LABEL NOISE FORGOTTEN IN LOSS PERSISTS IN COMPLEXITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Training loss can decline as a network memorizes incorrect labels. What remains observable about label noise after its initial loss response subsides? We identify an asymmetry in training trajectories: after controlled label-noise injections, loss returns toward its pre-injection course, while neural persistence, a statistic of normalized network weights, remains elevated through the recorded horizon. Matched clean additions do not produce the same sustained elevation. Separately, clean and noisy per-sample losses overlap strongly on real-world image data with injected label noise, despite clear separation on a simple synthetic dataset. We study a geometric mechanism for the complexity cost of fitting noise: maintaining a fixed positive output separation across nearby conflicting inputs requires increasing functional complexity. Explicit calibration and trajectory conditions connect this barrier to measured persistence and label recovery. We use the persistent signal in our Concept Recovery Test (CRT), combining classwise label-fit weights with a global complexity-level gate to aggregate training predictions. We prove finite-time recovery when the gate retains enough clean predictions, accounting for early errors, reference estimation, and the memorized tail. Across five benchmarks and four comparison methods, gated CRT improves rectification and detection over both the underlying method and loss-only CRT, and remains effective as oracle-tuned single-epoch loss thresholds deteriorate. These results show how a sustained structural response can complement loss-based evidence for label recovery.

1 INTRODUCTION

A network can fit an incorrect label well enough to assign it a small training loss. Declining loss can therefore accompany either learning class structure or memorizing annotation errors. Many approaches to label noise exploit an earlier stage, when clean examples tend to have smaller losses or more favorable prediction margins than corrupted examples (Jiang et al., 2018; Han et al., 2018; Arazo et al., 2019; Liu et al., 2020; Pleiss et al., 2020; Li et al., 2020). We study what remains observable about label noise as its initial loss response subsides, focusing on the structure of the trained model.

The empirical asymmetry: After controlled label-noise injections, the training-loss trajectory rises and then returns toward its pre-injection course. Neural persistence (Rieck et al., 2019), a topological statistic of normalized network weights, instead remains elevated through the recorded horizon (Figure 1). This elevation exceeds that produced by adding the same number of clean examples. Here, "forgotten in loss" describes the fading aggregate loss response while the corrupted labels remain in the training data. The corresponding structural response persists throughout the observed continuation of training. A separate comparison finds strongly overlapping clean and noisy per-sample losses on real-world image data with injected label noise, despite clear separation on a simple synthetic dataset. Section 2 presents these observations and their controls before the theoretical analysis.

The cost of fitting local conflicts: Consider a mislabeled example near correctly labeled examples of its true class in a fixed feature space. Confidently fitting the conflicting labels requires the predictor to vary over a short distance. We formalize this requirement as a *complexity barrier*: under explicit regularity conditions, maintaining a positive output separation requires complexity that diverges as the conflicting inputs approach each other. Hölder seminorms give an explicit inverse-distance rate. This geometric cost persists after fitting when functional complexity is nondecreasing along

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

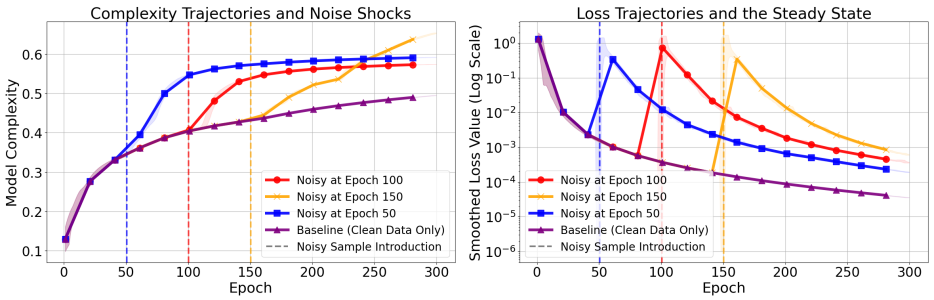


Figure 1: **Transient loss response and persistent structural change.** Dashed vertical lines mark label-noise injections. In each pair, training loss (right axis) returns toward its pre-injection course, whereas neural persistence remains elevated through the recorded horizon. Neural persistence measures the structure of normalized network weights; both traces follow the same training intervention.

the trajectory. Relating it to neural persistence additionally requires calibration, because persistence describes network parameters and can change under function-preserving reparameterization. The analysis distinguishes the geometric cost, its temporal persistence, and its measurement.

Using the persistent signal: The observation suggests weighting predictions according to when the model has learned class structure and when its measured complexity has risen. Our *Concept Recovery Test* (CRT) recovers labels by aggregating predictions across training. Its loss-only variant weights each prediction by the observed-label fit of its predicted class. Complexity-gated CRT adds a global level gate that decreases above an estimated persistence reference. The two factors downweight early inaccurate predictions and later memorization, respectively. Our recovery theorem applies to samples whose useful clean predictions receive enough weight under this shared gate; a two-mode model supplies one regime with the required temporal ordering.

The paper makes three contributions. First, controlled interventions establish a transient loss response and a sustained neural-persistence elevation, with matched clean additions and per-sample loss comparisons clarifying their interpretation. Second, a functional complexity barrier and explicit calibration and trajectory conditions characterize when fitting local conflicts can support a persistent measured signal and a label-recovery guarantee. Third, experiments across five benchmarks and four comparison methods demonstrate the utility of that signal in CRT, isolate the two gates' roles, and compare their predictions with oracle-tuned single-epoch loss thresholds.

2 Transient Loss and Persistent Complexity

We first establish the trajectory asymmetry motivating the paper. The injection studies use a three-layer ReLU multilayer perceptron (MLP) on MNIST; implementation details and supporting diagnostics appear in Sections C.1 and C.2. We measure structural complexity using neural persistence (Rieck et al., 2019). For an MLP, adjacent layers form a bipartite graph whose edge magnitudes are divided by the largest absolute weight in the network. Adding edges from largest to smallest records when connected components merge. Neural persistence aggregates the resulting lifetimes into normalized layer scores, then averages across layers. It summarizes the distribution and connectivity of weights within the fixed architecture; Section B.5 gives the formal definition. Its relation to functional complexity is developed in Section 3.

2.1 Loss and Persistence after Noise Injection

Starting from clean MNIST training, we add 20% noisy samples at epochs 50, 100, or 150. Loss rises after each injection and then declines toward its previous course, while persistence remains elevated through the recorded horizon (Figure 1). Adding the same proportion of clean samples at epoch 100 produces a smaller loss response without the sustained persistence increase (Figure 2). The clean control separates the noise response from an effect of increasing the training-set size.

Matched additions to a base set of $20{,}000$ samples also yield greater persistence at convergence for noisy than for clean additions. At fixed noise fraction, the excess over a matched clean control

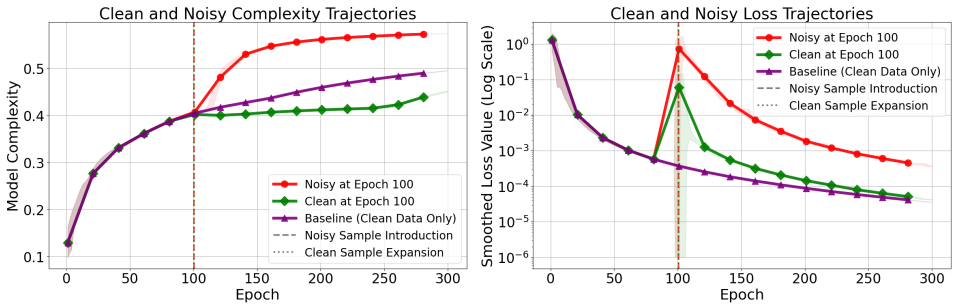


Figure 2: **Matched clean and noisy additions on MNIST.** At epoch 100 (vertical lines), 20% clean (green) or noisy (red) samples are added; purple denotes clean-only training. Left: neural persistence. Right: smoothed loss on a logarithmic scale. Both additions perturb loss, but only noisy additions produce a sustained persistence increase over the clean-only reference.

increases with dataset size. These diagnostics characterize the measured response to noise. Section 3 develops the functional barrier and the calibration and output-margin conditions needed to connect it to this response. Details appear in Section C.2. At low noise rates with early stopping, the ordering can reverse (Section C.4).

2.2 Per-Sample Loss Separation

The injection study concerns aggregate loss. To examine samplewise discrimination separately, we compare loss trajectories on a two-feature synthetic dataset and MNIST under 10% symmetric noise. With the same three-layer MLP architecture, clean and noisy losses separate early on the synthetic data but overlap throughout the 500-epoch MNIST run (Figure 3). The two studies identify complementary aspects of the problem: the aggregate loss response to an intervention subsides, and per-sample loss can offer limited clean/noisy separation. The sustained structural response motivates the analysis and recovery rule that follow.

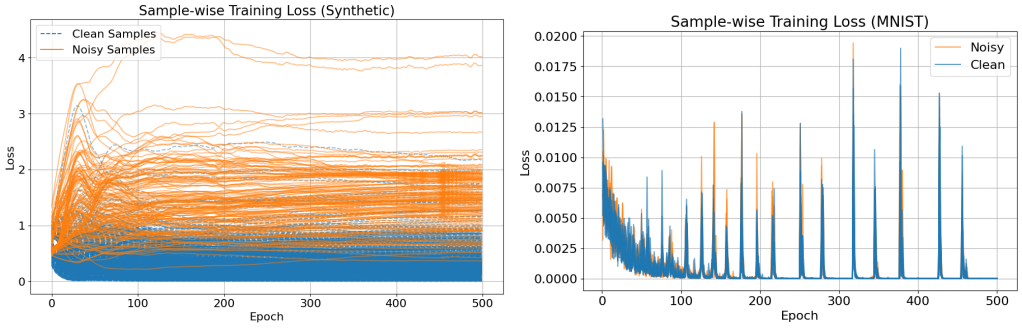


Figure 3: **Per-sample loss trajectories under 10% symmetric noise.** Clean samples are blue and noisy samples orange. Left: a synthetic dataset with two features and two classes. Right: MNIST. The same three-layer MLP architecture is monitored for 500 epochs. The synthetic populations separate early, whereas the MNIST trajectories overlap throughout the recorded run.

3 Theory: Complexity Barriers and Trajectory Recovery

The observed asymmetry raises two questions: what complexity is required to fit label conflicts, and under what conditions can that cost remain useful after the loss response subsides? We first establish a functional complexity barrier, then state the temporal and calibration conditions connecting it to a persistent measured level. Finally, we show when using that level to weight predictions recovers a clean label. A two-mode model realizes the required temporal ordering; full statements and proofs appear in Section B.

Learning setup: Let X be the input domain and Y the finite label alphabet. The clean dataset is $D^* = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^N$; the observed dataset D corrupts labels at rate η. A predictor $f : X \to \Delta(Y)$ outputs class probabilities, and $\hat{\mathbf{y}}_k(\mathbf{x})$ denotes its one-hot argmax prediction at epoch k. We define the clean concept $c(\mathbf{x})$ as the label predicted by the limiting model trained on D^* from the same initialization. This is the recovery target; it equals the ground-truth label wherever clean training predicts correctly. Section B.1 gives the full setup.

3.1 The Complexity Barrier

Consider a mislabeled input near a correctly labeled neighbor. Fitting both labels with a fixed positive output separation requires a sharper transition as their distance decreases. On a class $\mathcal{F}$ of continuous probability-valued predictors, we formalize this cost using a functional $\mathcal{C}$ that increases when the same output variations occur at smaller spatial scales. The axioms and compactness conditions in Section B.3 are satisfied by Hölder seminorms, including the Lipschitz seminorm (Section B.4). The geometric analysis uses fixed Euclidean feature coordinates $X = \mathbb{R}^m$.

Observation 1 (Complexity Barrier)**.** *Fix $\mathbf{x} \in X$, a unit direction $\mathbf{u}$, and an output-separation level $\mu > 0$. For $r > 0$, let $\mathbf{x}_r = \mathbf{x} + r\mathbf{u}$ and define $\mathcal{F}_r^{(\mu)} = \{f \in \mathcal{F} : \|f(\mathbf{x}) - f(\mathbf{x}_r)\| \geq \mu\}$ and*

$$C_\mu(r) = \inf\{\mathcal{C}(f) : f \in \mathcal{F}_r^{(\mu)}\}.$$

Then $0 < r_1 < r_2$ implies $C_\mu(r_1) \geq C_\mu(r_2)$, and $\lim_{r\to 0} C_\mu(r) = \infty$.

For Hölder complexity, any pair at distance r with output separation at least μ requires complexity at least μ/r^α (Corollary 1). Exact interpolation of distinct one-hot labels supplies a fixed μ; argmax disagreement alone does not. The abstract proof expands a fitted transition along the ray and uses compactness to rule out bounded complexity as the pair approaches (Section B.6). Consequently, a bounded-complexity class cannot fit these conflicts at arbitrarily small distances (Corollary 4).

Remark 1 (Label-consistent neighborhoods). Interpreting local conflicts as evidence of label noise assumes that clean data occupies label-consistent neighborhoods. Where true classes overlap, confident predictions for nearby clean examples can also incur the barrier, and the detector may flag ambiguous points alongside mislabeled ones. The geometry itself applies to either source of conflict.

3.2 Trajectory Conditions and the Clean-Prediction Window

The barrier describes the cost of fitting a conflict. Using this cost to recover labels also requires temporal structure. For a sample $\mathbf{x}$, suppose predictions equal $c(\mathbf{x})$ during a clean-prediction window $\mathcal{K}_c(\mathbf{x})$, before a later tail beginning at $t_0(\mathbf{x})$. Write $\mathcal{K}_c(\mathbf{x}) = \{\tau, \ldots, t^* - 1\}$, with $1 \leq \tau < t^* \leq t_0 \leq T$. A single complexity sequence and estimated reference $\widehat{C}_{\mathrm{ref}}$ are shared across samples. The *covered set* $\mathcal{R}$ consists of samples satisfying

$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \leq \epsilon_C, \qquad k \in \mathcal{K}_c(\mathbf{x}),$$
$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \geq \delta_C, \qquad k \geq t_0(\mathbf{x}),$$

for some $\epsilon_C \geq 0$ and $\delta_C > \epsilon_C$: useful epochs remain near the reference, while later epochs retain a positive excess. Assumptions 4 and 5 give the full conditions. They allow sample-specific windows; a late-learning sample may fall outside $\mathcal{R}$ if the shared gate suppresses its useful epochs.

The barrier supports the tail condition when trajectory complexity is nondecreasing (Assumption 3). If f_{t_0} separates a radial pair at distance r by at least μ, then

$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \;\geq\; C_\mu(r) - \widehat{C}_{\mathrm{ref}} \qquad \text{for every } k \geq t_0 \tag{1}$$

by Proposition 5. A positive excess therefore persists after such a fit. The finite-sample Hölder result connects this cost to conflicting-neighbor distance (Proposition 6).

A mechanism with separate timescales: The two-mode model in Section B.9 makes this temporal ordering explicit. At a mislabeled binary example with clean label $+1$ and observed label -1, its signed logit margin is $h_t = a(1 - e^{-\lambda_1 t}) + b(1 - e^{-\lambda_2 t})$, where $a > 0$, $b = -(1 + a)$, and $\lambda_1 > \lambda_2 > 0$. The clean mode is learned faster; the localized noise mode eventually supplies the negative contribution needed to fit the observed label. If $a\lambda_1 > |b|\lambda_2$, the margin is initially positive and crosses zero exactly once before converging to -1.

