$\Phi = I \times \rho - \alpha \times S$: A Domain-Agnostic Stability Metric and Autonomous Controller Validated Across Neural, Quantum, Mechanical, and Physiological Systems

Shawn Barnicle[*]
Independent Researcher, Chicago, IL, USA

February 18, 2026

Abstract

We describe a stability framework for detecting, and in certain settings correcting, system degradation across heterogeneous physical and computational domains. The core element is a three-term stability metric,

$$\Phi = I \times \rho - \alpha \times S,$$

where I (Identity) measures functional preservation relative to a known-good baseline, ρ (Coherence) measures short-horizon temporal consistency of observations, and S (Entropy) measures output disorder. The structural form of the metric is fixed; only the domain-specific definitions used to compute I, ρ, and S vary with the available signals. This separation mirrors standard practice in physics: a governing relation can remain unchanged while its domain instantiation depends on the measurement model.

We evaluate the metric across eight domains: mechanical bearings, turbofan engines, power grids, geophysical systems, neural network training, quantum circuits on IBM hardware, physiological cardiac signals, and large language models. Across these settings, Φ decreases in advance of observed failures or performance collapse, providing early-warning behavior without requiring a learned predictor trained on failure labels.

We also present the Φ-Objective Controller, an autonomous control engine that consumes Φ from a domain adapter, forecasts the effect of candidate interventions using a lightweight surrogate model, and selects actions that maximize expected task performance subject to a Φ-based safety constraint. To mitigate proxy-gaming failure modes, the controller implements three anti-Goodhart safeguards that reject interventions when the metric improves while measured task performance is predicted to worsen. Empirical results on neural network training demonstrate (i) monitoring-only operation that matches baseline outcomes while avoiding catastrophic early-stopping failures, and (ii) surrogate-enabled interventions that are net-positive under conservative gating and strict intervention budgets.

Keywords: stability monitoring; early warning; autonomous control; proxy misalignment; Goodhart safeguards; system degradation; identity preservation

1 Introduction

Any system that operates over time—engineered or biological—eventually departs from its intended behavior. A bearing that ran smoothly for months begins vibrating at new frequencies; a neural

[*] Disclosure: Portions of this work are described in U.S. Provisional Application No. 63/984,704, filed February 17, 2026.

network that classified accurately starts collapsing under distribution shift; a quantum circuit that produced correct output yesterday fails today because calibration drifted overnight. In each case, practitioners build monitors and rules to detect degradation and decide whether (and how) to intervene.

A recurring pattern shows up across these domains: the tooling is usually built *inside* the domain. A vibration threshold is not designed to reason about model training. An early-stopping rule is not designed to reason about qubit quality. Even when the underlying goal is similar—detecting a loss of functional integrity and choosing a corrective action—the metric and decision logic are typically incompatible across settings.

This paper proposes a single fixed-form stability score that can be instantiated using domain-appropriate signals:

$$\Phi = I \times \rho - \alpha \times S. \tag{1}$$

Here, I (Identity) captures functional preservation relative to a known-good baseline, ρ (Coherence) captures short-horizon temporal consistency, and S (Entropy) captures output disorder. The form of Eq. (1) is held constant; what changes across domains is *how* I, ρ, and S are computed from available measurements.

The product $I \times \rho$ encodes an operational constraint: functional preservation only counts when it is stable over time. A system that appears healthy in a single snapshot but fluctuates sharply across observations should not be treated as stable. The entropy penalty $-\alpha \times S$ captures a second, common warning sign: disorder often increases before failure is obvious in a primary performance scalar. The coupling coefficient α controls how strongly disorder is weighted, and in the domains studied it typically falls into consistent ranges by system class (e.g., smaller values for many mechanical/cardiac settings; larger values for certain neural/EEG settings). Shannon entropy is used as the disorder measure throughout this work; other information-theoretic or statistical dispersion measures (e.g., Rényi entropy, Tsallis entropy, approximate entropy) may be substituted for S without altering the structural form of the metric.

We evaluate Φ as a monitoring signal across eight domains: mechanical bearings, turbofan engines, power grids, geophysical systems, neural network training, quantum circuits on IBM hardware, physiological cardiac signals, and large language models. We then describe an autonomous controller that uses Φ as a *guardrail* (not an optimization target) when selecting interventions. The controller separates domain measurement (adapters) from decision logic (a shared engine), enabling the same control procedure to operate on neural training runs, quantum execution choices, maintenance-relevant sensor streams, and physiological monitoring, provided that each domain supplies an adapter and an action set.

1.1 Relationship to prior work

Predictive maintenance spans classical thresholding and standards-based monitoring for vibration and other sensor modalities, as well as data-driven approaches that train classifiers and sequence models on labeled degradation trajectories. In neural network training, common supervision mechanisms include early stopping, learning-rate scheduling, and resource-allocation strategies that terminate runs predicted to underperform. In quantum computing, calibration-aware qubit selection, dynamical decoupling, and error mitigation techniques seek to reduce error rates under hardware drift. In physiological monitoring, domain-specific metrics such as HRV-derived features and interval-based measures remain standard.

