FORECASTING WHEN TEST-TIME ADAPTATION WILL FAIL

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation (TTA) updates a pre-trained model on unlabeled test data. Many TTA methods recompute batch-normalization (BN) statistics on every test batch. On skewed batches such as pathology slides which are processed one at a time, this can sharply reduce accuracy. We show, in controlled experiments that keep the whole stream class-balanced, that skewed batches alone cause this failure, that recomputing the BN statistics without any gradient step reproduces it, and that the degradation comes from normalizing a batch with statistics dominated by its own class. We turn these findings into a gate that forecasts, before each TTA update, whether adapting on the current batch will help, based on the batch composition predicted by the model without adaptation and its response to the batch's statistics. The forecast is fitted once on held-out labeled source data and needs no target labels. Across CIFAR-10-C, CIFAR-100-C and two pathology datasets, the gate prevents the collapse of every BN-based method we evaluate on class- and slide-ordered streams, retains most of the benefit on shuffled streams, and improves methods designed for skewed streams on pathology. (Code will be made public upon acceptance)

1 INTRODUCTION

Test-time adaptation (TTA) adapts a trained model to shifted unlabeled test data, such as corrupted images or medical images that come from another hospital with different machines and settings. Many TTA methods rely on the batch-normalization (BN) layers, which normalize each feature channel with the mean and variance stored from training. At test time, these methods replace the stored statistics from training with those of the current test batch (Schneider et al., 2020), and most combine this with updating the layers' scale and shift parameters by minimizing the entropy of the model's own predictions, as done by TENT (Wang et al., 2021). Because both the statistics and the entropy update are computed from the current batch alone, adapting on the wrong batch can leave a model worse than no adaptation at all, and standard methods cannot tell in advance when this will happen (Section 3).

TTA is usually evaluated on shuffled test sets, but real-world test data often arrive in a structured order. In pathology, for example, patches are processed slide by slide, and because tissue classes cluster spatially, skewed batches are the norm. This creates batch-level changes in class composition even when the complete stream is balanced. Earlier work has observed that such streams can make adaptation worse than no adaptation at all (Gong et al., 2022; Yuan et al., 2023; Niu et al., 2023; Zhao et al., 2023b), but varies batch composition together with other variables.

In this paper, we show that a skewed batch composition alone can make adaptation harmful, even if the overall stream remains exactly class-balanced. In controlled experiments, we keep class proportions of the overall stream fixed, while only varying how the classes are spread across the individual batches. As the batches become more skewed, TENT's gain over the unadapted model drops from $+24$ to -33 percentage points on CIFAR-10-C. We show that recomputing the batch statistics alone, without changing any weights, already reproduces the failure, so the problem is not specific to entropy minimization. Methods that re-balance the statistics themselves, such as TRIBE (Su et al., 2024a), avoid this failure in the controlled composition sweep, yet can still fall far below the source model on class- and slide-ordered pathology streams (Section 5.3).

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.

Since the degradation originates in the batch itself, much of it can be anticipated before adapting. We therefore propose a simple gate that forecasts, for each incoming batch, whether adapting on it will help or degrade. The forecast reads label-free signals from a frozen copy of the source model (the model before adaptation): how skewed the batch's predicted classes are, and how strongly the model reacts to the batch's normalization statistics, combined with a linear rule fitted on held-out labeled source data, so no target labels are required. When the gate predicts degradation, it skips adaptation and returns the frozen model's prediction. The gate is therefore a safeguard, and it is complementary to methods that correct the normalization statistics themselves (Yuan et al., 2023; Su et al., 2024a).

Our contributions are as follows:

- We show, with a composition sweep that holds the whole stream class-balanced, that skewed batches alone make BN-based adaptation degrade accuracy without any weight update, and trace the failure to normalizing a batch with statistics dominated by its own class.
- We introduce a gate that forecasts, before each update and without target labels, whether adapting on the current batch will help or degrade. It weighs three measurements, whose relative importance differs across methods and datasets, with a linear forecast fitted once on held-out labeled source data. It wraps existing TTA methods at the cost of two extra forward passes per batch, and one calibration set per dataset serves every method we wrap.
- We show on CIFAR-10-C, CIFAR-100-C (Hendrycks & Dietterich, 2019) and two pathology benchmarks, Camelyon17-WILDS (Koh et al., 2021; Bandi et al., 2019) and NCT-CRC-HE-100K→CRC-VAL-HE-7K (Kather et al., 2019), that the gate prevents the collapse of every BN-based method we evaluate on class- and slide-ordered streams. It also substantially improves methods designed for skewed streams on slide- and class-ordered pathology streams (Section 5).

In Section 3, we analyze why BN-based adaptation fails on skewed batches. We present the proposed gate in Section 4, and in Section 5, we evaluate where the gate helps and where it costs.

2 Related Work

Test-time adaptation and batch normalization. BN-Adapt replaces the stored BN statistics with the statistics of the current batch, without updating the model weights (Schneider et al., 2020), and TENT (Wang et al., 2021) also updates the normalization scale and shift by minimizing prediction entropy. Later methods keep this per-batch normalization but add safeguards against unreliable updates. EATA (Niu et al., 2022) skips high-entropy and redundant samples and anchors source-important weights with Fisher regularization to prevent forgetting. SAR (Niu et al., 2023) combines reliable-sample selection with sharpness-aware entropy minimization and recommends batch-agnostic group or layer normalization when batches are small or non-i.i.d. DeYO (Lee et al., 2024a) filters and reweights samples using entropy and pseudo-label probability changes under data augmentations, and SoTTA (Gong et al., 2023) adapts on a class-balanced memory of high-confidence samples to screen out noisy inputs. These safeguard methods choose which samples enter the loss, whereas a skewed batch already harms a method that normalizes it with its own statistics in the forward pass, before any loss is computed (Section 3.2).

Class imbalance and non-i.i.d. streams. A second line of work changes how normalization statistics are formed when batches are class-imbalanced or temporally correlated. Vianna et al. (2025) show that recomputing BN statistics on a single class-imbalanced test batch can fail even without covariate shift, disproportionately harming the majority class. We instead hold the whole stream class-balanced and swap the normalizing statistics between batches. This isolates the cause as normalization with statistics dominated by the batch's own class: equally skewed statistics from a batch dominated by another class keep most of the gain, and most of the loss is carried by the variance, beyond the subtracted class mean through which Vianna et al. (2025) explain the failure (Section 3.3). Hybrid-TTN (Vianna et al., 2025) adapts only channels less sensitive to label shift. NOTE (Gong et al., 2022) and RoTTA (Yuan et al., 2023) address temporally correlated streams, where a skewed batch also sits inside a long run of similar batches. Both combine normalization changes with a

prediction-balanced memory: NOTE uses instance-aware normalization, while RoTTA uses robust moving-average statistics. UnMix-TNS (Tomar et al., 2024) mixes several normalization components, TRIBE (Su et al., 2024a) addresses batch-level and global imbalance and counters both with balanced normalization statistics, DELTA (Zhao et al., 2023a) reweights over-represented pseudo-labels and stabilizes its statistics, and TTN (Lim et al., 2023) interpolates source and test statistics according to channel sensitivity to domain shift. TEMA (Su et al., 2024b) attributes the errors of small i.i.d. batches mainly to their low class diversity and mixes averaged test statistics with source statistics layer by layer. LAME (Boudiaf et al., 2022) leaves the model unchanged and refines its outputs, while LSA (Park et al., 2023) and DART (Jang & Chung, 2025) adapt the model or its predictions to an estimated target label distribution. These methods *correct* for skewed batches. Our gate is complementary: it only decides whether the chosen TTA method runs on the batch and otherwise serves the source model.

Label-free signals for adaptation decisions. Several works estimate accuracy without target labels, from a confidence threshold fit on labeled source data (Garg et al., 2022) or from dataset feature statistics learned on a meta-dataset (Deng & Zheng, 2021). These estimate how accurate a model is, not how much adapting on the next batch would change its accuracy, which is what our gate forecasts. In TTA, AETTA (Lee et al., 2024b) estimates accuracy through disagreement with dropout predictions and uses these estimates to trigger resets, while ASR (Lim et al., 2026) uses prediction concentration to decide when to reset and which layers to restore. RDumb (Press et al., 2023) instead resets to the source weights after a fixed number of steps. However, reset methods act after an update has already been made. Schirmer et al. (2025) sequentially monitor an adapting model's risk without test labels and return an alert when a predefined rule is violated. MORPHEUS (Danilowski et al., 2026) is closer to forecasting: it predicts post-adaptation accuracy per method using entropy-based or neural-collapse-based regressors to choose a TTA method before adapting. In its evaluated protocol, the regressors are fitted on labeled adaptation outcomes from held-in corruption types of the same benchmark, and method selection occurs once per domain. Mutlu et al. (2026) decide per batch whether to adapt with a classifier trained on labeled adaptation outcomes from shifted test batches of the evaluated benchmarks. For a fixed method, our gate also decides before each update whether to adapt on the current batch, but it forecasts the gain of adapting. It reads three label-free measurements from a frozen copy of the source model and weighs them with a linear rule fitted only on labeled source images held out from training (Section 4). Like previous solutions in the TTA literature (Niu et al., 2022; Jang & Chung, 2025; Vianna et al., 2025), it thus needs source data beyond the trained model, here a small labeled calibration set.

3 How Does Batch Composition Affect Test-Time Adaptation?

We first show that skewed batches can degrade adaptation even when the whole stream is class-balanced (Section 3.1). We then show that recomputing BN statistics alone reproduces this degradation (Section 3.2). Finally, we use batch statistics swaps to trace the loss to normalization with batch statistics dominated by the batch's own class (Section 3.3).

Measurements. A model receives an unlabeled test stream $S = (\mathcal{B}_1, \ldots, \mathcal{B}_T)$ of batches, where batch $\mathcal{B}_t$ holds n_t inputs. Each batch is predicted with the current model state, which may then be updated before the next batch arrives. We call methods that normalize each batch with its own statistics before predicting it *BN-based*: BN-Adapt, TENT, SAR-BN (SAR on a BatchNorm model), EATA and DeYO. The composition of batch t, which the methods never see, is its vector π_t of empirical class proportions over the K classes. With π_{train} the class proportions of the training data, $d_t = \mathrm{JS}(\pi_t, \pi_{\text{train}}) \in [0, 1]$ (base-2 Jensen–Shannon divergence) measures how skewed batch t is, and

$$\bar{d}_B = \frac{\sum_t n_t\, d_t}{\sum_t n_t}, \qquad d_S = \mathrm{JS}\Big(\frac{\sum_t n_t\, \pi_t}{\sum_t n_t},\, \pi_{\text{train}}\Big) \tag{1}$$

are the mean batch divergence and the divergence of the whole stream. A stream with $d_S = 0$ is *prior-matched*. Such a stream is class-balanced when π_{train} is uniform. The adaptation gain $\Delta(S) = \mathrm{acc}_{\text{adapted}}(S) - \mathrm{acc}_{\text{source}}(S)$, in percentage points, compares the adapting model with the frozen source model on the same inputs in the same order. $\Delta > 0$ means adaptation helped.

Models and data. We use CIFAR-10-C (Hendrycks & Dietterich, 2019), which applies 15 synthetic corruptions (noise, blur, weather and digital distortions) to the CIFAR-10 test set (Krizhevsky, 2009) at five intensity levels, of which we use the strongest (severity 5). The source model is the Standard WRN-28-10 (Zagoruyko & Komodakis, 2016) of RobustBench (Croce et al., 2021), trained on all 50,000 training images. Unless stated otherwise, a stream holds 2,000 inputs in batches of $B = 200$, with one corruption at fixed severity, so that only the class composition varies. Every method starts from the source model and keeps its state across the stream. True labels are used only to construct and score the streams (Appendix C) and in the diagnostics of Sections 3.2 and 3.3.