The modes have Lipschitz constants $0 < L_1 \ll L_2$. Under the model's separated variation sectors, the trajectory's Lipschitz complexity equals the larger of the two growing mode contributions. When $|b|L_2 > aL_1$, its positive excess over the ideal clean-mode reference aL_1 increases monotonically to $|b|L_2 - aL_1 > 0$. Before either the prediction changes sign or complexity crosses that reference, predictions are clean and receive no complexity penalty. Later fitting retains the localized variation even after the observed label is predicted correctly. This construction realizes a clean-prediction window and persistent excess under explicit spectral and spatial assumptions (Proposition 7). Practical reference quality enters through the realized separation constants, and the fitted-loss factor must be controlled separately.

Measured persistence: The functional barrier motivates our use of neural persistence, a statistic of normalized network weights (Rieck et al., 2019). Neural persistence is stable to normalized weight perturbations and invariant to hidden-unit permutations and common rescaling, although neuron-wise ReLU rescaling can change it while preserving the function. Under the within-trajectory calibration

$$aM_p(\theta_k) + b = C^{\mathrm{fun}}(f_k) + e_k, \qquad |e_k| \le \varepsilon_{\mathrm{NP}},$$

with $a > 0$, functional separation transfers to measured persistence when $\delta_C - \epsilon_C > 2(\varepsilon_{\mathrm{NP}} + \varepsilon_{\mathrm{ref}})$, where $\varepsilon_{\mathrm{ref}}$ bounds the reference-calibration error (Proposition 4). Recovery can also be applied directly to a measured sequence satisfying shared-gate separation; the calibration is needed to connect that separation to the functional barrier.

3.3 Gated Concept Recovery

For a finite training trajectory $\{f_k\}_{k=1}^T$, we weight each prediction by two factors. The **fitted-loss gate** measures how well its predicted class has been learned from observed labels:

$$z_k(\mathbf{y}) \;=\; \frac{1}{|I_{\mathbf{y}}|} \sum_{i \in I_{\mathbf{y}}} \mathbf{1}[\,\hat{\mathbf{y}}_k(\mathbf{x}_i) = \mathbf{y}_i\,], \qquad z_k(\mathbf{x}) \;=\; z_k(\hat{\mathbf{y}}_k(\mathbf{x})), \tag{2}$$

where $I_{\mathbf{y}} = \{i : \mathbf{y}_i = \mathbf{y}\}$; classes absent from the observed training set are omitted from the output alphabet, so $|I_{\mathbf{y}}| > 0$. Thus "fitted" means zero observed-label 0–1 loss, which is exactly computable from recorded predictions and does not require cross-entropy to equal zero. For the fixed sample below, abbreviate $z_k = z_k(\mathbf{x})$. Define the shared **complexity gate**

$$g_k \;=\; \exp\!\big(-\lambda\,\big(\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}}\big)_+\big), \tag{3}$$

where $\lambda \ge 0$, $(\cdot)_+ = \max(\cdot, 0)$, and $\widehat{C}_{\mathrm{ref}}$ is estimated once from the global trajectory. Set $w_k = z_k g_k$ and

$$P_T(\mathbf{y}) \;=\; \frac{\sum_{k=1}^T w_k\,\mathbf{1}[\hat{\mathbf{y}}_k = \mathbf{y}]}{\sum_{k=1}^T w_k}.$$

Theorem 2 (Gated Concept Recovery)**.** *Fix $\mathbf{x} \in \mathcal{R}$ satisfying Assumptions 4 and 5. Let $m = |\mathcal{K}_c(\mathbf{x})|$, suppose $z_k \ge z_{\mathrm{in}} > 0$ on $\mathcal{K}_c(\mathbf{x})$, and define the early incorrect mass*

$$B_{\mathrm{early}}(\mathbf{x}) = \sum_{\substack{k < t_0(\mathbf{x}) \\ \hat{\mathbf{y}}_k \neq c(\mathbf{x})}} w_k.$$

Then, with $n_{\mathrm{tail}} = (T - t_0(\mathbf{x}) + 1)_+$,

$$1 - P_T(c(\mathbf{x})) \;\le\; \frac{B_{\mathrm{early}}(\mathbf{x}) + n_{\mathrm{tail}} e^{-\lambda\delta_C}}{m z_{\mathrm{in}} e^{-\lambda\epsilon_C} + B_{\mathrm{early}}(\mathbf{x}) + n_{\mathrm{tail}} e^{-\lambda\delta_C}}. \tag{4}$$

In particular, the argmax aggregate returns $c(\mathbf{x})$ whenever

$$m z_{\mathrm{in}} e^{-\lambda\epsilon_C} > B_{\mathrm{early}}(\mathbf{x}) + n_{\mathrm{tail}} e^{-\lambda\delta_C}.$$

The bound compares a lower bound on useful clean-vote mass with upper bounds on early incorrect mass and tail mass. Recovery follows when the clean mass exceeds their sum. It is deterministic conditional on the recorded trajectory and reference; Proposition 8 relates z_{in} to class fit, noise rate, and sample size.

Dependence on the training horizon: For a mislabeled sample whose tail predictions remain at the observed label and whose fitted-loss weights tend to one, loss-only aggregation eventually returns that observed label. With both gates, Corollary 5 gives a sufficient finite-horizon condition: the sharpness needed to suppress tail mass grows logarithmically with tail length, provided the same sharpness keeps early incorrect mass below the retained clean mass. The result compares the two CRT variants defined here and requires a feasible sharpness at the chosen horizon; one fixed gate is not guaranteed for arbitrarily long training.

4 METHOD: COMPLEXITY-GATED CONCEPT RECOVERY TEST

The sustained response in Section 2 motivates using complexity level to select informative training epochs. Our **Concept Recovery Test (CRT)** aggregates their weighted predictions to infer a label, then flags disagreement with the observed label. Complexity-gated CRT applies this rule to the recorded predictive trajectory used to evaluate a base detector, which we call the *substrate*. The augmentation requires an underlying model trajectory with predictions and weights available; CRT supplies the aggregation and label decision. The loss-only CRT variant uses $w_k = z_k$, while the full method multiplies this fitted-loss weight by the complexity gate.

1. **Train and record.** Configure the underlying predictive model according to the experimental protocol in Section C.1 to obtain $\{f_k\}_{k=1}^{T}$. At each epoch, record sample predictions, the classwise fitted fractions in Equation (2), and one neural-persistence score $M_k = M_p(\theta_k)$ for the whole model.
2. **Estimate the reference.** Compute $\Delta_k = M_k - M_{k-1}$, locate the earliest window of ω consecutive epochs with increments below threshold ϵ_Δ, and use its mean persistence as $\widehat{M}_{\mathrm{ref}}$. By default, $\omega = 10$ and ϵ_Δ is the 25th percentile of observed increments. MLP and GNN increments are used directly; ResNet increments are first smoothed with an exponential moving average of half-life approximately 5 epochs. This uses the observed trajectory; the low-noise calibration used in the most localized sweep setting is described in Section 5.2. Reference error enters the theory through the separation constants.
3. **Aggregate predictions.** When the median persistence excess after the detected window is positive, choose λ so that it receives complexity-gate weight 0.1. For each sample, form $w_k = z_k \exp(-\lambda(M_k - \widehat{M}_{\mathrm{ref}})_+)$ and compute
$$S(\mathbf{y}) = \sum_{k=1}^{T} w_k \, \mathbf{1}[\hat{\mathbf{y}}_k = \mathbf{y}], \qquad \hat{\mathbf{y}}^* = \arg\max_{\mathbf{y} \in Y} S(\mathbf{y}).$$
4. **Detect and rectify.** Flag the sample when $\hat{\mathbf{y}}^*$ differs from its observed label $\mathbf{y}$, and output $\hat{\mathbf{y}}^*$ as the rectified label.

Complementary roles of the gates: The fitted-loss factor downweights early predictions from classes not yet fit; the complexity factor downweights later epochs after the measured structural cost rises. Their product assigns substantial weight only when both factors do. The complexity gate is global, while predicted classes, fitted-loss weights, and votes remain sample-specific.

Implementation and scope: The gate requires low measured levels on useful epochs and a positive excess on the tail within each run. Neural persistence is our default, with alternative measures evaluated in Section 5.4. The rate-gated ablation $g_k = \exp(-\lambda\Delta_k)$ weights the size of the transition; when increments approach zero on a memorized tail, its weight returns to one. Further implementation details appear in Section C.1. The gate sharpness λ and reference window ω are evaluated in the sensitivity analysis. The fitted-loss gate has no threshold parameter. Where training-length conventions require a noise-rate estimate, we use Liu et al. (2023).

5 EVALUATING COMPLEXITY-GATED LABEL RECOVERY

Having established the trajectory asymmetry in Section 2, we test its usefulness for label recovery. We first isolate the roles of the fitted-loss and complexity-level gates, then examine their combination

as scalar-loss discrimination weakens. Broader benchmarks and sensitivity studies assess the resulting CRT procedure.

Setup and comparisons: Our main benchmark suite comprises MNIST (LeCun et al., 1998), CIFAR-10/100 (Krizhevsky, 2009), T-Finance (Tang et al., 2022), and Clothing1M (Xiao et al., 2015). T-Finance has structured asymmetric noise, with fraudulent nodes mislabeled as benign; Clothing1M supplies real-world instance-dependent noise. We compare four base detectors—negative label smoothing (NLS; Wei et al., 2022a), NoiseGPT (Wang et al., 2024), PLS-LSA+ (Albert et al., 2024), and Delora (Yuan et al., 2025)—with their CRT augmentations. Both loss-only CRT and complexity-gated CRT are variants of our method. Rectification error evaluates corrected labels; detection F1 evaluates the noisy-versus-clean decision. We report means and standard deviations over 5 runs. Neural persistence (Rieck et al., 2019) is measured once per epoch and serves as the default structural complexity signal. Training, reference estimation, and compute controls are described in Sections C.1 and C.7.

5.1 Complementary Contributions of the Gates

With NLS fixed as the base detector, we compare loss-only weighting ($w_k = z_k$), complexity-only weighting ($w_k = g_k$), their product, and a rate-gated product using $g_k = \exp(-\lambda \Delta_k)$. Under 20% symmetric noise, we evaluate the standard horizon, which ends at zero observed-label classification error, and a horizon three times as long. This tests the finite-horizon distinction in Corollary 5: a level gate can continue to suppress late votes after complexity increments become small.

Weight w_k	**MNIST**		**CIFAR-10**	
	Std. horizon	$3\times$ **horizon**	**Std. horizon**	$3\times$ **horizon**
NLS (base detector)	6.59 ± 0.86	8.50 ± 1.00	8.05 ± 1.02	9.90 ± 1.10
+CRT, loss gate only (z_k)	4.36 ± 0.21	7.90 ± 0.60	5.10 ± 0.20	8.80 ± 0.70
+CRT, complexity gate only (g_k)	5.42 ± 0.30	5.61 ± 0.32	6.35 ± 0.33	6.52 ± 0.35
+CRT, rate-gated product ($z_k e^{-\lambda \Delta_k}$)	4.15 ± 0.20	6.80 ± 0.48	4.90 ± 0.19	7.62 ± 0.55
+Gated CRT ($z_k g_k$, ours)	$\mathbf{3.98 \pm 0.16}$	$\mathbf{4.21 \pm 0.17}$	$\mathbf{4.62 \pm 0.16}$	$\mathbf{4.94 \pm 0.19}$

Table 1: Gate ablation with NLS at 20% symmetric noise. Rectification error (%; mean $\pm$ standard deviation over 5 runs; lower is better) is evaluated when training first reaches zero observed-label classification error and after extending that horizon by $3\times$. All CRT rows are variants of our method.

On MNIST, extending training raises loss-only CRT error from 4.36% to 7.90% and rate-gated error from 4.15% to 6.80%. Complexity-only weighting changes little with the horizon but has higher standard-horizon error (5.42%). The product gives the lowest error at both horizons (3.98% and 4.21%); CIFAR-10 shows the same pattern. These results are consistent with the separate contributions of early incorrect votes and memorized-tail votes in Equation (4).

5.2 Controlled Loss Entanglement

We hold the CIFAR-100 noise rate at 20% and vary flip locality β: each corrupted label is drawn from the nearest confusable class within its superclass with probability β, and uniformly at random otherwise. We measure loss entanglement as $1 - \mathrm{AUROC}^{\star}_{\mathrm{loss}}$, where the starred quantity is the best area under the receiver operating characteristic curve over all recorded epochs and both ranking directions of observed-label loss on a fixed NLS reference trajectory. We separately compute the best detection F1 over epoch, threshold, and direction using the true noise mask. This oracle benchmark covers every single-epoch scalar-loss threshold on that trajectory; richer temporal or representation-based detectors remain outside its scope (Section C.3).

As locality increases, oracle scalar-loss F1 decreases from 0.96 to 0.68, and both base detectors and loss-only CRT deteriorate. At the most localized setting, gated CRT reaches F1 0.88–0.90 across detectors, averaging 0.21 above the oracle benchmark. The initial average margin is -0.02. Thus the gated prediction history supplies useful detection information beyond the best single-epoch loss threshold in this comparison. Clean-only training separates the confused classes on the same split. At

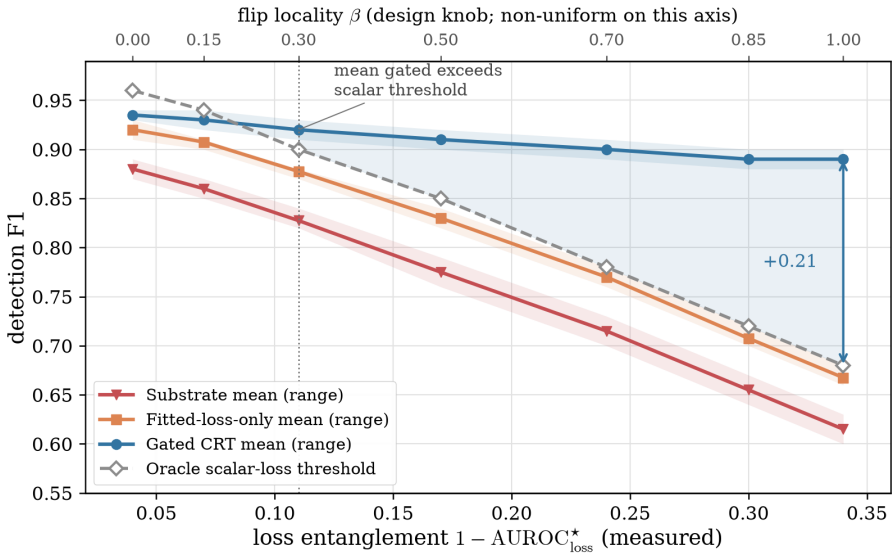


Figure 4: **Detection under increasing loss entanglement.** CIFAR-100 at fixed 20% noise. Curves and shaded bands give the mean and range across four base detectors for the detector alone (red), loss-only CRT (orange), and gated CRT (blue). Gray is the oracle scalar-loss F1 benchmark on the NLS trajectory. The horizontal coordinate is measured entanglement; the upper axis gives flip locality β.

the most localized setting, an indistinct plateau requires a low-noise calibration run for the persistence reference; this is an exception to estimation from the evaluated trajectory alone. CIFAR-100N (Wei et al., 2022b) provides a complementary evaluation with human-annotator noise (Section C.5).

5.3 Performance across Benchmarks

We compare each detector, its loss-only CRT augmentation, and its gated CRT augmentation under 10–40% symmetric and 40% asymmetric noise on MNIST and CIFAR, structured asymmetric noise on T-Finance, and native noise on Clothing1M. Complete results appear in Tables 3, 5 to 7 and 9.