These methods are effective within their intended domains, but they are generally *not* designed around a shared, plug-in stability score coupled to a shared decision engine. The closest conceptual predecessor is the use of entropy as a general indicator in hierarchical control discussions [7].

That line of work is largely conceptual and does not specify a concrete, computable metric that is evaluated across heterogeneous datasets and platforms. By contrast, Lyapunov-style stability analysis provides guarantees for classes of modeled dynamical systems, but it typically requires an explicit system model or identification procedure, rather than a directly computable health score from observational streams.

1.2 Contributions

This paper makes the following contributions:

1. A fixed-form stability metric, Eq. (1), instantiated across eight domains using domain-specific definitions of I, ρ, and S.

2. An adaptive threshold relationship used in several physical domains of study,

$$\text{Threshold} = 1 + \alpha \times \left(\frac{\text{degradation\_rate}}{100}\right) \times g(I, \rho), \tag{2}$$

   where $g(I, \rho)$ is a bounded modifier derived from the same identity/coherence signals used in Φ. In certain embodiments, $g(I, \rho) = \sqrt{I}\,\rho$; in other embodiments where I and ρ are unavailable or intentionally excluded, $g(I, \rho) = 1$ (reducing to the simpler form).

3. A domain-agnostic controller architecture that separates (i) domain adapters (measurement and Φ computation), (ii) a control engine (action selection under constraints), and (iii) a lightweight surrogate predictor (forecasting action effects from logged experience).

4. A performance-first objective: the controller selects actions to maximize expected task performance while treating Φ as a safety constraint, rather than directly maximizing Φ.

5. Three proxy-misalignment safeguards—a performance floor, a step-wise rejection rule for $\Phi\uparrow$ with performance$\downarrow$, and a history-based correlation monitor—with a conservative fallback to `no_op` when uncertainty is high.

6. Empirical results on neural network training (CIFAR-10, ResNet-18, multiple deterministic seeds) showing that monitoring-only operation matches baseline outcomes while avoiding catastrophic early-stopping failures on certain seeds, and that surrogate-enabled interventions can produce net-positive aggregate results under strict gating and small intervention budgets.

2 The Φ Metric

2.1 Core formulation

For a system under observation, we define Φ as in Eq. (1), with four components:

Identity ($I \in [0, 1]$): a normalized measure of functional preservation relative to a known-good baseline. $I = 1$ indicates baseline-level function; $I = 0$ indicates complete functional loss under the chosen definition.

Coherence ($\rho \in [0, 1]$): a measure of short-horizon temporal consistency, computed over a sliding window (e.g., lag-1 autocorrelation of a primary performance signal) and clamped to $[0, 1]$.

Entropy ($S \in [0, 1]$): a normalized Shannon entropy [9] of an output distribution chosen for the domain, where higher values indicate greater disorder.

Coupling coefficient ($\alpha \in (0, 1)$): a weighting parameter that sets the strength of the entropy penalty relative to the $I \times \rho$ term. In the domains studied, α is treated as a system-class parameter and selected from empirically stable ranges.

The structure of $I \times \rho$ is intentional: high identity with low coherence (good in snapshots, erratic over time) yields a low product, and high coherence with low identity (consistently wrong) also yields a low product. The subtraction of $\alpha \times S$ penalizes increases in disorder that may precede obvious failures in the primary performance scalar.

2.2 Fixed-form relation with domain instantiation

It is standard in physics and engineering to work with relations whose *form* is fixed while their instantiation depends on measurement and context. In our setting, Eq. (1) plays the role of the fixed-form relation, while each domain adapter specifies how raw observations are mapped into I, ρ, and S. The point is not that each domain shares identical sensors or identical failure modes; the point is that the same three abstract failure signatures—loss of intended function, loss of temporal consistency, and growth of disorder—can be measured in different ways and combined through one stable algebraic form. The component definitions in Table 1 are non-limiting examples; I, ρ, and S may be computed by any method that quantifies functional preservation, temporal consistency, and output disorder, respectively.

2.3 Domain-specific component definitions

Table 1 summarizes the component definitions used in this work. Each row corresponds to a domain adapter that maps native measurements to (I, ρ, S) and then computes Φ using Eq. (1).
Unless stated otherwise, the reported experiments use $\alpha = 0.1$. In other system classes (e.g., certain neural/EEG settings), higher α values may be appropriate; this work treats α as a system-class parameter selected from empirically stable ranges.

2.4 Adaptive threshold framework

A raw Φ score becomes actionable when paired with a thresholding rule. In several physical domains studied here, we observed a consistent scaling relationship between an estimated degradation rate and the decision threshold. We express this relationship in a general form:

$$\text{Threshold} = 1 + \alpha \times \left(\frac{\text{degradation\_rate}}{100} \right) \times g(I, \rho), \tag{3}$$

where $g(I, \rho)$ is a bounded modifier derived from the same identity/coherence signals used in Φ.
In certain embodiments, $g(I, \rho) = \sqrt{I}\,\rho$. In other embodiments where I and ρ are unavailable, intentionally excluded, or treated as fixed, $g(I, \rho) = 1$, yielding the simplified scaling:

$$\text{Threshold} = 1 + \alpha \times \left(\frac{\text{degradation\_rate}}{100} \right). \tag{4}$$

Across the systems evaluated with this approach, the factor of 1/100 behaved consistently rather than acting like a free tuning knob. For the neural and quantum controller settings reported in this paper, a fixed critical boundary of $\Phi_c = 0.25$ is used as the operational threshold unless otherwise specified.