3.1 Skewed batches make adaptation fail

We vary how the classes are spread over the batches. We draw the class counts of one batch from a Dirichlet distribution with concentration β and then rotate them cyclically over the K classes, with one rotation per batch. Every class therefore appears equally often within each cycle of K batches, thus the stream is prior-matched ($d_S = 0$). We randomize the order of the rotated batches and the samples within each batch. We then evaluate nine different values for β, eight draws and two seeds for all 15 corruptions. TENT's mean gain drops from $+23.9$ pp on nearly balanced batches ($\beta = 1000$) to -32.7 pp on the most skewed batches ($\beta = 0.01$) (Figure 1), and over all streams the Spearman correlation between $\bar{d}_B$ and gain is -0.79 (Table A12). Since every stream is class-balanced and its batches arrive in random order, the degradation is due to the composition of the batches, instead of the class proportions of the stream or the batch order. On the prior-matched streams, the skew at which adaptation stops helping depends on the corruption (Appendix Figure A2). Neither the overall class balance of the stream nor a single skew threshold can therefore tell us whether adaptation will help.

3.2 BN statistics alone reproduce the degradation

A skewed batch can act on adaptation through two channels, the normalization statistics and the parameter updates. To investigate whether the parameter updates are needed for the degradation we see, we compare TENT against BN-Adapt on the same source model and streams. BN-Adapt recomputes the batch statistics but does not apply an update. TENT's and BN-Adapt's curves almost exactly track each other across the tested skew levels (Figure 1). At the most skewed level they differ by 0.1 pp (-32.7 vs. -32.6 pp, Table A12). On these streams the degradation thus appears in full without any gradient update: it is caused by normalizing the batch, not by applying updates.

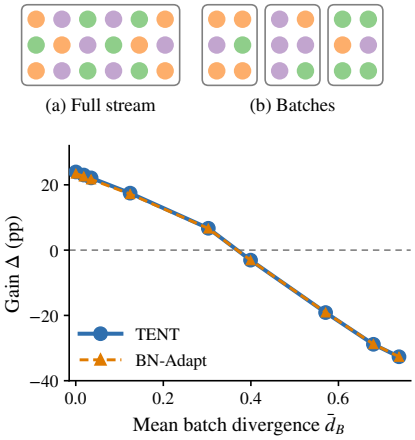


Figure 1: Prior-matched CIFAR-10-C streams ($B = 200$): (a) each full stream is class-balanced ($d_S = 0$), while (b) its batches are skewed, with colors denoting classes. Mean gain Δ over 240 streams per skew level against $\bar{d}_B$.

The other BN-based methods degrade in the same way: at $\beta = 0.01$, SAR-BN, EATA and DeYO also lose 32.6 pp (Table A12 in Appendix E.1). Their sample filters choose which inputs enter the loss, but the loss arises when the batch is normalized, and they still update on every batch after the first, on 65–85% of the inputs under skew (Appendix E.2). Methods that form their normalization statistics differently degrade less or not at all: RoTTA and TRIBE keep about $+20$ pp at both ends of the sweep, whereas Hybrid-TTN keeps only $+8.3$ of its $+20.0$ pp and TEMA falls to -17.3 pp. Separately trained GroupNorm and LayerNorm models on CIFAR-10-C, which do not normalize over the batch, barely respond to batch skew, but they also barely adapt (Appendix E.4).

At $\beta = 0.01$, the loss is similar across the three tested batch sizes: BN-Adapt's gain at $B = 16$ and 64 is within 0.5 pp of its gain at $B = 200$ (Appendix E.2). It also persists over long streams: after 150 batches of $B = 64$ in which no image is used twice, TENT stays within 0.6 pp of BN-Adapt on the most skewed streams, and after 1,000 batches (images reused) BN-Adapt's gain at $\beta = 0.01$

is still -31.5 pp. Gradient updates only add to this loss: by batch 1,000, TENT reaches -35.2 pp (Appendix E.2).

We next test whether the degradation extends to a larger label space. On ImageNet-C (1,000 classes) with a ResNet-50 at $B = 64$, BN-Adapt drops from $+13.5$ pp on shuffled batches to -12.6 pp when each batch holds two or three classes, and its gain drops in all 15 corruptions. The degradation on class-sorted streams is known (Niu et al., 2023; Zhao et al., 2023b), but here the batch order stays shuffled, so only the composition of the batches changes. The tested LayerNorm ViT-B/16, which also differs in architecture and pretraining, shows little change in mean TENT gain with batch composition: $+10.8$ pp on shuffled and $+11.2$ pp on the skewed streams (Appendix E.3).

3.3 Statistics dominated by the batch's own class cause the loss

To locate the cause by intervention, we keep the images and weights of each skewed batch fixed and replace only the statistics that normalize it, in every BN layer (Table 1, Appendix E.1). Each replacement tests one explanation. The statistics of other images with the same class counts test whether the particular images matter. Those of the previous batch, equally skewed but toward a different class, test whether any skew hurts. Those of a class-balanced batch test whether balance restores the gain. Replacing only the variance or only the mean by its balanced value tests which of the two carries the effect. These are offline diagnostics and labels are used only to build the reference batches.

The same class counts reproduce the loss (-32.6 pp at $\beta = 0.01$) and balanced statistics restore the gain ($+23.2$ pp). Therefore, the loss comes from the class composition of the statistics, not from particular images. The degradation appears when a batch is normalized with statistics dominated by its own class: equally skewed statistics from a batch dominated by a different class keep $+19.3$ pp, 83% of the balanced-statistics gain. Replacing the variance alone recovers 80% of this gap and the mean alone 26%. Most of the effect therefore lies in the variance. At $\beta = 0.01$, where the dominant class makes up 95% of a batch on average, BN-Adapt assigns only 21% of its predictions to that class, against 54% for the source model. Normalizing with its own statistics pushes predictions away from the class that dominates the batch.

Table 1: Statistics swap on the streams of Figure 1: gain Δ (pp) when each batch is normalized with the listed statistics, with images and weights fixed and no gradient step.

Statistics	β=1000	0.1	0.01
Own (BN-Adapt)	+23.5	−19.1	−32.6
Same class counts, other images	+23.1	−19.4	−32.6
Other dominant class	+23.1	+19.7	+19.3
Balanced	+23.1	+23.1	+23.2
Balanced variance only	+23.3	+15.3	+11.9
Balanced mean only	+23.4	−8.0	−17.9

4 Method: Forecasting When to Adapt

Section 3 links the degradation of BN-based methods to normalizing a skewed batch with its own statistics. At deployment, true class composition is unavailable, so we instead measure the frozen model's predictions and its response to current-batch normalization before adapting. On a labeled calibration set of held-out source images, we fit a linear forecast of the per-batch gain of adapting, freeze it, and deploy it as an online gate that adapts only when the forecast gain is positive (Figure 2).

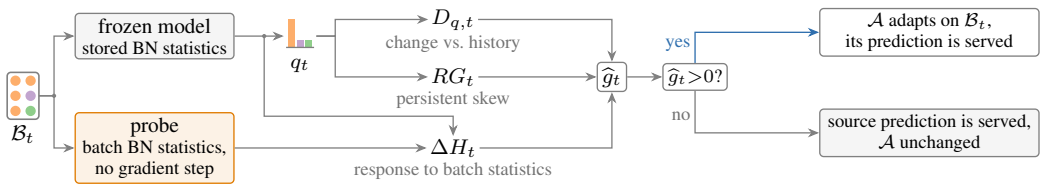


Figure 2: The gate. Before any update, a frozen observer measures the batch: the probe's entropy response ΔH_t to the batch's own BN statistics, and the deviations $D_{q,t}$ and RG_t of the predicted-class histogram q_t from the stream history and from a fixed reference. Which of them signals harm differs between methods and datasets, so a linear forecast $\widehat{g}_t$ of the per-batch gain of adapting weighs all three; it is fitted once on labeled source images under synthetic shifts, then frozen. The source model serves the first batch.

Setup. We consider the unlabeled test stream S of Section 3 and an existing TTA method $\mathcal{A}$. A frozen source model f_{θ_0} and an adapting model are initialized from the same parameters. Before applying $\mathcal{A}$, the gate chooses $a_t \in \{0, 1\}$ from measurements on the current and previous batches. When $a_t = 1$, we apply $\mathcal{A}$, use its predictions and keep its updated state. When $a_t = 0$, we serve the frozen source predictions and leave the adapting model, its optimizer and all adaptation state unchanged, including BN statistics, memory and counters. We call $\mathcal{A}$ *native* when it adapts on every batch after the first and *gated*, the proposed method, when the gate decides. The *on rate* is the fraction of batches after the first on which the gate chooses to adapt. The gate forecasts the per-batch gain $g_t = \mathrm{acc}_{\mathrm{adapted}}(\mathcal{B}_t) - \mathrm{acc}_{\mathrm{source}}(\mathcal{B}_t)$, both as fractions. The forecast is fitted on a *calibration set* of labeled source images held out from training and checkpoint selection. No target batch enters the fit.

Forecast function. The gate forecasts g_t with an affine function of three label-free measurements:

$$\widehat{g}_t = b_0 - b_H \Delta H_t - b_D D_{q,t} - b_R R G_t, \qquad b_H, b_D, b_R \geq 0. \tag{2}$$

ΔH measures the frozen model's entropy response to normalizing the current batch with its own statistics, the operation that Section 3.3 identifies as the cause of the degradation. We add two composition proxies: D_q measures how much the current batch deviates from the prediction history, and RG measures deviation from a fixed reference histogram, weighted by a competence proxy based on prototype distances. All four coefficients are fitted once per dataset, method and batch size on the calibration set and then frozen. The three measurements are large together on most degrading batches, so we constrain their weights to be nonnegative, which keeps correlated measurements from taking opposite signs, and the fit may set a weight to zero (Appendix F.4). Since the skew at which adaptation stops helping depends on the corruption (Section 3.1), we fit how the measurements trade off rather than threshold a single skew measure. For $t > 1$, the model adapts when $\widehat{g}_t > 0$ and serves the source predictions otherwise. The source model always predicts the first batch, and $\mathcal{A}$ does not adapt on it, since D_q has no preceding batches to compare against yet. We use the frozen model's predicted-class histogram q_t as a proxy for the batch composition: $q_{t,c}$ is the fraction of inputs in $\mathcal{B}_t$ that the frozen model assigns to class c.

Entropy change under batch normalization. The *probe* is the frozen model run with the current batch's BN statistics in place of its running statistics. We define $\Delta H_t = H_t^{\mathrm{BN}} - H_t^{\mathrm{source}}$, where H_t^{source} and H_t^{BN} are the mean entropies, divided by $\log K$, of the frozen model's and the probe's predictions over $\mathcal{B}_t$. The probe typically becomes less confident than the frozen model on a label-skewed batch and changes little or gains confidence under shift alone (Appendix F.3). q_t cannot separate the two, since a corruption can also concentrate the frozen model's predictions (Section 5.4).

History deviation. We compare q_t with the pooled predictions of all preceding batches:

$$D_{q,t} = \|q_t - h_t\|_2^2 - E_{0,t}, \qquad h_t = \frac{\sum_{s<t} n_s q_s}{\sum_{s<t} n_s}, \tag{3}$$

where $E_{0,t}$ is the expected squared deviation under random regrouping of the pooled predictions into groups of the same sizes. Subtracting $E_{0,t}$ keeps $D_{q,t}$ near zero under pure sampling noise.

Reference deviation. History deviation misses persistent skew once all preceding batches share a similar composition. We compare q_t against a fixed reference $r = \mathbf{C}_{\mathrm{source}} \pi_{\mathrm{train}}$, what the frozen model would predict under the source-training class proportions, through $G_t = \|q_t - r\|_2^2 - V_t$. Here $\mathbf{C}_{\mathrm{source}}$ is the frozen model's confusion matrix on clean calibration images, with $(\mathbf{C}_{\mathrm{source}})_{ij} = P(\text{predict } i \mid \text{true } j)$, and V_t subtracts the sampling variance of q_t (closed forms for V_t and $E_{0,t}$ in Appendix F). G_t also responds to corruption, which can shift q_t even on a prior-matched batch. The forecast therefore uses $RG_t = A_t G_t$, which discounts G_t when the frozen model is likely wrong: the competence proxy $A_t \in [0, 1]$ decreases with the batch-mean cosine distance between each input's features and the source prototype of its predicted class (Appendix F.1).

Calibration. We fit the four coefficients on labeled streams built from the calibration set, where the gain of adapting on each batch can be measured: each image receives a synthetic shift implemented separately from the test corruptions (Table A15), and the images are arranged into shuffled, temporally drifting, class-ordered and long-tail streams that share no images with the test streams.