Dataset	NLS	+CRT, loss only	+Gated CRT
MNIST	8.76 ± 1.07	5.01 ± 0.31	$\mathbf{4.28 \pm 0.24}$
CIFAR-10	12.51 ± 1.12	6.97 ± 0.37	$\mathbf{5.98 \pm 0.30}$
CIFAR-100	18.01 ± 1.55	14.60 ± 0.71	$\mathbf{12.85 \pm 0.55}$
T-Finance	18.81 ± 2.91	15.33 ± 0.65	$\mathbf{13.05 \pm 0.50}$
Clothing1M	11.55 ± 2.36	8.31 ± 0.40	$\mathbf{7.45 \pm 0.32}$

Table 2: Cross-benchmark summary with NLS fixed as the base detector. Rectification error (%; mean $\pm$ standard deviation over 5 runs; lower is better) at 40% symmetric noise on MNIST/CIFAR, 40% structured asymmetric noise on T-Finance, and native noise on Clothing1M. Both CRT variants are ours. These entries are drawn from the full benchmark tables.

Gated CRT reduces mean rectification error relative to both alternatives in every reported configuration. Table 2 summarizes the NLS comparison across all five datasets. For NLS at 40% symmetric noise, MNIST error decreases from 8.76% to 5.01% with loss-only CRT and to 4.28% with the complexity gate. The corresponding CIFAR-100 errors are 18.01%, 14.60%, and 12.85%. On Clothing1M, the complexity gate reduces loss-only CRT error by 0.81–0.97 percentage points across the four detectors. Detection F1 also improves throughout the reported tables. With NLS, the augmentation costs 1.3–1.6$\times$ the base runtime; the NLS control given the additional compute improves rectification error by at most 0.30 percentage points (Table 12).

5.4 Sensitivity to Gates and Noise-Rate Estimates

On MNIST with NLS at 20% symmetric noise, error stays within 0.15 percentage points of the default for $\lambda \in [\lambda^*/4, 4\lambda^*]$ and plateau windows $\omega \in \{5, 10, 20\}$. Weight-displacement and path-norm-increment gates give errors of 4.10% and 4.15%, compared with 3.98% for neural persistence and 4.36% for loss-only CRT. Replacing oracle noise rates with estimates changes error by at most 0.2 percentage points in the evaluated settings. Full settings and results appear in Section C.6; these comparisons assess the chosen reference and gates without estimating the surrogate-calibration error.

6 Related Work

Learning and detection under label noise: MentorNet (Jiang et al., 2018) and Co-Teaching (Han et al., 2018) use training dynamics to select examples. Loss-mixture methods distinguish clean and noisy populations (Arazo et al., 2019); DivideMix (Li et al., 2020) integrates this separation into semi-supervised training, while Early-Learning Regularization (ELR) (Liu et al., 2020) discourages memorization. Confident Learning (Northcutt et al., 2021) estimates label-error statistics, and Area Under the Margin (AUM) (Pleiss et al., 2020) aggregates prediction margins across training. Our structural trajectory signal complements these sources of evidence. The scalar-loss oracle specifically benchmarks single-epoch thresholds.

Learning dynamics and complexity: Spectral bias describes preferential learning of lower-frequency components (Xu et al., 2019; Rahaman et al., 2019; Xu et al., 2020). Neural persistence summarizes weight-graph topology (Rieck et al., 2019); weight dispersion and concentration influence its interpretation (Girrbach et al., 2023). We examine its sustained response to label-noise interventions and its use in CRT; further context appears in Section A.

7 Conclusion

The recorded trajectories show that label noise can produce a transient loss response and a sustained structural change. After noise injections, loss returns toward its pre-injection course, while neural persistence remains elevated relative to both clean-only training and matched clean additions. Declining loss can therefore coexist with a measurable structural response to the corrupted training labels.

The functional barrier identifies a geometric cost of confidently fitting local label conflicts. Trajectory monotonicity can preserve that cost, and calibration specifies when it transfers to measured persistence. CRT demonstrates the practical value of the measured signal: combining a classwise fit weight with a complexity-level gate improves label recovery in the reported benchmarks. The recovery bound connects this use of the persistent signal to the amount of retained clean-prediction mass, early errors, and later memorization.

Scope and limitations: The geometric analysis uses fixed features and label-consistent neighborhoods. Genuine class overlap can also create local conflicts, and argmax disagreement alone does not supply the required output margin. Recovery is conditional on useful clean predictions and their coverage by the shared gate, which may exclude late-learning examples. The barrier-based temporal explanation assumes monotone functional complexity; measured persistence need only satisfy the gate-separation condition. Our experiments assess persistence as a trajectory statistic without estimating its calibration error to a functional seminorm. The low-noise, early-stopping study also finds an inverted persistence ordering (Section C.4). Finally, horizon control requires both tail suppression and bounded early leakage at the chosen sharpness; one fixed gate is not guaranteed to remain effective under arbitrary training extensions.

Directions for further analysis: Sample-specific structural attribution could improve coverage of late-learning examples. Deriving useful prediction windows and reference-estimation guarantees for nonlinear training would strengthen the link between geometry and the operational gate. Quantifying calibration on convolutional and graph architectures would clarify when the same structural measurement supports the functional explanation.

Ethics Statement

CRT identifies potentially mislabeled examples and proposes corrected labels. Under class overlap, it can also flag genuinely ambiguous examples; its predictions depend on the representation and training data. In applications where incorrect relabeling has substantial consequences, proposed corrections should be reviewed with domain expertise and evaluated for class-specific errors and bias.

Reproducibility Statement

The theoretical assumptions and recovery result appear in Section 3, with proofs and the full surrogate analysis in Section B. The CRT procedure is specified in Section 4. Training conventions, feature sources, complexity measurement, reference estimation, computational costs, and sensitivity studies are described in Sections C.1 and C.7. Experiments use established public benchmarks.

AI Use Statement

Generative AI tools were used to assist with checking mathematical proofs, running experiments, and revising the manuscript. Manuscript assistance included organization, exposition, and checks of notation, cross-references, and bibliography entries.

References

Paul Albert, Jack Valmadre, Eric Arazo, Tarun Krishna, Noel E. O'Connor, and Kevin McGuinness. An accurate detection is not all you need to combat label noise in web-noisy datasets. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2407.05528.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, 2019.

Christopher M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37:103–120, 2007.

Lihui Deng, Bo Yang, Zhongfeng Kang, and Yanping Xiang. Invariant feature based label correction for dnn when learning with noisy labels. *Neural Networks*, 2024.

Leander Girrbach, Anders Christensen, Ole Winther, Zeynep Akata, and A. Sophia Koepke. Addressing caveats of neural persistence with deep graph persistence. *Transactions on Machine Learning Research*, 2023.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

Adarsh Kappiyath, Abhra Chaudhuri, AJAY KUMAR JAISWAL, Ziquan Liu, Yunpeng Li, Xiatian Zhu, and Lu Yin. SEBRA : Debiasing through self-guided bias ranking. In *ICLR*, 2025.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *STOC*, 1993.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html. Technical Report.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

Defu Liu, Ivor W. Tsang, and Guowu Yang. A convergence path to deep learning on noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020.

Yang Liu, Hao Cheng, and Kun Zhang. Identifiability of label noise transition matrix. In *ICML*, 2023.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. doi: 10.1613/jair.1.12125.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *ICML*, 2019.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *ICLR*, 2019.

Charlie Tan, Inés García-Redondo, Qiquan Wang, Michael M. Bronstein, and Anthea Monod. On the limitations of fractal dimension as a measure of generalization. In *NeurIPS*, 2024.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *ICML*, 2022.

Brendan van Rooyen and Robert C. Williamson. A theory of learning with corrupted labels. *JMLR*, 2018.

Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*, 2015.

Haoyu Wang, Zhuo Huang, Zhiwei Lin, and Tongliang Liu. Noisegpt: Label noise detection and rectification through probability curvature. In *Advances in Neural Information Processing Systems*, 2024.

Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. To smooth or not? when label smoothing meets noisy labels. In *ICML*, 2022a.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *ICLR*, 2022b. URL `https://openreview.net/forum?id=TBWA6PLJZQm`.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, 2019.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 2020.

Yu Yang, Eric Gan, Gintare Karolina Dziugaite, and Baharan Mirzasoleiman. Identifying spurious biases early in training through the lens of simplicity bias. *AISTATS*, 2024.

Bo Yuan, Yulin Chen, and Yin Zhang. Weed out, then harvest: Dual low-rank adaptation is an effective noisy label detector for noise-robust learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15292–15311, 2025. doi: 10.18653/v1/2025.findings-acl.792. URL https://aclanthology.org/2025.findings-acl.792/.

Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Toward understanding the importance of noise in training neural networks. In *ICML*, 2019.

A Extended Literature Review

Noisy Labels: Label-noise methods draw on both feature structure and learning dynamics. Invariant-feature-based correction (Deng et al., 2024) uses stable feature information to revise labels. Studies of learning dynamics (Yang et al., 2024; Kappiyath et al., 2025) and optimization under noise (Zhou et al., 2019; Liu et al., 2024) examine how optimization and learned structure evolve during training. Our question concerns the signals available after this acquisition: whether the resulting structural change can still help select useful predictions when loss has become less informative. Classical analyses of corrupted-label learning (Kearns, 1993; Natarajan et al., 2013; van Rooyen & Williamson, 2018), including symmetric-noise theory (van Rooyen et al., 2015), provide the broader learnability context for this trajectory-based question.

Complexity measures for neural networks: Neural persistence (Rieck et al., 2019) measures the topological structure of a trained MLP's weight graph via persistent homology; subsequent work identifies caveats in interpreting it as functional complexity (Girrbach et al., 2023). Fractal-dimension-based measures have likewise been studied as generalization proxies, with known limitations (Tan et al., 2024). Our operational gate requires within-run level separation rather than a universal generalization measure (Section 4); the additional conditions that connect this structural statistic to the functional barrier are isolated in Section B.5.

B Theoretical Results

B.1 Learning Setup and Recovery Target

Preliminaries: Let X be an input domain containing the training inputs, and let Y be the common finite alphabet for clean labels, observed labels, and predictions. For each $\mathbf{x} \in X$, the (unobserved) clean label is $\mathbf{y}^* \in Y$ and the observed (possibly noisy) label is $\mathbf{y} \in Y$, with $\mathbf{y} = \mathbf{y}^*$ holding with probability $1 - \eta$ and $\mathbf{y} \neq \mathbf{y}^*$ with probability η, where η is the noise rate. We denote the clean dataset $D^* = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^N$ and the noisy dataset $D = T(D^*)$, where T is the label-noise transformation. The clean and noisy concepts are $c, \tilde{c} : X \to Y$. Labels are one-hot unless stated otherwise; $f(\mathbf{x}) \in \Delta(Y)$ denotes a predictive distribution and $\hat{\mathbf{y}}(\mathbf{x}) = \mathrm{onehot}(\arg\max_{\mathbf{y} \in Y} f_{\mathbf{y}}(\mathbf{x}))$ its one-hot argmax prediction.

Effective function class and clean concept: Let $\mathcal{F}$ denote a hypothesis space of measurable functions $f : X \to \Delta(Y)$. Fix an initialization $f_0 \in \mathcal{F}$ and gradient-flow dynamics $\frac{d}{dt} f_t = -\nabla \mathcal{L}_D(f_t)$ for training set D. The resulting trajectory defines the *effective, data-induced function class*:

$$\mathcal{F}_{\text{eff}}(D) \;=\; \left\{\, f_t : t \geq 0,\; \dot{f}_t = -\nabla \mathcal{L}_D(f_t) \,\right\} \;\subseteq\; \mathcal{F}.$$

Let f^* be the predictor reached by gradient flow on the clean dataset, and let the **clean concept** c denote its induced label map:

$$f^* \;:=\; \lim_{t \to \infty} f_t, \qquad c(\mathbf{x}) \;:=\; \mathrm{onehot}\left(\arg\max_{\mathbf{y} \in Y} f^*_{\mathbf{y}}(\mathbf{x}) \right), \qquad \dot{f}_t = -\nabla \mathcal{L}_{D^*}(f_t).$$

On noisy data, gradient flow traverses $\mathcal{F}_{\text{eff}}(D)$, whose limit generally differs from f^*. The recovery target is the label map $c(\mathbf{x})$ induced by clean training. Identifying this target with the ground-truth label requires clean training to predict that label correctly.

B.2 Effective Feature Space

Let ϕ map raw inputs into the fixed feature coordinates used by the learner. For the geometric results, we identify X with the ambient coordinate space $\mathbb{R}^m$ and set $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2$; the finite set $\{\phi(\mathbf{x}_i)\}$ lies inside this domain, and the classifier head is defined on the ambient domain rather than only on the sampled points. Thus the local inconsistency conditions underlying the barrier refer to nearby effective-coordinate pairs with different labels. The Euclidean ambient domain permits the radial contractions used in Lemma 1.

Remark 2 (Static metric). Throughout the analysis, ϕ and hence d are treated as fixed over the analyzed trajectory. This matches the experiments that train detection models on frozen pretrained features (Section C.1). When the representation is learned jointly, ϕ evolves and could increase the

distance between a conflicting pair, relaxing the barrier at that pair; persistence then becomes an additional empirical question outside this static-metric theorem.

B.3 Complexity Axioms

Let $\mathcal{F}$ be a hypothesis class of continuous functions $f : X \to \Delta(Y)$, and let

$$\mathcal{C} : \mathcal{F} \to [0, \infty]$$

be a complexity functional. The following *complexity axioms* formalize the spatial variation measured by the barrier analysis.

Axiom 1 (Well-definedness). For all $f \in \mathcal{F}$,

$$\mathcal{C}(f) \in [0, \infty].$$

Axiom 2 (Continuity). Let $(f_n)_{n\in\mathbb{N}}$ be a sequence in $\mathcal{F}$ converging to f in the chosen topology on $\mathcal{F}$. Then

$$f_n \to f \quad \Longrightarrow \quad \mathcal{C}(f_n) \to \mathcal{C}(f).$$

(Lower semicontinuity, $\mathcal{C}(f) \le \liminf_n \mathcal{C}(f_n)$, suffices for all results below.)

Axiom 3 (Monotonicity). Let the *oscillation* of f on a set $S \subseteq X$ be

$$\operatorname{osc}(f, S) \;=\; \sup_{\mathbf{z},\mathbf{z}'\in S} \|f(\mathbf{z}) - f(\mathbf{z}')\|,$$

and for each output variation $\omega > 0$ define the *oscillation scale* of f at ω as the smallest spatial scale at which f realizes that variation:

$$s_f(\omega) \;=\; \inf\{\, \operatorname{diam}(S) \,:\, S \subseteq X,\ \operatorname{osc}(f, S) \ge \omega \,\} \qquad (\inf \emptyset = \infty).$$

We say f *scale-dominates* g (written $f \succeq g$) if $s_f(\omega) \le s_g(\omega)$ for all $\omega > 0$, *i.e.*, f realizes every output variation at spatial scales no larger than g does; and f is *strictly more oscillatory* than g (written $f \succ g$) if additionally $s_f(\omega) < s_g(\omega)$ for some ω. Then:

$$f \succeq g \quad \Longrightarrow \quad \mathcal{C}(f) \ge \mathcal{C}(g).$$

Remark 3 (Why scale dominance). A single local comparison need not order global smoothness: a function can match another's variation on one small set while varying more slowly elsewhere. Scale dominance compares the entire profile $\omega \mapsto s_f(\omega)$, requiring the ordering at every output variation. It is a preorder on functions and a partial order on their oscillation profiles. Standard smoothness seminorms are monotone under this comparison (Section B.4).

Axiom 4 (Non-degeneracy).

$$\mathcal{C}(f) = 0 \quad \Longleftrightarrow \quad f \text{ is constant on } X.$$

The proof also uses compactness and closure under spatial contraction. We impose these as regularity conditions on the pair $(\mathcal{F}, \mathcal{C})$, in addition to the four axioms:

Assumption 1 (Regularity)**.** (i) $\mathcal{F}$ consists of continuous functions $X \to \Delta(Y)$, and any sequence in $\mathcal{F}$ with uniformly bounded complexity admits a subsequence that converges uniformly on compact subsets of X to a limit in $\mathcal{F}$. (ii) On the Euclidean effective-coordinate domain $X = \mathbb{R}^m$, $\mathcal{F}$ is closed under precomposition with radial affine contractions: for any $f \in \mathcal{F}$, $\mathbf{x}_0 \in X$, and $\gamma \in (0, 1]$, the map $\mathbf{z} \mapsto f(\mathbf{x}_0 + \gamma(\mathbf{z} - \mathbf{x}_0))$ belongs to $\mathcal{F}$.

