Mining Point-of-No-Return Boundaries in Constrained Dynamical Systems via Counterfactual Auditing

Jia Liu
South China University of Technology
Guangzhou, China
Pazhou Laboratory
Guangzhou, China
Huazhong University of Science and Technology
Wuhan, China
jialiu0330@hust.edu.cn

Jiaxin Luo
South China University of Technology
Guangzhou, China
csljx1127@mail.scut.edu.cn

Lejun Ai
South China University of Technology
Guangzhou, China
cs_ailejun@mail.scut.edu.cn

Yue Wang
South China University of Technology
Guangzhou, China
csyuewang@mail.scut.edu.cn

Enpeng Lan
South China University of Technology
Guangzhou, China
202521043827@mail.scut.edu.cn

Yulong Li
South China University of Technology
Guangzhou, China
csyulongli@mail.scut.edu.cn

Zongyu Li
South China University of Technology
Guangzhou, China
zongyuli@scut.edu.cn

Fuxin Zhang*
South China University of Technology
Guangzhou, China
fuxinzhang@scut.edu.cn

Siyuan Liu*
South China University of Technology
Guangzhou, China
liu@scut.edu.cn

Long Hu
Huazhong University of Science and Technology
Wuhan, China
hulong@hust.edu.cn

Yixue Hao
Huazhong University of Science and Technology
Wuhan, China
yixuehao@hust.edu.cn

Min Chen*
South China University of Technology
Guangzhou, China
Pazhou Laboratory
Guangzhou, China
minchen@ieee.org

Abstract

Safety-critical failures in constrained dynamical environments are often detected only after a violation occurs, while outcome metrics (e.g., success rate, time-to-failure) conflate structural inevitability with decision-induced errors. We formulate failure diagnosis as recoverability boundary discovery and define the Point-of-No-Return (PoNR) as the last recoverable snapshot before violation becomes unavoidable under an audited intervention class. We introduce Counterfactual PoNR Auditing (CPA), a snapshot-replay framework that produces matched counterfactual rollouts and is accelerated by a neural operator surrogate (UNO) for a 3.09× end-to-end speedup. CPA yields Evidence Objects (EOs) that support structural–policy decomposition and enable Failure Atlas mining via contrastive analysis of event sequences within the policy-induced gap. CPA also serves as an offline supervision generator, allowing lightweight predictors to learn early inevitability signals beyond static observational baselines. Experiments on a high-fidelity hydrodynamic control testbed and a canonical bistable system show that PoNR precedes observable violation, remains stable under up to 30% multiplicative surrogate noise, and reduces auditing cost through surrogate-guided search.

CCS Concepts

• **Information systems** → **Data mining**; • **Computing methodologies** → *Artificial intelligence*; Causal reasoning and diagnostics; Modeling and simulation.

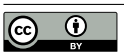
Keywords

Constrained Dynamical Systems, Point-of-No-Return (PoNR), Counterfactual Auditing, Recoverability Boundaries, Evidence Objects, Contrastive Sequential Pattern Mining, Failure Motifs

ACM Reference Format:

Jia Liu, Jiaxin Luo, Lejun Ai, Yue Wang, Enpeng Lan, Yulong Li, Zongyu Li, Fuxin Zhang, Siyuan Liu, Long Hu, Yixue Hao, and Min Chen. 2026. Mining Point-of-No-Return Boundaries in Constrained Dynamical Systems via Counterfactual Auditing. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3818127

*Corresponding authors: Fuxin Zhang (fuxinzhang@scut.edu.cn), Siyuan Liu (liu@scut.edu.cn), and Min Chen (minchen@ieee.org).



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3818127

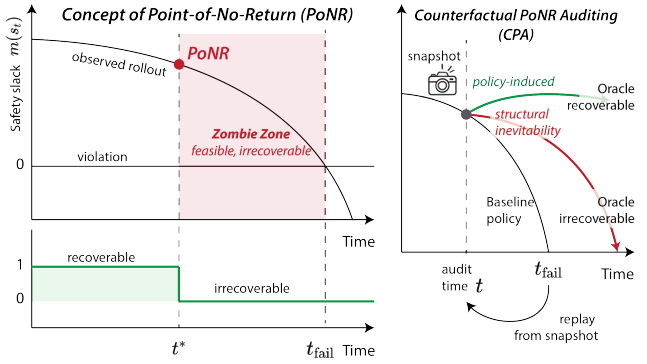


Figure 1: Point-of-No-Return (PoNR) and counterfactual auditing. Left: PoNR t^* is the last recoverable time; the interval $[t^*, t_{\text{fail}}]$ is feasible but irrecoverable (Zombie Zone). Right: CPA replays a snapshot at time t and branches rollouts to separate policy-induced failure (oracle recoverable) from structural inevitability (oracle irrecoverable).

1 Introduction

Autonomous decision-making systems are increasingly deployed in high-stakes constrained dynamical environments [15, 17, 23]. An episode may remain feasible in observation, yet already be irrecoverable under stochastic disturbances, creating a *Zombie Zone*: a feasible-but-irrecoverable window where sensors report no unsafe indications while the system has crossed a recoverability boundary (Fig. 1).

Outcome metrics such as success rate and time-to-failure are typically lagging indicators that conflate two distinct sources of failure. The first is structural irrecoverability induced by environment dynamics and disturbances [6, 13]. The second is policy-induced irrecoverability, where recovery was structurally feasible but became unattainable due to the decision-maker's choices [5, 9]. This distinction is critical in agentic evaluations, where outcomes are often highly sensitive to randomness and implementation details [10]. Unlike per-timestep failure prediction, which treats each step as an i.i.d. classification instance [12], diagnosing irrecoverability is a boundary discovery problem over sequential trajectories. This perspective aligns naturally with rare-event analysis and contrastive pattern mining [7].

To make irrecoverability explicit, we propose Point-of-No-Return (PoNR) boundary mining. PoNR identifies the last snapshot where an episode remains recoverable under an audited intervention class; beyond this point, eventual constraint violation becomes unavoidable. Notably, this boundary can precede observable failure: constraints may still be satisfied, yet the remaining slack is insufficient to enable recovery. As an offline supervision label produced by counterfactual auditing, PoNR can be distilled into learnable early-warning monitors. We operationalize this via Counterfactual PoNR Auditing (CPA), which leverages replayable simulator snapshots to branch controlled counterfactual rollouts. CPA yields Evidence Objects (EOs): snapshot-aligned counterfactual pairs that support structural–policy attribution and reduce sensitivity to environmental randomness through paired replay.

Building on EOs, we introduce Failure Atlas Mining to extract reusable failure motifs. Inspired by counterfactual contrastive analysis [4, 16], we analyze event sequences within the policy-induced gap and identify an Oscillation Trap motif via snapshot-matched replay. To address the computational bottleneck of counterfactual auditing, we integrate a U-Net Operator (UNO) surrogate, reducing the marginal cost of counterfactual rollouts by orders of magnitude (8ms vs. 72min) with negligible accuracy loss (NSE > 0.99; Appendix D). We evaluate the framework on a high-fidelity hydrodynamic control testbed and a canonical bistable dynamical system, and demonstrate stability under up to 30% surrogate noise (Appendix D).

Our contributions are as follows.

- We formalize PoNR as a recoverability boundary and introduce CPA to produce snapshot-aligned EOs, enabling counterfactual attribution of structural inevitability versus decision-induced errors.
- We use CPA as an offline label-generation engine for downstream tasks, including **training** lightweight early-warning monitors (AUC > 0.95) and contrastively **mining** interpretable failure motifs (Failure Atlas).
- We validate CPA on a canonical bistable dynamical system, showing that the Zombie Zone can arise generically in constrained dynamics beyond a specific control testbed.

2 Related Work

2.1 From outcome metrics to boundary auditing

Evaluation in high-stakes constrained dynamical environments is commonly outcome-centric, reporting success rate, constraint violations, and cost, often combined with runtime guardrails that treat failure as a terminal event [1, 22, 25]. In parallel, safe reinforcement learning and constrained control prioritize prevention, either by optimizing CMDP-style objectives or enforcing invariance-based constraints during training and execution [8, 11]. A complementary line of work focuses on prediction: early-warning models and survival-style analyses estimate statistical risk to forecast if and when failure will occur [19]. Our work targets a diagnostic objective that is orthogonal to both prevention and prediction: we audit a deployed decision policy—which may be a classical controller, an RL policy, or an agentic system—to localize the time at which recoverability is lost under feasible interventions, i.e., the point-of-no-return [2, 24]. This perspective reframes failure analysis from outcome summaries or risk scoring to discovering a dynamical recoverability boundary embedded in closed-loop interactions.

