true violation rate crosses α, and the dotted line is the test level 0.1.

time-domain simulation trajectories with 50 features, produced using DIgSILENT PowerFactory (DIgSILENT GmbH, n.d.) with Python-based automation. The benchmark complements the simulation experiments in two ways: the context is 50-dimensional, and the evaluation varies the class proportions of the test set, a label shift under which Assumption 1 holds by construction.

To cover diverse operating conditions, generation and load levels are varied from 80% to 120% in 5% increments. For each operating point, three-phase short-circuit faults are applied at seven locations (0%, 10%, 20%, 50%, 80%, 90%, 100%) along transmission lines, with clearing times ranging from 0.1s to 0.3s. Each simulation runs for 10s, and post-fault trajectories are sampled at 0.01s resolution from the fault clearance time to 0.6s afterward.

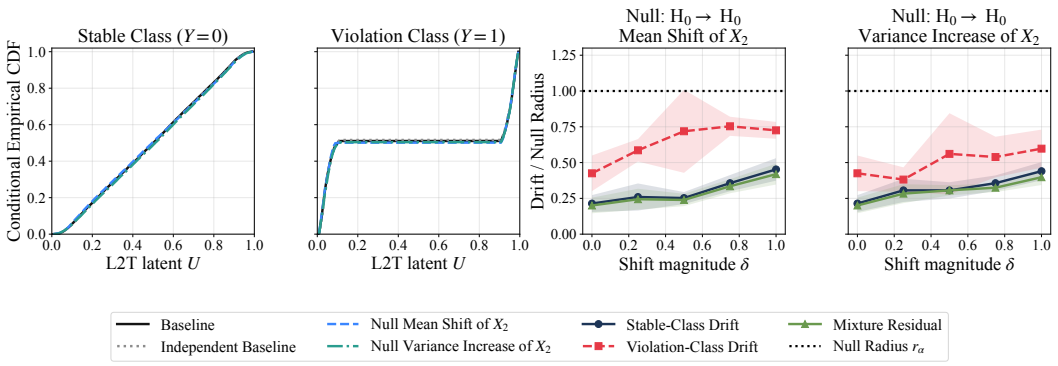


Figure 8: Empirical check of Assumption 1 for the trained `L2T` encoder on the mass–spring–damper system. Left two panels: class-conditional empirical CDFs of the latent for the baseline, an independent baseline sample, and the two null shifts of Figure 4a at $\delta = 1$. Right two panels: class-conditional drifts and mixture residual (MMD), divided by the null radius $r_\alpha = (\alpha/2)D(R_0, R_1)$, along the mean shift and the variance increase of X_2 (columns 1 and 2 of Figure 4a), where the violation rate stays at 8.0–8.1% and 7.6–8.0%; $\delta = 0$ is the independent-baseline sampling floor.

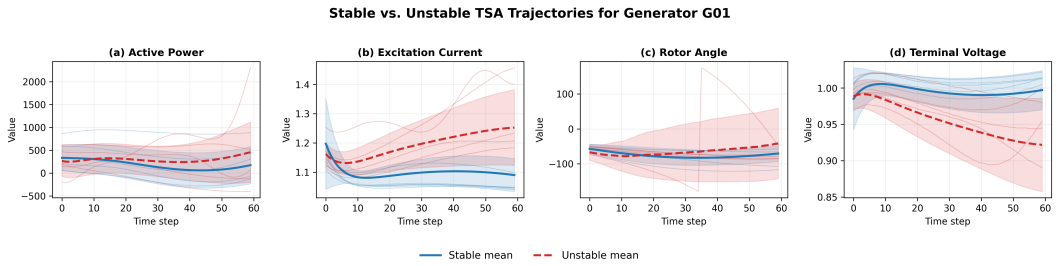


Figure 9: Trajectory visualization for generator G01 across four system variables: (a) active power, (b) excitation current, (c) rotor angle, and (d) terminal voltage. Blue curves denote stable trajectories and red curves denote unstable ones. Thin lines correspond to individual sampled trajectories, while solid lines represent the mean trajectories and shaded regions indicate one standard deviation. Stable trajectories remain tightly concentrated and exhibit smooth, consistent dynamics, whereas unstable trajectories show larger dispersion and systematic deviations, particularly in rotor angle and terminal voltage. These differences highlight the distributional characteristics that enable detection of instability.