Condition (i) is an Arzelà–Ascoli-type compactness requirement. It holds, for example, for bounded Hölder sublevel sets when $\mathcal{F}$ is closed under uniform-on-compacta limits: the complexity bound gives equicontinuity and $\Delta(Y)$ gives pointwise boundedness. Condition (ii) holds for architecture families in which precomposition with a fixed affine map can be absorbed into the first layer. The general barrier is conditional on both properties.

B.4 Concrete Complexity Measures

Hölder seminorms give concrete instances of the axioms and an explicit barrier rate. Comparing them with total variation and parameter count clarifies why the relevant quantity is spatial variation at a given scale.

Proposition 1 (Hölder seminorms satisfy the axioms)**.** *For $\alpha \in (0, 1]$, the Hölder seminorm*

$$|f|_\alpha \;=\; \sup_{\mathbf{z}\neq\mathbf{z}'} \frac{\|f(\mathbf{z}) - f(\mathbf{z}')\|}{d(\mathbf{z}, \mathbf{z}')^\alpha}$$

(the Lipschitz seminorm at $\alpha = 1$) satisfies Well-definedness, Continuity in the lower-semicontinuous form, Monotonicity under scale dominance, and Non-degeneracy. Moreover it admits the scale-profile representation

$$|f|_\alpha \;=\; \sup_{\omega>0} \frac{\omega}{s_f(\omega)^\alpha}.$$

Proof. Well-definedness and Non-degeneracy ($|f|_\alpha = 0$ iff f constant) are immediate, and lower semicontinuity under uniform-on-compacta convergence is standard (a supremum of functionals continuous in f pointwise over pairs). For the representation: for any S with $\mathrm{osc}(f, S) \geq \omega$ there are $\mathbf{z}, \mathbf{z}' \in S$ with $\|f(\mathbf{z}) - f(\mathbf{z}')\| \geq \omega - \epsilon$ and $d(\mathbf{z}, \mathbf{z}') \leq \mathrm{diam}(S)$, so $|f|_\alpha \geq (\omega - \epsilon)/\mathrm{diam}(S)^\alpha$; infimizing over S and letting $\epsilon \to 0$ gives $|f|_\alpha \geq \omega/s_f(\omega)^\alpha$ for every ω. Conversely, for any pair $(\mathbf{z}, \mathbf{z}')$ with $\omega_0 = \|f(\mathbf{z}) - f(\mathbf{z}')\|$ and $d_0 = d(\mathbf{z}, \mathbf{z}')$, the two-point set shows $s_f(\omega_0) \leq d_0$, hence $\sup_\omega \omega/s_f(\omega)^\alpha \geq \omega_0/d_0^\alpha$; supremizing over pairs gives the reverse inequality. Monotonicity is now immediate: $f \succeq g$ means $s_f(\omega) \leq s_g(\omega)$ pointwise, and ω/s^α is nonincreasing in s, so $|f|_\alpha \geq |g|_\alpha$. □

The seminorm definition in Proposition 1 gives the following pairwise lower bound.

Corollary 1 (Explicit pairwise barrier rate)**.** *For $\mathcal{C} = |\cdot|_\alpha$, any $f \in \mathcal{F}$ and pair $\mathbf{z} \neq \mathbf{z}'$ with $\|f(\mathbf{z}) - f(\mathbf{z}')\| \geq \mu > 0$ satisfy*

$$\mathcal{C}(f) \;\geq\; \frac{\mu}{d(\mathbf{z}, \mathbf{z}')^\alpha}.$$

Consequently, the radial quantity in Observation 1 obeys $C_\mu(r) \geq \mu/r^\alpha$. This pairwise rate requires no compactness argument.

Instructive non-examples: *Total variation* fails scale-dominance Monotonicity: a single sharp bump can scale-dominate a train of many gentle bumps of the same height while having smaller total variation. Total variation accumulates changes across the domain, whereas the barrier concerns their spatial scale. *Parameter count* measures an architecture rather than its realized function: equivalent functions can have different parameter counts, and a fixed architecture assigns the same count to constant and nonconstant realizations. It therefore does not supply the required functional Well-definedness and Non-degeneracy. Neural persistence is also computed from network parameters; the next section specifies its structural properties and the conditions needed for a functional interpretation.

B.5 Neural Persistence as a Structural Surrogate

Neural persistence summarizes the structure of a parameterized network's weights. To use it alongside the functional barrier, we establish its invariances and stability, identify the limits of its function-space interpretation, and give a calibration condition that transfers functional separation to the measured trajectory.

Fix $p \in [1, \infty)$ and a fully connected architecture with L weight layers and nonzero edge-weight vector $w(\theta)$. Following Rieck et al. (2019), assign each edge the normalized filtration value

$$\nu_e(\theta) = \frac{|w_e(\theta)|}{\|w(\theta)\|_\infty} \in [0, 1].$$

All vertices enter at filtration value one and edges enter in descending order of ν_e. For the bipartite graph between layers ℓ and $\ell + 1$, let $\mathcal{D}_\ell^{\mathrm{fin}}(\theta)$ be the multiset of finite zero-dimensional merge pairs, retaining zero-lifetime pairs as diagonal padding and omitting the one essential component. Its cardinality is $q_\ell = |V_\ell| + |V_{\ell+1}| - 1$. The mean normalized neural persistence is

$$M_p(\theta) = \frac{1}{L} \sum_{\ell=0}^{L-1} \frac{1}{q_\ell^{1/p}} \left(\sum_{(b,d)\in\mathcal{D}_\ell^{\mathrm{fin}}(\theta)} |b - d|^p \right)^{1/p} \in [0, 1]. \tag{5}$$

Equivalently, the finite death edges form a maximum spanning tree of each layer graph. Thus M_p summarizes the distribution and connectivity of large normalized weights, rather than directly evaluating the input–output map. This distinction is material: later analyses identify weight dispersion and concentration as principal drivers of the score (Girrbach et al., 2023).

Proposition 2 (Structural invariance and stability of neural persistence)**.** *For a fixed architecture, M_p has the following properties.*

1. *It is invariant under every layer-compatible relabeling of hidden units.*
2. *It is invariant under multiplication of all edge weights by one common nonzero scalar.*
3. *For two nonzero parameter vectors θ, θ' of that architecture, set $d_\nu(\theta, \theta') = \|\nu(\theta) - \nu(\theta')\|_\infty$. Then*

$$|M_p(\theta) - M_p(\theta')| \le 2d_\nu(\theta, \theta'). \tag{6}$$

In particular, with $m = \min\{\|w(\theta)\|_\infty, \|w(\theta')\|_\infty\} > 0$,

$$|M_p(\theta) - M_p(\theta')| \le \frac{4\|w(\theta) - w(\theta')\|_\infty}{m}.$$

Proof. A layer-compatible neuron relabeling induces a weighted-graph isomorphism in every layer, so it leaves every persistence diagram unchanged. Common rescaling cancels in ν_e. For stability, the standard bottleneck stability theorem for persistence diagrams gives a matching between $\mathcal{D}_\ell^{\mathrm{fin}}(\theta)$ and $\mathcal{D}_\ell^{\mathrm{fin}}(\theta')$, padded on the diagonal to q_ℓ pairs, whose endpoint displacement is at most $d_\nu(\theta, \theta')$ (Cohen-Steiner et al., 2007). Each matched lifetime therefore changes by at most $2d_\nu$; the reverse triangle inequality for the ℓ_p norm and division by $q_\ell^{1/p}$ give the first bound after averaging over layers. Finally,

$$\left\| \frac{|w(\theta)|}{\|w(\theta)\|_\infty} - \frac{|w(\theta')|}{\|w(\theta')\|_\infty} \right\|_\infty \le \frac{2\|w(\theta) - w(\theta')\|_\infty}{m},$$

which gives the second bound. □

The invariance and stability bounds in Proposition 2 control changes in the normalized weight graph. Functional stability additionally requires nearby functions to admit nearby canonical parameters, as formalized next.

Corollary 2 (Functional stability on an identifiable section)**.** *Let $R : \Theta \to \mathcal{F}$ be the network realization map and let $s : \mathcal{K} \to \Theta$ be a right inverse on $\mathcal{K} \subseteq \mathcal{F}$. Suppose the admissible canonical representative of each function is unique up to layer-compatible neuron permutations and, for a modulus ρ with $\rho(\delta) \to 0$ as $\delta \to 0$,*

$$\inf_\pi \|\nu(s(f)) - \pi\nu(s(g))\|_\infty \le \rho(d_{\mathcal{F}}(f, g)),$$

where π ranges over those permutations. Then $M_p \circ s$ is a well-defined continuous functional on $\mathcal{K}$, and

$$|M_p(s(f)) - M_p(s(g))| \le 2\rho(d_{\mathcal{F}}(f, g)).$$

Thus graph changes induced by a functional perturbation of size at most δ change neural persistence by at most $2\rho(\delta)$.

Proof. Permutation invariance makes the score independent of the remaining ambiguity in the section. Apply Equation (6) after the minimizing relabeling and then use the assumed inverse modulus. □

The next counterexample shows why the canonical-parameter assumption is needed: different parameterizations of the same function can have different neural persistence.

Proposition 3 (No unconditional function-space interpretation)**.** *For ReLU networks, raw normalized neural persistence is not, in general, a functional $\mathcal{F} \to [0, \infty)$ and does not satisfy the four complexity axioms or Assumption 1 on the unrestricted realized function class.*

Proof. Consider the two-hidden-unit scalar-output ReLU network

$$R_\theta(\mathbf{x}) = (1,1)\, r(I_2\mathbf{x}) = r(x_1) + r(x_2), \qquad r(t) = \max\{0, t\}.$$

For any $a > 1$, positive homogeneity gives the same function under

$$W_1' = \mathrm{diag}(a, 1), \qquad W_2' = (a^{-1}, 1),$$

because $a^{-1}r(ax_1) = r(x_1)$. For the original parameters, the first-layer lifetimes are $(0, 0, 1)$ and the second-layer lifetimes are $(0, 0)$, so

$$M_p(\theta) = \frac{1}{2\,3^{1/p}}.$$

Global normalization of the rescaled parameters instead gives first-layer weights $\mathrm{diag}(1, a^{-1})$ and second-layer weights (a^{-2}, a^{-1}). Consequently,

$$M_p(\theta') = \frac{1}{2}\left[\frac{\left(1 + (1-a^{-1})^p\right)^{1/p}}{3^{1/p}} + \frac{\left((1-a^{-1})^p + (1-a^{-2})^p\right)^{1/p}}{2^{1/p}}\right] > M_p(\theta).$$

Composing both realizations with the same fixed map from the scalar logit to a two-class probability gives identical elements of $\mathcal{F}$ with different scores. Therefore neural persistence violates functional Well-definedness.

Non-degeneracy and compact-sublevel regularity fail simultaneously. On $X = [-1, 1]$, consider the one-layer two-logit networks

$$f_n(x) = \mathrm{softmax}(nx, -nx), \qquad n \in \mathbb{N}.$$

Both normalized edge magnitudes equal one, so every finite persistence pair has zero lifetime and $M_p = 0$, although every f_n is nonconstant. Moreover, every subsequence converges pointwise away from zero to opposite one-hot vectors on the two half intervals, while $f_n(0) = (1/2, 1/2)$; hence no subsequence converges uniformly to a continuous function. Thus bounded-persistence sublevels are not compact, contrary to Assumption 1(i). Since the raw score is not a function on $\mathcal{F}$ in the first place, functional Continuity and scale-dominance Monotonicity cannot be asserted without restricting the realization map and adding calibration. □

Along a recorded training run, the optimizer supplies a particular parameter sequence θ_k. The score can therefore be calibrated on that sequence even though it varies across functionally equivalent parameterizations. The following assumption quantifies the calibration needed to transfer a functional separation margin to the measured gate.

Assumption 2 (Trajectory calibration of neural persistence)**.** Let θ_k realize f_k, let $M_k = M_p(\theta_k)$, and let C^{fun} be a functional complexity satisfying the axioms and regularity conditions above. There exist $a > 0$, $b \in \mathbb{R}$, and $\varepsilon_{\mathrm{NP}} \geq 0$ such that, over the analyzed trajectory,

$$aM_k + b = C^{\mathrm{fun}}(f_k) + e_k, \qquad |e_k| \leq \varepsilon_{\mathrm{NP}}.$$

For the trajectory-derived persistence reference $\widehat{M}_{\mathrm{ref}}$ and a corresponding functional reference $C^{\mathrm{fun}}_{\mathrm{ref}}$, assume likewise

$$a\widehat{M}_{\mathrm{ref}} + b = C^{\mathrm{fun}}_{\mathrm{ref}} + e_{\mathrm{ref}}, \qquad |e_{\mathrm{ref}}| \leq \varepsilon_{\mathrm{ref}}.$$

The calibration is specific to one run: the offset b cancels from level differences, and the scale a can be absorbed into gate sharpness. Its uniform error bound is stronger than observing that both quantities increase together. Establishing that bound requires paired functional-complexity and persistence measurements.

Proposition 4 (Surrogate transfer to the shared gate)**.** *Suppose the functional trajectory has shared-gate separation relative to* $C^{\mathrm{fun}}_{\mathrm{ref}}$*:*

$$C^{\mathrm{fun}}(f_k) - C^{\mathrm{fun}}_{\mathrm{ref}} \leq \epsilon_C, \qquad k \in \mathcal{K}_c(\mathbf{x}),$$
$$C^{\mathrm{fun}}(f_k) - C^{\mathrm{fun}}_{\mathrm{ref}} \geq \delta_C, \qquad k \geq t_0(\mathbf{x}).$$

Under Assumption 2, the measured persistence trajectory satisfies

$$M_k - \widehat{M}_{\mathrm{ref}} \le \frac{\epsilon_C + \varepsilon_{\mathrm{NP}} + \varepsilon_{\mathrm{ref}}}{a}, \qquad k \in \mathcal{K}_c(\mathbf{x}),$$
$$M_k - \widehat{M}_{\mathrm{ref}} \ge \frac{\delta_C - \varepsilon_{\mathrm{NP}} - \varepsilon_{\mathrm{ref}}}{a}, \qquad k \ge t_0(\mathbf{x}).$$

Consequently the persistence gate retains a positive inside-versus-tail separation whenever

$$\delta_C - \epsilon_C > 2(\varepsilon_{\mathrm{NP}} + \varepsilon_{\mathrm{ref}}). \tag{7}$$

With these degraded constants, Theorem 2 and Corollary 5 apply unchanged.

Proof. Subtract the reference calibration equation from the epoch-k equation:

$$a(M_k - \widehat{M}_{\mathrm{ref}}) = C^{\mathrm{fun}}(f_k) - C^{\mathrm{fun}}_{\mathrm{ref}} + (e_k - e_{\mathrm{ref}}).$$

The two bounds follow from $|e_k - e_{\mathrm{ref}}| \le \varepsilon_{\mathrm{NP}} + \varepsilon_{\mathrm{ref}}$. Their gap is positive exactly under Equation (7); substituting the resulting persistence-scale constants into the recovery bounds proves the final claim. □

Exact order calibration also identifies conditions under which a canonical persistence score inherits the functional axioms.