2.2 Counterfactual alignment for sequential mining and scalable auditing

Counterfactual explanation and auditing methods are most often developed for static prediction settings, where counterfactuals modify inputs to change a model output [3, 14, 18]. In contrast, CPA operates in sequential closed-loop systems by branching rollouts from replayable snapshots. This snapshot-replay protocol produces aligned Evidence Objects that support matched comparisons and help separate decision-induced vulnerability from structural inevitability.

Building on these aligned traces, our Failure Atlas performs contrastive sequential pattern mining, differing from observational log mining where correlations can confound attribution [20, 21]. To mitigate the cost of boundary auditing in expensive simulators, we leverage surrogate guidance to prioritize boundary-relevant queries and report the resulting efficiency behavior in the experiments.

2.3 Relation to reachability, safety verification, and failure prediction.

Reachability analysis and safety verification provide ex-ante, system-level guarantees by characterizing recoverable regions in state space [6, 13], while failure prediction estimates statistical risk to forecast if and when a violation is likely [19]. In contrast, CPA performs post-hoc, episode-level auditing to pinpoint when recoverability is already lost under an audited intervention class—a retroactive attribution target that prior methods do not address.

3 Problem Formulation

We study counterfactual auditing of safety-critical risk episodes in constrained dynamical environments. Our objective is not to optimize control performance, but to identify the recoverability boundary—the latest time at which an intervention can still prevent future constraint violation.

3.1 Episodes and replayable snapshots

We consider a risk episode recorded as a discrete-time snapshot sequence $\{x_t\}_{t=0}^{T}$, where each snapshot x_t is replayable under a fixed simulator and disturbance model. Each snapshot contains the information required to restart simulation from time t, enabling counterfactual rollouts from the same starting point. Let s_t denote the underlying physical state contained in x_t.

Safety is defined by a slack function $m : \mathcal{S} \to \mathbb{R}$:

$$\begin{aligned} m(s_t) \geq 0 &\ \text{ indicates feasibility,} \\ m(s_t) < 0 &\ \text{ indicates violation.} \end{aligned} \tag{1}$$

In our hydrodynamic testbed, $m(s_t)$ is instantiated as the safety margin to the dyke/limit water level (i.e., the remaining clearance before overtopping).

We define the observed failure time for an episode as

$$t_{\text{fail}} \triangleq \min\{t \in \{0, \dots, T\} \mid m(s_t) < 0\}, \tag{2}$$

if a violation occurs within the observation window.

3.2 Recoverability and point of no return

We define the intervention space as a functional space $\mathcal{U}$. In our hydrodynamic setting, the action at time t is the normalized drainage intensity $u_t \in [0, D_{\max}]$. CPA evaluates interventions as do-operators applied at a snapshot x_t. Let ι denote a fixed audited recovery capability.

Specifically, we adopt the maximal bang-bang control $\iota_{\text{oracle}} = \{u(t) \equiv D_{\max}\}$ as the reference protocol. In this dissipative system, the water level is monotonically non-increasing with respect to drainage intensity; thus, ι_{oracle} provably yields the optimal recovery trajectory without requiring complex MPC search.

We verified that richer oracle families (delayed, ramped, capped, front-loaded, pulsed) do not outperform constant max-drain on 22 Track-A episodes (0 Win / 13 Tie / 9 Loss; mean shift −0.5h; worst −2h), confirming max-drain as a tight upper-bound.

Given a horizon H and tolerance $\eta \in [0, 1)$, we define future-only (η, H)-recoverability:

$$\begin{aligned} y(x_t; \iota) = \mathbb{I}\Big[\mathbb{P}\big(\forall k \in [t+1, t+H], \\ m(s_k) \geq 0 \mid x_t, do(\iota)\big) \geq 1 - \eta\Big], \end{aligned} \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

While formulated probabilistically to accommodate surrogate uncertainty, for deterministic simulator auditing (as in our primary experiments), the probability collapses to a binary check (i.e., we set $\eta = 0$).

Along a risk episode, $y(x_t; \iota)$ typically transitions from 1 (still savable) to 0 (already lost). We define the Point-of-No-Return as the latest savable snapshot time:

$$t^*(\iota) \triangleq \max\{t \in \{0, \dots, T\} \mid y(x_t; \iota) = 1\}. \tag{4}$$

In all experiments, the reported t_{PoNR} is the estimate $\hat{t}^*(\iota)$ returned by CPA (Sec. 4.2).

The Actionable Lead Time is the temporal gap between PoNR and the observed violation:

$$\Delta \triangleq \max(0,\ t_{\text{fail}} - \hat{t}^*(\iota)). \tag{5}$$

When $\Delta > 0$, the interval $[\hat{t}^*(\iota),\ t_{\text{fail}}]$ is the *Zombie Zone*, a phase where the system remains feasible yet has already lost recoverability under ι.

3.3 Structural and policy decomposition

To separate system limits from decision quality, we compare PoNR under two audited capability settings: a deployed setting ι_{obs} and an upper-bound setting ι_{oracle} that represents enhanced recovery while keeping the same environment dynamics fixed. Let $\hat{t}^*_{\text{obs}}$ and $\hat{t}^*_{\text{oracle}}$ denote the CPA estimates under these two settings.

We define the policy-induced gap as

$$\Delta_{\text{gap}} \triangleq \hat{t}^*_{\text{oracle}} - \hat{t}^*_{\text{obs}} \ \geq\ 0. \tag{6}$$

Intuitively, Δ_{gap} measures how much earlier the episode becomes irrecoverable under the deployed setting compared to the upper bound, thus localizing the segment where improved decision-making could have extended viability.

4 Methodology

Having formalized PoNR and its decomposition, we now introduce **Counterfactual PoNR Auditing (CPA)**, a framework to estimate these boundaries and mine failure patterns. CPA operates on *snapshot-based time-travel*: it forks simulation from a recorded snapshot x_t to explore counterfactual futures under controlled interventions. Throughout, we assume replayable snapshots $x_t = (s_t, h_t, b_t, r_t)$ as defined in Section 3.

4.1 Evidence Objects: The Mining Unit

To enable reproducible auditing and downstream mining, we define an **Evidence Object (EO)** as a compact, replayable record of a snapshot-based counterfactual test:

$$E \triangleq \langle x_{\text{start}}, \iota, \tau', \mathbf{m}', \mathbf{u}', y, t_{\text{fail}}, c, \text{meta} \rangle, \tag{7}$$

where x_{start} is the branching snapshot (at step t_{start}), and $\iota \in \mathcal{I}$ specifies the intervention (e.g., switching to an oracle recovery policy). The counterfactual rollout τ' records the resulting trajectory; recoverability is evaluated on an H-step prefix, with slack sequence $\mathbf{m}' = \{m(s_k)\}_{k=t_{\text{start}}}^{t_{\text{start}}+H}$. $\mathbf{u}'$ denotes the tool-call/event stream, $y \in \{0,1\}$ is the (ir)recoverability outcome, t_{fail} is the first violation time if it occurs, and c summarizes auditing cost. Finally, meta stores reproducibility identifiers (domain, simulator version, policy ID, seed).

Unlike raw logs, EOs are *snapshot-aligned* across interventions (sharing the same x_{start}) and therefore comparable instances for matched counterfactual analysis. We aggregate EOs into a dataset $\mathcal{D}$, which serves as the input to both causal auditing and Failure Atlas mining.

4.2 Algorithm: Counterfactual PoNR Auditing (CPA)

CPA estimates the *Point-of-No-Return* as a **transition time** on the snapshot timeline of a risk episode. Concretely, for each auditable episode (Track A; Sec. 5.1), we consider the discrete snapshot sequence $\{x_t\}_{t=0}^{T}$ (1-hour resolution in our setting) and a fixed audited recovery capability ι (e.g., max-drain oracle). CPA searches for the latest snapshot time t^* such that starting the intervention at x_{t^*} still yields a feasible future for a horizon H. The resulting estimate $\hat{t}^*$ is reported as t_{PoNR} in all experimental figures and tables (e.g., Fig. 3, Fig. 6).

Recoverability as a binary transition. The future-only (η, H)-recoverability indicator $y(x_t; \iota)$ is defined as in Equation (3). In inertia-dominated systems, $y(x_t; \iota)$ typically exhibits a one-way transition from 1 (savable) to 0 (lost) along the episode timeline. CPA therefore estimates PoNR by locating the **boundary index** of this transition.

Track taxonomy and censoring (aligned with experiments). We integrate the experimental episode taxonomy directly into the audit protocol:

- **Track B (Instant Fail):** episodes with observed failure within two snapshots ($t_{\text{fail}} < 2$ hours) are excluded from boundary search and reported separately (Sec. 5.1).
- **Left-censored:** if $y(x_{t_0}; \iota) = 0$ at the earliest auditable time t_0 (episode start), the episode is unsavable under the audited capability class; we record $\hat{t}^*$ as left-censored.
- **Right-censored:** if $y(x_{t_{\max}}; \iota) = 1$ at the latest snapshot in the observation window, PoNR lies beyond the window; we record $\hat{t}^*$ as right-censored.
- **Observed:** otherwise, PoNR is identifiable within-window and we return a finite $\hat{t}^*$ used in lead-time statistics.