Table 3: TSA results on Type-I and Type-II errors.

Method	Type-I @ 0.00	Type-II @ 0.20
L2T (flow, $d_v = 8$)	0.100 ± 0.035	0.005 ± 0.008
L2T-MMD ($d_v = 1$)	0.145 ± 0.041	0.435 ± 0.058
Reconstruction Only	0.110 ± 0.036	0.000 ± 0.000
Regularization Only	0.115 ± 0.037	0.710 ± 0.053
Context Test	0.010 ± 0.011	0.175 ± 0.044
Binary Classifier	1.000 ± 0.000	0.000 ± 0.000

Each trajectory records generator-level signals for 10 generators, including active power, terminal voltage, excitation current, rotor speed, and rotor angle (relative to G02, the reference machine). The dataset encodes stable trajectories as 1 and unstable trajectories as 0; this raw-data convention is the complement of the violation indicator y in Definition 1. Approximately 42% of the cases are unstable.

Data Split. We adopt a stable-versus-unstable testing protocol. Among 12,852 trajectories, 7,460 are stable and 5,392 are unstable. For training, we sample 5,000 trajectories consisting of 95% stable

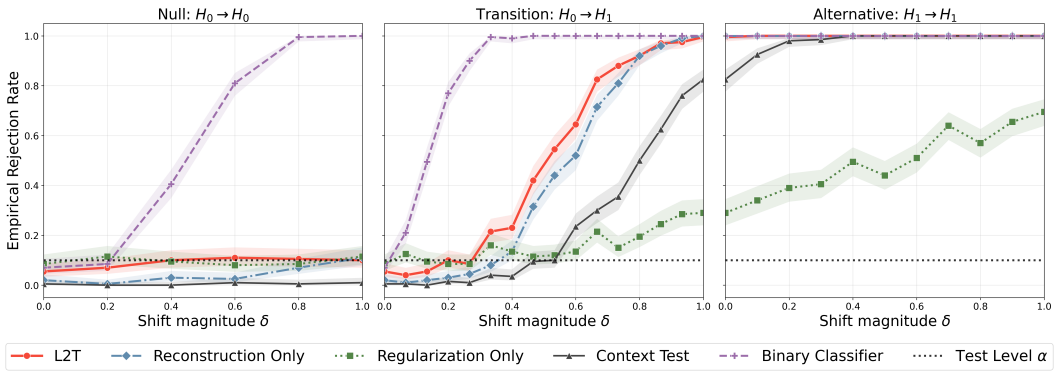


Figure 10: TSA rejection curves under three distribution-shift regimes. The null setting is defined by the reference mixture with unstable ratio 0.05. Left: null shift from H_0 to H_0, where the unstable ratio decreases from 0.05 to 0.00. Middle: transition shift from H_0 to H_1, where the unstable ratio increases from 0.05 to 0.20. Right: alternative-regime shift from H_1 to H_1, where the unstable ratio varies from 0.20 to 0.30. The horizontal axis shows the normalized shift magnitude $\delta \in [0, 1]$ for each regime. The empirical rejection rate of L2T remains close to the nominal test level under the null shift, increases smoothly through the transition regime, and is close to one in the alternative regime, indicating a favorable balance between Type-I error control and detection power.

and 5% unstable samples, resulting in $4{,}750$ stable and 250 unstable trajectories. The remaining stable and unstable samples are reserved for evaluation.