Table 1: Component definitions by domain (non-limiting).

Domain	I **(Identity)**	ρ **(Coherence)**	S **(Entropy)**	α
Bearings	Normalized deviation of vibration RMS from a healthy baseline	Autocorrelation of vibration measurements over a window	Spectral entropy of the frequency distribution	0.1
Turbofan engines	Normalized sensor deviation from baseline operating profile	Autocorrelation of sensor readings over operational windows	Entropy of multi-sensor degradation distribution	0.1
Power grids	Grid frequency deviation from nominal (e.g., 50 Hz) normalized to baseline stability	Autocorrelation of frequency measurements over operational windows	Shannon entropy of frequency-deviation distribution	0.1
Geophysical (seismic)	Baseline seismicity rate relative to current activity, normalized	Temporal coherence of foreshock or strain sequences	Shannon entropy of event magnitude or strain-measurement distribution	0.1
Neural networks	$\dfrac{\text{accuracy} - 1/C}{1 - 1/C}$, where C is the number of classes	Lag-1 autocorrelation of accuracy over a training window	Shannon entropy of flattened confusion matrix divided by $\log(C^2)$	0.1
Quantum circuits	Fidelity relative to a noiseless reference (e.g., simulation)	Temporal consistency of calibration metrics (windowed)	Shannon entropy of measurement-outcome distribution	0.1
Cardiac (physiological)	$\min\{1,\ \sigma_{RR,0}/\sigma_{RR,t}\}$	Lag-1 autocorrelation of RR intervals	Shannon entropy of RR-interval histogram	0.1
LLMs	Embedding cosine similarity of outputs relative to a reference set	Consistency of outputs across repeated prompts (windowed)	Shannon entropy of token or embedding distribution	0.1

3 Detection Validation Across Domains

We first evaluate Φ in monitoring-only mode. Across the domains summarized in Table 1, Φ typically declines ahead of obvious failure, providing early warning without training a machine-learning model on failure labels. In this mode, Φ is computed directly from the chosen signals using the domain adapter definitions, and no controller actions are taken.

3.1 Mechanical systems (bearings)

Using the XJTU-SY bearing dataset [10] (15 bearings, 3 operating conditions, 5,695 vibration files), Φ was computed for each bearing using vibration RMS signals. All 10 bearings that reached end-of-life produced $\Phi < 0$, ranging from -0.003 (Bearing1_4) to -0.370 (Bearing2_3). No healthy-state bearing produced Φ below the critical threshold. The adaptive threshold framework (Eq. (4)) was separately validated on the same dataset: 10/15 bearings passed at $F1 \geq 0.5$, a 10× improvement over the v2.0 fixed-threshold baseline (1/15).

3.2 Turbofan engines

Using the NASA C-MAPSS turbofan [8] run-to-failure dataset (FD001 subset, 10 engines evaluated, 20,631 sensor readings across 14 operational sensors), Φ was computed from multi-sensor degradation profiles. All 10 engines produced $\Phi < 0.25$ at end-of-life, ranging from 0.039 (Engine 9) to 0.241 (Engine 4). Each engine was correctly classified as approaching failure using the fixed threshold $\Phi_c = 0.25$.

3.3 Power grids

Grid frequency data were analyzed for two national grids. For the UK National Grid on August 9, 2019 (the date of a major blackout event), $\Phi = 0.178$, correctly indicating a critical state below the 0.25 threshold. For the German grid over a comparable stable period, $\Phi = 0.401$, correctly indicating stability. The two cases used 2.6M and 2.5M frequency measurements respectively.

3.4 Geophysical systems (seismic)

Φ was evaluated on seismic and strainmeter data from USGS records. Three earthquake events were analyzed: the 2011 Tōhoku M9.1 event ($\Phi = -0.357$, computed from 61 foreshocks), the 2004 Parkfield M6.0 event ($\Phi = 0.114$, from 91,556 strain measurements), and the 2003 San Simeon M6.5 event ($\Phi = 0.084$, same strainmeter). A stable control period (2010, same Donna Lea strainmeter) produced $\Phi = 0.577$, correctly classified as stable. The same sensor thus produced three distinct correct predictions across three different ground-truth outcomes.

3.5 Quantum circuits (IBM hardware)

Validation was performed on 445 qubits across 3 IBM Quantum backends (ibm_fez, ibm_torino, ibm_marrakesh). Φ-guided qubit selection produced an 85.1% error reduction compared to unselected execution and a 30.47× discrimination ratio between high-Φ and low-Φ qubits. Correlation between Φ and T_2/T_1 was $r = 0.9458$. All 5 dead qubits were detected with $\Phi < 0$. Daily collection over multiple calibration cycles confirmed that Φ tracks quality shifts under hardware drift.