Transition localization over snapshots. We localize the recoverable-to-irrecoverable transition by bisection over snapshot indices. A lightweight surrogate may propose a near-boundary bracket $[t_a, t_b]$, but the final decision is always verified by high-fidelity counterfactual rollouts from the candidate snapshot. Each oracle query produces an Evidence Object (Sec. 4.1), so the audit outputs both a PoNR estimate and a replayable evidence dataset.

In our audited setting, recoverability predominantly exhibits a one-way 1⟶0 transition (97.5% strict monotone over 240 pairs; 98.3% bisection–full-scan match (Table 1). We emphasize that monotonicity is a *protocol-dependent empirical condition*, not a universal axiom of CPA.

Table 1: Validation of monotonicity and bisection correctness (240 pairs).

Metric	Result
Strict 1⟶0 monotone transitions	234/240 (97.5%)
Bisection exactly matches full scan	236/240 (98.3%)

Output semantics (hard alignment to experiments). For each non-censored Track A episode, CPA outputs $\hat{t}^*$, which is the t_{PoNR} used in Fig. 3 and to compute lead time Δ. Episodes are categorized by Δ into *Structural* (< 0.5h), *Policy Gap* (intermediate), and *Manageable* (> 2.5h), matching the composition plot in Fig. 5. Censoring flags determine inclusion in lead-time statistics and regime composition.

4.3 Causal Estimands

CPA produces snapshot-aligned Evidence Objects (Sec. 4.1) by branching multiple do-operator interventions from the same replayable snapshot x_t. This alignment enables paired causal comparisons under identical starting conditions, so that differences are attributable to interventions rather than episode selection.

PoNR boundary and recoverability. For an intervention $\iota \in \mathcal{I}$, CPA returns (i) a PoNR estimate $\hat{t}^*(\iota)$, defined as the last recoverable snapshot index under the (η, H) test used by Algorithm 1, and (ii) a recoverability outcome $y(\iota) \in \{0,1\}$ evaluated on the same auditable snapshot set. In the experiments, $\hat{t}^*(\iota)$ corresponds to the PoNR time reported in phase-diagram figures (e.g., t_{PoNR}), while $y(\iota)$ corresponds to whether the episode is recoverable under ι given the auditing horizon.

Paired average treatment effects. Let ι_0 denote the baseline capability (e.g., the deployed policy class). We focus on two estimands that directly map to the experimental evidence chain:

$$\text{ATE}_{t^*}(\iota) \triangleq \mathbb{E}\left[\hat{t}^*(\iota) - \hat{t}^*(\iota_0)\right], \tag{8}$$

$$\text{ATE}_{\text{RSR}}(\iota) \triangleq \mathbb{E}[y(\iota) - y(\iota_0)]. \tag{9}$$

Expectations are taken over audited episodes and simulator randomness; differences are computed on snapshot-matched EO groups produced by counterfactual parallelism. ATE_{t^*} measures how an intervention shifts the recoverability boundary (earlier vs. later PoNR), while ATE_{RSR} measures the corresponding change in recoverability probability.

Policy-induced gap and actionable lead time. A central quantity in our decomposition is the *policy-induced gap*, Δ_{gap}, which was defined in Eq. (6) as the expected difference between the PoNR under an oracle (ι_{oracle}) and the deployed (ι_{obs}) capability classes. A larger Δ_{gap} indicates that failures are, to a greater extent, attributable to decision quality (recoverability lost earlier under the deployed class) rather than structural inevitability. Correspondingly, the *actionable lead time*, Δ, was defined in Eq. (5) and quantifies the window

between the structural PoNR (under ι_{oracle}) and the observed violation. This is the operational window that is invisible to reactive thresholding yet detectable by recoverability auditing.

Cost reporting. CPA admits a natural cost notion through the number of oracle calls induced by boundary search. When detailed runtime is not the focus, we summarize cost primarily by oracle-call counts and the logarithmic bisection behavior implied by Algorithm 1, and emphasize boundary/lead-time effects as the primary evidence.

4.4 Identification and Robustness

CPA targets counterfactual estimands that are identified by design in a replay-based setting. The central device is *counterfactual parallelism*: multiple interventions branch from the same replayable snapshot x_t, yielding aligned EOs. Accordingly, all paired effects in Sec. 4.3 are interpreted *on the auditable population* and under the oracle simulator.

Auditable population and episode filtering (Track A/B).. CPA is executed only on episodes for which replayable snapshots exist and oracle rollouts can be completed. We denote this auditable subset as **Track A**. Episodes that cannot be audited due to non-replayable conditions (e.g., instant failure at the first snapshot, missing replay state, or other execution failures) are categorized as **Track B** and excluded from the EO dataset used for effect estimation. All estimands in Sec. 4.3, including ATE_{t^*}, ATE_{RSR}, and Δ_{gap}, are therefore defined with respect to Track A.

Censoring and well-defined EO outcomes. For each snapshot x_t and intervention ι, the recoverability label $y(x_t;\iota)$ is defined via the finite-horizon (η, H) test using K oracle rollouts. If oracle verification cannot be completed for a branch (e.g., truncated rollouts or missing simulator outputs), the corresponding EO is treated as *censored* and excluded from snapshot-matched paired comparisons. This convention ensures that both $\hat{t}^*(\iota)$ and $y(x_t;\iota)$ entering Sec. 4.3 are computed only from fully evaluated counterfactual tests, avoiding ambiguous outcomes.

Identification assumptions. Within Track A, identification follows from standard assumptions that are operationally enforced by the snapshot-replay protocol. First, *consistency* holds because counterfactual outcomes are generated by running the oracle simulator from x_t under the specified do-operator $do(\iota)$. Second, *exchangeability* is ensured by construction: interventions share the same starting snapshot, and randomness arises only from independent disturbance draws (and agent sampling noise, when applicable) across oracle rollouts. Third, *no interference* holds because counterfactual branches are executed independently across episodes.

Robustness and Model Fidelity. We complement the design-based identification with rigorous stress tests to verify that the mined boundary is not an artifact of surrogate approximation. In particular, we examine the stability of $\hat{t}^*$ under surrogate model uncertainty by injecting multiplicative Gaussian noise (up to $\sigma = 30\%$) into the rollout dynamics (detailed in Appendix D). The estimated PoNR boundary shifts by less than ±1 hour under moderate noise, confirming that the Zombie Zone is a stable structural feature robust to high-frequency model errors. Additionally, we verify causal effects using within-group label permutation to ensure that paired differences concentrate near zero under randomized interventions. Together, these properties justify interpreting ATE_{t^*} as a structural shift in recoverability, robust to both simulator imperfections and procedural choices. A sensitivity analysis over audit horizon and actuator delay is provided in Table 9.

5 Experiments

We evaluate CPA on a high-fidelity hydrodynamic control testbed of storm-surge dynamics in the Yangtze Estuary. This domain exemplifies **inertia-dominated** constrained systems: safety violations are rare but catastrophic, and interventions exhibit delayed consequences, making purely observational alarms prone to lagging behavior.

Our goal is to challenge a common monitoring paradigm in such systems: instead of treating safety as a property of the *current state*, we test whether *recoverability*—a dynamic property assessed via counterfactual futures—provides a more reliable **leading signal** of impending constraint violation. Beyond early warning, we ask when such signals are operationally useful: can CPA separate *decision-sensitive* failures (where improved actions could still matter) from *structural* inevitability under a fixed audited recovery capability?

The experiments follow a tightly coupled evidence chain:

H1 (Existence): Zombie Zone. There exist episodes that remain observationally feasible yet have already become irrecoverable under the audited capability class. We establish this by case-level evidence (Fig. 2).

H2 (Leading signal & baseline failure): PoNR precedes violation. The structural boundary mined by CPA statistically precedes observed violation, yielding an actionable window. We validate this via population-level phase diagrams (Fig. 3) and show that static threshold alarms can be systematically late across sensitivity settings (Table 3).

H3 (Operational regimes): value depends on capacity. We test whether increasing recovery capacity is associated with a shift from structurally inevitable failures toward decision-sensitive regimes where auditing provides practical debugging value (Fig. 6).

Finally, we provide a mechanistic explanation for why static alarms can fail: early-stage trajectories can be deceptively similar across regimes, while short-horizon dynamics (momentum features) enable accurate early warning (Fig. 4).