Corollary 3 (When a persistence surrogate inherits the axioms)**.** *In the setting of Corollary 2, suppose $\mathcal{K}$ is closed in the topology used by Assumption 1 and C^{fun} is finite on $\mathcal{K}$. Suppose further that C^{fun} satisfies the four axioms and there is a continuous strictly increasing map $h : [0,\infty) \to [0,\infty)$ with $h(0) = 0$ such that*

$$M_p(s(f)) = h(C^{\mathrm{fun}}(f)) \qquad \textit{for every } f \in \mathcal{K}.$$

Then $M_p \circ s$ satisfies Well-definedness, Continuity (or lower semicontinuity), scale-dominance Monotonicity, and Non-degeneracy on $\mathcal{K}$. Compact-sublevel regularity also transfers if h is proper, meaning $h(c) \to \infty$ as $c \to \infty$.

Proof. Well-definedness follows from the section and permutation invariance. Continuity follows by composition; strict increase preserves the scale-dominance order; and $h(c) = 0$ iff $c = 0$, which transfers Non-degeneracy. If h is proper, every bounded sublevel of $h \circ C^{\mathrm{fun}}$ lies in a bounded sublevel of C^{fun}, so compact-sublevel regularity transfers as well. □

Normalized neural persistence is bounded, so it cannot instantiate a globally unbounded functional barrier. Its role in the method is a structural statistic along one training trajectory. Proposition 4 supplies a conditional connection to functional complexity; the recovery theorem can also be applied directly when the measured trajectory satisfies shared-gate separation.

B.6 Unboundedness under Vanishing Separation

The proof has two steps. Spatial contraction orders the minimum complexity across pair distances, and compactness rules out a finite limiting cost as the distance vanishes.

Throughout this subsection, fix $\mathbf{x} \in X$, an output separation $\mu > 0$, and a unit direction $\mathbf{u}$ in the effective feature coordinates. For each $r > 0$, let $\mathbf{x}_r := \mathbf{x} + r\,\mathbf{u}$, so the points $\{\mathbf{x}_r\}_{r>0}$ are colinear and $d(\mathbf{x}, \mathbf{x}_r) = r$. Define

$$\mathcal{F}_r^{(\mu)} = \{f \in \mathcal{F} : \|f(\mathbf{x}) - f(\mathbf{x}_r)\| \ge \mu\}, \qquad C_\mu(r) = \inf\{\mathcal{C}(f) : f \in \mathcal{F}_r^{(\mu)}\}, \qquad (\inf \emptyset = \infty).$$

Lemma 1 (Oscillation Growth)**.** *For $0 < r_1 < r_2$ and any $f \in \mathcal{F}_{r_1}^{(\mu)}$, there exists $g \in \mathcal{F}_{r_2}^{(\mu)}$ with $f \succeq g$; consequently $C_\mu(r_1) \ge C_\mu(r_2)$.*

Proof. Let $\rho : X \to X$ be the radial contraction toward $\mathbf{x}$ with ratio $\gamma = r_1/r_2 \in (0,1)$:

$$\rho(\mathbf{z}) \;=\; \mathbf{x} + \gamma\,(\mathbf{z} - \mathbf{x}), \qquad \text{so} \qquad d(\mathbf{x}, \rho(\mathbf{z})) = \gamma\, d(\mathbf{x}, \mathbf{z}) \quad \text{and} \quad \rho(\mathbf{x}_{r_2}) = \mathbf{x}_{r_1}.$$

Given $f \in \mathcal{F}_{r_1}^{(\mu)}$, define $g := f \circ \rho$; by Assumption 1(ii), $g \in \mathcal{F}$. Since $g(\mathbf{x}) = f(\mathbf{x})$ and $g(\mathbf{x}_{r_2}) = f(\mathbf{x}_{r_1})$, their output separation is at least μ, so $g \in \mathcal{F}_{r_2}^{(\mu)}$.

We verify $f \succeq g$ under scale dominance. For any set $S \subseteq X$, $\operatorname{osc}(g, S) = \operatorname{osc}(f, \rho(S))$ and $\operatorname{diam}(\rho(S)) = \gamma \operatorname{diam}(S)$; since witness sets for g correspond to subsets of $\rho(X) \subseteq X$ for f, the oscillation-scale profiles satisfy

$$s_g(\omega) \;\geq\; \frac{1}{\gamma}\, s_f(\omega) \qquad \text{for all } \omega > 0$$

(the infimum defining s_g ranges over a γ-dilation of a restricted family of witness sets). Hence $s_f(\omega) \leq \gamma\, s_g(\omega) \leq s_g(\omega)$ for all ω, so $f \succeq g$. By the Monotonicity axiom, $\mathcal{C}(f) \geq \mathcal{C}(g) \geq C_\mu(r_2)$. Taking the infimum over $f \in \mathcal{F}_{r_1}^{(\mu)}$ yields $C_\mu(r_1) \geq C_\mu(r_2)$. □

Lemma 2 (Unboundedness under Vanishing Separation)**.** *Under the axioms and Assumption 1,*

$$\lim_{r \to 0} C_\mu(r) = \infty.$$

Proof. Suppose for contradiction that $\sup_{r>0} C_\mu(r) \leq M < \infty$ (the supremum is approached as $r \to 0$ by Lemma 1). Then for each $n \in \mathbb{N}$ there exists $f_n \in \mathcal{F}_{1/n}^{(\mu)}$ with $\mathcal{C}(f_n) \leq M + 1$, satisfying

$$\|f_n(\mathbf{x}) - f_n(\mathbf{x}_{1/n})\| \geq \mu, \qquad d(\mathbf{x}, \mathbf{x}_{1/n}) = 1/n.$$

By Assumption 1(i), (f_n) admits a subsequence (f_{n_k}) converging uniformly on compact sets to some $f^* \in \mathcal{F}$, and by the Continuity axiom $\mathcal{C}(f^*) \leq M + 1 < \infty$.

Fix a compact K containing $\mathbf{x}$ and all $\mathbf{x}_{1/n_k}$. Uniform convergence on K and continuity of f^* give

$$\|f_{n_k}(\mathbf{x}) - f_{n_k}(\mathbf{x}_{1/n_k})\| \;\leq\; 2\|f_{n_k} - f^*\|_{\infty,K} + \|f^*(\mathbf{x}) - f^*(\mathbf{x}_{1/n_k})\| \;\longrightarrow\; 0,$$

contradicting the lower bound μ. Therefore $\lim_{r \to 0} C_\mu(r) = \infty$. □

Remark 4. Together, the lemmas show that retaining a fixed output separation at smaller distances raises the minimum complexity, with compactness forcing this minimum to diverge. The result concerns a family of fitted functions indexed by pair distance; its use along an optimization trajectory requires the additional conditions below.

B.7 Complexity Barrier and Unreachability

Proof of Observation 1: The Observation restates Lemmas 1 and 2 in the form used in the main text: monotonicity of $C_\mu(\cdot)$ follows from the radial contraction construction, and $\lim_{r \to 0} C_\mu(r) = \infty$ because uniform-on-compacta convergence to a continuous limit is incompatible with retaining output separation μ across a pair whose distance vanishes.

Corollary 4 (Radial unreachability)**.** *Let $\mathcal{F}'$ be a hypothesis class with $\sup_{f \in \mathcal{F}'} \mathcal{C}(f) \leq K < \infty$, and fix the ray and $\mu > 0$ from Observation 1. Then there exists $r_{K,\mu} > 0$ such that no $f \in \mathcal{F}'$ belongs to $\mathcal{F}_r^{(\mu)}$ for $r \leq r_{K,\mu}$. In particular, no gradient-flow limit in $\mathcal{F}'$ exactly interpolates two distinct one-hot labels along this family at arbitrarily small distances.*

Proof of Corollary 4: Let $\mathcal{F}'$ satisfy $\sup_{f \in \mathcal{F}'} \mathcal{C}(f) \leq K < \infty$. By Lemma 2, there exists $r_{K,\mu} > 0$ such that $C_\mu(r) > K$ for all $r \leq r_{K,\mu}$. Any f retaining output separation μ on the specified radial pair at such a distance has $\mathcal{C}(f) \geq C_\mu(r) > K$ and hence lies outside $\mathcal{F}'$. Exact interpolation of distinct one-hot labels is the special case $\mu = \|\mathbf{y} - \mathbf{y}'\|$. The conclusion concerns positive output separation on the fixed radial family in a bounded-complexity class.

B.8 Trajectory Conditions and Barrier Consequences

The recovery analysis uses three properties: a useful interval of clean predictions, separation of that interval from the later tail by one shared gate, and sufficient fitted-loss weight inside the interval. The first two are stated here; Proposition 8 addresses the third. Monotone functional complexity links a fitted geometric conflict to a persistent tail excess.

Assumption 3 (Monotone trajectory complexity)**.**

$$t_1 < t_2 \quad \Longrightarrow \quad \mathcal{C}(f_{t_1}) \le \mathcal{C}(f_{t_2}), \tag{8}$$

for the trajectory under analysis.

Assumption 4 (Clean-prediction window and memorizing tail)**.** For a recoverable sample $\mathbf{x}$, there exist $1 \le \tau(\mathbf{x}) < t^*(\mathbf{x}) \le t_0(\mathbf{x}) \le T$ such that

$$\hat{\mathbf{y}}_k = c(\mathbf{x}) \quad \text{for } \tau(\mathbf{x}) \le k < t^*(\mathbf{x}).$$

For the long-horizon statement only, if $\mathbf{x}$ is mislabeled, we additionally require

$$\hat{\mathbf{y}}_k = \tilde{c}(\mathbf{x}) \neq c(\mathbf{x}) \quad \text{for all } k \ge t_0(\mathbf{x}).$$

We call $\mathcal{K}_c(\mathbf{x}) = \{\tau, \dots, t^* - 1\}$ a **clean-prediction window**. The finite-time recovery theorem needs only a nonempty set of clean votes; it does not require a single transition or assume that every sample has such a window. The two-mode model in Section B.9 gives sufficient conditions for both clauses.

The implementation uses one global complexity sequence and one estimated reference $\widehat{C}_{\mathrm{ref}}$ for all samples. Its required alignment with the sample-specific windows is therefore a coverage condition:

Assumption 5 (Shared-gate separation)**.** For every sample $\mathbf{x}$ in a covered set $\mathcal{R}$, there are constants $\epsilon_C \ge 0$ and $\delta_C > \epsilon_C$ such that

$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \le \epsilon_C, \qquad k \in \mathcal{K}_c(\mathbf{x}),$$
$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \ge \delta_C, \qquad k \ge t_0(\mathbf{x}).$$

The constants may be chosen uniformly to obtain a guarantee over all of $\mathcal{R}$. Samples whose clean-prediction windows are truncated by the shared gate simply fall outside the guaranteed set.

The reference is computed from the observed trajectory as described in Section 4. Its alignment with the sample-specific clean windows is quantified by the realized constants ϵ_C and δ_C. These assumptions define the scope of the general recovery theorem; Section B.9 derives the temporal structure in a two-mode model.

Proposition 5 (Barrier-implied persistent margin)**.** *Suppose Assumption 3 holds and, at step t_0, f_{t_0} has output separation at least μ on the radial pair at distance r. For any trajectory-derived reference $\widehat{C}_{\mathrm{ref}}$ and every $k \ge t_0$,*

$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \;\ge\; C_\mu(r) - \widehat{C}_{\mathrm{ref}}.$$

Consequently, if $C_\mu(r) - \widehat{C}_{\mathrm{ref}} > 0$, the second inequality of Assumption 5 holds with any $\delta_C \le C_\mu(r) - \widehat{C}_{\mathrm{ref}}$.

Proof. The definition of $C_\mu(r)$ gives $\mathcal{C}(f_{t_0}) \ge C_\mu(r)$. Monotone trajectory complexity gives $\mathcal{C}(f_k) \ge \mathcal{C}(f_{t_0})$ for every $k \ge t_0$. Subtracting the fixed, data-dependent value $\widehat{C}_{\mathrm{ref}}$ proves the claim. □

The proposition connects a realized fit to the tail-separation condition: the barrier lower-bounds its complexity, monotonicity preserves that cost, and the estimated reference determines the remaining excess.

Proposition 6 (Finite-sample barrier)**.** *Let $\mathcal{C} = |\cdot|_\alpha$, fix $\mu > 0$, and condition on a noisily labeled sample $\mathbf{x}$. Suppose its true class $c(\mathbf{x})$ has N_c additional clean samples drawn i.i.d. from $P_{c(\mathbf{x})}$, whose density is bounded below on a neighborhood of $\mathbf{x}$ in the (d'-dimensional) effective feature space. Let $r_{N_c} = d(\mathbf{x}, \mathbf{x}_{\mathrm{nn}})$ be the distance to the nearest of these distinct clean samples. Then $r_{N_c} = O_P\big(N_c^{-1/d'}\big)$, and any f satisfying $\|f(\mathbf{x}) - f(\mathbf{x}_{\mathrm{nn}})\| \ge \mu$ obeys*

$$\mathcal{C}(f) \;\ge\; \frac{\mu}{r_{N_c}^{\alpha}} \;=\; \Omega_P\big(N_c^{\alpha/d'}\big).$$

Exact interpolation of the clean and noisy one-hot labels yields $\mu = \|c(\mathbf{x}) - \tilde{c}(\mathbf{x})\|$; correct argmax predictions alone require an additional uniform output-margin premise to supply such a μ. If such a function occurs at step t_0 of a monotone-complexity trajectory, then for every $k \ge t_0$,

$$\mathcal{C}(f_k) - \widehat{C}_{\mathrm{ref}} \;\ge\; \frac{\mu}{r_{N_c}^{\alpha}} - \widehat{C}_{\mathrm{ref}}.$$

In particular, a growing persistent margin $\delta_C = \Omega_P(N_c^{\alpha/d'})$ follows when $\widehat{C}_{\mathrm{ref}} = o_P(N_c^{\alpha/d'})$; without a condition on the reference, the proposition guarantees growth of the fitting complexity but not of its difference from the reference.

Proof. The pairwise Hölder bound in Corollary 1 gives $\mathcal{C}(f) \geq \mu / r_{N_c}^{\alpha}$. The density lower bound supplies a constant $q > 0$ such that, for each fixed $A > 0$ and sufficiently large N_c, a ball of radius $AN_c^{-1/d'}$ around $\mathbf{x}$ has probability mass at least $qA^{d'}/N_c$. Independence therefore gives $\Pr(r_{N_c} > AN_c^{-1/d'}) \leq (1 - qA^{d'}/N_c)^{N_c} \leq e^{-qA^{d'}}$. Letting A increase establishes $r_{N_c} = O_P(N_c^{-1/d'})$. Monotone trajectory complexity preserves the complexity lower bound after step t_0; the final rate follows after subtracting a reference of smaller asymptotic order. □

Remark 5 (Sample density and reference growth). Proposition 6 predicts increasing fitting complexity as clean samples become denser, provided output separation remains bounded below. A growing excess over the reference additionally requires $\widehat{C}_{\mathrm{ref}} = o_P(N_c^{\alpha/d'})$. The shock protocol in Section 2.1 measures the observed excess directly; it does not estimate μ and hence tests the trajectory signature rather than this scaling rate. The relevant finite-data quantity is the nearest-conflicting-neighbor distance.

B.9 Exact Analysis in a Tractable Two-Mode Model

We now give a linearized, neural-tangent-kernel-style model in which explicit spectral conditions yield a clean-prediction window, monotone complexity, and a persistent tail excess. The construction represents a fast clean mode and a slower localized noise mode, motivated by spectral-bias analyses (Rahaman et al., 2019; Xu et al., 2019).