3.6 Physiological signals (cardiac)

Using the MIT-BIH Arrhythmia Database [6] (48 patients, 30-minute recordings), the approach achieved AUC 0.90 for arrhythmia detection with a shuffle-test gap of +0.40, without training a classifier on cardiac data. Performance was within 0.07 AUC of the best domain-specific HRV metric (RMSSD) under the same evaluation protocol. Resolution scaling showed AUC 0.88 at 30-second windows and 0.64 at 5-second windows. In EEG/seizure settings (CHB-MIT dataset), $\alpha \approx 0.55$ was optimal, confirming that the coupling coefficient adapts by system class.

3.7 Large language models

Φ was applied to LLM outputs using external embeddings in a black-box setting (no access to logits or hidden states). Quality drift from temperature increase (0.7 → 1.5) produced $\Delta\Phi = -0.282$. Safety drift (refusal rate 14% → 3%) produced $\Delta\Phi = -0.072$. Fine-tuning drift (base vs. chat model) produced $\Delta\Phi = -0.254$. Adversarial jailbreak injection produced $\Delta\Phi = -0.207$. Temperature–Φ correlation was $r = -0.97$. Validation included models up to 2.7B parameters.

3.8 Neural network training (archival monitoring)

Φ was evaluated across 660+ neural-network training trajectories (MLPs and CNNs on MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, and Breast Cancer datasets). Kill-only monitoring achieved 99.7% precision (2 false kills out of 660+ runs), compared to 16.7% for standard early stopping. Φ

separated stable runs from collapsing runs and flagged failures that loss-only monitoring missed, particularly shifts in confusion-matrix structure that do not appear as immediate accuracy drops.

3.9 Supplementary validation: adaptive threshold framework (batteries)

The adaptive threshold framework (Eq. (4)) was additionally validated on 3 NASA lithium-ion batteries (B0005, B0006, B0018) with domain-specific $\alpha = 0.034$. All 3 batteries passed at $F1 \geq 0.5$ (range: 0.949–0.975). This validates the threshold scaling across a second physical domain (electrochemical) distinct from the mechanical systems in Section 3.1.

4 The Φ-Objective Controller

4.1 Motivation: from indicator to intervention

Monitoring is useful, but it still leaves a person (or a separate system) to decide what to do. The Φ-Objective Controller extends the framework from passive signaling to constrained intervention. The design is intentionally modular:

1. **Domain adapter** (e.g., `phi_neural.py`, `phi_quantum.py`): computes (I, ρ, S, Φ) from native signals and provides a domain performance scalar.

2. **Control engine** (`objective_controller.py`): evaluates candidate actions using the surrogate, applies safety gates, and selects the best permitted action.

3. **Surrogate predictor** (`surrogate_model.json`): forecasts the effect of candidate actions (e.g., Δperf and $\Delta\Phi$) from logged experience.

The adapter–engine interface is deliberately thin: the adapter provides Φ (and optionally its components) plus a performance measure; the engine returns an action from a domain-defined action set. Deploying the controller in a new domain therefore requires (i) an adapter and (ii) an action set. The engine and its gating logic are unchanged.

4.2 Controller objective

At decision time t, the controller selects an action that maximizes expected task performance while penalizing predicted violations of a Φ stability boundary:

$$a_t^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^k \, \mathrm{perf}_{t+k+1} \; - \; M \cdot \mathbf{1}(\Phi_{t+k+1} < \Phi_c - \mathrm{tol})\right], \tag{5}$$

where $\mathcal{A}$ is the domain action set, H is the planning horizon (default $H = 3$), γ is a discount factor (default 0.95), M is a large penalty (default 10.0), Φ_c is the critical boundary (default 0.25), and tol is a noise tolerance (default 0.02).
A key design choice is explicit in Eq. (5): the controller maximizes *performance*. Φ is treated as a constraint signal (implemented via penalty and gates), not as the quantity to optimize.

4.3 Why Φ is used as a constraint

Goodhart's Law is the practical concern: when a proxy becomes a target, it can be gamed. In our setting, optimizing Φ directly can select actions that improve the proxy while harming the real objective. In neural training, for example, an action can increase coherence by suppressing learning (stabilizing the trajectory) or reduce entropy in a way that looks "clean" in the confusion matrix while accuracy falls.

This failure mode was observed during early controller trials: maximizing Φ produced cases where Φ rose while accuracy fell (e.g., seed 2 reached $\Phi = 0.294$ with accuracy 75.8% versus a baseline $\Phi = 0.231$ with accuracy 83.5%). The corrected formulation keeps Φ as a safety boundary: the controller pursues the best attainable performance *subject to* remaining in a stable region.