5.1 Auditing Protocol & Experimental Setup

We study replayable hydrodynamic control in the Yangtze Estuary under fixed exogenous forcing. The underlying dynamics are defined by a high-fidelity simulator (Delft3D). To accelerate counterfactual boundary search, we use a learned neural operator (UNO) and reserve oracle simulator runs for final verification (Sec. 5.4).

We construct a corpus of risk episodes with snapshots recorded at $\Delta t = 1$h and partition them by temporal resolution. Track A contains auditable episodes with sufficient observation windows ($t_{\text{fail}} \geq 2\Delta t$) to localize a boundary via bracketing; Track B contains instant failures ($t_{\text{fail}} < 2\Delta t$) and is excluded from boundary mining because binary search cannot be bracketed. This is distinct from

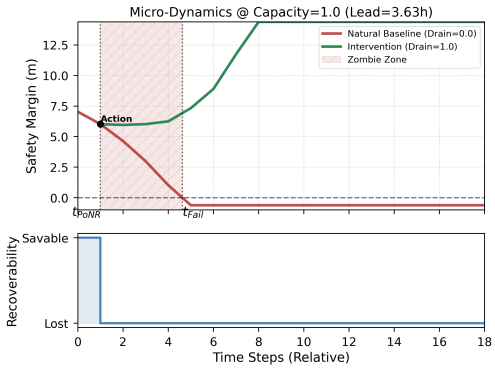


Figure 2: A representative failure episode illustrating the Zombie Zone. Top: Safety margin under baseline (red) and oracle intervention (green). Bottom: Recoverability indicator transition from Savable to Lost. Shaded region = Zombie Zone $[t_{\text{PoNR}}, t_{\text{fail}})$.

left-censored episodes in Track A, which are valid long episodes but structurally unrecoverable even at $t = 0$. Dataset statistics are reported in Appendix B.

The action space is normalized drainage intensity $\mathcal{A} = \{a_t \in [0, D_{\max}]\}$. We instantiate CPA with a single maximal-intensity intervention $\iota_{\max}$ where $a_t \equiv D_{\max}$, treating the intervention class ι as a fixed policy. In this dissipative setting, we adopt a protocol-level monotonicity assumption: within the audited class, larger drainage does not decrease the safety margin over the fixed horizon, so $\iota_{\max}$ serves as an upper bound on recoverability and avoids MPC-style search when defining the boundary.

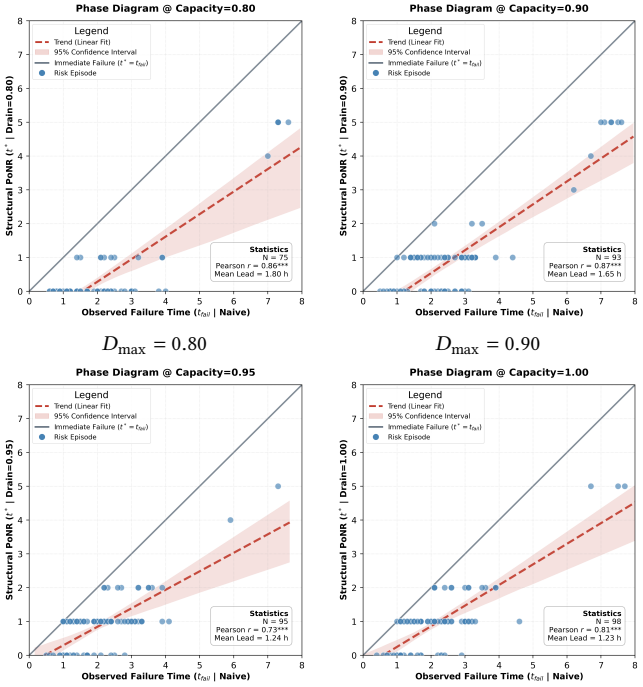
While Eq. 3 defines a general stochastic predicate, our primary audit exploits deterministic snapshot replay. We therefore instantiate recoverability with K=1 and η=0 (up to a small numerical tolerance), and reserve probabilistic auditing (K>1) for surrogate-uncertainty analysis under noise injection (Appendix D, Fig. 11). For each valid episode, PoNR t^* is the latest snapshot such that initiating $\iota_{\max}$ keeps the system feasible over a fixed horizon H=24 hours.

We report baseline failure time t_{fail}, PoNR time t^*, and actionable lead time $\Delta = \max(0, t_{\text{fail}} - t^*)$ in absolute time units, and summarize population-level actionable rate $\mathbb{P}[\Delta \geq 1 \text{ hour}]$. All auditing logs are released to enable deterministic replay.

5.2 Zombie Zone: Existence and Precedence

5.2.1 Micro-Dynamics (H1). We first provide micro-level evidence for the proposed *Zombie Zone*: a critical interval in which the system remains observationally feasible (positive safety margin) yet has already become irrecoverable under the audited capability class.

Figure 2 visualizes a representative failure episode, revealing a systematic lag between *observed feasibility* and *counterfactual recoverability*. Although the safety margin (slack) remains positive for several hours, the recoverability signal drops instantly at t_{PoNR} (denoted as t^* in the audit). Crucially, even the maximum-strength intervention (green trajectory), despite visibly deflecting the slack trajectory, fails to avert the violation at t_{fail}. This decoupling is



$D_{\max} = 0.80$ $D_{\max} = 0.90$

$D_{\max} = 0.95$ $D_{\max} = 1.00$

Figure 3: Phase diagrams across audited intervention capacities. Each point is a replayable risk episode, plotting the structural boundary time t_{PoNR} against the observed failure time t_{fail}. The gray diagonal indicates zero lead time ($t_{\text{PoNR}} = t_{\text{fail}}$).

characteristic of inertia-dominated systems: exogenous forcing and accumulated momentum can render failure inevitable well before the state variable crosses the safety threshold.

Operationally, this confirms that threshold-based monitoring on slack is a lagging indicator. The interval gap Δ constitutes the *Actionable Lead Time* exposed by our recoverability audit. This micro-level anatomy motivates the population-level precedence analyses in Section 5.2.2.

5.2.2 Population-Level Precedence (H2). We extend the analysis from micro-dynamics to the population level by quantifying the temporal precedence of the structural boundary (t_{PoNR}) relative to the observational failure (t_{fail}) across all valid risk episodes.

Precedence and Actionability. Figure 3 plots t_{PoNR} against t_{fail} under multiple audited intervention capacities. Our audit reveals that in many episodes the system enters the irrecoverable regime earlier ($t_{\text{PoNR}} < t_{\text{fail}}$), yielding positive lead time Δ. This separation confirms hydro-dynamical inertia creates a "blind spot" for instantaneous constraints, which our look-ahead mechanism successfully exposes. Quantitatively, the induced lead time is consistently non-zero: across threshold ratios, the median Δ remains around 1–2 hours with a non-trivial actionable rate (Table 2).

Robustness to Safety Thresholds. To assess whether the leading-signal effect depends on a particular slack threshold, we fix the audited capability at the maximum oracle ($D_{\max} = 1.0$) and vary

Table 2: Sensitivity of actionable lead time Δ to threshold ratios (risk episodes at $D_{\max} = 1.0$).

Threshold Ratio	N	Pearson r	Mean Δ (h)	Median Δ (h)	$\mathbb{P}[\Delta \geq 1\text{h}]$
0.80	75	0.86	1.80 ± 0.81	1.70 (IQR 1.15)	0.84
0.85	82	0.85	1.54 ± 0.76	1.50 (IQR 1.30)	0.73
0.90	93	0.87	1.65 ± 0.83	1.90 (IQR 1.40)	0.73
0.95	95	0.73	1.24 ± 0.73	1.20 (IQR 1.10)	0.65
1.00	98	0.81	1.23 ± 0.74	1.10 (IQR 1.15)	0.64

the violation ratio from 0.80 to 1.00 relative to historical peaks, isolating thresholding from capacity effects. Table 2 reports the resulting lead-time and association statistics.

Across all ratios, the lead time Δ exhibits a consistently positive central tendency with a non-trivial actionable rate, indicating that the PoNR boundary is not a threshold-tuned artifact. While stricter thresholds naturally trigger earlier declarations and hence larger leads, the effect persists even at the peak-referenced setting (ratio = 1.00), where the mean lead time remains above 1 hour.

Table 3: Comparison of Late-Alarm Rate and Alarm Lag between CPA and static threshold baselines across intervention capacities.