Setting: Binary labels $Y = \{+1, -1\}$; a noisily labeled point $\mathbf{x}_0$ with clean concept $c(\mathbf{x}_0) = +1$ and noisy label $\tilde{c}(\mathbf{x}_0) = -1$. In this subsection, f_t denotes the scalar signed logit margin of the binary classifier, rather than the probability-vector predictor used in the general framework. In the linearized regime, the trajectory at $\mathbf{x}_0$ evolves as a sum over eigenmodes with rates given by the kernel eigenvalues; we retain the two relevant ones:

$$f_t(\mathbf{x}_0) \;=\; a\big(1 - e^{-\lambda_1 t}\big) \;+\; b\big(1 - e^{-\lambda_2 t}\big), \qquad f_t \;=\; a\big(1 - e^{-\lambda_1 t}\big)\,\varphi_1 \;+\; b\big(1 - e^{-\lambda_2 t}\big)\,\varphi_2,$$

where φ_1 is the smooth population mode through which the surrounding clean class pulls $f(\mathbf{x}_0)$ toward $+1$ (coefficient $a > 0$), φ_2 is the localized mode through which the point-mass noisy term completes the fit ($a + b = -1$, so $b = -(1 + a) < -a$), and $\lambda_1 > \lambda_2 > 0$ is the spectral gap (low-frequency modes fit first). The modes have Lipschitz constants $0 < L_1 = \mathrm{Lip}(\varphi_1) \ll L_2 = \mathrm{Lip}(\varphi_2)$ (the noise mode is spatially localized). As part of the two-mode model, assume their variation sectors are separated so that

$$\mathrm{Lip}(f_t) = \max\big(a(1 - e^{-\lambda_1 t})L_1,\ |b|(1 - e^{-\lambda_2 t})L_2\big).$$

Proposition 7 (Window and separation in the two-mode model)**.** *In the setting above, suppose the spectral-gap condition $\lambda_1/\lambda_2 > |b|/a$ holds and $|b|L_2 > aL_1$. Then:*

(i) (Monotone complexity.) *$t \mapsto \mathrm{Lip}(f_t)$ is nondecreasing.*

(ii) (Single transition.) *$f_t(\mathbf{x}_0) > 0$ on an interval $(0, t^*)$ and $f_t(\mathbf{x}_0) < 0$ for all $t > t^*$, for a unique t^*: the prediction $\mathrm{sign}(f_t(\mathbf{x}_0))$ equals the clean concept on $(0, t^*)$ and transitions exactly once, to the noisy label.*

(iii) (Long window under a widening spectral gap.) *Write $R = \lambda_1/\lambda_2$ and $u^* = \lambda_2 t^*$. Then u^* is the unique positive root of $a(1 - e^{-Ru}) = |b|(1 - e^{-u})$, and as $R \to \infty$,*

$$u^* \longrightarrow \log \frac{|b|}{|b| - a} = \log(1 + a).$$

Hence $t^* \sim \log(1 + a)/\lambda_2$ in the large-gap limit, while the clean mode saturates on the shorter timescale $1/\lambda_1$.

(iv) (Persistent excess.) *With the ideal clean-mode reference $C_0 = aL_1$, the excess $\big(\mathrm{Lip}(f_t) - C_0\big)_+$ is nondecreasing and converges to $|b|L_2 - aL_1 > 0$: it crosses the reference at t_C defined below and never reverts.*

(v) (A gate-aligned subwindow.) *Define*

$$t_C = -\frac{1}{\lambda_2}\log\left(1 - \frac{aL_1}{|b|L_2}\right) > 0.$$

For any $0 < \tau < t^\dagger := \min\{t^*, t_C\}$, predictions are clean and $\mathrm{Lip}(f_t) - C_0 \le 0$ on $[\tau, t^\dagger)$. For any $t_0 > \max\{t^*, t_C\}$, predictions are noisy and

$$\mathrm{Lip}(f_t) - C_0 \ge \delta_C(t_0) := |b|(1 - e^{-\lambda_2 t_0})L_2 - aL_1 > 0 \qquad (t \ge t_0).$$

Thus the prediction-window and shared-complexity-gate conditions hold on a nonempty subwindow with $\epsilon_C = 0$; a lower bound on the separate fitted-loss factor z_k is still required.

Proof. (i) Both factors $a(1 - e^{-\lambda_1 t})L_1$ and $|b|(1 - e^{-\lambda_2 t})L_2$ are nondecreasing in t, hence so is their maximum.

(ii) Write $h(t) = f_t(\mathbf{x}_0)$; $h(0) = 0$ and $h(\infty) = a + b = -1 < 0$. Its derivative is

$$h'(t) \;=\; a\lambda_1 e^{-\lambda_1 t} + b\lambda_2 e^{-\lambda_2 t} \;=\; e^{-\lambda_2 t}\big(\, a\lambda_1 e^{-(\lambda_1 - \lambda_2)t} + b\lambda_2 \,\big).$$

The bracket is strictly decreasing (since $\lambda_1 > \lambda_2$), starts at $a\lambda_1 + b\lambda_2 > 0$ by the spectral-gap condition, and tends to $b\lambda_2 < 0$; hence h' has exactly one zero t_{peak}, with h strictly increasing before it and strictly decreasing after. Thus $h > 0$ on $(0, t_{\mathrm{peak}}]$, and on $(t_{\mathrm{peak}}, \infty)$ the strictly decreasing h crosses zero exactly once, at t^*, after which $h < 0$ permanently.

(iii) Substitute $u = \lambda_2 t$ and $R = \lambda_1/\lambda_2$ into the positive-root equation $h(t^*) = 0$ to obtain

$$a(1 - e^{-Ru^*}) = |b|(1 - e^{-u^*}).$$

Part (ii) gives uniqueness of this positive root. As $R \to \infty$, the left side converges to a for every fixed $u > 0$. Values immediately below and above $\log(|b|/(|b| - a))$ have opposite limiting signs in the root equation, so uniqueness brackets u^* and gives $u^* \to \log(|b|/(|b| - a)) = \log(1 + a)$. Thus $t^* \sim \log(1 + a)/\lambda_2$, whereas the clean-mode timescale is $1/\lambda_1 = 1/(R\lambda_2)$; their ratio diverges with R.

(iv) The clean branch satisfies $a(1 - e^{-\lambda_1 t})L_1 \le aL_1 = C_0$, so it contributes no positive excess above C_0. Therefore

$$\big(\mathrm{Lip}(f_t) - C_0\big)_+ = \big(|b|(1 - e^{-\lambda_2 t})L_2 - aL_1\big)_+,$$

which is nondecreasing and converges to $|b|L_2 - aL_1 > 0$.

(v) The localized branch equals C_0 exactly at t_C and is smaller before that time. Together with part (ii), this gives the clean, below-reference interval $(0, t^\dagger)$. For $t_0 > \max\{t^*, t_C\}$, part (ii) gives the noisy tail and monotonicity of the localized branch gives, for every $t \ge t_0$,

$$\mathrm{Lip}(f_t) - C_0 = |b|(1 - e^{-\lambda_2 t})L_2 - aL_1 \ge \delta_C(t_0) > 0.$$

This is shared-gate separation for the ideal reference; the model does not specify the classwise fitted fraction used to lower-bound z_k. □

Remark 6. The model supplies a clean-prediction window and a monotone localized complexity contribution under its spectral and variation-sector assumptions. It uses an ideal clean-mode reference and leaves the fitted-loss factor to be controlled separately. The aggregate loss-shock response in nonlinear training is evaluated empirically in Section 2.1.

B.10 Finite-Time Guarantee for Gated Concept Recovery

For a fixed sample $\mathbf{x}$, the fit factor z_k depends on its predicted class, while the complexity factor g_k is the same global sequence for every sample. The guarantee below concerns their sample-specific product $w_k = z_k g_k$; only the complexity gate is shared.

Proof of Theorem 2: Let $W_c = \sum_{k:\hat{\mathbf{y}}_k = c(\mathbf{x})} w_k$ and $W_{\neg c} = \sum_{k:\hat{\mathbf{y}}_k \neq c(\mathbf{x})} w_k$. On $\mathcal{K}_c(\mathbf{x})$, Assumption 5 gives $g_k \geq e^{-\lambda \epsilon_C}$, so

$$W_c \geq \sum_{k \in \mathcal{K}_c(\mathbf{x})} w_k \geq m z_{\text{in}} e^{-\lambda \epsilon_C}.$$

The incorrect mass before t_0 is $B_{\text{early}}(\mathbf{x})$ by definition. For $k \geq t_0$, shared-gate separation gives $g_k \leq e^{-\lambda \delta_C}$ and $z_k \leq 1$, hence

$$W_{\neg c} \leq B_{\text{early}}(\mathbf{x}) + n_{\text{tail}} e^{-\lambda \delta_C}.$$

Because $1 - P_T(c(\mathbf{x})) = W_{\neg c}/(W_c + W_{\neg c})$ and this ratio decreases with W_c and increases with $W_{\neg c}$, substitution proves Equation (4). If the displayed clean lower bound exceeds the total incorrect upper bound, it also exceeds the mass assigned to each individual competing class, so the argmax returns $c(\mathbf{x})$.

The next proposition makes the in-window constant z_{in} explicit.

Proposition 8 (Explicit gate constants)**.** *Fix a window step $k \in \mathcal{K}_c(\mathbf{x})$ and write $\mathbf{y} = c(\mathbf{x})$ for the predicted class. Assume $|I_{\mathbf{y}}| > 0$, and decompose $I_{\mathbf{y}} = I_{\mathbf{y}}^{\text{clean}} \cup I_{\mathbf{y}}^{\text{noisy}}$ (samples carrying observed label $\mathbf{y}$ whose label is correct, resp. corrupted), with empirical noisy fraction $\hat{\eta}_{\mathbf{y}} = |I_{\mathbf{y}}^{\text{noisy}}|/|I_{\mathbf{y}}|$. Suppose at least a $1 - \beta$ fraction of $I_{\mathbf{y}}^{\text{clean}}$ is correctly classified at every step in the window. Then*

$$z_k \;\geq\; (1 - \hat{\eta}_{\mathbf{y}})\,(1 - \beta), \qquad z_{\text{in}} \;\geq\; (1 - \hat{\eta}_{\mathbf{y}})\,(1 - \beta).$$

Moreover, assume independent class-conditional corruption within each observed class and $|I_{\mathbf{y}}| > 0$ for every class included in the following union bound. With probability at least $1 - \zeta$ over the label-corruption draw, simultaneously for those classes,

$$\hat{\eta}_{\mathbf{y}} \;\leq\; \tilde{\eta}_{\mathbf{y}} + \sqrt{\frac{\log(|Y|/\zeta)}{2\,|I_{\mathbf{y}}|}},$$

where $\tilde{\eta}_{\mathbf{y}}$ is the population probability that a sample with observed label $\mathbf{y}$ is mislabeled. In particular, under balanced symmetric noise and with $n_{\min} = \min_{\mathbf{y}} |I_{\mathbf{y}}| \geq 1$,

$$z_{\text{in}} \;\geq\; \left(1 - \eta - O\left(\sqrt{\frac{\log |Y|}{n_{\min}}}\right)\right)(1 - \beta).$$

Proof. The fitted clean members alone contribute at least $|I_{\mathbf{y}}^{\text{clean}}|(1 - \beta)$ to the numerator of Equation (2); any fitted corrupted members only increase the gate. Dividing by $|I_{\mathbf{y}}|$ gives the deterministic bound. For the concentration statement, assume independent class-conditional corruption within each observed class. Hoeffding's inequality applied to the $|I_{\mathbf{y}}|$ corruption indicators, followed by a union bound over $\mathbf{y} \in Y$, gives the stated deviation. Under balanced symmetric noise, $\tilde{\eta}_{\mathbf{y}} = \eta$. □

Remark 7 (Dependence on horizon, sample size, and noise rate). Noise rate and class sample size control the lower bound on useful mass through z_{in}. The finite-sample barrier can enlarge δ_C when the reference grows more slowly than the cost of fitting a conflict (Proposition 6). The horizon determines the number of tail votes, and reference quality enters through $\delta_C - \epsilon_C$. Equation (4) keeps these dependencies explicit.

Corollary 5 (Finite-horizon tail control)**.** *Consider extending a trajectory satisfying the tail clause of Assumption 4. If $z_k \to 1$ on the memorized tail, loss-only aggregation $w_k = z_k$ converges to the observed noisy label. For the gated aggregate, let*

$$M_\lambda = m z_{\text{in}} e^{-\lambda \epsilon_C}$$

and write $B_{\text{early}}^{(\lambda)}(\mathbf{x})$ to emphasize that the early mass in Theorem 2 depends on the same gate sharpness. Fix $\kappa < 1$ and $\varepsilon > 0$ with $\kappa + \varepsilon < 1$. At any finite horizon T, every $\lambda \geq 0$ satisfying both

$$B_{\text{early}}^{(\lambda)}(\mathbf{x}) \leq \kappa M_\lambda,$$

$$\lambda \geq \frac{1}{\delta_C - \epsilon_C} \log \frac{n_{\text{tail}}}{\varepsilon m z_{\text{in}}}$$

makes the tail contribution at most εM_λ and guarantees recovery by Theorem 2. Thus the tail-suppression requirement grows only logarithmically with the horizon, but recovery additionally requires early-mass feasibility at the chosen λ.

Proof of Corollary 5: For the loss-only aggregate, the tail clause of Assumption 4 fixes $\hat{\mathbf{y}}_k = \tilde{c}(\mathbf{x})$ after t_0. If $z_k \to 1$, its cumulative tail mass is $T - t_0 + o(T)$, whereas all pre-tail labels receive only $O(1)$ mass. Hence $P_T(\tilde{c}(\mathbf{x})) \to 1$.

For the gated aggregate, divide the tail upper bound by the clean-window lower bound:

$$\frac{n_{\text{tail}} e^{-\lambda\delta_C}}{m z_{\text{in}} e^{-\lambda\epsilon_C}} = \frac{n_{\text{tail}}}{m z_{\text{in}}} e^{-\lambda(\delta_C - \epsilon_C)}.$$

The lower bound on λ makes this ratio at most ε. The simultaneous early-mass condition gives total incorrect mass at most $(\kappa + \varepsilon)M_\lambda < M_\lambda$, so Theorem 2 returns the clean label. Both conditions concern the same λ; increasing sharpness improves the tail ratio but can also reduce surviving clean-window mass when $\epsilon_C > 0$. The resulting feasibility condition depends on both reference quality and early incorrect mass.

Remark 8 (Rate gating on a memorized tail). If $\Delta_k \to 0$ on a memorized tail, the rate gate $g_k = \exp(-\lambda\Delta_k)$ tends to one there. Under the same tail assumptions used above, its aggregate therefore has the same asymptotic failure as the loss-only aggregate. The level gate remains small under shared-gate separation, with its finite-horizon guarantee governed by Corollary 5.

C Experiments Continued

C.1 Implementation Details

Substrate protocol: We compare NLS (Wei et al., 2022a), NoiseGPT (Wang et al., 2024), PLS-LSA+ (Albert et al., 2024), and Delora (Yuan et al., 2025) with their CRT augmentations. The reported trajectory experiments on MNIST and CIFAR train MLP heads on pretrained ResNet34 (He et al., 2016) features obtained from Wei et al. (2022a); T-Finance uses pretrained features from Tang et al. (2022). Training continues to zero observed-label classification error. A high initial learning rate accelerates fitting and is decayed by a factor of 1.1 every 1500 epochs until this criterion is reached; the target epoch–learning-rate configuration is $(30579,\ 1.6\text{e}{-3})$. For Clothing1M, we follow the standard benchmark baseline protocol (Xiao et al., 2015; Li et al., 2020). Loss-only CRT and gated CRT use the corresponding recorded prediction histories; both are variants of our method.

Complexity measurement: Neural persistence (Rieck et al., 2019) is computed once per epoch and supplies one global scalar for all samples. MLPs use the original weight-graph formulation. ResNet and GNN measurements follow the same weight-graph construction and aggregation principle; the original work describes the convolutional construction through sparse unrolled layer matrices. These measurements provide within-run structural scores. Relating their level separation to the functional barrier requires Assumption 2, whose calibration error is not estimated in these experiments. The dedicated MNIST complexity analyses (Sections 2.1, 2.2 and C.4) use a three-layer ReLU MLP $[128, 64, 10]$, to which the original formulation applies directly.