We run $\mathcal{A}$ natively on each stream and fit $\widehat{g}_t$ to the measured gains by weighted least squares with nonnegative slopes (Appendix F.1). The gains are exact for stateless methods such as BN-Adapt and approximate for stateful ones; Appendix F.3 tests the forecast on the states that gated TENT actually visits.

Deployment. A declined batch still enters the prediction history used by D_q. The gate decides before the update and serves the source model on declined batches because a BN-based method would already normalize with that batch's statistics (Section 3.2), a loss that a later reset cannot undo. The gate adds two forward passes per batch and no backward pass, one with stored statistics and one with current-batch statistics, and keeps one extra copy of the source model. The decision threshold stays at zero, where adapting and serving the source model have equal forecast accuracy, and is never tuned on target data.

5 EXPERIMENTS

Datasets and models. We evaluate on CIFAR-10-C and CIFAR-100-C at severity 5 with a WideResNet-28-10 and a ResNeXt-29, on Camelyon17-WILDS hospital 2 with a DenseNet-121 trained from scratch on the source hospitals, and on the NCT-CRC-HE-100K→CRC-VAL-HE-7K transfer with an ImageNet-initialized ResNet-50. Unlike the RobustBench model of Section 3, our CIFAR models hold out 5,000 of the 50,000 training images as the calibration set (Appendix B).

Methods. We wrap five BN-based methods: BN-Adapt, TENT, SAR-BN, EATA, and DeYO (Schneider et al., 2020; Wang et al., 2021; Niu et al., 2023; 2022; Lee et al., 2024a). Each native/gated pair shares the source model, objective, batch order and hyperparameters. We also evaluate native and gated RoTTA, UnMix-TNS, TRIBE, TEMA, and Hybrid-TTN (Yuan et al., 2023; Tomar et al., 2024; Su et al., 2024a;b; Vianna et al., 2025), which change how the normalization statistics are formed, each with its own forecast fitted on the calibration set (settings in Appendix D). Methods that do not normalize with the batch's statistics, such as LAME (Boudiaf et al., 2022), lie outside the failure the gate targets (Appendix A.6).

Streams. *Shuffled*: a random ordering of the test pool. *Temporal Dirichlet* (δ=0.1): class prevalences drift over time, following the NOTE/RoTTA construction. *Class order* (R=∞): the stream sorted by class, the limit of SAR's majority-block construction. *Persistent skew* (α=0.03), a stable label shift: one Dirichlet composition repeated until a required class pool is exhausted. *Recovery* (CIFAR and NCT): shuffled–class-ordered–shuffled. *Slide order* (Camelyon): patches grouped by slide and ordered by spatial coordinates. *Within-slide shuffled* keeps the slide sequence but shuffles the patches inside each slide. Only the shuffled, temporal and class-order types also occur in calibration (Appendix C).

5.1 GATING ADAPTATION ACROSS BATCH COMPOSITIONS

On CIFAR-10-C the gate separates three regimes (Table 2). On shuffled streams it lets the BN-based methods adapt on almost every batch and keeps about 97% of their gain. Under severe skew, where native adaptation collapses to 21–25%, it declines almost every batch and returns to the source model; for every BN-based method its gain over native adaptation has a 95% bootstrap interval above zero on every skewed stream type (Appendix A.4). Under moderate skew it does more than abstain: on temporal drift and recovery streams every gated BN-based method ends above both the source model and its native version, and averaged over the five stream types gated TENT beats labeled oracles that choose between native TENT and the source model for each stream (67.3% vs. 65.9%, Appendix F.2). CIFAR-100-C shows the same pattern: the gate repairs the class-order collapse and raises TENT's mean over the five stream types from 27.6% to 37.7% (Appendix A.5).

5.2 TRANSFER TO PATHOLOGY DATASETS

Camelyon17: Presenting patches one slide at a time is exactly the setting our analysis targets: batches come predominantly from single slides, with spatially contiguous runs dominated by one class. Native BN-Adapt, TENT and EATA fall below the source model (Table 3). The gate adapts on

Table 2: Gating adaptation on CIFAR-10-C, B=64: native → gated full-stream accuracy (%), averaged over 15 corruptions and five draws. Top: the five BN-based methods; bottom: other methods, also gated. Source is the frozen model, which also predicts the first batch of every run. CIFAR-100-C results are in Appendix A.5, B=200 results and on rates in Appendix A, and per-corruption TENT results in Table A6.

Method	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery
Source	59.53	59.53	59.53	62.44	59.53
BN-Adapt	79.94 → 79.26	54.51 → 63.91	22.69 → 59.80	24.75 → 62.39	45.69 → 67.55
TENT	82.45 → 81.89	51.05 → 63.99	21.14 → 59.81	24.55 → 62.39	43.97 → 68.31
SAR-BN	80.40 → 79.77	55.02 → 64.19	22.79 → 59.81	24.69 → 62.37	45.85 → 67.74
EATA	83.38 → 82.58	51.71 → 63.47	22.11 → 59.81	23.50 → 62.32	45.52 → 68.42
DeYO	82.93 → 82.33	52.57 → 64.29	21.74 → 59.81	24.77 → 62.39	45.37 → 68.46
RoTTA	79.93 → 79.93	76.70 → 76.79	71.45 → 73.08	58.40 → 63.93	78.04 → 77.85
UnMix-TNS	77.67 → 77.67	76.62 → 76.72	74.09 → 74.60	65.32 → 66.32	76.65 → 76.75
TRIBE	81.92 → 81.90	83.89 → 83.69	81.64 → 78.20	77.31 → 75.40	81.50 → 79.57
TEMA	80.45 → 80.35	61.17 → 66.34	31.80 → 60.14	41.32 → 62.90	50.82 → 68.16
Hybrid-TTN	77.97 → 77.97	74.82 → 74.82	68.82 → 67.67	70.97 → 71.53	72.55 → 71.86

Table 3: Transfer to pathology, B=64: native → gated accuracy (%) on Camelyon17 (DenseNet-121) and NCT→CRC-VAL (ResNet-50). Top: BN-based methods; bottom: other methods, also gated.

	Camelyon17			NCT		
Method	Shuffled	Slide order	Within-slide shuffled	Shuffled	Class order	Recovery
Source	72.91	72.91	72.91	95.72	95.72	95.72
BN-Adapt	93.48 → 91.26	63.68 → 78.65	79.35 → 81.78	91.97 → 94.81	15.27 → 95.72	45.91 → 95.36
TENT	93.31 → 89.76	56.03 → 75.01	51.74 → 75.02	95.19 → 95.67	12.82 → 95.72	41.96 → 95.68
SAR-BN	93.48 → 91.26	63.68 → 78.65	79.35 → 81.78	92.48 → 94.91	15.34 → 95.72	45.43 → 95.38
EATA	93.71 → 90.24	64.22 → 77.53	79.30 → 80.72	94.81 → 95.69	14.44 → 95.72	46.05 → 95.72
DeYO	53.17 → 72.91	50.53 → 72.91	44.52 → 72.91	87.00 → 95.48	14.10 → 95.72	43.38 → 95.67
RoTTA	92.50 → 92.50	76.28 → 85.45	78.09 → 84.39	96.11 → 96.11	81.60 → 86.62	95.73 → 95.92
UnMix-TNS	92.62 → 92.62	82.09 → 84.12	84.42 → 85.56	92.17 → 92.17	78.43 → 94.94	88.43 → 92.42
TRIBE	93.43 → 90.52	48.26 → 67.56	62.66 → 82.51	95.31 → 95.30	43.28 → 95.42	94.61 → 95.78
TEMA	93.02 → 91.06	62.80 → 78.41	78.96 → 81.41	94.42 → 94.48	21.13 → 95.72	51.92 → 95.26
Hybrid-TTN	93.51 → 93.51	75.99 → 80.56	87.12 → 84.57	95.26 → 95.26	51.32 → 95.72	69.43 → 95.50

a minority of these batches, for TENT almost only on tumor-rich ones (Figure A1), and lifts all three methods above the source model (75–79% against 72.9%). It keeps the source model's specificity and exceeds its tumor sensitivity, while the native methods label more than half of the normal patches as tumor (Table A4). These gains hold in all ten further random slide orders (Appendix A.2). In those orders, the same number of randomly placed TENT updates falls below the source model every time, and gated RoTTA beats native UnMix-TNS, the best ungated method in Table 3, in all ten. On shuffled patches the gate keeps most of the roughly 20-point gain of the BN-based methods. **NCT:** The source model already reaches 95.7%, so NCT tests whether the gate does harm. Class ordering collapses every BN-based method. The gate declines these batches and returns the source model's accuracy, also on recovery streams, where it declines the class-ordered middle.

5.3 Gating methods designed for skewed streams

RoTTA, UnMix-TNS, TRIBE and Hybrid-TTN handle the ordered CIFAR-10-C streams better than any gated BN-based method (Table 2), but no method that re-forms its normalization statistics is safe on every dataset: each falls below the source model on some stream, most severely TEMA, TRIBE and Hybrid-TTN on class-ordered NCT and TRIBE on slide-ordered Camelyon17 (Tables 2 and 3). Wrapped around these methods, the gate repairs most of these failures: it lifts TRIBE from 43.3% to 95.4% on class-ordered NCT and from 48.3% to 67.6% on slide-ordered Camelyon17, and makes gated RoTTA the most accurate method on slide-ordered Camelyon17 (85.45%). Where a method already copes with the skew, the gate mostly leaves it alone, at a cost of at most 2.6 points, except for TRIBE, which loses up to 3.4 points where it already works but gains the most on pathology.

5.4 Ablations

Measurements. No single measurement carries every dataset: the fit relies on ΔH on CIFAR-10-C and NCT, on D_q on CIFAR-100-C and on RG on Camelyon17, and without ΔH the CIFAR-10-C gate ends 7.0 points below the source model on its worst stream type (Appendix F.4). **Resets.** Resets do not prevent the collapse: TENT reset after every batch predicts exactly like BN-Adapt, and TENT with AETTA resets (Lee et al., 2024b) still collapses to 22.6% under class order, 37 points below gated TENT (Appendix F.2). **Forecast quality.** For BN-Adapt the gate separates beneficial from harmful batches with an AUROC of 0.88–0.95 across all four datasets, and 0.94–0.95 for SAR-BN, EATA, DeYO and TEMA on CIFAR-10-C (Appendix F.3). Nonlinear forecasters gain at most 0.4 points except on Camelyon17, and a threshold on the largest predicted-class share falls below the source model on CIFAR-100. **Labels and calibration.** The label-free rule $\Delta H_t \leq 0$ avoids the collapse of the BN-based methods, which confirms that ΔH reads the failure of Section 3.3. But it reads only the batch, makes the same decisions for every method, and cannot tell a method that a skewed batch harms from one built to gain from it. Calibration learns each method's own response, and the calibrated gate beats the label-free rule in 10 of the 11 method–dataset pairs of Table A18. On CIFAR-10-C shuffled streams it gives up 0.6 points of TENT's 22.9-point gain where the label-free rule gives up 4.6, and it wins by 6.8–12.8 points for Hybrid-TTN, RoTTA, UnMix-TNS and TRIBE, which the label-free rule leaves below their native versions. On Camelyon17 over ten slide orders it wins for RoTTA (85.6% vs. 81.1%), whose gate relies on RG, a measurement the label-free rule ignores. The label-free rule is higher only for TENT on Camelyon17, where the calibrated gate defaults to the source model too often. One fit transfers across methods (BN-Adapt's fit serves TENT, SAR-BN, EATA and DeYO within 0.12 points) and batch sizes (within 0.2 points from B=64 to B=200), stays within 0.15 points without the class-order and long-tail calibration streams, and loses at most 0.45 points when calibrated on 1,000 instead of 5,000 images (Appendices F.4 and A.1).

6 Discussion

Whether test-time adaptation helps is decided batch by batch, and it can be forecast before adapting. Overall class balance does not protect BN-based adaptation from skewed batches: recomputing the statistics alone reproduces the degradation (Section 3.2), and the harm comes from normalizing a batch with statistics dominated by its own class (Section 3.3). The gate reads this failure before adapting. The frozen model's entropy response to the batch's own statistics avoids the collapses of the BN-based methods without any labels, but it forgoes gain where strong corruption also raises it and leaves methods that re-form their normalization statistics, such as RoTTA and TRIBE, below their native versions. The linear forecast, calibrated once on held-out source data, weighs it with the composition measurements and is what lets one procedure serve every method and dataset we evaluate. Across CIFAR-10-C, CIFAR-100-C and two pathology datasets the gate prevents the collapse of every BN-based method we evaluate on class- and slide-ordered streams while retaining most of the gain on shuffled streams, also on corruption types and a hospital absent from calibration (Appendix F.4). The gate does not compete with methods designed for skewed streams but makes them safer: on skewed pathology streams it improves nearly every method it wraps, and the best gated method beats the best native one on slide-ordered Camelyon17 and class-ordered NCT (Table 3).