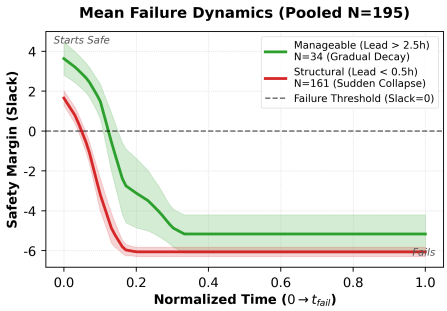
Capacity	Aggressive ($\tau = 0.5$m)		Conservative ($\tau = 1.0$m)	
($D_{\max}$)	Late Rate	Lag (h)	Late Rate	Lag (h)
0.80	98.7%	2.01	81.3%	1.72
0.85	96.3%	1.77	73.2%	1.70
0.90	95.7%	1.90	73.1%	1.71
0.95	85.3%	1.59	63.2%	1.42
1.00	**83.7%**	**1.60**	**58.2%**	**1.44**

5.2.3 Comparative Analysis: Failure of Static Baselines. Table 3 reveals a fundamental limitation of threshold-based monitoring: even with conservative threshold ($\tau = 1.0$m), static alarms lag behind the structural boundary in **58.2%** of episodes at maximum capacity, while CPA provides **1.4–1.6 hours** of additional reaction time. CPA-supervised monitors further outperform stronger observational time-series baselines (Table 8).

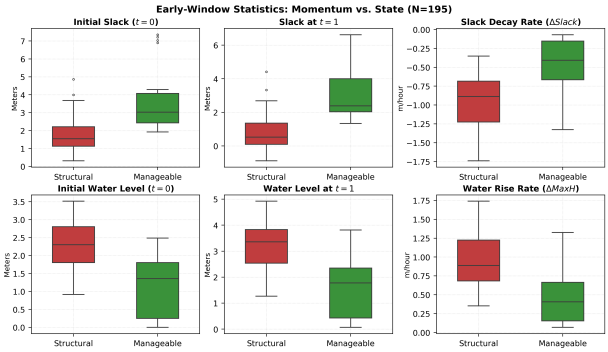
5.2.4 Deceptive Early States and Momentum Signals. Static baselines fail because early states are often *deceptive*: manageable and irrecoverable episodes can exhibit similar slack trajectories before the recoverability transition. Figure 4**(a)** shows substantial early overlap between Structural and Manageable groups, explaining why fixed thresholds incur late alarms. However, short-horizon *momentum* signals (e.g., ΔSlack, ΔMaxH) are more separable than instantaneous state values (Fig. 4**(b)**), suggesting that irrecoverability is encoded in early acceleration patterns rather than slack alone.

5.3 Operational Utility: Regime Taxonomy, Early Warning, and Motif Mining

5.3.1 Operational Regimes (H3). We analyze how failure composition changes with intervention capacity to locate where CPA provides the highest operational value. Figure 5 shows a clear regime



(a) Mean slack dynamics: early overlap.



(b) Early-window stats: static vs. momentum features.

Figure 4: Deceptive early states and momentum signals. (a) Structural and Manageable episodes exhibit substantial early overlap in slack trajectories, causing static thresholds to trigger late alarms. (b) Momentum features (e.g., ΔSlack, ΔMaxH) provide stronger early separability than static state features.

shift: at low capacity, most risk episodes are structural (near-zero lead time, < 0.5h), indicating physically inevitable failures. As capacity increases, the mass moves into the Policy Gap region, where recovery is feasible in principle but missed by the baseline policy. A small fraction becomes clearly manageable (large lead time, > 2.5h), reflecting substantial slack. This trend implies that higher capacity does not eliminate risk; it converts inevitability into policy-sensitive, windowed failures—precisely the setting where snapshot-aligned counterfactual auditing is most informative. Figure 6 visualizes the same transition on a single episode under identical snapshots.

5.3.2 Learnability of PoNR Labels. **CPA is an offline audit, not an online predictor.** Its role is to *produce supervision*: PoNR labels and aligned Evidence Objects that can be learned by lightweight models. Using only the first two snapshots (2h window), a simple Random Forest classifier predicts whether $t_{\text{PoNR}} \leq 1.5$h with high AUC (Fig. 7). An ablation without the capacity feature remains strong, indicating that learnability is not merely driven by control authority but also by short-horizon dynamics.

5.3.3 Failure Atlas Excerpt. To substantiate the mining capability enabled by EOs, we highlight an intuitive Zombie-Zone failure motif, the Oscillation Trap, where control degenerates into high-frequency bang–bang adjustments under shrinking safety margins. We run a snapshot-aligned counterfactual replay on N=30 identified risk episodes: from the same branching snapshot and under

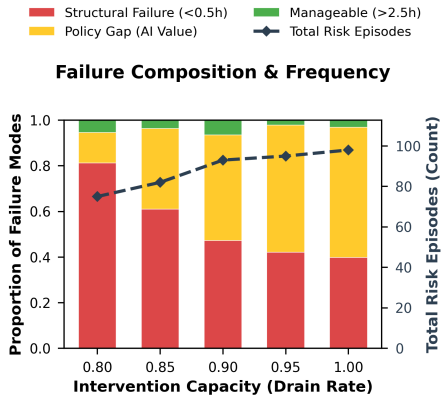


Figure 5: Failure composition vs. capacity. Stacked bars show regime proportions by lead time Δ: Structural ($\Delta < 0.5$h), Policy Gap (intermediate), and Manageable ($\Delta > 2.5$h). Dashed line (right axis): number of risk episodes.

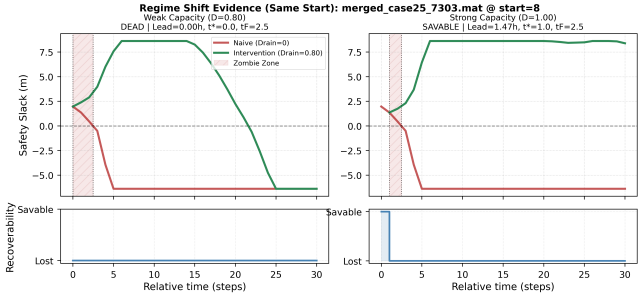


Figure 6: Case Study: Capacity-Dependent Regime Shift. Same snapshot ($t_{\text{start}} = 8$) under two limits: $D_{\max} = 0.80$ (left) remains Structural; $D_{\max} = 1.00$ (right) becomes recoverable under the oracle. Only $D_{\max}$ changes.

identical forcing, we replay (i) a *Panic* controller that induces rapid action fluctuation and (ii) a *Smooth* reference controller. Control instability is quantified by fluctuation intensity $|\Delta u|_1$.

Figure 8 shows a clear distributional separation: Panic exhibits higher oscillation (Mean=0.430) than the Smooth reference (Mean=0.265) on the same snapshot set, indicating that aligned EOs support contrastive motif extraction attributable to behavioral differences rather than episode selection.

Beyond the scalar metric, the aligned EO pairs enable *contrastive sequential pattern mining*. We tokenize each rollout using a physics-informed discretization operator (Appendix E), mapping the normalized slack ratio $r_t = (\theta - h_t)/\theta$ to safety contexts (Safe/Warn/Near/Crit) and mapping action changes $\Delta u_t = u_t - u_{t-1}$ to dynamics symbols (Up++/Up/Dn++/Dn/St), forming tokens such as Warn.St. We then mine N-gram motifs enriched in the failure-inducing branch (Panic) relative to the reference (Smooth) under identical episode conditions. The full procedure (episode-level support, one-sided Fisher test, and Benjamini–Hochberg FDR control) is summarized in Algorithm 2 and detailed in Appendix E.

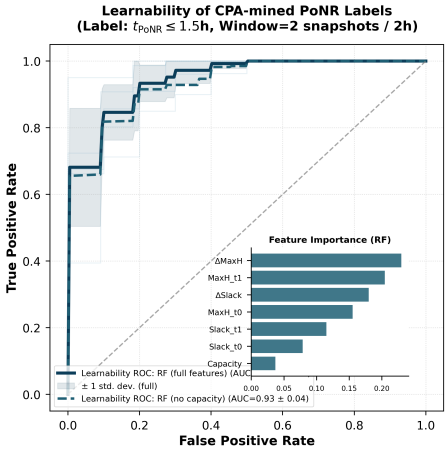


Figure 7: Learnability of CPA-mined PoNR labels. A Random Forest trained on two early snapshots achieves strong ROC-AUC for predicting $t_{\text{PoNR}} \le 1.5$h, and remains competitive without capacity as an input feature. This suggests PoNR provides a learnable supervision signal rather than a hand-crafted threshold.

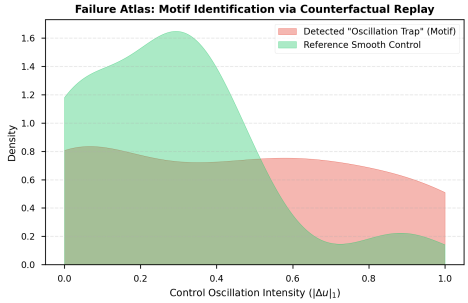


Figure 8: Failure Atlas motif discovery via snapshot-aligned counterfactual replay. Two controllers are replayed from identical branching snapshots across N=30 risk episodes and evaluated by $|\Delta u|_1$. Panic exhibits higher oscillation (Mean=0.430) than Smooth (Mean=0.265), yielding a clear separation.