Plateau detection and gate hyperparameters: We estimate $\widehat{M}_{\text{ref}}$ by finding the earliest window of ω consecutive epochs whose persistence increments Δ_k lie below threshold ϵ_Δ, then averaging persistence over that window. By default, $\omega = 10$ and ϵ_Δ is the 25th percentile of observed increments. When the median persistence excess after the detected window is positive, we choose λ so that it gives gate value 0.1. MLP and GNN increments are used without smoothing; ResNet increments are first smoothed with an exponential moving average of half-life approximately 5 epochs. Sensitivity results appear in Section C.6. The localized-noise sweep includes the auxiliary-reference exception described in Section C.3.

Fitted-loss gate: For a sample currently predicted as class y, the gate z_k uses the fraction of training examples with observed label y that the model classifies correctly against that observed label (Equation (2)). It is computed from recorded argmax predictions and introduces neither a held-out set nor a loss threshold. Where training-length conventions require a noise-rate estimate, we use the estimator of Liu et al. (2023) (Table 8).

C.2 Additional Trajectory Diagnostics

Matched sample additions: To measure the response to sample type while controlling sample count, we progressively add clean or noisy samples to a base set of 20,000 MNIST examples and measure persistence at convergence. Figure 5 compares these additions and the 20% noise-injection protocol with clean-only and clean-augmented references. Noisy additions produce larger persistence values than matched clean additions, with greater separation as more samples are added.

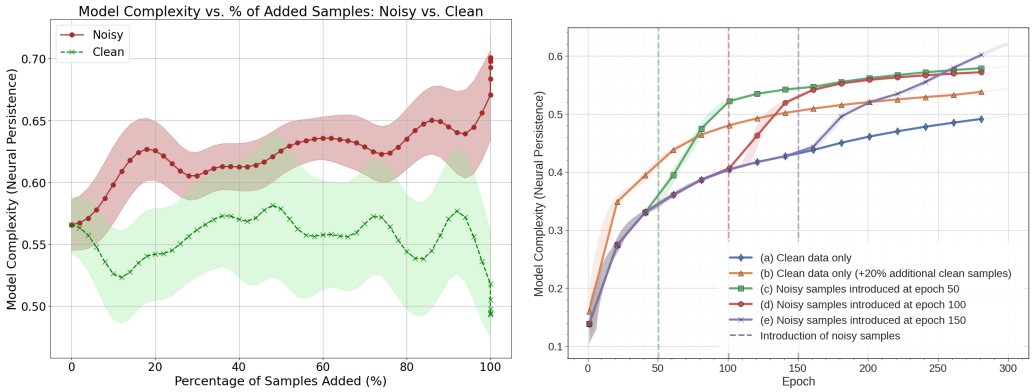


Figure 5: **Persistence under matched data additions on MNIST.** Left: persistence at convergence as clean (green) or noisy (red) samples are added to a base set of 20,000. Right: persistence after adding 20% noisy samples at epochs 50, 100, or 150, compared with clean-only training and training with 20% additional clean samples. These finite trajectories measure a structural surrogate; divergence of functional complexity is a separate theoretical statement.

Dataset-size scaling: For Hölder complexity, Proposition 6 bounds the cost of maintaining an N-independent positive output separation from increasingly dense clean neighbors by order $N^{\alpha/d'}$. An increasing excess over a reference additionally requires the reference to grow more slowly. We repeat the shock protocol at fixed noise fraction, vary the base dataset size, and measure the asymptotic persistence excess over a matched clean-augmented control. The measured excess increases as $0.045 \to 0.061 \to 0.083 \to 0.108$ for $N \in \{5\text{k}, 10\text{k}, 20\text{k}, 40\text{k}\}$, with log–log slope approximately 0.4. This ordering is consistent with the geometric motivation. The slope describes the measured surrogate; estimating the Hölder exponent would additionally require output-margin and surrogate-calibration measurements.

Per-sample loss trajectories: We train a three-layer MLP ($[128, 64, 10]$, ReLU) on a two-feature, two-class synthetic dataset and on MNIST, each with 10% symmetric noise, and record per-sample losses for 500 epochs. Figure 3 shows early separation between clean and noisy samples on the synthetic data and persistent overlap on MNIST. Together with the injection experiment, this motivates evaluating information beyond a single loss value; aggregate loss and samplewise separability are distinct measurements.

C.3 Controlled Entanglement Protocol

At a fixed 20% noise rate on CIFAR-100, a corrupted label is assigned to the nearest confusable class within the same superclass with probability β and uniformly at random with probability $1 - \beta$. The sweep uses $\beta \in \{0, 0.15, 0.30, 0.50, 0.70, 0.85, 1.0\}$. Increasing β makes flips more local in class structure. The horizontal coordinate in Figure 4 measures the resulting scalar-loss separability rather than assuming it from β.

We use the NLS trajectory as the fixed reference and observed-label negative log-likelihood $\mathcal{L}_k(\mathbf{x}) = -\log f_{k,\mathbf{y}}(\mathbf{x})$ as its scalar statistic. Using the true noise mask, we optimize the epoch and ranking direction for AUROC, then separately optimize the epoch, threshold, and direction for F1:

$$\text{AUROC}^{\star}_{\text{loss}} = \max_{k \le T,\, s \in \{-1,+1\}} \text{AUROC}\big(\{s\mathcal{L}_k(\mathbf{x})\}_{\mathbf{x}},\ \text{noise mask}\big),$$

$$\text{F1}^{\star}_{\text{loss}} = \max_{k,\theta,s \in \{-1,+1\}} \text{F1}\big(\mathbf{1}\left[s\big(\mathcal{L}_k(\mathbf{x}) - \theta\big) \ge 0\right],\ \text{noise mask}\big).$$

The entanglement coordinate is $1 - \mathrm{AUROC}^{\star}_{\mathrm{loss}}$. The F1 benchmark is an upper envelope of single-epoch thresholds on the recorded NLS losses. It leaves open the performance of multiepoch loss features, margins, mixtures, gradients, and representations.

Clean-only training on the same split separates the confused classes at each β, supporting the recoverable-noise setting of Remark 1. At $\beta = 1$, slow learning of fine-grained classes can obscure the plateau; in this case we estimate $\widehat{M}_{\mathrm{ref}}$ from a low-noise calibration run. The reported sweep therefore includes an auxiliary-reference case. Local class confusions motivate the geometric interpretation: if they reduce conflicting-neighbor distances while preserving output separation, the Hölder lower bound increases. Neither distance nor output margin is measured by β itself.

Sweep outcomes: The average gated margin over the scalar-loss benchmark changes from -0.02 at low entanglement to $+0.21$ at $\beta = 1$, becoming positive near $\beta \approx 0.30$. The oracle benchmark decreases from F1 0.96 to 0.68, while gated F1 is approximately 0.88–0.90 at the localized end of the sweep. This compares the gated prediction histories with the specified scalar-loss benchmark on a fixed reference trajectory.

C.4 Complexity Gap at Low Noise Rates

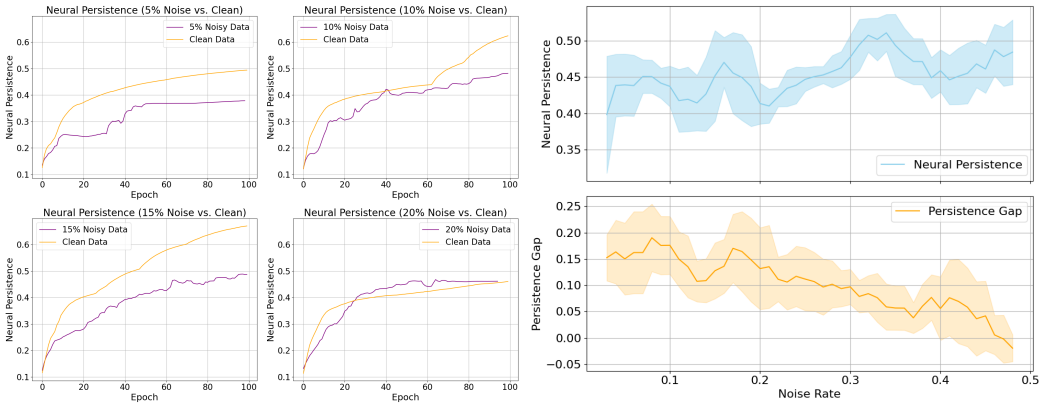


Figure 6: **Persistence ordering at low noise rates on MNIST.** Left: clean (orange) and noisy (purple) trajectories at 5%, 10%, 15%, and 20% noise over 100 epochs. Right: noisy-model persistence (top) and the clean-minus-noisy persistence gap (bottom) as the noise rate varies. Positive gaps mean that the clean model has greater measured persistence.

Setting and observations: At low noise rates (0–15%) with early stopping at 100 epochs, clean MNIST models can have higher persistence than noisy models (Figure 6, left). At 20%, the displayed trajectories end at similar persistence values. The accompanying noise-rate sweep shows the clean-minus-noisy gap and the corresponding noisy-model persistence (right). The broader sweep shows an overall decrease in the gap and increase in noisy-model persistence, with fluctuations across rates.

Interpretation: The early-stopped, low-noise ordering limits the use of persistence level as a universal indicator of corruption. The functional barrier concerns a maintained positive output separation and transfers to this structural statistic only under calibration; those conditions are not established by early stopping. Related analyses address regularization by small input noise (Bishop, 1995) and noise-assisted optimization (Zhou et al., 2019). Establishing whether analogous mechanisms explain this label-noise ordering, and evaluating gate coverage at very low noise rates, remain open empirical questions.

C.5 Additional Benchmarks

Real human-annotator noise (CIFAR-100N): CIFAR-100N (Wei et al., 2022b) supplies a complementary test with real annotator errors. We compare each base detector with loss-only and gated CRT, and compute the oracle single-epoch scalar-loss benchmark on the NLS reference trajectory (Table 4). The oracle benchmark reaches F1 71.0%, while gated CRT reaches 86.90–88.30% across

Method	Symmetric (%)				Asym. (%)
	10	20	30	40	40
NLS (Wei et al., 2022a)	6.10 ± 0.95	6.59 ± 0.86	7.11 ± 0.98	8.76 ± 1.07	8.95 ± 1.05
+CRT (loss gate only)	3.80 ± 0.11	4.36 ± 0.21	4.87 ± 0.23	5.01 ± 0.31	4.85 ± 0.24
+Gated CRT (ours)	**3.45 ± 0.09**	**3.98 ± 0.16**	**4.34 ± 0.18**	**4.28 ± 0.24**	**4.10 ± 0.19**
NoiseGPT (Wang et al., 2024)	4.89 ± 0.73	5.60 ± 0.81	6.63 ± 0.90	7.21 ± 1.30	7.10 ± 1.25
+CRT (loss gate only)	4.22 ± 0.33	4.55 ± 0.21	4.80 ± 0.19	6.08 ± 0.25	6.39 ± 0.18
+Gated CRT (ours)	**3.90 ± 0.25**	**4.12 ± 0.17**	**4.31 ± 0.15**	**5.22 ± 0.20**	**5.45 ± 0.15**
PLS-LSA+ (Albert et al., 2024)	5.15 ± 0.80	5.80 ± 0.95	7.10 ± 1.13	7.48 ± 1.21	7.50 ± 1.15
+CRT (loss gate only)	4.35 ± 0.12	5.20 ± 0.09	6.71 ± 0.22	6.20 ± 0.25	6.05 ± 0.18
+Gated CRT (ours)	**4.02 ± 0.10**	**4.71 ± 0.08**	**5.95 ± 0.18**	**5.35 ± 0.20**	**5.18 ± 0.14**
Delora (Yuan et al., 2025)	5.30 ± 0.86	5.48 ± 0.71	6.40 ± 0.90	8.40 ± 1.02	8.25 ± 1.15
+CRT (loss gate only)	4.88 ± 0.15	5.26 ± 0.22	5.35 ± 0.23	6.02 ± 0.28	6.30 ± 0.25
+Gated CRT (ours)	**4.50 ± 0.12**	**4.78 ± 0.18**	**4.80 ± 0.19**	**5.15 ± 0.22**	**5.42 ± 0.20**

Table 3: MNIST rectification error (%, lower is better; mean ± standard deviation over 5 runs). Each base detector is compared with its loss-only and complexity-gated CRT augmentations, both variants of our method. Bold marks the lowest mean error within each detector group.

detectors. Base-detector F1 ranges from 64.20% to 67.85%, and loss-only CRT from 68.05% to 70.20%. The comparison supports complementarity to the same scalar-loss benchmark used in the controlled sweep.

Method	Rect. Err. (%) ↓	F1 (%) ↑	Recall (%) ↑
Oracle scalar-loss threshold (F1): 71.0			
NLS (Wei et al., 2022a)	18.90 ± 2.15	64.20 ± 1.85	66.10 ± 2.05
+CRT (loss only)	16.75 ± 0.72	68.05 ± 0.68	70.30 ± 0.75
+Gated CRT	**12.40 ± 0.55**	**87.60 ± 0.48**	**89.20 ± 0.52**
NoiseGPT (Wang et al., 2024)	17.60 ± 1.95	66.80 ± 1.70	65.40 ± 1.88
+CRT (loss only)	15.85 ± 0.68	69.40 ± 0.62	71.05 ± 0.70
+Gated CRT	**12.05 ± 0.52**	**86.90 ± 0.50**	**88.40 ± 0.55**
PLS-LSA+ (Albert et al., 2024)	17.20 ± 2.05	67.10 ± 1.80	66.85 ± 1.95
+CRT (loss only)	15.40 ± 0.65	69.85 ± 0.60	70.60 ± 0.68
+Gated CRT	**11.90 ± 0.50**	**88.05 ± 0.46**	**89.65 ± 0.50**
Delora (Yuan et al., 2025)	16.95 ± 1.78	67.85 ± 1.65	68.90 ± 1.75
+CRT (loss only)	15.10 ± 0.62	70.20 ± 0.58	72.10 ± 0.64
+Gated CRT	**11.75 ± 0.48**	**88.30 ± 0.44**	**90.05 ± 0.48**

Table 4: CIFAR-100N with human-annotator noise: rectification error, detection F1, and recall (%; mean ± standard deviation over 5 runs). The oracle single-epoch scalar-loss threshold on the NLS trajectory reaches F1 71.0%. Bold marks the best mean within each detector group.

Detection F1 across benchmarks: Rectification error evaluates the recovered labels. Table 9 reports the complementary detection F1 for deciding which observed labels are noisy on MNIST, CIFAR-10/100, and T-Finance; Clothing1M F1 appears in Table 7. Symmetric entries average noise rates 10–40%, while asymmetric entries use 40% noise.