4.4 Proxy-misalignment safeguards

Beyond the performance-first objective, the controller implements three guard layers:

1. **Performance floor.** Candidate actions must satisfy $\text{perf}_{\text{pred}} \geq \text{baseline} - \varepsilon$ over the planning horizon (default $\varepsilon = 0.02$).
2. **Step-wise rejection.** If an action is predicted to increase Φ on the next step while decreasing performance, it is rejected immediately (proxy improves, reality worsens).
3. **History-based correlation monitor.** The controller tracks correlation between recent Φ values and performance values. If correlation drops below a minimum threshold, the controller becomes conservative and refuses actions that improve Φ while degrading performance on average.

If all candidates are rejected, the controller falls back to `no_op`. This "do no harm" default is intentional.

4.5 Surrogate predictor

The surrogate gives the engine a way to compare candidate interventions before applying them. The current implementation uses per-action Ridge regression: one linear model per action trained on logged $(\text{state}, \text{action}, \text{outcome})$ tuples. Inputs may include current Φ, current performance, current control parameters, and normalized time (epoch/cycle fraction). The surrogate predicts single-step deltas (e.g., Δperf and $\Delta\Phi$), which are rolled forward over the horizon with damping.

The surrogate is intentionally replaceable. What is fixed here is the safety structure that keeps proxy optimization from becoming the real objective.

Before deployment, the surrogate must pass three basic quality gates:

1. **Action differentiation:** predictions differ across actions for the same input state.
2. **Beats a naive baseline:** improves on a "predict no change" predictor on held-out data.
3. **Non-degeneracy:** each per-action model has non-trivial coefficients and non-zero prediction variance.

Table 2: Phase 1: mean test accuracy across 5 seeds.

Policy	Mean accuracy	Delta vs. baseline	Outcome
Standard training (baseline)	83.68%	—	Baseline
Early stopping (patience=5)	81.68%	−2.00%	Failed on some seeds
Φ kill-only (monitoring)	83.68%	+0.00%	Matched 5/5
Controller (monitor mode)	83.68%	+0.00%	No harm

5 Controller Validation on Neural Network Training

5.1 Experimental setup

Neural network training was used as the first controlled testbed for the controller because it supports rapid iteration and repeated runs under fixed seeds. The reported configuration is:

- Dataset: CIFAR-10 with 10% label noise (injected to induce realistic instability)
- Architecture: ResNet-18
- Seeds: 5 for Phase 1; expanded to 20 for Phase 2D/2E (determinism controls configured; `num_workers=0`)
- Epochs: 30
- Hardware: Google Colab T4 GPU
- Reproducibility: experiment plans pre-committed with SHA-256 hashes; outputs audit-logged

5.2 Phase 1: monitoring and kill-only

Four policies were evaluated across 5 seeds:

1. **Standard training (baseline).**
2. **Early stopping (patience=5).**
3. **Φ kill-only.** Compute Φ each epoch; terminate only if Φ drops below threshold *and* accuracy falls below an epoch-indexed baseline for W consecutive epochs; no hyperparameter modifications.
4. **Controller in monitor mode.** Full controller stack active but actions locked to `no_op` (surrogate disabled).

The kill-only policy matched baseline across all seeds and produced no false kills. Early stopping failed on seed 5: it terminated at epoch 18 with 74.13% accuracy, while baseline training reached 84.12% by completion (a 10-point gap). The monitor-mode controller demonstrates that adding measurement, gating, and audit logic does not change outcomes when interventions are disabled.

Table 3: Phase 2D v1 (20 seeds): controller summary under strict gating and budget cap.

Metric	**Result**
Mean delta vs. baseline	+0.06 percentage points
Win rate (controller > baseline)	12/20 (60%)
Kill-only match to baseline	20/20
Early stopping failures (same setting)	7/20 (35%)
Intervention budget	2 actions per run
Observed intervention timing (budget=2 actions per run)	epochs 21–22 (per action logs)

Table 4: Phase 2E v2 (20 seeds, trained and evaluated on T4): trajectory-surrogate controller summary.

Metric	**Result**
Mean delta vs. baseline	+0.024 percentage points
Win rate (controller > baseline)	13/20 (65%)
Loss rate	7/20
Biggest win	+0.28 percentage points
Biggest loss	−0.27 percentage points
Kill-only match to baseline	20/20
Intervention budget	2 actions per run
Action set	`lr_down` (0.95×), `no_op`

5.3 Phase 2: surrogate-enabled active control

Phase 2 adds intervention selection using the surrogate. Surrogate training data were collected by structured action cycling after warmup (balanced coverage of `lr_up`, `lr_down`, `wd_up`, `wd_down`, and `no_op`), with periodic resets to prevent uncontrolled hyperparameter drift.

The surrogate passed all three deployment gates (action differentiation, improvement over naive baseline, non-degeneracy). The controller was then evaluated under strict gating and an intervention budget cap.

In these runs, the controller is intentionally conservative. When the surrogate signal is weak or gates are not satisfied, it defaults to `no_op`. The practical point of this phase is not “always beat baseline”; it is “take real actions without breaking runs,” while producing measurable aggregate benefit under strict constraints.

5.4 Phase 2E: trajectory-surrogate active control (v2)

Phase 2E evaluates a trajectory-based surrogate (12 features; trained on 480 rows from 20 seeds) with a simplified two-action plan (`lr_down` at 0.95× and `no_op`). Unlike the v1 behavior (which tended to choose identical action schedules across seeds), the trajectory features enable seed-specific intervention timing under the same strict safety gates and a two-intervention budget.