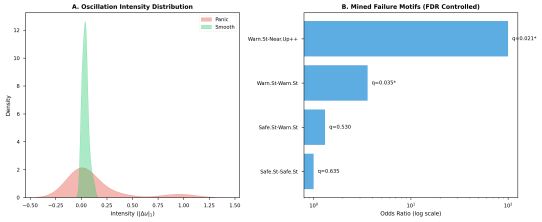
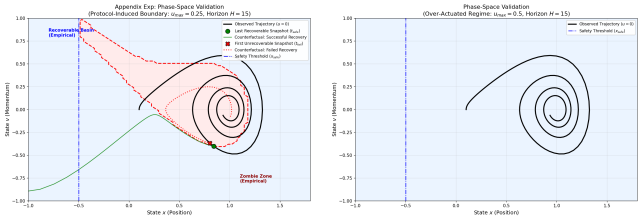


Figure 9: Failure Atlas Mining via snapshot-aligned Evidence Objects. (A) Micro-dynamics view: Panic exhibits higher oscillation intensity than Smooth under identical starting snapshots. (B) Mining view: contrastive 2-gram motifs mined via one-sided Fisher exact test with Benjamini–Hochberg FDR control (Algorithm 2; Appendix E). Odds ratios are log-scaled; infinite values are capped for visualization.



(a) Constrained regime ($u_{\max} = 0.25$) (b) Over-actuated regime ($u_{\max} = 0.5$)

Figure 10: Cross-domain phase-space validation. (a) Empirical recoverability map under the audited protocol $(\mathcal{U}, H)$ (blue: recoverable; pink: unrecoverable/Zombie Zone); dashed contour is the induced boundary. CPA returns a PoNR bracket along the nominal trajectory ($u = 0$): last-recoverable (circle) vs. first-unrecoverable (cross), up to grid/time discretization. (b) Over-actuated regime yields (nearly) uniform recoverability in the plotted region, so CPA finds no PoNR bracket. Definitions and protocol are in Appendix F.

5.4 Scalability and Generalization

5.4.1 Efficiency via Surrogates. Boundary mining in high-fidelity dynamical systems is bottlenecked by expensive simulator rollouts. We benchmark CPA on 22 long-horizon episodes (audit horizon > 3 steps, 1-hour resolution) using measured latencies: **72.0 min** per oracle rollout (Delft3D) versus **8.42 ms** per surrogate step (UNO). Compared to a standard binary-search baseline (**3.09** oracle queries/episode), surrogate-guided localization reduces verification to **1.00** oracle query/episode (**67.6%** fewer queries), yielding an estimated wall-clock reduction from 222.5 to **72.0** minutes per episode (**3.09**× speedup).

Surrogate–oracle boundary fidelity. To verify that the speedup does not materially shift the boundary, we compared surrogate-guided localization against oracle PoNR labels on 22 long-horizon replayable episodes.

Surrogate-guided localization agrees with oracle PoNR within ±1h on 86.4% of episodes (MedAE 1.00h, MeanAE 0.91h; 22 long-horizon episodes), confirming that the 3.09× speedup does not materially shift the boundary.

5.4.2 Cross-domain Validation. To test whether PoNR captures a *protocol-dependent* recoverability transition beyond hydrodynamics, we apply CPA to a canonical damped bistable system with bounded intervention and finite horizon H (see Appendix F for the dynamics, protocol $\mathcal{U}$, and the recoverability predicate). Figure 10 contrasts a constrained regime where a recoverability boundary is identifiable, with an over-actuated regime where PoNR is not identifiable in the plotted region.

6 Discussion

Outcome-centric metrics (e.g., success rate, time-to-failure) are lagging for constrained dynamics: they register violations only after recoverability is already lost. PoNR reframes failure analysis as boundary discovery on the snapshot timeline under an audited intervention class and finite horizon. The interval between PoNR and the first violation constitutes an *actionable window* when it exists at the simulator's temporal resolution.

CPA operationalizes this idea via snapshot replay and bisection on a recoverability predicate, yielding snapshot-aligned Evidence Objects that support structural–policy decomposition through the policy-induced gap and make counterfactual contrastive mining well-posed.

Two practical concerns are audit cost and the replay assumption. CPA is designed as an offline auditing procedure in a replayable simulator (digital twin), not an online controller; its PoNR labels can be distilled into lightweight monitors for deployment (Sec. 5.3), separating expensive supervision generation from real-time operation. We further evaluate scalability (Sec. 5.4) and validate that PoNR localization remains stable under controlled perturbations to surrogate rollouts (Appendix D), which serves as a stress test for error accumulation over the audit horizon. To verify cross-domain behavior, we include a canonical bistable system where dense phase-space auditing is feasible (Appendix F), illustrating PoNR as a protocol-dependent recoverability transition beyond hydrodynamics. Finally, all claims remain conditional on the audited intervention class and observation design; we report a coastline-restricted observation variant as a negative result in the supplementary material.

7 Conclusion

We proposed PoNR boundary mining as a leading-boundary view of failure in constrained dynamical environments and introduced CPA to estimate PoNR via snapshot-replay interventions. CPA produces snapshot-aligned Evidence Objects that support causal attribution, structural–policy decomposition, and downstream motif discovery from counterfactual traces.

Experiments on a high-fidelity hydrodynamic control testbed show that PoNR can precede observable violation and provide actionable lead time in risk episodes. Additional analyses on a canonical bistable system confirm that the Zombie Zone is not unique to the hydrodynamic testbed, though broader cross-domain empirical validation remains future work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Key Project, Grant No. 52539005), in part by the National Key Research and Development Program of China (Grant No. 2025YFE0213400), in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62276109, and in part by the Interdisciplinary Research Program of HUST under Grant No. 5003210069.

The authors thank the anonymous reviewers for their constructive feedback. AI assistants were used in a limited manner for language polishing and editorial refinement of the manuscript, but not for generating results, conducting analyses, or making scientific decisions.

References

[1] Mattijs Baert, Pietro Mazzaglia, Sam Leroux, and Pieter Simoens. 2025. Maximum causal entropy inverse constrained reinforcement learning. *Machine Learning* 114, 4 (2025), 103.

[2] Thomas Banker and Ali Mesbah. 2025. Model-free Reinforcement Learning for Model-based Control: Towards Safe, Interpretable and Sample-efficient Agents. *arXiv e-prints* (2025), arXiv–2507.

[3] S. Baron. 2023. Explainable AI and Causal Understanding: Counterfactual Approaches Considered. *Minds and Machines* 33 (2023), 347 – 377. doi:10.1007/

