

CASE-ID: Constraint-Aware State Estimation and Instability Detection

Craig Kyrle Strachan Davidson
Independent Researcher, Petit-Mars, France
ORCID: [0009-0007-2990-8716](https://orcid.org/0009-0007-2990-8716)
cksd.nantes@gmail.com

Abstract

English. Deep learning systems lack reliable early-warning indicators for instability during training and deployment. Standard metrics such as loss, gradient norms, and weight statistics react only after degradation has begun. This paper introduces CASE-ID (Constraint-Aware State Estimation and Instability Detection), a lightweight framework that models neural networks as latent stochastic dynamical systems and detects structural shifts in representation space over time before performance collapses. CASE-ID defines a latent state S_t , observed activations X_t , and a representation state Z_t whose distributional evolution is tracked through a Gaussian approximation. Instability is quantified using a KL-based drift measure combined with constraint-aware penalties that capture violations of expected representation geometry. Experiments on CIFAR-100 with ResNet-50 show that CASE-ID provides early warnings 120–180 steps before loss-based triggers (median ≈ 150 steps) and reduces false positive rate by approximately 25–40% relative to gradient-norm heuristics. The method is architecture-agnostic, computationally inexpensive ($< 2\%$ overhead), and requires no modification to model weights or training dynamics. CASE-ID complements standard metrics by providing an earlier, representation-level indicator of instability that is not observable in loss or gradient statistics.

Abstract

Français. Les systèmes d'apprentissage profond ne disposent pas de mécanismes fiables permettant d'anticiper les instabilités, que ce soit durant l'entraînement ou lors du déploiement. Les indicateurs usuels réagissent tardivement. Ce travail présente CASE-ID, un cadre léger fondé sur la modélisation des réseaux neuronaux comme systèmes dynamiques stochastiques latents. Le modèle introduit un état latent S_t , des activations observées X_t et un état de représentation Z_t dont l'évolution distributionnelle est suivie via une approximation gaussienne. Les expériences menées sur CIFAR-100 avec ResNet-50 montrent que CASE-ID fournit des signaux d'alerte 120 à 180 itérations avant les déclencheurs basés sur la perte (médiane ≈ 150) et réduit le taux de faux positifs d'environ 25 à 40%. Le cadre est indépendant de l'architecture, peu coûteux ($< 2\%$), et ne nécessite aucune modification des poids ou de la dynamique d'entraînement.

1 Introduction

Neural networks operate in regimes where instability can emerge abruptly: distribution shift, overfitting, catastrophic forgetting, gradient explosion, or internal representation collapse. Existing monitoring tools detect these events only after they manifest in performance metrics. A

proactive approach requires estimating the internal state of the model and detecting structural deviations before they propagate. CASE-ID provides such an early-warning mechanism. It models the network as a latent stochastic dynamical system and monitors the evolution of internal representations through compact statistical descriptors. Instability is defined as a measurable deviation from expected representation dynamics. **Contributions.**

- A latent-state formulation for neural network observability.
- A Gaussian representation-state estimator based on activation manifolds.
- A KL-based instability signal with constraint-aware geometric penalties.
- A statistically calibrated, persistence-based detection rule.
- A lightweight implementation suitable for real-time monitoring ($< 2\%$ overhead).
- Empirical evidence demonstrating substantial lead-time over loss-based metrics.

CASE-ID complements standard metrics by providing an earlier, representation-level indicator of instability that is not observable in loss or gradient statistics.

2 Background and Motivation

Neural networks exhibit structured internal dynamics: activations cluster by class, layer-wise covariance evolves smoothly, and representation geometry stabilizes as training converges. Instability disrupts these patterns. Monitoring these structures provides a richer signal than loss alone. Let X_t denote activations at time t . A representation state Z_t is defined as a compact summary of X_t , typically:

- mean vector μ_t
- covariance matrix Σ_t (or a diagonal/low-rank approximation)
- optional low-rank projections

These quantities evolve over time and can be modeled as a stochastic process. By treating the network as a dynamical system, we can apply principles from control theory to observe "state drift" before "system failure."

3 Mathematical Formulation

3.1 Latent Dynamical System

The network is modeled as:

$$S_{t+1} = f_\theta(S_t) + \epsilon_t$$

where f_θ is an unknown nonlinear transition and ϵ_t represents stochastic perturbations. The representation state is approximated as a Gaussian distribution:

$$Z_t \sim \mathcal{N}(\mu_t, \Sigma_t)$$

Instability is operationally defined as a **sustained increase** in an instability score I_t relative to a calibrated baseline, rather than a single threshold crossing.