At test time, the nominal reference distribution is defined as a $95{:}5$ stable-to-unstable mixture, which serves as a representative point within the null family that includes nearby mixtures such as the fully stable setting. We then evaluate controlled shifts by varying the unstable component of the test set, including a downward null-preserving shift, a transition shift from the nominal mixture toward a higher-unstable mixture, and an alternative-regime sweep.

All features are normalized using statistics computed from the training set only, across both sample and time dimensions. The same normalization is applied to all evaluation data to avoid leakage.

Trajectory Characteristics. To provide intuition on the TSA dataset, we visualize a subset of trajectories from generator G01 across four representative variables: active power, excitation current, rotor angle, and terminal voltage. For each variable, we randomly sample 5 stable and 5 unstable trajectories. As shown in Figure 9, stable trajectories exhibit relatively consistent patterns in both magnitude and temporal dynamics, forming a concentrated and structured region in trajectory space. In contrast, unstable trajectories display significantly higher variability, with diverse shapes, amplitudes, and temporal behaviors.

Context Construction. Each trajectory has length 60 with 50 features per time step. In our setting, the exogenous context is defined as the first observation of each trajectory:

$$X_i = s_{i,1} \in \mathbb{R}^{50}.$$

This corresponds to using the initial system state as the context variable. Since trajectories in this dataset begin immediately after fault clearance, this observation represents the post-fault system state at the onset of transient dynamics. The remaining trajectory is used to construct a single training pair per sample. Specifically, each trajectory is evenly split into two halves:

$$o_i^{\text{hist}} = (s_{i,1}, \ldots, s_{i,T/2}), \quad o_i^{\text{next}} = (s_{i,T/2+1}, \ldots, s_{i,T}),$$

where $T = 60$. This yields one $(o^{\text{hist}}, o^{\text{next}})$ pair per trajectory.

Model Training. We train a conditional GRU model to predict future dynamics conditioned on the context and past trajectory segment. For the main L2T-flow model, we use a latent dimension $d_v = 8$ and add a normalizing-flow regularization term to the reconstruction-based training objective. We also report an L2T-MMD variant with $d_v = 1$, which replaces the flow regularizer with the MMD-based regularization used in the original baseline. All models are trained on GPU (CUDA).

Evaluation Protocol. At test time, each trajectory is mapped to a latent representation using only its context X_i. The reference set contains 950 stable trajectories and 50 unstable trajectories, corresponding to the nominal unstable ratio 0.05. For each test ratio r, we keep the 950 reference-stable trajectories fixed and sample

$$n_{\text{unstable}}(r) = \text{round}\left(\frac{r}{1-r} \cdot 950\right)$$

unstable trajectories from the held-out unstable pool. The resulting test batch therefore has unstable ratio approximately r, with the stable component fixed and the unstable component resampled across repeats.

For MMD-based methods, the rejection threshold is calibrated as the empirical $(1-\beta)$-quantile of the available null calibration statistics with $\beta = 0.1$; this is the finite-calibration approximation of the composite critical value over the represented null settings. For the context baseline, we use the projected KS two-sample test on raw context vectors. For the Binary Classifier baseline, we compare the predicted-probability vectors of the reference and test batches using Welch's two-sample t-test. Each setting is repeated 200 times, and we report the empirical rejection rate.

Additional TSA Results. Table 3 reports the Type-I error under the fully stable setting ($r = 0.00$) and the Type-II error at the 80:20 stable-to-unstable mixture ($r = 0.20$). L2T-flow maintains Type-I error close to the target level while achieving near-zero Type-II error. Compared with L2T-MMD, the flow-based variant better preserves detection power on this high-dimensional TSA benchmark: L2T-MMD has noticeably higher Type-II error at $r = 0.20$, while L2T-flow remains both calibrated and powerful. Some baselines achieve low Type-II error but suffer from over-rejection under the fully stable null, which is further reflected in the shift curves in Figure 10. On this high-dimensional benchmark, where Assumption 1 holds, `L2T` with a sufficiently expressive latent is therefore both calibrated and powerful.