s11023-023-09637-x
[4] Zhenhua Dong, Hong Zhu, Pengxiang Cheng, Xinhua Feng, Guohao Cai, Xiuqiang He, Jun Xu, and Ji-Rong Wen. 2020. Counterfactual learning for recommender system. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020). doi:10.1145/3383313.3411552
[5] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. 2021. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese* 198, Suppl 27 (2021), 6435–6467.
[6] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. 2018. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Automat. Control* 64, 7 (2018), 2737–2752.
[7] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S Yu. 2019. A survey of parallel sequential pattern mining. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 3 (2019), 1–34.
[8] Kunal Garg, Songyuan Zhang, Oswin So, Charles Dawson, and Chuchu Fan. 2024. Learning safe control for multi-robot systems: Methods, verification, and open challenges. *Annu. Rev. Control.* 57 (2024), 100948. doi:10.1016/j.arcontrol.2024.100948
[9] Nathan Grinsztajn, Johan Ferret, Olivier Pietquin, Matthieu Geist, et al. 2021. There is no turning back: A self-supervised approach for reversibility-aware reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 1898–1911.
[10] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
[11] Ankita Kushwaha, Kiran Ravish, Preeti Lamba, and Pawan Kumar. 2025. A Survey of Safe Reinforcement Learning and Constrained MDPs: A Technical Survey on Single-Agent and Multi-Agent Safety. *ArXiv* abs/2505.17342 (2025). doi:10.48550/arxiv.2505.17342
[12] Huihan Liu, Shivin Dass, Roberto Martín-Martín, and Yuke Zhu. 2024. Model-based runtime monitoring with interactive imitation learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4154–4161.
[13] Diego Manzanas Lopez, Patrick Musau, Nathaniel P Hamilton, and Taylor T Johnson. 2022. Reachability analysis of a general class of neural ordinary differential equations. In *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 258–277.
[14] Murali Krishna Pasupuleti. 2025. Auditing Black-Box AI Systems Using Counterfactual Explanations. *International Journal of Academic and Industrial Research Innovations(IJAIRI)* (2025). doi:10.62311/nesx/rphcr20
[15] Matěj Petrlík, Tomáš Báča, Daniel Heřt, Matouš Vrba, Tomáš Krajník, and Martin Saska. 2020. A robust UAV system for operations in a constrained environment. *IEEE Robotics and Automation Letters* 5, 2 (2020), 2169–2176.
[16] Silviu Pitis, Elliot Creager, and Animesh Garg. 2020. Counterfactual Data Augmentation using Locally Factored Dynamics. *ArXiv* abs/2007.02863 (2020).
[17] Karlo Rado, Mirko Baglioni, and Anahita Jamshidnejad. 2025. Enabling robots to autonomously search dynamic cluttered post-disaster environments. *Scientific Reports* 15, 1 (2025), 34778.
[18] Stratis Tsirtsis, A. De, and M. Gomez-Rodriguez. 2021. Counterfactual Explanations in Sequential Decision Making Under Uncertainty. (2021), 30127–30139.
[19] Simon Wiegrebe, Philipp Kopper, R. Sonabend, and Andreas Bender. 2023. Deep learning for survival analysis: a review. *Artificial Intelligence Review* 57 (2023). doi:10.1007/s10462-023-10681-3
[20] Youxi Wu, Yufei Meng, Yan Li, Lei Guo, Xingquan Zhu, Philippe Fournier-Viger, and Xindong Wu. 2024. COPP-Miner: Top-k Contrast Order-Preserving Pattern Mining for Time Series Classification. *IEEE Transactions on Knowledge and Data Engineering* 36 (2024), 2372–2387. doi:10.1109/tkde.2023.3321749
[21] Youxi Wu, Yuehua Wang, Yan Li, Xingquan Zhu, and Xindong Wu. 2021. Top-k self-adaptive contrast sequential pattern mining. *IEEE transactions on cybernetics* 52, 11 (2021), 11819–11833.
[22] Qisong Yang, Thiago D Simão, Simon H Tindemans, and Matthijs TJ Spaan. 2023. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning* 112, 3 (2023), 859–887.
[23] Xiang Yin, Bingzhao Gao, and Xiao Yu. 2024. Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges. *Annual Reviews in Control* 57 (2024), 100940.
[24] Linrui Zhang, Q. Zhang, Li Shen, Bo Yuan, Xueqian Wang, and Dacheng Tao. 2022. Evaluating Model-free Reinforcement Learning toward Safety-critical Tasks. *ArXiv* abs/2212.05727 (2022). doi:10.48550/arxiv.2212.05727
[25] Qiyuan Zhang, Shu Leng, Xiaoteng Ma, Qihan Liu, Xueqian Wang, Bin Liang, Yu Liu, and Jun Yang. 2024. CVaR-Constrained Policy Optimization for Safe Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* 36 (2024), 830–841. doi:10.1109/tnnls.2023.3331304

A Notation

Table 4: Summary of Key Notation

Symbol	Description
Core Time & Audit Variables	
t_{fail}	Observed failure time (first constraint violation).
t_{PoNR}	Point of No-Return: last time recoverable under audited capability (conceptual).
t^*	Estimated PoNR time from CPA (used in algorithms and results).
$\hat{t}^*$	Formal estimator of PoNR (output of Algorithm 1).
Δ	Actionable lead time: $\Delta = \max(0, t_{\text{fail}} - t^*)$.
Δ_{gap}	Policy-induced gap: $\Delta_{\text{gap}} = \hat{t}^*_{\text{oracle}} - \hat{t}^*_{\text{obs}}$.
Capacity, Thresholds & Metrics	
$D_{\max}$	Maximum intervention capacity (normalized drainage intensity).
θ_{safe}	Episode safety threshold (water level).
η	Tolerance parameter for probabilistic recoverability.
H	Audit / rollout horizon (hours).
Framework & Data Components	
x_t, s_t	Replayable system snapshot / physical state at time t.
$y(x_t; \iota)$	Recoverability indicator (1=recoverable, 0=lost) under intervention ι.
$\mathcal{D}$	Dataset of Evidence Objects (EOs).
Track A	Auditable episodes (PoNR localization meaningful).
Track B	Instant-failure episodes ($t_{\text{fail}} < 2$ h).
Zombie Zone	Interval $[t_{\text{PoNR}}, t_{\text{fail}})$: feasible but irrecoverable.
General Mathematical Symbols	
$m(s_t)$	Safety slack: $m(s_t) = \theta_{\text{safe}} - \max(H_t)$.
$\mathbb{I}[\cdot]$	Indicator function.
$\mathbb{E}[\cdot], \mathbb{P}(\cdot)$	Expectation and probability operators.
$\mathcal{U}$	Intervention space (set of admissible control actions).
K	Number of rollout repetitions for stochastic auditing.

B Dataset Characteristics

Surrogate model fidelity. To ensure physical fidelity, surrogate errors are evaluated only over valid water cells, excluding land regions where hydrodynamic states are not meaningful. The valid water mask is extracted directly from environment input channels. The water-only MAE is 0.149 m, indicating reliable spatial patterns for constraint evaluation.

Episode extraction and Track A/B taxonomy. Track A contains auditable episodes with $t_{\text{fail}} \geq 2$h; Track B contains instant failures. Within Track A, episodes are classified as observed, left-censored, or right-censored (Sec. 4.4).

Dataset Statistics Atlas. Figure 12 summarizes the audited episodes: peak water levels, baseline failure times t_{fail}, actionable lead times $\Delta = t_{\text{fail}} - t^*$, and episode type composition (Track A/B, censoring categories).

C CPA Implementation and Evidence Objects

Evidence Object schema. Each EO contains: initial snapshot, forcing schedule, naive-policy trace, oracle outcomes, and PoNR localization results (Table 5).

Counterfactual PoNR Auditing via Binary Search. Algorithm 1 provides the full procedure.

Table 5: Evidence Object fields (summary).

Field	Description
`snapshot_0`	Initial state snapshot
`forcing_seq`	Recorded forcing schedule
`t_start`	Absolute start time index
`naive_trace`	Slack/trajectory under naive policy
`t_fail`	First violation time (absolute)
`oracle_scan[1.0]`	Oracle audit under max-strength intervention
`t_star`	Localized PoNR time t^* (absolute)
`censoring_type`	left-/right-censored or observed

Algorithm 1 Counterfactual PoNR Auditing on Track-A Episodes

Input: Track-A episodes $\{x_t\}_{t=0}^{T}$, capability ι, horizon H, tolerance η, budget K
Output: PoNR estimate $\hat{t}^*$, censoring flag(s), EO dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$
for episode e in Track A **do**
  obtain $\{x_t\}_{t=0}^{T}$ and t_{fail}
  if $t_{\text{fail}} < 2$ **then**
    continue
  end if
  $t_a \leftarrow 0,\ t_b \leftarrow T$
  query $y(x_{t_a}; \iota)$; record EO
  if $y(x_{t_a}; \iota) = 0$ **then** mark **left-censored**; **continue**
  end if
  query $y(x_{t_b}; \iota)$; record EO
  if $y(x_{t_b}; \iota) = 1$ **then** mark **right-censored**; **continue**
  end if
  while $t_b - t_a > 1$ **do**
    $t_{\text{mid}} \leftarrow \lfloor (t_a + t_b)/2 \rfloor$
    run K rollouts from $x_{t_{\text{mid}}}$ under $do(\iota)$
    compute $\hat{p}_{\text{mid}}$; $y_{\text{mid}} \leftarrow \mathbb{I}(\hat{p}_{\text{mid}} \geq 1 - \eta)$; record EO
    if $y_{\text{mid}} = 1$ **then** $t_a \leftarrow t_{\text{mid}}$
    else $t_b \leftarrow t_{\text{mid}}$
    end if
  end while
  $\hat{t}^* \leftarrow t_a$; record as t_{PoNR}
end for
return $\{\hat{t}^*\}, \mathcal{D}$

D Surrogate Reliability for Counterfactual Auditing

Physics-informed hydrodynamic emulation. We adopt a U-shaped Neural Operator (U-NO) with spectral convolutions. Training uses $\mathcal{L}_{total} = \mathcal{L}_{data} + \lambda \mathcal{L}_{mass}$.

Table 6: U-NO emulator performance against Delft3D.

Variable	NSE	RMSE	MAE	Rel L_2	MSE
Water Depth (H)	0.9955	0.0342	0.0156	0.0307	3.78×10^{-3}
Flow Velocity (U)	0.9948	0.0169	0.0078	0.0361	9.54×10^{-4}
Flow Velocity (V)	0.9814	0.0326	0.0142	0.0549	4.92×10^{-3}
Overall	**0.9927**	0.0296	0.0125	0.0358	3.22×10^{-3}

Robustness of PoNR under noise. Injecting multiplicative Gaussian noise up to $\sigma = 30\%$, the boundary shifts by less than ±1 hour; degradation occurs only at $\sigma > 40\%$ (Fig. 11).