3.2 KL Divergence

Instability is defined as drift in representation space, quantified via the Kullback-Leibler (KL) Divergence:

$$D_t = \text{KL}(\mathcal{N}(\mu_t, \Sigma_t) \parallel \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}))$$

For Gaussian distributions, this has a closed form:

$$D_t = \frac{1}{2} \left[\text{tr}(\Sigma_{t-1}^{-1} \Sigma_t) + (\mu_t - \mu_{t-1})^T \Sigma_{t-1}^{-1} (\mu_t - \mu_{t-1}) - k + \ln \left(\frac{\det \Sigma_{t-1}}{\det \Sigma_t} \right) \right]$$

Interpretation. The trace term captures covariance inflation, the quadratic term captures mean displacement (centroid drift), and the log-determinant term penalizes volume contraction (representation collapse).

3.3 Constraint Penalty

A geometric penalty captures structural deformation that might be under-weighted by pure probabilistic measures:

$$C_t = \alpha \|\mu_t - \mu_{t-1}\|^2 + \beta \|\Sigma_t - \Sigma_{t-1}\|_F^2$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Coefficients α and β normalize the contributions of mean and covariance terms to comparable scales based on initial training statistics. The final instability score is:

$$I_t = D_t + C_t$$

Both terms are commensurate after normalization and represent complementary views of distributional change: probabilistic divergence (KL) and first-order geometric deformation.

4 Unified CASE-ID Framework

4.1 Training Regime and Calibration

A threshold τ is calibrated statistically during the "stable" phase of early training. During a stable window $t \in [t_0, t_1]$, compute:

$$\mu_I = \mathbb{E}[I_t], \quad \sigma_I = \text{Std}[I_t]$$

and set:

$$\tau = \mu_I + k\sigma_I$$

To ensure comparability across layers and training regimes, I_t is normalized by σ_I . A persistence parameter m is used to suppress transient fluctuations (e.g., stochastic batch noise):

Flag instability only if $I_t > \tau$ for m consecutive steps.

4.2 Constraint Envelope

Expected representation geometry is encoded as:

- smooth covariance evolution (no abrupt rank collapse)

- stable class-conditional clusters
- bounded drift in the latent manifold

These constraints are operationalized through penalty terms applied to deviations in mean and covariance dynamics.

5 Implementation

5.1 Hooks and Monitoring

Forward hooks are used to extract activations from selected layers (e.g., the final residual block). Monitoring overhead is 1–2 ms per step on ResNet-50 ($< 2\%$), making it suitable for production use. The method requires no modification to model weights or training dynamics.

5.2 Online Estimation

Means and covariances are updated using exponential moving averages (EMA) to maintain a running "view" of the state:

$$\begin{aligned}\mu_t &= (1 - \lambda)\mu_{t-1} + \lambda x_t \\ \Sigma_t &= (1 - \lambda)\Sigma_{t-1} + \lambda(x_t - \mu_t)(x_t - \mu_t)^T\end{aligned}$$

While EMA-based statistics are biased, they preserve the **relative changes** and **temporal correlations** necessary for drift detection.

6 Experiments

6.1 Setup

- **Dataset:** CIFAR-100
- **Model:** ResNet-50
- **Monitoring layers:** Final residual block
- **Baselines:** Loss-based triggers, gradient-norm heuristics, weight-change metrics

6.2 Lead-Time Definition

Lead time is defined as the number of steps between the first persistent threshold crossing of I_t and the onset of a significant loss spike ($\geq 20\%$ increase from the local rolling average).

6.3 Results

CASE-ID detects instability 120–180 steps earlier than loss-based metrics, with a median lead time of approximately 150 steps. Furthermore, the persistence-based rule reduces the false positive rate (FPR) by 25–40% compared to raw gradient-norm monitoring.

Layer Monitored	Mean Lead Time (Steps)	Standard Deviation
Final Block	152	18
Intermediate Block	131	22

Table 1: Lead time across 10 independent seeds.

7 Discussion

CASE-ID provides a compact, interpretable signal for instability. It bridges the gap between statistical monitoring and control-theoretic constraint enforcement.

- **Architecture Agnostic:** In transformers, Z_t can be defined over attention outputs or token embeddings.
- **Computational Efficiency:** The use of diagonal covariance approximations allows for scaling to very wide layers without the $O(k^3)$ cost of matrix inversion.

CASE-ID acts as a **minimal observability operator** over representation manifolds, providing the "smoke detector" necessary for safe and efficient deep learning.

8 Conclusion

CASE-ID offers a practical, mathematically grounded approach to early instability detection. By modeling representation dynamics and quantifying structural drift, it provides actionable signals that enable intervention—such as early stopping or learning rate adjustment—before instability propagates to observable failure.

Funding

This research received no external funding.

References

- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition.
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning.
- Tishby, N., & Zaslavsky, N. (2015). Deep Learning and the Information Bottleneck Principle.

A Technical Implementation Details

A.1 Numerical Stability

To ensure the invertibility of Σ_t in the KL term, a small Tikhonov regularization term δI is added:

$$\Sigma_t \leftarrow \Sigma_t + \delta I$$

A.2 Covariance Approximation

For high-dimensional layers, we recommend the diagonal approximation ($O(k)$ complexity) or a low-rank Factor Analysis approach ($O(kr)$ complexity) to maintain the $< 2\%$ computational overhead target.