Limitations and future work. The gate requires a small labeled calibration set held out from source training. Without labeled source data, the label-free rule $\Delta H_t \leq 0$ still avoids the collapse of the BN-based methods, but methods designed for skewed streams need a fully unsupervised gate. The gate chooses between the wrapped method and the source model and forecasts accuracy, so under sustained skew a gated BN-based method stays close to the source model, and class-wise metrics such as tumor sensitivity are not optimized directly. A graded decision and a class-weighted or risk-bounded forecast are natural extensions. Both the failure and the probe that detects it are specific to BatchNorm: a LayerNorm ViT-B/16 keeps its adaptation gain on skewed batches (Section 3.2). On such models the failure we diagnose does not arise, and a gate would need a different signal. Our pathology evidence covers patch classification on two benchmarks with constructed slide orders and a single Camelyon17 test hospital. Medical imaging is moving toward large pretrained models without BatchNorm layers. Whether test-time adaptation is safe in that setting, and whether a forecast can work without batch statistics, remain open questions.

AI USE STATEMENT

We used generative AI tools for writing assistance, improving paper structure, coding assistance, method implementation, feedback on the methodology and experiments, figure and table generation, and finding related work. We have reviewed all AI-assisted work: AI-written code was reviewed and tested by the authors, all reported numbers were checked against the experiment records, and suggested related work was checked against the original papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The pathology experiments evaluate patch classification on existing research datasets. The constructed test orders and observed adaptation gains do not establish clinical safety or patient-level diagnostic performance.

REPRODUCIBILITY STATEMENT

Appendices B–D describe the source models, data partitions, stream constructions and adaptation settings. Appendix F specifies the calibration perturbations, fitted targets and online decision rule. Appendix A gives detailed results, and the execution conventions used in the comparisons will be released with the code.

REFERENCES

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019. doi: 10.1109/TMI.2018.2867350.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8344–8353, 2022.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Michal Danilowski, Alexander Murphy, Young D. Kwon, and Abhirup Ghosh. MORPHEUS: Meta test-time adaptation via neural collapse geometry. In *ICLR 2026 Workshop on Test-Time Updates*, 2026. URL https://openreview.net/forum?id=YfyApJH1Rj.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15069–15078, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations (ICLR)*, 2022.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *arXiv*, 2023. doi: 10.48550/arxiv.2310.10074.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Minguk Jang and Hye Won Chung. Label distribution shift-aware prediction refinement for test-time adaptation. *Transactions on Machine Learning Research*, February 2025. URL https://openreview.net/forum?id=c7AAHdEYz5.

Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLOS Medicine*, 16(1):e1002730, 2019. doi: 10.1371/journal.pmed.1002730.

Eungyeup Kim, Mingjie Sun, Christina Baek, Aditi Raghunathan, and J. Zico Kolter. Test-time adaptation induces stronger accuracy and agreement-on-the-line. In *Advances in Neural Information Processing Systems*, 2024. URL https://papers.neurips.cc/paper_files/paper/2024/file/d96fcc07d623a9eba68616629911143a-Paper-Conference.pdf.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=9w3iw8wDuE.

Taeckyung Lee, Sorn Chottananurak, Taesik Gong, and Sung-Ju Lee. AETTA: Label-free accuracy estimation for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28643–28652, June 2024b.

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. TTN: A domain-shift aware batch normalization in test-time adaptation. In *International Conference on Learning Representations*, 2023.

Taejun Lim, Joong-Won Hwang, and Kibok Lee. When and where to reset matters for long-term test-time adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0JayjvOKxt.

Onur Cezmi Mutlu, Mohammadmahdi Honarmand, Parnian Azizian, Saimourya Surabhi, and Dennis P. Wall. Topology-guided test-time adaptation via persistent homology: From affective behavior analysis to autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 5331–5340, June 2026.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

Sunghyun Park, Seunghan Yang, Jaegul Choo, and Sungrack Yun. Label Shift Adapter for Test-Time Adaptation under Covariate and Label Shifts. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16375–16385, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.01505. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01505.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. RDumb: A simple approach that questions our progress in continual test-time adaptation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VfP6VTVsHc.

Mona Schirmer, Metod Jazbec, Christian A. Naesseth, and Eric Nalisnick. Monitoring risks in test-time adaptation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TzHX2RWUdE.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022.

Yongyi Su, Xun Xu, and Kui Jia. Towards real-world test-time adaptation: Tri-net self-training with balanced normalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13): 15126–15135, Mar. 2024a. doi: 10.1609/aaai.v38i13.29435.

Zixian Su, Jingwei Guo, Kai Yao, Xi Yang, Qiufeng Wang, and Kaizhu Huang. Unraveling batch normalization for realistic test-time adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):15136–15144, 2024b. doi: 10.1609/aaai.v38i13.29436.

Devavrat Tomar, Guillaume Vray, Jean-Philippe Thiran, and Behzad Bozorgtabar. Un-mixing test-time normalization statistics: Combatting label temporal correlation. In *International Conference on Learning Representations (ICLR)*, 2024.

Pedro Vianna, Muawiz Sajjad Chaudhary, Paria Mehrbod, An Tang, Guy Cloutier, Guy Wolf, Michael Eickenberg, and Eugene Belilovsky. Channel-selective normalization for label-shift robust test-time adaptation. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 514–533. PMLR, 2025. URL https://proceedings.mlr.press/v274/vianna25a.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

Wenyu Zhang, Li Shen, Wanyue Zhang, and Chuan-Sheng Foo. Few-shot adaptation of pre-trained networks for domain shift. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 1665–1671, 2022. URL https://www.ijcai.org/proceedings/2022/0232.pdf.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. DELTA: Degradation-free fully test-time adaptation. In *International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=eGm22rqG93.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42058–42080. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/zhao23d.html.

A Additional Results

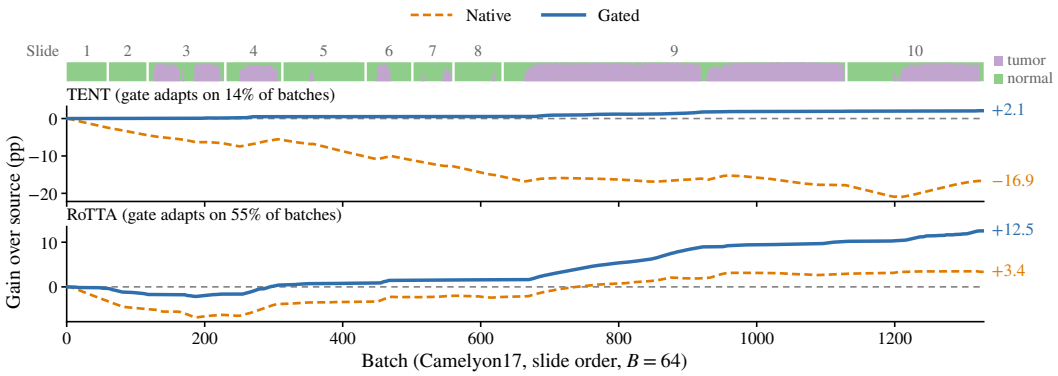


Figure A1: Camelyon17 in slide order ($B = 64$). Top: share of tumor patches per batch, for illustration only (the gate uses no labels). Below: gain over the source model for TENT and RoTTA, accumulated batch by batch. The right end equals the full-stream gain in Table 3. The gated curve is flat where the gate declines and serves the source prediction. The TENT gate adapts almost only on tumor-rich batches.

A.1 Calibration size and model seeds

A small calibration set leaves more data for training. Over three independently trained CIFAR-10 models, gated TENT at $B = 64$ reaches $66.67 \pm 0.59\%$ against $44.63 \pm 0.01\%$ native, and calibrating on 1,000 instead of 5,000 images changes the gated means by at most 0.45 points (Table A1). On CIFAR-100-C, 10 to 50 calibration images per class give gated means within 0.6 points of each other (Table A2(a)).

Table A1: Calibration size and model seeds on CIFAR-10-C: accuracy (%) averaged over the five stream types, mean ± population standard deviation over three independently trained WRN-28-10 models. Seed 0 is the model of Table 2. The source model predicts the first batch of every run. Most of the spread comes from persistent skew, where gated TENT (B=64, 5,000 calibration images) reaches 62.39, 57.31 and 62.07% for seeds 0, 1 and 2 (source 62.44, 60.56 and 62.51%). *RG* receives nonzero weight only in some seed-2 fits ($b_R < 0.17$).

B	Calibration images	BN-Adapt native	BN-Adapt gated	TENT native	TENT gated
64	5,000	45.44 ± 0.15	65.96 ± 0.53	44.63 ± 0.01	66.67 ± 0.59
64	2,500	45.44 ± 0.15	65.95 ± 0.56	44.64 ± 0.02	66.63 ± 0.59
64	1,000	45.44 ± 0.15	65.86 ± 0.83	44.63 ± 0.00	66.50 ± 0.82
200	5,000	46.91 ± 0.19	66.34 ± 0.62	47.51 ± 0.07	67.16 ± 0.65
200	2,500	46.91 ± 0.19	66.18 ± 0.92	47.51 ± 0.07	66.98 ± 0.91
200	1,000	46.91 ± 0.19	65.90 ± 1.37	47.51 ± 0.07	66.71 ± 1.34

A.2 Ten further slide orders

The Camelyon17 results are not specific to one slide order. We replay the test stream in ten further orders, each presenting the ten slides of the test hospital in a random order with the patches of a slide sorted by column then row or by row then column, for BN-Adapt, TENT and EATA, for RoTTA, the best gated method, and for native UnMix-TNS, the best native method of Table 3, with the gates of Table 3 unchanged. Gated BN-Adapt, TENT, EATA and RoTTA beat the source model in every order, and their native versions in 39 of the 40 method–order pairs (Table A3). Gated RoTTA averages 85.6% and beats native UnMix-TNS in all ten orders; over four seeds on the slide order of Table 3 it spans 84.8–85.6%, against 65.3–82.1% for native RoTTA. The same number of TENT updates placed at random batches stays below the source model in all 30 runs (ten orders, three seeds), and the gate beats random timing in every order for both TENT and RoTTA.

Table A2: CIFAR-100-C calibration size for gated TENT at B=64: full-stream accuracy (%) with 10 to 200 calibration images per class, for ResNeXt-29 source models trained on (a) 45,000 and (b) 30,000 images. Native rows come from the 50-per-class (a) and 200-per-class (b) runs; the other runs differ from them by at most 0.08 points. In (b), the 200-per-class TENT fit gives ΔH almost no weight and adapts on only 10% of shuffled batches, while BN-Adapt's fit on the same calibration images adapts on 99% of them, so this is one degenerate fit rather than an effect of calibration size.

Policy	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery	Mean
(a) Source model trained on 45,000 images						
Source	31.33	31.33	31.33	31.37	31.33	31.34
Native	54.66	33.18	3.10	30.01	17.02	27.59
Gated, 10 per class	54.59	35.06	31.39	29.90	40.09	38.21
Gated, 25 per class	54.62	34.32	31.27	29.89	39.76	37.97
Gated, 50 per class	54.62	33.56	30.99	30.00	39.11	37.65
(b) Source model trained on 30,000 images						
Source	28.30	28.30	28.30	29.41	28.30	28.52
Native	47.44	26.42	2.79	28.98	14.83	24.09
Gated, 25 per class	47.49	27.44	28.14	29.00	35.28	33.47
Gated, 50 per class	47.46	27.09	27.98	29.07	34.83	33.28
Gated, 100 per class	47.47	27.92	28.27	28.93	35.89	33.69
Gated, 200 per class	31.68	29.19	28.30	28.58	29.67	29.49

Table A3: Camelyon17 over ten further random slide orders (B=64; source model 72.91%), for BN-Adapt, TENT and EATA, which collapse on slide order, and RoTTA, the best gated method. Accuracy gain in points against each alternative: mean over the orders, gain in the worst order, and the number of orders in which the gated method is better. Random batches: as many updates as the gate, at randomly chosen batches (TENT: mean of three seeds). UnMix-TNS is the best native method on the slide order of Table 3.