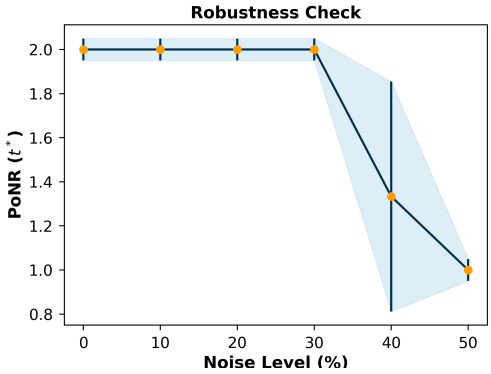


Figure 11: Robustness of PoNR identification under surrogate noise.

Structured bias and control degradation. Under systematic/spatial surrogate biases (±0.05 offset), boundary movement remains limited and oracle agreement stays high (> 91%). Under actuator degradation (10–20% capacity reduction), actionability remains 100% with monotonically earlier PoNR shifts.

E Environment Details and Structured Robustness

Deterministic forcing replay. To ensure strict counterfactual alignment, the environment maintains an absolute time index $t_{\text{abs}} = t_{\text{start}} + t$. At each step: (i) apply action to (H, U, V), (ii) run UNO and add increments, (iii) mask and clamp, (iv) overwrite Channels 0–3 with recorded forcing at $t^+_{\text{abs}} = t_{\text{start}} + t + 1$. This guarantees all branches share identical exogenous conditions.

Action semantics: drainage and momentum damping. The intervention $a_t \in [0, D_{\max}]$ is applied before the surrogate update:

$$H_t \leftarrow H_t - \alpha \cdot a_t, \tag{10}$$

$$U_t \leftarrow U_t \cdot (1 - \beta \cdot a_t), \tag{11}$$

$$V_t \leftarrow V_t \cdot (1 - \beta \cdot a_t), \tag{12}$$

with $\alpha = 1.0$ and $\beta = 0.2$. This conservative configuration induces strong inertia: single-step drainage has limited effect and requires sustained intervention.

Safety slack and termination. The episode-level threshold is $\theta_{\text{safe}} = 0.85 \times \max(H_{\text{historical}})$. Slack is $m(S_t) = \theta_{\text{safe}} - \max_{x,y} H_t(x, y)$; violation triggers when $m(S_t) < -0.01$. The environment enforces a strict cap of $T = 24$ steps.

Robustness under structured perturbations. Beyond Gaussian noise (Sec. D), we test structured surrogate biases and control-side degradation (Table 7). Systematic/spatial biases (±0.05 offset) yield limited boundary movement and high oracle agreement (> 91%). Actuator degradation (10–20% capacity reduction) preserves 100% actionability with monotonically earlier PoNR shifts.

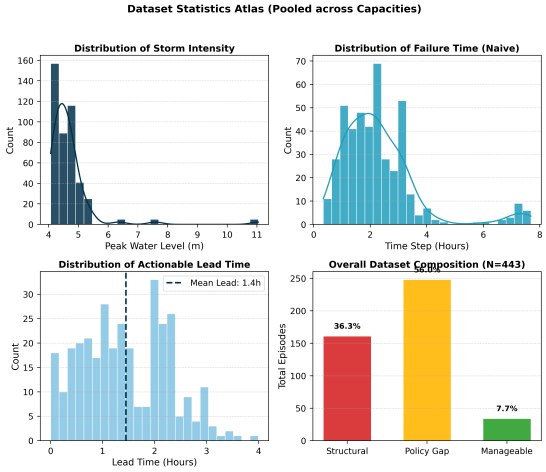


Figure 12: Dataset Statistics Atlas. Episode intensities, failure times, lead times, and dataset composition.

Table 8: Stronger early-warning baselines and CPA-supervised monitors (135 episodes).

Method	LAR↓	ML↓	MAO↓
Observational			
Static	51.1	+0.46	1.00
Momentum	48.9	+0.39	0.92
Trend	48.1	+0.36	0.91
CPA-supervised (Ours)			
Logistic Reg.	32.6	−0.24	0.90
Random Forest	32.6	−0.03	0.73

LAR: Late Alarm Rate (%); **ML**: Mean Lag (h); **MAO**: Mean Abs Offset (h).

Table 9: Horizon and delay sensitivity (240 episode-capacity pairs).

Setting	Shift↓	RC↓	LS↓
Horizon (6h–18h)	0	0.4	0
Delay 1h	0	20.4	0
Delay 2h	−1.0	31.2	0
Delay 3h	−1.0	47.8	6.7
Delay 4h	−1.0	50.9	9.2
Delay 5h	−1.0	53.7	20.8

Shift: Median Shift (h); **RC**: Regime Change (%); **LS**: Lost Salvageable (%).

Table 7: Structured robustness: surrogate bias and control degradation.

Setting	Aff.(%)	Reg.Chg.(%)	Lost(%)	Agree.(%)
Systematic bias (+0.05)	4.2	1.2	0.0	91.7
Spatial bias (+0.05)	1.2	0.0	0.0	92.5
Degradation	**Act.**	**Reg.Chg.**	**Aff.**	—
10% cap. red.	100%	5.0%	17.9%	—
20% cap. red.	100%	15.0%	45.0%	—

Motif mining algorithm. Algorithm 2 details the contrastive sequential pattern mining operator used to construct the Failure Atlas (Fig. 9). Each rollout is tokenized into safety contexts (Safe/Warn/Near/Crit) and dynamics symbols (Up++/Up/Dn++/Dn/St), forming tokens such as `Warn.St`. Episode-level support (document frequency) prevents long rollouts from dominating statistics.

Algorithm 2 Contrastive Failure Motif Mining on Snapshot-Aligned EOs (Fisher + BH-FDR)

Input: Target EO rollouts $\mathcal{D}_{\text{tar}}$ (e.g., Panic), reference $\mathcal{D}_{\text{ref}}$ (e.g., Smooth); N-gram length N; min support τ
Output: Ranked motifs with $(w, \text{OR}(w), p(w), q(w))$

Tokenization: convert each rollout into token sequence
for rollout ρ in $\mathcal{D}_{\text{tar}} \cup \mathcal{D}_{\text{ref}}$ **do**
  $S(\rho) \leftarrow [\,]$
  for $t = 1$ to $T(\rho)$ **do**
    $r_t \leftarrow (\theta - h_t)/\theta$; map to $s_t \in$ {`Safe, Warn, Near, Crit`}
    $\Delta u_t \leftarrow u_t - u_{t-1}$; map to $a_t \in$ {`Up++, Up, Dn++, Dn, St`}
    append $\sigma_t \leftarrow s_t.a_t$ to $S(\rho)$
  end for
end for
Episode-level support: initialize $a(\cdot), b(\cdot) \leftarrow 0$
for ρ in $\mathcal{D}_{\text{tar}}$ **do**
  for w in ngrams$(S(\rho), N)$ **do**
    $a(w) \leftarrow a(w) + 1$
  end for
end for
for ρ in $\mathcal{D}_{\text{ref}}$ **do**
  for w in ngrams$(S(\rho), N)$ **do**
    $b(w) \leftarrow b(w) + 1$
  end for
end for
Fisher exact test: for each w with $a(w) \geq \tau$, compute $(\text{OR}(w), p(w))$
BH-FDR: apply Benjamini–Hochberg to obtain $q(w)$
return motifs sorted by $q(w) \uparrow$, $\text{OR}(w) \downarrow$

F Cross-domain Validation Details

We consider a damped bistable system:

$$\dot{x} = v, \qquad \dot{v} = \frac{x - x^3 - \gamma v + u}{m}, \qquad |u| \le u_{\max}, \tag{13}$$

with $m = 1.0$, $\gamma = 0.2$, safe set $\mathcal{S} = \{(x, v) : x \le x_{\text{safe}} = -0.5\}$, and audit horizon $H = 15$s. The audited intervention class is maximal recovery $\mathcal{U} = \{u(t) \equiv -u_{\max}\}$ with $u_{\max} = 0.25$.

The recoverability predicate is:

$$R_{\mathcal{U},H}(s_0) = \mathbb{I}\Big[\min_{t\in[0,H]} x(t; s_0, u) \le x_{\text{safe}} \text{ under } u(\cdot) \in \mathcal{U}\Big]. \tag{14}$$

For visualization, we compute an empirical phase-space map by brute-force grid evaluation (100×100). CPA independently applies bisection to $R_{\mathcal{U},H}(s_t)$ along the nominal trajectory ($u = 0$) and returns a discrete bracket: last recoverable (circle) vs. first unrecoverable (cross), up to discretization.