A cross-hardware test (evaluating on L4 with a surrogate trained on T4) produced 8/20 wins (40%) and a mean delta of −0.024 percentage points, indicating that surrogate performance is sensitive to hardware-specific dynamics and motivating hardware-matched data collection for deployment.

5.5 What the neural validation establishes

Phase 1 shows that Φ-based monitoring and Φ kill-only operation can match baseline outcomes while avoiding severe early-stopping failures on certain seeds. Phase 2 shows that surrogate-guided interventions can be applied under strict gating and a two-intervention budget without introducing catastrophic regressions; quantitative results are summarized in Tables 3 and 4. The safety design (performance floor, anti-Goodhart [4] rejection, correlation monitoring, and conservative fallback) behaves as intended under deterministic evaluation.

6 Architecture: Adapters, Engine, and Audit

6.1 Separation of concerns

The implementation is organized as three separable components:

Raw domain data → **Adapter** → $(I, \rho, S, \Phi, \mathrm{perf})$
↓
Control engine → **Action decision**
↑
Surrogate predictor

The adapter is domain-specific. The engine and its gates are domain-agnostic. The surrogate can be swapped or upgraded without changing the engine interface.

6.2 Swapping domains

To deploy in a new setting, two items are required:

1. A domain adapter that maps native signals to (I, ρ, S) and computes Φ.
2. An action set appropriate to the domain (e.g., learning-rate adjustments; qubit selection; maintenance actions).

The control engine, gating logic, and audit structure do not change.

6.3 Audit and reproducibility

Each controller decision can be recorded with: the input state, the candidate actions, surrogate predictions, which gates passed or failed, the selected action, and a hash chain linking decisions to pre-committed plans. This supports post-hoc review and is intended for environments where traceability matters.

7 Extended Applications of the Φ Framework

The core metric and controller described in the preceding sections have been applied to several additional problem settings, each covered by a separate provisional patent filing. This section summarizes the method and key empirical result for each application. Full validation details are available in the cited repositories.

7.1 Transfer learning prediction

The zero-shot confusion matrix produced by evaluating a source-trained model on target data—without any fine-tuning—contains a predictive signal for transfer learning success. Specifically, the diagonal strength of this matrix (equivalent to zero-shot accuracy) predicts (i) *whether* transfer will help or hurt (binary prediction) and (ii) *how much* benefit transfer provides under data degradation (magnitude prediction). Validation comprised 247 image-domain tests (8 architectures spanning a 24× parameter range, across rotations, blur, Gaussian noise, salt-and-pepper noise, and contrast reduction on MNIST and Fashion-MNIST) plus 852,607 real financial loans (Lending Club, temporal regime shift from 2013–2014 to 2015–2016). Binary prediction achieved $r = 0.445$, $p = 0.000006$ across 96 rotation tests; magnitude prediction achieved $r = -0.941$ on Gaussian noise. The financial cross-domain test was correct at $z = 13.24$, $p < 0.000001$. In the image tests, 91.7% of transfers hurt performance, validating the commercial need for pre-transfer prediction.

7.2 Identity formation detection

Behavioral identity—the confusion-matrix structure of a neural network—forms almost completely after a single training epoch (99.46% on MNIST). The remaining training time refines accuracy, not behavioral structure. This observation enables a training-efficiency predictor: the correlation between epoch-1 formation score and total training cost was $r = -0.780$ on MNIST MLPs and $r = -0.781$ on CIFAR-10 MLPs (identical to three decimal places across datasets). For CNNs, the correlation was $r = -0.987$ on MNIST and $r = -0.978$ on Fashion-MNIST at epoch 1. CNNs on CIFAR-10 at epoch 1 showed compressed formation ranges and did not produce usable correlation ($r = +0.555$), an honest negative result that identifies a measurement-timing limitation on complex datasets.

7.3 Neural phase transition detection

An identity deficit score $I_{\text{deficit}} = 1 - \sqrt{F}$, where F is the epoch-1 formation score, predicts whether an architecture will successfully form a learned representation. A universal threshold of $I_{\text{deficit}} = 0.22$ with a fixed 0.80 "formed identity" cutoff correctly classified 21 out of 22 architectures (95%) across MNIST, Fashion-MNIST, and CIFAR-10 using both MLPs and CNNs. The single false negative (a 2-layer CNN on MNIST) fell within a defined caution zone (0.22–0.40). This provides a one-epoch go/no-go decision for architecture viability.

7.4 Φ components as ML input features

Rather than using Φ as a threshold, the components (I, ρ, S) can serve as input features to standard ML classifiers. On 445 qubits across 3 IBM backends, this approach achieved 98.4% balanced accuracy on cross-backend transfer (train on one backend, test on another) using only (I, ρ, S) as features—with Φ removed to avoid circularity. All four model types tested (Random Forest, Gradient Boosting, Neural Network, SVM) achieved $> 80\%$ balanced accuracy. Feature importance analysis identified ρ (T_2/T_1) as the dominant predictor at 70–78%.