	Mean gain	Worst order	Orders won
Gated BN-Adapt against			
source model	+5.61	+4.99	10/10
native BN-Adapt	+14.08	+12.81	10/10
Gated TENT against			
source model	+1.72	+1.26	10/10
native TENT	+19.23	+13.90	10/10
TENT on random batches	+3.66	+2.98	10/10
Gated EATA against			
source model	+4.26	+3.06	10/10
native EATA	+12.69	+11.54	10/10
Gated RoTTA against			
source model	+12.65	+9.65	10/10
native RoTTA	+3.80	−0.17	9/10
RoTTA on random batches	+6.02	+3.08	10/10
native UnMix-TNS	+4.61	+1.10	10/10

A.3 CLASS METRICS, ON RATES AND PER-CORRUPTION RESULTS

Table A4 splits the Camelyon17 slide-order accuracy into tumor sensitivity and specificity, Table A5 gives the share of batches on which each gate adapts, Table A6 gives TENT per corruption, and Tables A7 and A8 repeat Tables 2 and 3 at $B = 200$.

Table A4: Camelyon17 slide order, B=64: accuracy, tumor sensitivity and specificity (%), native → gated. SAR-BN equals BN-Adapt in every cell, and gated DeYO equals the source model. Appendix A.2 gives the gains over ten further slide orders, with the worst order and the number of orders won.

Method	Accuracy	Sensitivity	Specificity
Source	72.9	50.8	95.1
BN-Adapt	63.7 → 78.6	85.6 → 63.2	41.7 → 94.1
TENT	56.0 → 75.0	65.1 → 54.9	46.9 → 95.1
EATA	64.2 → 77.5	84.7 → 60.3	43.7 → 94.7
TRIBE	48.3 → 67.6	1.6 → 39.8	94.9 → 95.3

Table A5: Share of batches (%, rounded) on which the gate adapts in Tables 2 and 3 (B=64). The share includes the first batch, which the gate never adapts. The on rate of Section 4 excludes it and is $n/(n-1)$ times higher for a stream of n batches. This matters only for the persistent streams (about 15 batches). TRIBE values are recorded as whole percents.

	CIFAR-10-C					Camelyon17			NCT		
Method	Shuf.	Temp.	Class	Pers.	Rec.	Shuf.	Slide	Within slide	Shuf.	Class	Rec.
BN-Adapt	96	28	4	5	40	90	22	46	48	0	18
TENT	97	31	4	6	41	84	14	39	19	0	7
SAR-BN	96	29	4	6	41	90	22	46	45	0	17
EATA	96	29	4	6	41	85	19	45	13	0	5
DeYO	97	31	4	6	41	0	0	0	22	0	8
RoTTA	99	97	68	69	80	100	55	68	99	30	61
UnMix-TNS	99	98	76	76	86	100	44	56	99	2	41
LAME	99	99	99	93	99	100	100	100	99	99	99
TRIBE	99	97	65	68	79	87	23	46	99	2	41
TEMA	99	39	5	9	43	91	20	45	98	0	38
Hybrid-TTN	99	91	41	45	64	100	42	61	99	0	38

Table A6: Per-corruption TENT accuracy (%) on CIFAR-10-C, $B = 64$, native → gated. The Mean row equals the TENT row of Table 2.

Corruption	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery
Gaussian noise	75.87 → 69.32	46.04 → 35.76	18.91 → 28.91	20.44 → 22.63	40.55 → 44.88
Shot noise	78.52 → 77.44	47.59 → 43.42	19.36 → 35.87	20.77 → 31.48	42.05 → 51.73
Impulse noise	69.80 → 69.55	40.69 → 43.90	17.49 → 33.48	19.91 → 36.60	36.45 → 46.40
Defocus blur	87.94 → 87.85	55.56 → 62.12	23.31 → 54.12	25.55 → 57.22	48.05 → 67.76
Glass blur	70.07 → 69.96	41.09 → 55.14	17.80 → 52.47	21.04 → 54.26	35.71 → 58.20
Motion blur	87.28 → 87.32	55.09 → 71.04	23.21 → 69.42	26.05 → 67.43	46.63 → 76.56
Zoom blur	88.90 → 88.99	56.82 → 67.49	23.07 → 62.79	24.79 → 57.00	47.71 → 73.24
Snow	85.99 → 85.68	53.70 → 80.06	22.17 → 80.19	26.13 → 83.12	45.93 → 82.46
Frost	85.28 → 85.05	53.40 → 70.94	22.26 → 68.53	25.94 → 67.10	45.77 → 74.91
Fog	87.65 → 87.55	56.69 → 75.90	24.40 → 75.47	27.70 → 78.77	47.89 → 80.35
Brightness	92.05 → 92.14	60.08 → 92.10	24.59 → 92.25	29.45 → 92.52	50.59 → 92.32
Contrast	87.64 → 87.67	53.61 → 60.71	20.65 → 49.08	26.73 → 59.27	44.40 → 64.17
Elastic transform	78.58 → 78.66	47.54 → 75.00	19.74 → 75.44	24.04 → 74.39	41.43 → 76.52
Pixelate	84.23 → 84.24	51.28 → 56.40	21.09 → 48.25	25.62 → 73.32	45.35 → 62.31
JPEG compression	76.90 → 76.93	46.53 → 69.91	19.04 → 70.87	24.14 → 80.79	41.06 → 72.82
Mean	**82.45 → 81.89**	**51.05 → 63.99**	**21.14 → 59.81**	**24.55 → 62.39**	**43.97 → 68.31**

Table A7: CIFAR-10-C at B=200, companion to Table 2: native → gated full-stream accuracy (%), averaged over 15 corruptions and five draws. The source model predicts the first batch of every run. TEMA and Hybrid-TTN were run at B=64 only.

$B = 200$; *native → gated*					
Method	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery
Source	59.53	59.53	59.53	62.52	59.53
BN-Adapt	80.24 → 80.19	56.26 → 64.63	22.36 → 59.81	30.36 → 62.50	45.94 → 67.79
TENT	83.09 → 83.07	55.41 → 64.93	22.23 → 59.84	30.27 → 62.50	46.47 → 68.49
SAR-BN	80.51 → 80.46	56.52 → 64.74	22.53 → 59.81	30.38 → 62.50	46.11 → 67.83
EATA	83.65 → 83.64	56.98 → 64.89	23.13 → 59.85	30.14 → 62.50	47.17 → 68.61
DeYO	83.32 → 83.31	55.91 → 65.07	22.48 → 59.83	30.31 → 62.51	46.96 → 68.56
Source	59.53	59.53	59.53	62.52	59.53
RoTTA	79.46 → 79.46	76.98 → 77.06	73.65 → 74.23	61.06 → 63.82	78.52 → 78.42
UnMix-TNS	77.06 → 77.06	76.41 → 76.49	74.79 → 75.04	65.01 → 65.59	76.64 → 76.81
LAME	58.92 → 58.92	61.12 → 61.12	62.58 → 62.58	65.62 → 65.62	61.08 → 61.08
TRIBE	81.67 → 81.67	79.57 → 79.61	69.08 → 69.07	72.54 → 69.09	76.14 → 75.26

Table A8: Pathology at B=200, companion to Table 3: native → gated accuracy (%) on Camelyon17 (DenseNet-121) and NCT→CRC-VAL (ResNet-50). Top: BN-based methods; bottom: other methods, also gated. Source is the frozen model. TEMA and Hybrid-TTN were run at B=64 only.

	Camelyon17			NCT		
Method	Shuffled	Slide order	Within-slide shuffled	Shuffled	Class order	Recovery
Source	72.91	72.91	72.91	95.72	95.72	95.72
BN-Adapt	93.57 → 93.38	65.54 → 79.83	80.02 → 82.06	93.54 → 94.61	18.70 → 95.72	48.22 → 95.30
TENT	95.20 → 94.72	57.49 → 75.28	70.14 → 76.81	96.73 → 96.06	16.89 → 95.72	48.44 → 95.76
SAR-BN	93.57 → 93.38	65.54 → 79.83	80.02 → 82.06	93.65 → 94.69	18.79 → 95.72	48.11 → 95.31
EATA	93.67 → 93.21	65.85 → 78.61	79.60 → 80.51	96.51 → 96.00	18.56 → 95.72	49.28 → 95.72
DeYO	52.79 → 62.20	51.49 → 74.17	50.71 → 67.66	93.97 → 95.24	18.20 → 95.72	47.86 → 95.54
RoTTA	91.66 → 91.66	81.83 → 84.73	81.83 → 83.17	96.08 → 96.08	83.97 → 87.46	95.92 → 95.86
UnMix-TNS	92.63 → 92.63	82.47 → 84.04	84.57 → 85.79	91.66 → 91.66	80.11 → 95.54	89.03 → 92.26
LAME	72.71 → 72.71	73.57 → 73.57	73.47 → 73.47	95.46 → 95.46	97.43 → 97.43	96.83 → 96.83
TRIBE	93.22 → 93.21	57.43 → 83.03	84.40 → 83.20	95.59 → 95.57	59.41 → 94.55	94.76 → 95.70

A.4 UNCERTAINTY OF THE CIFAR-10-C RESULTS

Table A9 adds paired 95% bootstrap intervals to Table 2, resampling the 15 corruptions with their five draws as in Appendix F.2; Table A1 gives the spread over source models.

Table A9: Table 2 with uncertainty (CIFAR-10-C, B=64). For all five BN-based methods the gate is significantly more accurate than native adaptation on every skewed stream type: every 95% interval lies above zero, with the lowest lower bound at $+3.36$ points. On shuffled streams the gate costs these methods 0.56 to 0.79 points, a small but also significant cost, and gating costs TRIBE 0.2 to 3.4 points on the skewed stream types. Each cell: gated minus native full-stream accuracy (points) as mean ± SD (SD to one decimal) over the 75 (corruption, draw) cells and, below it, the paired 95% corruption-cluster bootstrap interval of the mean (10,000 resamples of the 15 corruptions, each keeping its five draws; conditional on the test pool and the model seed). Mean: the five-type mean, resampling whole corruptions across the five stream types.