CIFAR-10					
Method	**Symmetric (%)**				**Asym. (%)**
	10	**20**	**30**	**40**	**40**
NLS (Wei et al., 2022a)	6.64 ± 0.85	8.05 ± 1.02	10.57 ± 0.98	12.51 ± 1.12	11.72 ± 1.02
+CRT (loss gate only)	4.06 ± 0.12	5.10 ± 0.20	5.85 ± 0.24	6.97 ± 0.37	6.11 ± 0.32
+Gated CRT (ours)	$\mathbf{3.72 \pm 0.10}$	$\mathbf{4.62 \pm 0.16}$	$\mathbf{5.20 \pm 0.20}$	$\mathbf{5.98 \pm 0.30}$	$\mathbf{5.25 \pm 0.26}$
NoiseGPT (Wang et al., 2024)	6.83 ± 0.48	7.42 ± 0.77	8.10 ± 0.83	9.40 ± 0.92	8.98 ± 1.02
+CRT (loss gate only)	4.98 ± 0.06	5.35 ± 0.07	5.02 ± 0.24	6.98 ± 0.28	6.15 ± 0.25
+Gated CRT (ours)	$\mathbf{4.58 \pm 0.05}$	$\mathbf{4.86 \pm 0.06}$	$\mathbf{4.52 \pm 0.19}$	$\mathbf{6.02 \pm 0.23}$	$\mathbf{5.30 \pm 0.20}$
PLS-LSA+ (Albert et al., 2024)	7.55 ± 1.16	9.39 ± 1.31	9.30 ± 1.44	11.31 ± 1.28	10.68 ± 1.14
+CRT (loss gate only)	5.56 ± 0.20	5.91 ± 0.18	6.88 ± 0.22	7.35 ± 0.27	6.90 ± 0.23
+Gated CRT (ours)	$\mathbf{5.12 \pm 0.16}$	$\mathbf{5.36 \pm 0.14}$	$\mathbf{6.12 \pm 0.18}$	$\mathbf{6.32 \pm 0.22}$	$\mathbf{5.95 \pm 0.18}$
Delora (Yuan et al., 2025)	7.61 ± 0.72	8.37 ± 0.96	9.63 ± 1.15	10.71 ± 1.37	9.87 ± 1.20
+CRT (loss gate only)	5.45 ± 0.11	6.28 ± 0.15	6.55 ± 0.23	6.87 ± 0.20	6.70 ± 0.14
+Gated CRT (ours)	$\mathbf{5.02 \pm 0.09}$	$\mathbf{5.70 \pm 0.12}$	$\mathbf{5.85 \pm 0.18}$	$\mathbf{5.90 \pm 0.16}$	$\mathbf{5.78 \pm 0.11}$

CIFAR-100					
Method	**Symmetric (%)**				**Asym. (%)**
	10	**20**	**30**	**40**	**40**
NLS (Wei et al., 2022a)	7.36 ± 1.15	14.21 ± 1.10	15.57 ± 1.38	18.01 ± 1.55	15.41 ± 1.55
+CRT (loss gate only)	6.50 ± 0.25	12.03 ± 0.50	12.98 ± 0.63	14.60 ± 0.71	10.45 ± 0.68
+Gated CRT (ours)	$\mathbf{6.02 \pm 0.20}$	$\mathbf{10.95 \pm 0.40}$	$\mathbf{11.70 \pm 0.50}$	$\mathbf{12.85 \pm 0.55}$	$\mathbf{9.10 \pm 0.52}$
NoiseGPT (Wang et al., 2024)	8.95 ± 0.93	16.13 ± 1.05	16.67 ± 1.25	17.25 ± 1.55	13.61 ± 1.20
+CRT (loss gate only)	7.98 ± 0.28	14.60 ± 0.45	15.08 ± 0.41	15.51 ± 0.66	12.90 ± 0.71
+Gated CRT (ours)	$\mathbf{7.35 \pm 0.22}$	$\mathbf{13.30 \pm 0.36}$	$\mathbf{13.65 \pm 0.33}$	$\mathbf{13.60 \pm 0.52}$	$\mathbf{11.20 \pm 0.55}$
PLS-LSA+ (Albert et al., 2024)	7.80 ± 1.09	14.05 ± 1.16	16.88 ± 1.40	19.37 ± 1.38	15.80 ± 1.28
+CRT (loss gate only)	7.01 ± 0.28	12.33 ± 0.44	15.50 ± 0.51	15.70 ± 0.71	11.80 ± 0.42
+Gated CRT (ours)	$\mathbf{6.48 \pm 0.22}$	$\mathbf{11.20 \pm 0.35}$	$\mathbf{13.95 \pm 0.41}$	$\mathbf{13.75 \pm 0.55}$	$\mathbf{10.25 \pm 0.34}$
Delora (Yuan et al., 2025)	7.90 ± 0.98	14.20 ± 1.07	15.97 ± 1.22	17.65 ± 1.20	14.27 ± 1.51
+CRT (loss gate only)	7.25 ± 0.15	13.86 ± 0.38	14.22 ± 0.52	14.98 ± 0.44	12.40 ± 0.47
+Gated CRT (ours)	$\mathbf{6.70 \pm 0.12}$	$\mathbf{12.60 \pm 0.30}$	$\mathbf{12.80 \pm 0.42}$	$\mathbf{13.10 \pm 0.35}$	$\mathbf{10.80 \pm 0.38}$

Table 5: CIFAR-10 and CIFAR-100 rectification error (%, lower is better; mean $\pm$ standard deviation over 5 runs). Each base detector is compared with loss-only and gated CRT. Bold marks the lowest mean within each detector group; Asym. denotes asymmetric noise.

C.6 Sensitivity to Gate Choices

Table 10 varies gate sharpness λ and the plateau window ω on MNIST with NLS at 20% symmetric noise. For $\omega \in \{5, 10, 20\}$, errors lie within 0.15 percentage points of the default; the largest error over the full grid is 0.20 points above it. As $\lambda \to 0$, the method recovers loss-only CRT. Extreme sharpness increases error by approximately 0.3 percentage points, consistent with truncating useful predictions too early.

Table 11 replaces neural persistence with weight-displacement norm or path-norm increments. Both alternatives improve over loss-only CRT but have higher mean error than neural persistence. These results show that useful gates can be constructed from several structural signals in this setting. Because an exponential gate depends on the magnitude of the measured excess and its sharpness, the comparison does not establish invariance to arbitrary monotone transformations of a score.

Replacing oracle noise rates with the estimates of Liu et al. (2023) changes error by at most 0.2 percentage points in Table 8. The rate enters through training-length conventions; the fitted-loss gate itself has no noise-rate parameter.

Method	Noise Rates			
	10%	20%	30%	40%
NLS (Wei et al., 2022a)	7.77 ± 1.85	11.55 ± 1.55	16.36 ± 2.78	18.81 ± 2.91
+CRT (loss gate only)	5.35 ± 0.49	9.88 ± 0.51	13.15 ± 0.61	15.33 ± 0.65
+Gated CRT (ours)	**4.75 ± 0.38**	**8.70 ± 0.40**	**11.45 ± 0.48**	**13.05 ± 0.50**
NoiseGPT (Wang et al., 2024)	8.61 ± 1.03	11.02 ± 2.76	14.41 ± 2.66	18.88 ± 3.29
+CRT (loss gate only)	6.27 ± 0.40	10.51 ± 0.65	12.58 ± 0.66	15.27 ± 0.71
+Gated CRT (ours)	**5.55 ± 0.32**	**9.25 ± 0.50**	**10.95 ± 0.52**	**13.00 ± 0.55**
PLS-LSA+ (Albert et al., 2024)	8.22 ± 2.80	12.66 ± 1.79	15.37 ± 2.55	19.89 ± 3.08
+CRT (loss gate only)	6.08 ± 0.39	11.18 ± 0.42	13.86 ± 0.38	15.41 ± 0.41
+Gated CRT (ours)	**5.40 ± 0.30**	**9.85 ± 0.33**	**12.05 ± 0.30**	**13.10 ± 0.32**
Delora (Yuan et al., 2025)	7.56 ± 1.76	11.77 ± 2.88	18.78 ± 2.91	20.92 ± 2.95
+CRT (loss gate only)	6.25 ± 0.40	10.30 ± 0.51	14.80 ± 0.49	16.77 ± 0.53
+Gated CRT (ours)	**5.55 ± 0.32**	**9.05 ± 0.40**	**12.85 ± 0.38**	**14.25 ± 0.42**

Table 6: T-Finance rectification error (%, lower is better; mean ± standard deviation over 5 runs) under structured asymmetric noise. Fraudulent nodes are mislabeled as benign. Bold marks the lowest mean within each detector group.

Method	Rect. Err. (%) ↓	F1 (%) ↑	Recall (%) ↑
NLS (Wei et al., 2022a)	11.55 ± 2.36	88.28 ± 1.60	89.59 ± 1.87
+CRT (loss gate only)	8.31 ± 0.40	91.18 ± 0.28	95.40 ± 0.20
+Gated CRT (ours)	**7.45 ± 0.32**	**92.60 ± 0.24**	**95.90 ± 0.18**
NoiseGPT (Wang et al., 2024)	9.26 ± 1.98	89.21 ± 1.75	87.02 ± 1.80
+CRT (loss gate only)	8.11 ± 0.55	90.27 ± 0.28	91.30 ± 0.66
+Gated CRT (ours)	**7.30 ± 0.44**	**91.75 ± 0.25**	**92.60 ± 0.52**
PLS-LSA+ (Albert et al., 2024)	9.37 ± 2.29	90.08 ± 1.96	88.68 ± 1.80
+CRT (loss gate only)	9.17 ± 0.40	92.06 ± 0.42	90.55 ± 0.29
+Gated CRT (ours)	**8.20 ± 0.33**	**93.10 ± 0.35**	**91.80 ± 0.26**
Delora (Yuan et al., 2025)	9.11 ± 1.58	90.51 ± 1.80	92.55 ± 1.86
+CRT (loss gate only)	9.05 ± 0.55	92.50 ± 0.61	93.95 ± 0.45
+Gated CRT (ours)	**8.15 ± 0.45**	**93.55 ± 0.50**	**94.85 ± 0.36**

Table 7: Clothing1M under native instance-dependent noise: rectification error, detection F1, and recall (%; mean ± standard deviation over 5 runs). Bold marks the best mean within each detector group.

Dataset	20% (S)	40% (S)	40% (A)
MNIST	4.12 (3.98)	4.41 (4.28)	4.22 (4.10)
CIFAR-10	4.75 (4.62)	6.12 (5.98)	5.41 (5.25)
CIFAR-100	11.10 (10.95)	13.02 (12.85)	9.28 (9.10)
T-Finance	–	–	13.22 (13.05)

Table 8: Gated CRT with NLS: rectification error (%) using estimated noise rates (Liu et al., 2023), with oracle-rate results in parentheses. S/A denote symmetric/asymmetric noise; symmetric noise is not evaluated on T-Finance.

Method	MNIST		CIFAR-10		CIFAR-100		T-Finance
	Sym.	Asym.	Sym.	Asym.	Sym.	Asym.	Asym.
NLS (Wei et al., 2022a)	89.20 ± 0.90	86.55 ± 1.31	83.69 ± 0.89	82.55 ± 1.24	84.79 ± 0.95	79.21 ± 1.20	85.60 ± 1.15
+CRT (loss only)	91.55 ± 0.42	94.90 ± 0.35	92.20 ± 0.61	90.44 ± 0.68	85.86 ± 0.51	84.71 ± 0.71	88.20 ± 0.61
+Gated CRT	**92.80 ± 0.30**	**95.70 ± 0.26**	**93.10 ± 0.45**	**91.60 ± 0.50**	**86.95 ± 0.40**	**86.10 ± 0.52**	**90.05 ± 0.45**
NoiseGPT (Wang et al., 2024)	87.20 ± 1.20	86.38 ± 1.35	88.51 ± 1.12	87.38 ± 0.90	85.89 ± 1.25	85.40 ± 1.29	82.70 ± 1.15
+CRT (loss only)	90.26 ± 0.21	89.28 ± 0.45	88.90 ± 0.30	91.77 ± 0.21	87.68 ± 0.55	86.25 ± 0.51	87.31 ± 0.72
+Gated CRT	**91.40 ± 0.24**	**91.20 ± 0.38**	**90.05 ± 0.28**	**92.60 ± 0.20**	**88.75 ± 0.44**	**87.55 ± 0.45**	**88.90 ± 0.55**
PLS-LSA+ (Albert et al., 2024)	89.50 ± 1.05	88.60 ± 1.20	87.32 ± 1.08	85.21 ± 1.35	86.86 ± 1.16	84.31 ± 1.20	83.37 ± 1.45
+CRT (loss only)	91.57 ± 0.40	93.66 ± 0.45	89.70 ± 0.50	90.58 ± 0.42	87.20 ± 0.70	88.50 ± 0.55	87.50 ± 0.60
+Gated CRT	**92.70 ± 0.28**	**94.40 ± 0.33**	**90.80 ± 0.38**	**91.75 ± 0.35**	**88.25 ± 0.52**	**89.30 ± 0.44**	**88.85 ± 0.48**
Delora (Yuan et al., 2025)	89.20 ± 1.36	87.50 ± 1.32	88.50 ± 1.40	89.20 ± 1.27	85.77 ± 1.37	86.30 ± 1.05	84.29 ± 1.22
+CRT (loss only)	89.80 ± 0.56	92.40 ± 0.20	90.57 ± 0.54	89.86 ± 0.35	88.71 ± 0.71	89.33 ± 0.46	89.20 ± 0.51
+Gated CRT	**90.95 ± 0.34**	**93.30 ± 0.22**	**91.60 ± 0.40**	**91.05 ± 0.30**	**89.60 ± 0.55**	**90.20 ± 0.40**	**90.35 ± 0.42**

Table 9: Noise-detection F1 (%, higher is better; mean ± standard deviation over 5 runs). Symmetric columns average noise rates 10–40%; asymmetric columns use 40%. Each base detector is compared with loss-only and gated CRT; bold marks the highest mean within each group.

λ	Plateau window ω			
	5	**10**	**20**	**40**
$\lambda^*/4$	4.09	4.06	4.08	4.14
$\lambda^*/2$	4.02	3.99	4.01	4.07
λ^*	3.99	**3.98**	3.99	4.05
$2\lambda^*$	4.00	3.99	4.02	4.09
$4\lambda^*$	4.08	4.06	4.10	4.18

Table 10: Rectification error (%) versus sharpness λ and plateau window ω on MNIST with NLS at 20% symmetric noise. The default $(\lambda^*, \omega = 10)$ is bold. Errors range from 3.98% to 4.18% across the full grid.

Complexity gate measure	Rect. Err. (%)
Neural persistence (default)	**3.98 ± 0.16**
Weight-displacement norm	4.10 ± 0.18
Path-norm increment	4.15 ± 0.20
No complexity gate (loss only)	4.36 ± 0.21

Table 11: Rectification error (%) on MNIST with NLS at 20% symmetric noise for three complexity signals. Both alternative signals improve over loss-only CRT; neural persistence gives the lowest mean error.

C.7 Practical Considerations

Computational overhead: Gated CRT records per-epoch predictions and classwise observed-label accuracies, measures one global persistence score per epoch, and uses the recorded training trajectory for aggregation. Table 12 reports NLS runtimes: persistence measurement adds 4–7% per epoch, while the full augmented procedure costs 1.3–$1.6\times$ the base run. A compute-matched NLS control receives the additional budget through extended training or additional seeds. Its rectification-error improvement is 0.12–0.30 percentage points, smaller than the gains from CRT in the reported comparisons.

Benchmark	Substrate (h)	Persist./epoch	D+Gated (h)	Overhead	Ctrl. recovers
MNIST	0.42	+4%	0.58	1.38×	0.12
CIFAR-10	1.15	+5%	1.63	1.42×	0.18
CIFAR-100	1.28	+6%	1.94	1.52×	0.22
T-Finance	0.36	+5%	0.52	1.44×	0.20
Clothing1M	6.80	+7%	10.75	1.58×	0.30

Table 12: Runtime with NLS, measured on the same hardware over 5 runs. "Substrate" is the base runtime in hours; "Persist./epoch" gives persistence overhead per epoch. "D+Gated" and "Overhead" give augmented runtime and its ratio to the base run. "Ctrl. recovers" gives the compute-matched NLS control's reduction in rectification error, in percentage points.

Scope of comparison: The benchmark tables measure the benefit of adding CRT to each comparison method at the reported budgets. The NLS ablation compares weighting rules on a fixed training configuration, and the entanglement sweep compares the resulting decisions with the oracle scalar-loss benchmark. The compute control quantifies the benefit of the additional NLS budget; it does not establish equal-compute comparisons for every method.