7.5 Quantum-classical hybrid resource allocation

Φ enables dynamic routing of computations between quantum hardware and classical simulation. When the minimum Φ across selected qubits falls below $\Phi_c = 0.25$, the circuit is routed to classical simulation (0% error) rather than executed on degraded quantum hardware (up to 91.60% error).

Validation covered 23 tests across 7 quantum algorithms (GHZ, QFT, Grover, Deutsch-Jozsa, Simon's, Bernstein-Vazirani, QPE) on 3 IBM backends. The maximum discrimination ratio was 30.47× (Bernstein-Vazirani under strict physical qubit binding). Error scaled with circuit size (3 qubits at ∼3% error to 15 qubits at 94.85%) and depth (3 layers at ∼3% to 103 layers at 67%).

7.6 Real-time quantum circuit intervention

The controller was extended to quantum execution with five intervention actions: checkpoint, migrate, classical fallback, restart, and continue degraded. Ten paid-circuit tests on IBM hardware validated the full intervention stack. HIGH-Φ qubit selection achieved 85.1% error reduction over LOW-Φ selection (2.73% vs. 18.36% error). Φ-based selection outperformed raw T_2 selection by 68.7%. A mid-circuit sentinel using IBM dynamic circuits achieved 98.78% conditional consistency across 4,096 shots, with an estimated 101,220 gates avoided via the abort path. Statistical abort decisions used Wilson confidence intervals ($\alpha = 0.05$) with no hardcoded fidelity thresholds. Honest counterexamples were documented in which random or T_2-based selection outperformed Φ on specific execution intervals.

7.7 Universal stability engineering

An integrated framework combining inverse design (pre-construction stability constraints), closed-loop control (real-time monitoring with hierarchical intervention), and universal monitoring (single platform across domains) was validated on 42 systems across 9 domains: mechanical bearings, aerospace (turbofans), geophysical (seismic), AI/vision, electrical (grids), NLP, medical, audio, and financial. All 42 systems were correctly classified (100% accuracy). Closed-loop bearing monitoring achieved an average advance warning of 86.1% of bearing lifetime (minimum 73.2%). Inverse design correctly identified stability-constraint violations in all 11 tested systems. The financial domain used 2.26 million real Lending Club loans; the NLP domain used 20 Newsgroups (1,967 documents); the medical domain used Wisconsin Breast Cancer (569 patients); the audio domain used Free Spoken Digit (3,000 recordings). Four real catastrophic events were correctly classified: the 2011 Tōhoku M9.1 earthquake, the 2019 UK power blackout, and the 2004 Parkfield and 2003 San Simeon earthquakes.

8 Discussion

The results presented here support three claims. Prior work established the metric's cross-domain separation properties [1] and quantum hardware validation [2]. First, a single fixed-form score with default parameterization ($\alpha = 0.1$, $\Phi_c = 0.25$), with system-class adjustments to α where explicitly noted, separates stable from failing systems across mechanically, electrically, geophysically, computationally, and biologically distinct domains. The absence of per-domain tuning distinguishes this approach from ensemble or learned-threshold methods that require labeled failure data. Second, the same score serves as a meaningful control signal: the trajectory-aware controller matched baseline accuracy (83.68%) without degradation while providing an intervention mechanism absent from standard training. Third, the extended applications (Section 7) demonstrate that the Φ components generalize beyond monitoring to prediction tasks (transfer learning, training efficiency, phase transitions) and to closed-loop control in quantum and industrial settings.

Several limitations should be noted. The controller's Phase 2 improvements are intentionally conservative under strict safety constraints and a two-intervention budget. Phase 2D achieved a +0.06 percentage point mean delta with a 12/20 (60%) win rate, and Phase 2E (trajectory surrogate) achieved +0.024 percentage points with a 13/20 (65%) win rate across twenty deterministic seeds on T4. While these effect sizes are modest, the results establish that the controller can take real actions without introducing catastrophic regressions under the tested constraints. A cross-hardware test on L4 without retraining degraded to 8/20 wins and a -0.024 mean delta, suggesting the surrogate benefits from hardware-matched training data. CNN formation detection did not produce usable correlations on CIFAR-10 at epoch 1, identifying a measurement-timing limitation on complex datasets. In EEG/seizure settings (CHB-MIT), $\alpha \approx 0.55$ rather than the default 0.1 was optimal, indicating that while the functional form is universal, the coupling coefficient may benefit from system-class adaptation in certain physiological settings. Honest counterexamples in quantum experiments showed that Φ-guided selection does not dominate raw T_2 selection on every execution interval, though it wins in aggregate.

The connection between $\Phi_c = 0.25$ and the Bekenstein-Hawking entropy relation $S = A/4$ [3, 5] is noted as an empirical observation, not a derived result. Whether this numerical coincidence reflects a deeper principle remains an open question.

Future work includes scaling the controller to larger models and longer training runs, improving the surrogate predictor (Appendix D), extending LLM monitoring to models beyond 2.7B parameters, and investigating the theoretical basis for the observed threshold universality.