Method	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery	Mean
BN-Adapt	-0.68 ± 2.0 [−1.79, −0.02]	$+9.40 \pm 11.8$ [+3.63, +15.12]	$+37.12 \pm 16.8$ [+28.61, +45.51]	$+37.64 \pm 23.0$ [+28.63, +45.96]	$+21.86 \pm 10.5$ [+16.46, +27.06]	$+21.07 \pm 11.6$ [+15.27, +26.57]
TENT	-0.56 ± 1.7 [−1.47, −0.03]	$+12.94 \pm 12.3$ [+6.95, +18.75]	$+38.67 \pm 16.7$ [+30.26, +46.96]	$+37.84 \pm 23.0$ [+28.85, +46.14]	$+24.33 \pm 10.7$ [+18.87, +29.59]	$+22.65 \pm 11.7$ [+16.82, +28.16]
SAR-BN	-0.62 ± 1.8 [−1.63, −0.01]	$+9.17 \pm 11.9$ [+3.36, +14.85]	$+37.03 \pm 16.8$ [+28.52, +45.38]	$+37.68 \pm 23.0$ [+28.72, +45.96]	$+21.89 \pm 10.5$ [+16.50, +27.07]	$+21.03 \pm 11.6$ [+15.24, +26.50]
EATA	-0.79 ± 2.1 [−1.96, −0.09]	$+11.76 \pm 12.5$ [+5.69, +17.71]	$+37.70 \pm 16.8$ [+29.18, +46.06]	$+38.82 \pm 23.0$ [+29.84, +47.07]	$+22.90 \pm 10.7$ [+17.42, +28.18]	$+22.08 \pm 11.8$ [+16.19, +27.62]
DeYO	-0.59 ± 1.8 [−1.56, −0.01]	$+11.72 \pm 12.3$ [+5.76, +17.52]	$+38.07 \pm 16.8$ [+29.56, +46.43]	$+37.61 \pm 22.9$ [+28.67, +45.86]	$+23.09 \pm 10.5$ [+17.69, +28.30]	$+21.98 \pm 11.7$ [+16.15, +27.45]
RoTTA	0.00 ± 0.0 [0.00, +0.01]	$+0.09 \pm 0.2$ [+0.03, +0.16]	$+1.63 \pm 2.2$ [+0.68, +2.61]	$+5.53 \pm 8.7$ [+3.14, +8.20]	-0.20 ± 1.3 [−0.84, +0.40]	$+1.41 \pm 1.8$ [+0.83, +2.03]
UnMix-TNS	0.00 ± 0.0 [0.00, 0.00]	$+0.10 \pm 0.3$ [+0.05, +0.17]	$+0.51 \pm 1.8$ [−0.21, +1.15]	$+0.99 \pm 3.1$ [+0.36, +1.70]	$+0.11 \pm 0.9$ [−0.31, +0.53]	$+0.34 \pm 0.8$ [+0.07, +0.61]
TRIBE	-0.01 ± 0.2 [−0.07, +0.04]	-0.20 ± 0.6 [−0.27, −0.12]	-3.44 ± 5.8 [−5.40, −1.33]	-1.91 ± 5.4 [−3.02, −0.74]	-1.94 ± 2.2 [−2.74, −1.11]	-1.50 ± 1.7 [−2.06, −0.90]
TEMA	-0.10 ± 0.3 [−0.29, 0.00]	$+5.17 \pm 9.4$ [+0.53, +9.66]	$+28.34 \pm 15.6$ [+20.39, +36.17]	$+21.59 \pm 19.6$ [+13.58, +28.86]	$+17.34 \pm 9.5$ [+12.49, +22.13]	$+14.47 \pm 10.0$ [+9.46, +19.26]
Hybrid-TTN	0.00 ± 0.0 [0.00, 0.00]	0.00 ± 0.8 [−0.32, +0.32]	-1.15 ± 5.6 [−3.95, +1.70]	$+0.55 \pm 8.2$ [−2.07, +2.89]	-0.69 ± 3.4 [−2.40, +1.02]	-0.26 ± 3.1 [−1.71, +1.14]

A.5 CIFAR-100-C

Table A10 repeats the stream types of Section 5 on CIFAR-100-C. At $B = 64$ the gate repairs the class-order collapse of BN-Adapt and TENT ($6.16\% \rightarrow 31.21\%$ and $3.10\% \rightarrow 30.99\%$, source 31.33%), raises recovery by 14 and 22 points and keeps shuffled and temporal accuracy. Averaged over the five stream types, gated TENT reaches 37.65% against 27.59% native and 31.34% for the source model (41.02% vs. 33.35% at $B = 200$). Gated TRIBE improves four of the five stream types at $B = 64$. On longer persistent streams (Appendix C; 156 batches at $B = 64$, drawn with replacement), the CIFAR-10-C gates adapt on about 6% of the batches and stay at the source level (gated TENT 62.20% against 23.05% native, source 62.37%); the CIFAR-100-C gates adapt on nearly every batch and match native BN-Adapt and TENT (27.29% and 34.74% against 27.30% and 34.81%, source 31.37%).

A.6 METHODS OUTSIDE THE GATE'S TARGET

LAME (Boudiaf et al., 2022) refines the source model's predictions within each batch without changing its weights or its normalization statistics, so it lies outside the failure of Section 3. Its gate adapts on nearly every batch (Table A5), and gated and native LAME coincide in every cell, at B=64 and at B=200 (Tables A7 and A8). SAR on a GroupNorm model, which does not normalize over the batch either, changes accuracy by less than half a point on every stream type (Appendix E.4).

Table A10: CIFAR-100-C (ResNeXt-29 trained on 45,000 images, 50 calibration images per class): native → gated full-stream accuracy (%), mean over 15 corruptions and five draws; the source model predicts the first batch. At B=64 persistent streams have about four batches, and the BN-Adapt and TENT gates adapt on 77% of them, nearly every batch after the first.

Method	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery
$B = 64$					
Source	31.33	31.33	31.33	31.37	31.33
BN-Adapt	53.23 → 53.21	43.53 → 43.52	6.16 → 31.21	29.12 → 29.12	25.97 → 39.97
TENT	54.66 → 54.62	33.18 → 33.56	3.10 → 30.99	30.01 → 30.00	17.02 → 39.11
TRIBE	40.67 → 41.98	39.02 → 40.13	23.25 → 31.36	37.08 → 33.17	33.57 → 34.69
$B = 200$					
Source	31.33	31.33	31.33	30.83	31.33
BN-Adapt	54.06 → 54.06	44.21 → 44.34	9.10 → 31.76	30.22 → 30.22	32.59 → 41.16
TENT	58.87 → 58.87	42.22 → 42.59	6.24 → 31.60	30.22 → 30.22	29.22 → 41.80
TRIBE	46.95 → 46.94	42.60 → 42.44	37.42 → 36.42	30.83 → 30.83	42.86 → 42.64

B Datasets and Source Models

Table A11: Forecasting datasets. Fit images are held-out labeled source images. Target labels are used only to order the streams and to score the results.

Dataset	Classes	Fit images	Target images	Model
CIFAR-10-C	10	5,000	10,000/corruption	WRN-28-10
CIFAR-100-C	100	5,000	10,000/corruption	ResNeXt-29
Camelyon17	2	26,845	85,054	DenseNet-121
NCT→CRC-VAL	9	9,995	7,180	ResNet-50

B.1 CIFAR

For training we use 45,000 of the 50,000 CIFAR training images and keep the other 5,000 for calibration (Table A11), 500 per class on CIFAR-10 and 50 on CIFAR-100. The primary forecast results use seed 0. Training runs for 100 epochs on CIFAR-10 and 200 on CIFAR-100 with SGD (momentum 0.9, Nesterov, batch size 128, weight decay 5×10^{-4}), a 5-epoch warmup to learning rate 0.1 followed by cosine decay, and random 32×32 crops with padding 4 and horizontal flips. The primary corruption streams use one corruption type at severity 5. The analysis uses the standard RobustBench WRN-28-10 checkpoint and separately retrained normalization variants.

B.2 Camelyon17-WILDS

A DenseNet-121 is trained from scratch to separate tumor from normal tissue in 96×96 RGB lymph-node patches from the official training hospitals (Hospitals 0, 3, 4) (Koh et al., 2021; Bandi et al., 2019). The official in-distribution validation pool is split into a forecast-fitting pool (80%, 26,845 patches) and a checkpoint-selection set (20%). The test hospital is Hospital 2 (85,054 patches from ten slides and nine patients). Inputs use ImageNet mean/std normalization. Training runs for 10 epochs with SGD (learning rate 10^{-3}, momentum 0.9, weight decay 0.01, batch size 32).

B.3 NCT-CRC-HE-100K to CRC-VAL-HE-7K

An ImageNet-initialized ResNet-50 classifies tissue patches into nine classes (Kather et al., 2019). Within each class, a seeded shuffle assigns the first 10% of the source images to checkpoint selection, the next 10% to forecast fitting and the rest to training, which gives 9,995 images each for selection and fitting. Training runs for eight epochs with SGD (learning rate 3×10^{-4}, momentum 0.9, weight decay 10^{-4}, batch size 360) and random horizontal and vertical flips. The 7,180 CRC-VAL tiles form the external test set.

C STREAM CONSTRUCTION

C.1 DIAGNOSIS STREAMS

The prior-matched streams of Section 3.1 hold 2,000 samples at $B = 200$. They round the Dirichlet composition to integer counts before rotating it, and reuse samples when a class's requested supply exceeds its pool.

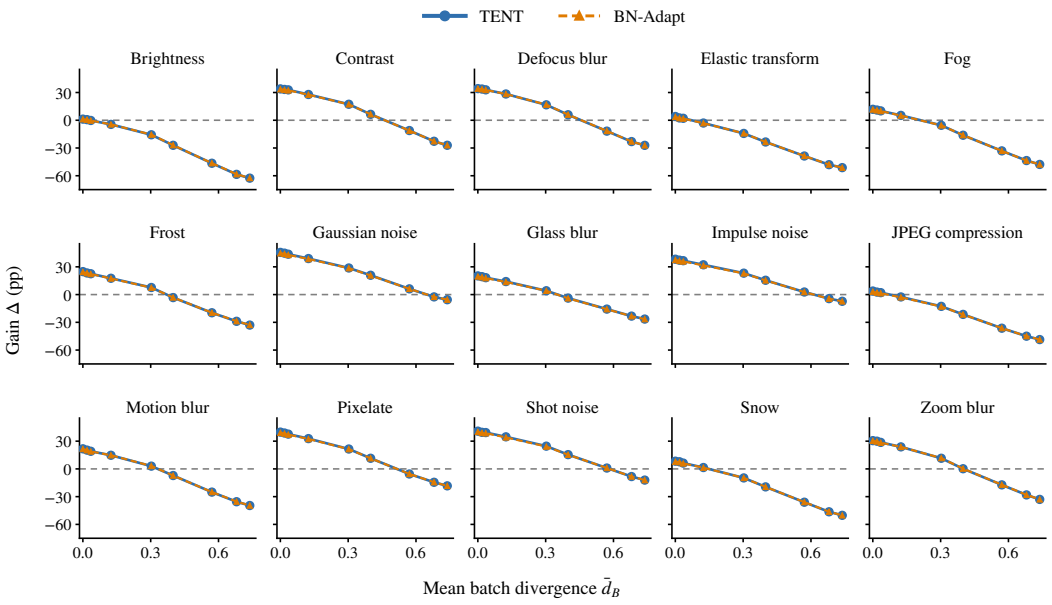


Figure A2: Figure 1 per corruption: mean gain Δ of TENT and BN-Adapt over 16 streams (eight draws, two seeds) per skew level, against the mean batch divergence $\bar{d}_B$.

C.2 FULL-POOL CIFAR-10 FORECASTING

Each corruption provides 1,000 images per class. Streams sample without replacement unless stated otherwise, so an image never appears twice in one stream but can recur across realizations. The shuffled, temporal, class-order and recovery streams contain 156 complete batches at $B = 64$ (9,984 samples) and 50 at $B = 200$ (10,000 samples); at $B = 64$ the BN-based and TRIBE runs score the 16 remaining images with the source model.

Shuffled. A random permutation of the class-balanced pool.

Temporal Dirichlet. A Dirichlet with $\delta = 0.1$ splits each class's images over ten consecutive segments, so each class is concentrated in a few; images are shuffled within each segment and cut into batches.

Class order. SAR's ten 1,000-image majority blocks (majority probability $R/(R+K-1)$, shuffled class order) at $R = \infty$, so each block holds one class.

Persistent skew. One composition drawn at $\alpha = 0.03$ is repeated until a class would be exhausted: 15 to 17 batches (15.4 on average) at $B = 64$, five at $B = 200$.

Recovery. Shuffle the pool, then sort the middle 60% by a random class order and keep the first and last 20% shuffled.

CIFAR-100-C uses the same constructions (100 classes, 100 images per class and corruption), so its persistent streams end after about four batches at $B = 64$ and one or two at $B = 200$. For Appendix A.5, longer CIFAR-10-C and CIFAR-100-C persistent streams draw images with replacement, up to 156 batches at $B = 64$ or 50 at $B = 200$. Calibration uses shuffled, temporal, class-order

and long-tail streams (Appendix F); persistent skew resembles long-tail, while recovery and the slide orders never occur in calibration.

C.3 MEDICAL ORDERS

Each **Camelyon** stream presents all 85,054 hospital-2 patches once, in shuffled, slide or within-slide-shuffled order. Slide order uses slide metadata, not tumor labels, and sorts each slide by column x and then row y. WILDS keeps every tumor patch and subsamples normal ones to balance the hospital, so neither this prevalence nor the spatial order is a clinical arrival process. **NCT** streams use five random draws of each order over all 7,180 tiles; class order groups tiles by tissue label, recovery sorts the middle 60% by class as for CIFAR, and there is no patient-level ordering. The labels behind these orders are hidden from the methods and the gate. A final partial batch is predicted by the source model and counts toward full-stream accuracy.

D TRAINING AND ADAPTATION SETTINGS

Each native/gated pair uses the same adaptation settings. The effective configurations, including method-specific overrides, will be released with the code.

- BN-Adapt normalizes with current-batch statistics and makes no gradient updates.
- TENT (Wang et al., 2021) updates the normalization affine parameters to minimize the mean prediction entropy $-B^{-1}\sum_{x,c} p_c(x)\log p_c(x)$. It uses Adam at 10^{-3} on CIFAR and NCT. On Camelyon it uses SGD at 10^{-3} with momentum 0.9, zero weight decay and no Nesterov acceleration, one step per batch and no episodic reset. Momentum SGD and this learning rate follow the Camelyon/DenseNet evaluation of Zhang et al. (2022), and Kim et al. (2024) also adapt with SGD momentum 0.9.
- SAR-BN (Niu et al., 2023) runs SAR on the BatchNorm model: SAM with SGD, learning rate 10^{-3}, momentum 0.9, radius 0.05, entropy threshold $0.4\log K$ and reset threshold 0.2. Native SAR-GN has its own source checkpoint and baseline (Appendix E.4).
- EATA (Niu et al., 2022): Adam at 10^{-3}, Fisher coefficient 1, entropy threshold $0.4\log K$ and diversity threshold 0.4.
- DeYO (Lee et al., 2024a): Adam at 10^{-3}, entropy thresholds $0.5\log K$ and $0.4\log K$, PLPD threshold 0.2, patch-shuffling augmentation on a 4×4 grid.
- RoTTA (Yuan et al., 2023): Adam at 10^{-3}, betas $(0.9, 0.999)$, zero weight decay, memory size and update frequency 64, $\nu = 0.001$, $\alpha = 0.05$, and time and uncertainty weights both 1.
- UnMix-TNS uses 16 components and no gradient updates. For source BN momentum m, its batch-size correction is $1-(1-m)^{B/64}$.
- LAME uses a binary five-nearest-neighbor affinity without forced symmetrization, at most 100 optimization iterations and source BN statistics.
- TRIBE (Su et al., 2024a): Adam at 10^{-3}, normalization update rate $\eta = 0.01$, $\gamma = 0$, entropy threshold $0.05\log K$ and anchor-loss weight 0.5.
- TEMA (Su et al., 2024b): official code, no gradient updates. Test-statistics momentum by the authors' rule, $m = 1$ on CIFAR-10 and Camelyon17 and $m = 0.1$ on NCT at $B = 64$, with the test statistics initialized to the source statistics at the start of each stream.
- Hybrid-TTN (Vianna et al., 2025): implemented from the paper. Channel sensitivities scored once on single-class batches of up to 4,500 labeled source-training images per class; the share of channels kept at source statistics grows linearly with depth; two passes, the second reweighted by the first pass's predicted-class histogram.
- NOTE (Gong et al., 2022) is not evaluated: its published setup uses separately trained instance-aware batch-normalization checkpoints, whereas our BN comparisons use a shared source checkpoint.

E Additional Analysis

E.1 Statistics swap

In Table 1 (Section 3.3), on all streams of Figure 1, the *balanced* statistics of a batch b come from 200 other images of the same corruption type, 20 per class, none of them in b. The *same-count* statistics come from 200 other images of the same corruption type with exactly the class counts of b. The *other-class* statistics come from the previous batch of the same stream (cyclically), which has the same class counts but a different dominant class. The mean-only and variance-only swaps replace the mean or the variance of b by that of its whole class-balanced stream. Table A12 gives the gains of all nine methods at both ends of the same composition sweep (Section 3.2).

Table A12: Gain Δ (pp) at the near-balanced (β=1000) and most skewed (β=0.01) ends of the CIFAR-10-C composition sweep, as mean $\pm$ standard deviation over 240 ten-batch streams (15 corruptions, eight draws, two seeds), and Spearman ρ between $\bar{d}_B$ and gain over all streams.

Method	ρ	Δ at $\beta = 1000$	Δ at $\beta = 0.01$
TENT	-0.787	$+23.9 \pm 14.6$	-32.7 ± 18.3
BN-Adapt	-0.785	$+23.5 \pm 14.5$	-32.6 ± 18.3
SAR-BN	-0.785	$+23.5 \pm 14.5$	-32.6 ± 18.3
EATA	-0.787	$+24.0 \pm 14.7$	-32.6 ± 18.3
DeYO	-0.787	$+24.0 \pm 14.6$	-32.6 ± 18.3
TRIBE	-0.011	$+20.2 \pm 12.4$	$+20.1 \pm 12.4$
RoTTA	$+0.067$	$+19.1 \pm 11.8$	$+20.9 \pm 12.4$
TEMA	-0.695	$+23.4 \pm 14.3$	-17.3 ± 16.5
Hybrid-TTN	-0.340	$+20.0 \pm 13.2$	$+8.3 \pm 13.1$

E.2 Batch size and stream length

Table A13 repeats the sweep of Figure 1 with 1,920 inputs per stream at $B = 16$ and $B = 64$, and over longer streams at $B = 64$.

Table A13: Batch size and stream length on the composition-controlled CIFAR-10-C streams (RobustBench WRN-28-10). Gain Δ (pp) over the source model, mean over 15 corruptions. The 10-batch rows are those of Figure 1.

Stream	Method	β=1000	1	0.3	0.1	0.03	0.01
10 batches, B=200	BN-Adapt	$+23.5$	$+17.2$	$+6.6$	-19.1	-28.8	-32.6
	TENT	$+23.9$	$+17.5$	$+6.7$	-19.1	-28.8	-32.7
120 batches, B=16	BN-Adapt	$+24.8$	$+17.4$	$+5.3$	-12.8	-26.7	-32.9
30 batches, B=64	BN-Adapt	$+23.8$	$+17.0$	$+7.4$	-12.6	-24.6	-33.1
150 batches, B=64, no image reuse	BN-Adapt	$+23.8$	$+16.5$	$+4.2$	-12.5	-26.3	-32.9
	TENT	$+25.9$	$+18.1$	$+4.7$	-13.1	-26.9	-33.3
1,000 batches, B=64, images reused	BN-Adapt	$+23.9$	$+16.2$	$+4.3$	-11.4	-25.7	-31.5
	TENT	$+26.5$	$+16.3$	-0.8	-19.8	-30.8	-35.2

On 1,000-batch streams with the WRN-28-10 of Section 5 ($B = 64$, $\beta = 0.03$, mean over 15 corruptions and five draws), the sample filters do not prevent adaptation: SAR, EATA and DeYO take an optimizer step on every batch after the first, on 65%, 85% and 81% of the inputs.

E.3 ImageNet-C

We use the ImageNet-C (Hendrycks & Dietterich, 2019) validation images at severity 5 for all 15 corruptions and three pre-trained models: a torchvision ResNet-50 with BatchNorm (He et al., 2016), a ViT-B/16 with LayerNorm (Dosovitskiy et al., 2021; Steiner et al., 2022) using its own input normalization, and a ResNet-50 with GroupNorm (Wu & He, 2018). To vary only batch

composition, we split the 1,000 classes into contiguous windows of c classes in a random order, shuffle images within each window, cut the stream into batches of $B = 64$ and shuffle the batch order ($c = 1000$, the shuffled stream: 62 classes per batch on average, $c = 1$: about 2.2). A reference stream in unshuffled class order follows SAR (Niu et al., 2023). Each stream uses every image at most once, with one draw per stream. TENT uses SGD with momentum 0.9 at learning rate 2.5×10^{-4} for the ResNets and 10^{-3} for the ViT, updating the affine parameters of the normalization layers.

Table A14: ImageNet-C, severity 5, $B = 64$. Gain Δ (pp) over each model's source accuracy, mean over 15 corruptions, on shuffled streams, on class-window streams with $c = 1$ (batch order shuffled) and in SAR's class order; mean classes per batch in parentheses. Source accuracy is 18.2% for the ResNet-50, 51.6% for the ViT-B/16 and 31.4% for the ResNet-50-GN.

	Shuffled (62)	Class windows c=1 (2.2)	SAR class order (2.2)
ResNet-50 (BN), BN-Adapt	+13.5	−12.6	−13.9
ResNet-50 (BN), TENT	+24.6	−13.4	−14.9
ViT-B/16 (LN), TENT	+10.8	+11.2	+11.9
ResNet-50 (GN), TENT	−2.2	−2.4	−1.8

BN-Adapt's gain falls in all 15 corruptions from shuffled to $c = 1$ streams (-26.1 pp on average, 95% interval $[-32.8, -19.8]$ over corruptions). TENT on the GroupNorm ResNet-50 is insensitive to composition but does not gain at the one learning rate we tested. At $K = 1000$, ΔH separates beneficial from harmful batches less clearly than on CIFAR-10 (AUROC 0.65 to 0.75), and we did not run the gate on ImageNet-C.

E.4 Normalization backbones on CIFAR-10-C

Separately trained GroupNorm and LayerNorm WRN-28-10 models stay within a few points of zero gain at every batch divergence, while the BatchNorm model of the same architecture swings by more than 60 pp (Figure A3). At $B = 200$, SAR on its own GroupNorm source (source accuracy 68.6%) changes accuracy by less than half a point in every stream regime. These models barely gain even on balanced batches, and they start from different accuracies, so this is not a controlled comparison of normalization layers.

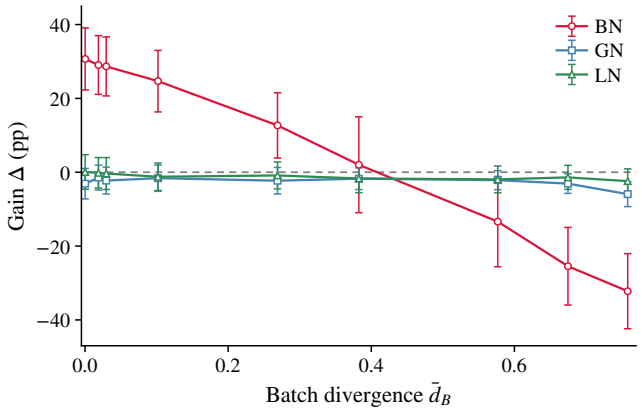


Figure A3: Gain Δ against mean batch divergence $\bar{d}_B$ on CIFAR-10-C for WRN-28-10 models with BatchNorm, GroupNorm and LayerNorm. Means over the 15 corruptions, bars span the corruptions.

F FORECASTING DETAILS

F.1 FORECAST DEFINITIONS AND CALIBRATION

Debiased deviations. For class counts $n_{t,c} = n_t q_{t,c}$, G_t subtracts an unbiased estimate of the current histogram's multinomial variance,

$$V_t = \frac{1}{n_t}\left(1 - \sum_c \frac{n_{t,c}(n_{t,c}-1)}{n_t(n_t-1)}\right) \quad (n_t > 1),$$

and $V_t = 0$ otherwise. $D_{q,t}$ subtracts the exact expected deviation under random regrouping of the pooled counts. With $N_h = \sum_{s<t} n_s$, $N = N_h + n_t$ and $s_c = (N_h h_{t,c} + n_t q_{t,c})/N$,

$$E_{0,t} = \frac{N^2}{n_t N_h (N-1)} \sum_c s_c(1-s_c) \quad (t > 1).$$

Corrected deviations may be negative and are not clipped.

Perturbations. The perturbations of Table A15 act on RGB values in $[0, 1]$ before input normalization; stain shifts rescale and offset the three stain concentrations of an optical-density decomposition, with factors drawn once per condition.

Fit. The coefficients are fitted to the gains g_t of every batch after the first by weighted least squares with b_0 free and $b_D, b_R, b_H \geq 0$, with the four stream types weighted equally. The competence proxy A_t is a non-increasing isotonic fit of the frozen model's balanced accuracy under each appearance condition to the mean cosine distance between a sample's penultimate features and the prototype of its predicted class, with prototypes taken as class means over the clean calibration images.

Table A15: Calibration setup. (a) Perturbation families and levels: five levels per CIFAR family and four per pathology family, which with the clean images give 26 appearance conditions on CIFAR and 21 on each pathology dataset. (b) Streams built from each appearance condition, 25 in total, so each fit uses 650 streams on CIFAR and 525 on each pathology dataset. Streams draw from the calibration pool of 5,000 CIFAR images, 26,845 Camelyon17 patches or 9,995 NCT tiles, on pathology as a class-balanced sample of 2,500 patches or 555 tiles per class.