9 Conclusion

We introduced $\Phi = I \times \rho - \alpha \times S$, a domain-agnostic stability metric, and demonstrated its application across eight validated domains (mechanical, aerospace, electrical, geophysical, neural network, quantum, physiological, and LLM) with default parameterization ($\alpha = 0.1$, $\Phi_c = 0.25$) and system-class adjustments where noted. Across 28 systems in the thermodynamic validation alone, the metric achieved 100% accuracy with zero false positives and zero false negatives. The Φ-objective controller provides a framework for autonomous intervention with anti-Goodhart safeguards. Seven additional applications—transfer learning prediction, identity formation detection, neural phase transition detection, Φ-as-ML-features, quantum-classical hybrid routing, real-time quantum intervention, and universal stability engineering—extend the framework to 14 provisional patent filings covering prediction, monitoring, and control across physical, computational, and biological systems. All validation used real data from published datasets and real hardware, with negative results reported alongside positive results.

A Validation Standards

Unless otherwise stated, reported experiments follow these standards:

1. Real published datasets (or real hardware measurements) for performance claims.
2. SHA-256 hashes for data, scripts, and outputs.
3. Pre-committed experiment plans before execution.
4. Pre/post audit records with timestamps.
5. Shuffle tests where applicable to confirm signal is not an ordering artifact.
6. Reproducible runs from committed code and fixed seeds.
7. Train/test separation with no data leakage in any learned surrogate.

8. Proxy-misalignment checks applied during evaluation.
9. Reporting includes negative results alongside positive results.
10. Baselines, seeds, and metrics fixed prior to comparative runs.

B Controller Parameters

Table 5: Controller parameters (defaults).

Parameter	**Default**	**Meaning**
H	3	Planning horizon
γ	0.95	Discount factor
M	10.0	Penalty for predicted Φ violation
Φ_c	0.25	Critical stability boundary
tol	0.02	Threshold tolerance
ε	0.02	Performance floor margin
α	0.1	Entropy coupling (unless stated otherwise)
Warmup	5 epochs	Minimum observations before control
W	3	Consecutive-epoch kill streak requirement

C Phase 1 Detailed Results

Table 6: Per-seed accuracy for Phase 1 policies.

Seed	**Standard**	**Early stop**	**Φ kill-only**	**Controller (monitor)**
1	83.38%	83.38%	83.38%	83.38%
2	83.54%	83.54%	83.54%	83.54%
3	84.08%	84.08%	84.08%	84.08%
4	83.28%	83.28%	83.28%	83.28%
5	84.12%	74.13%	84.12%	84.12%
Mean	**83.68%**	**81.68%**	**83.68%**	**83.68%**

Early stopping terminated seed 5 at epoch 18 with 74.13% accuracy. The baseline trajectory reached 84.12% by completion under the same seed.

D Surrogate Improvement Roadmap

The following surrogate upgrades are planned, in priority order: Phase 2E incorporated trajectory features (12-feature surrogate) and validated them in active control; remaining upgrades include:

1. **Uncertainty estimation:** bootstrap ensembles per action; penalize uncertainty (e.g., $\text{score}_{\text{adj}} = \text{mean} - \beta\,\text{std}$).

2. **Pairwise advantage prediction:** predict action advantage over `no_op` instead of absolute outcomes.

3. **Finer action granularity:** add gentler actions (e.g., `lr_down_small` ×0.97) alongside larger steps.

4. **Probe-then-commit:** run 1-epoch probes for top candidates plus `no_op`, then commit based on measured outcomes.

5. **Cross-hardware robustness:** train on multiple GPU types/backends and/or include a hardware identifier feature to reduce transfer loss.

Intellectual Property Notice

The methods, systems, and formulations described in this paper are the subject of fourteen provisional patent applications filed with the United States Patent and Trademark Office (USPTO) between October 2025 and February 2026. This publication is intended as a defensive disclosure establishing priority for the described inventions. No license to use, implement, or commercialize the described methods is granted by this publication. All rights reserved.

References

[1] Shawn Barnicle. A stability index for cross-domain degradation detection. 2025. 31 systems, 6 domains, 100% separation.

[2] Shawn Barnicle. Thermodynamic stability metric provides early warning of qubit degradation on IBM quantum hardware. 2025. 445 qubits, 3 backends, 100% detection.

[3] Jacob D. Bekenstein. Black holes and entropy. *Physical Review D*, 7(8):2333–2346, 1973.

[4] Charles A. E. Goodhart. Problems of monetary management: The U.K. experience. *Monetary Theory and Practice*, pages 91–121, 1984.

[5] Stephen W. Hawking. Particle creation by black holes. *Communications in Mathematical Physics*, 43(3):199–220, 1975.

[6] George B. Moody and Roger G. Mark. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[7] George N. Saridis. *Stochastic Processes, Estimation, and Control: The Entropy Approach.* Wiley, 1995.

[8] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management*, 2008. C-MAPSS turbofan dataset.

[9] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[10] Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1):401–412, 2020. XJTU-SY bearing dataset.