(a) Perturbations

Family	Parameter	Levels				
CIFAR						
Gaussian noise	standard deviation	0.02	0.06	0.12	0.25	0.40
Gaussian blur	standard deviation	0.3	0.8	1.6	3.2	6.0
Contrast	multiplier	0.9	0.7	0.45	0.2	0.05
JPEG	quality	95	75	50	25	5
Occlusion	area fraction	0.02	0.06	0.12	0.25	0.40
Camelyon17 and NCT						
Stain	strength s	0.05	0.1	0.2	0.3	
Hue	offset	-0.1	-0.05	0.05	0.1	
Saturation	offset	-0.3	-0.15	0.15	0.3	
Brightness	multiplier	0.65	0.8	1.2	1.35	
Contrast	multiplier	0.5	0.75	1.25	1.5	

(b) Calibration streams per appearance condition

Stream type	Construction	Streams
Shuffled	random order	5
Temporal (Gong et al., 2022)	each class spread over 10 consecutive segments, Dirichlet $\delta = 0.1$	5
Class order (Niu et al., 2023)	classes in random order	5
Long-tail (Zhao et al., 2023a)	one exponential class profile, $\rho = 10$ or 100, until a class runs out	10

F.2 ORACLES AND RESETS

Gated TENT (67.28%, the mean over the five stream types of Table 2) exceeds labeled oracles that pick the better of native TENT and the source model for each stream type (64.70%) and for each

stream (65.88%), by 2.58 [1.51, 3.69] and 1.40 [0.39, 2.34] points (95% corruption-cluster bootstrap intervals, 10,000 resamples, conditional on the test pool and the model seed).

Accuracy-estimation resets. AETTA (Lee et al., 2024b) resets TENT to the source weights when its dropout-disagreement accuracy estimate falls below 20% or its five-batch mean drops more than two points below that of the preceding five batches. We port it to the WRN-28-10, with dropout after the three residual stages, and run TENT with AETTA natively and inside the gate (Table A16). Transition streams hold five balanced batches, then batches of one α=0.1 composition whose majority moves to the next class at every batch (155 batches at B=64, 50 at B=200), then five balanced batches. With AETTA, TENT still collapses under class order, and on transition streams gated TENT exceeds it by 12.72 and 11.85 points; adding AETTA to the gate changes its accuracy by at most 0.78 points.

Table A16: TENT with AETTA resets on CIFAR-10-C, native and gated: full-stream accuracy (%), mean over 15 corruptions and five draws.

	B=64			B=200		
Policy	Shuffled	Class order	Transition	Shuffled	Class order	Transition
Source	59.53	59.53	59.53	59.53	59.53	59.53
TENT	82.46	21.09	55.54	83.09	22.23	58.70
TENT+AETTA	81.68	22.55	56.39	83.07	22.31	57.98
Gated TENT	81.92	59.81	69.11	83.07	59.84	69.83
Gated TENT+AETTA	81.14	59.81	68.84	83.06	59.84	69.63

F.3 FORECAST QUALITY

Table A17 scores the forecast on the test batches, with every gate fitted on the calibration streams only. The AUROC is the probability that a batch on which adapting helps ($g_t > 0$) receives a higher forecast than a batch on which it hurts ($g_t < 0$): 1 means every helpful batch is ranked above every harmful one, 0.5 is chance. Degrading is the share of all test batches on which adapting lowers accuracy, the risk of always adapting; Degrading ON is that share among the batches the gate adapts on. For BN-Adapt the AUROC is 0.88 to 0.95 on all four datasets, and on CIFAR and Camelyon17 the gate cuts the share of harmful updates from 29–53% to 3–15%. On NCT-CRC, where the source model is already at 95.7% and 89% of the batches degrade, most updates the gate still makes are harmful, but together they cost 0.4 points.

Table A17: Forecast quality on the test batches, B=64. AUROC: separation by the forecast of batches on which the method gains ($g_t > 0$) from those on which it degrades accuracy ($g_t < 0$). Degrading: share of all batches with $g_t < 0$; Degrading ON: the same share among the batches the gate adapts on. For methods other than BN-Adapt, g_t uses the native method's predictions.

Method	Dataset	Degrading (%)	AUROC	Degrading ON (%)
BN-Adapt	CIFAR-10-C	53.0	0.95	9.3
	CIFAR-100-C	37.9	0.93	15.1
	Camelyon17	28.8	0.88	3.4
	NCT-CRC	88.8	0.90	64.2
SAR-BN	CIFAR-10-C	52.7	0.95	9.2
EATA	CIFAR-10-C	53.7	0.95	9.8
DeYO	CIFAR-10-C	53.4	0.95	9.8
TEMA	CIFAR-10-C	47.4	0.94	8.6
	Camelyon17	19.1	0.88	2.5
Hybrid-TTN	CIFAR-10-C	22.3	0.83	11.8
	Camelyon17	16.3	0.83	5.6
	NCT-CRC	72.2	0.89	40.0
TRIBE	CIFAR-10-C	10.5	0.72	8.2
	CIFAR-100-C	45.0	0.78	19.0
	NCT-CRC	56.5	0.67	43.6

Why the forecast works, and for which methods. The probe's loss of confidence carries the signal: $\Delta H_t > 0$ on 94–100% of the test batches on which BN-Adapt degrades accuracy. Its median is $+0.17$ to $+0.57$ on test batches with at least 90% of one class, against -0.11 to $+0.05$ on shuffled test batches under shift. The forecast carries over to SAR-BN, EATA, DeYO and TEMA (AUROC 0.94–0.95 on CIFAR-10-C, Table A17). It also holds on the states that gated TENT actually visits, although it is fitted on native trajectories: scored against the gain of one adaptation step from the gated state, which is then restored, its AUROC is 0.95 on CIFAR-10-C and 0.89 on CIFAR-100-C (0.94 and 0.86 on native trajectories), and refitting it on calibration streams run under the gate changes gated accuracy by at most 0.33 points on any stream type.

F.4 Forecast ablations

We compare the linear gate with fewer measurements, single-threshold proxies and nonlinear forecasters on BN-Adapt, whose gate can be replayed exactly for every variant, all fitted on the calibration streams only with the gate's weighting. The fitted weights (b_H, b_D, b_R) are $(1.61, 0.31, 0)$ on CIFAR-10-C, $(0.16, 1.85, 0)$ on CIFAR-100-C, $(0, 0.22, 1.04)$ on Camelyon17 and $(1.13, 0.28, 0.08)$ on NCT-CRC: ΔH carries the CIFAR-10-C and NCT-CRC gates, D_q the CIFAR-100-C gate and RG the Camelyon17 gate. Without ΔH the CIFAR-10-C gate ends 7.0 points below the source model on its worst stream type, and D_q alone collapses there (37.7 points below it). Logistic regression on the sign of g_t and an unconstrained gradient-boosted model gain at most 0.40 points over the linear gate outside Camelyon17 (up to 2.0 there), and three further ones (logistic regression weighted by $|g_t|$, a quadratic model with interactions and a monotone gradient-boosted model) at most 0.33 points on any dataset. A threshold on the frozen model's entropy falls 11.1 and 12.6 points below the source model on CIFAR-10-C and CIFAR-100-C, and one on the largest predicted-class share 4.5 points below it on CIFAR-100-C, because a strong corruption also concentrates the frozen model's predictions on batches where adapting helps: on CIFAR-100 calibration batches whose largest predicted-class share exceeds 0.7, the median true share of the most frequent class is 0.11.

Calibration coverage. The gate does not depend on calibration resembling the test data. Without the class-order and long-tail calibration streams, the refitted gate stays within 0.15 points of the full gate on the class-order and persistent test streams of all four datasets. On the six CIFAR-10-C corruptions without a calibration counterpart (snow, frost, fog, brightness, elastic transform, pixelate), the gated BN-based methods keep 99.6% of their shuffled-stream gain and raise the mean accuracy over the five stream types from 47.61% to 77.91% (source 74.29%). On Camelyon17 the gate is calibrated on patches from the training hospitals only and tested on Hospital 2 (Appendix B).

Label-free rule. The rule $\Delta H_t \leq 0$ adapts only when the probe is at least as confident as the frozen model. It needs no labels and no calibration, but because ΔH_t is measured by the frozen model, it makes the same decisions whichever method it wraps: on CIFAR-10-C it adapts on 70.5%, 18.2%, 2.7%, 3.4% and 29.5% of the shuffled, temporal, class-order, persistent and recovery batches for every method. The calibrated gate instead learns from each method's own gains how that method responds to the same measurements. It is more accurate than the label-free rule for every method of Table A18 except TENT on Camelyon17: by 1.1–1.6 points for TENT, SAR-BN, EATA, DeYO and TEMA on CIFAR-10-C, and by 6.8–12.8 points for Hybrid-TTN, RoTTA, UnMix-TNS and TRIBE, which gain on skewed batches and which the label-free rule leaves below their native versions. On Camelyon17, RoTTA's gate relies on RG alone and reaches 85.6% over ten slide orders, against 81.1% for the label-free rule and 81.9% for native RoTTA. With BN-Adapt's frozen gate, TENT, SAR-BN, EATA and DeYO reach 67.17, 66.73, 67.27 and 67.33% on CIFAR-10-C, within 0.12 points of their own gates, so one calibration per dataset serves every BN-based method.

Small batches. The gate does not need recalibration for the batch size: calibrated at $B{=}64$ and applied unchanged to $B{=}200$ streams, the gate applied to BN-Adapt reaches 66.98%, 40.13%, 85.03% and 95.16% on CIFAR-10-C, CIFAR-100-C, Camelyon17 and NCT-CRC, within 0.2 points of the gate calibrated at $B{=}200$ (66.98%, 40.31%, 85.09% and 95.21%). Recalibrated at $B{=}16$ and $B{=}8$ on the same calibration images, the gate keeps working where native adaptation degrades further (Table A19): gated TENT reaches 64.70% and 62.69% over the five stream types, against 36.98% and 30.45% native (source 60.11%), and at $B{=}8$ it is more accurate than native TENT even

on shuffled streams (68.27% against 62.12%). Under persistent skew the gates end 0.9 to 1.8 points below the source model.

Table A18: Calibrated gate against the label-free rule, B=64: full-stream accuracy (%). Calibrated: each method's own gate (Tables 2 and 3). Label-free: adapt only if $\Delta H_t \leq 0$ (stateful methods: < 0, which differs only at ties), with no labels and no calibration. Means are over the test stream types of Tables 2 and 3. The CIFAR-10-C means of the stateful methods come from one replay per method in which the calibrated gate makes the same decisions as in Table 2 (within 0.14 points, 0.55 for TRIBE); RoTTA and UnMix-TNS from a rerun that reproduces Table 2 within 0.06 points. The Camelyon17 rows come from the runs in which the label-free rule was replayed and can differ from Tables 3 and A3 (RoTTA, identity order: $76.28 \rightarrow 85.45$ in Table 3).

Method	Stream	Source	Native	Calibrated	Label-free
TENT, CIFAR-10-C	shuffled	59.53	82.45	81.89	77.90
	temporal	59.53	51.05	63.99	62.85
	class order	59.53	21.14	59.81	59.73
	persistent	62.44	24.55	62.39	62.50
	recovery	59.53	43.97	68.31	66.72
	mean	60.11	44.63	67.28	65.94
TENT, Camelyon17	identity slide order	72.91	56.03	75.01	76.29
	mean, 10 random orders	72.91	54.75	74.73	76.39
	worst random order	72.91	51.82	74.32	74.57
RoTTA, Camelyon17	identity slide order	72.91	73.14	84.84	82.11
	mean, 10 random orders	72.91	81.94	85.55	81.06
	worst random order	72.91	70.24	82.56	79.34
SAR-BN, CIFAR-10-C	mean	60.11	45.74	66.78	65.67
EATA, CIFAR-10-C	mean	60.11	45.23	67.33	66.01
DeYO, CIFAR-10-C	mean	60.11	45.50	67.45	66.04
TRIBE, CIFAR-10-C	mean	60.11	81.31	79.63	66.84
TEMA, CIFAR-10-C	mean	60.11	53.11	67.58	65.95
Hybrid-TTN, CIFAR-10-C	mean	60.11	73.03	72.77	65.95
RoTTA, CIFAR-10-C	mean	60.11	72.87	74.28	66.25
UnMix-TNS, CIFAR-10-C	mean	60.11	74.05	74.39	65.43

Table A19: Small batches on CIFAR-10-C: native $\rightarrow$ gated full-stream accuracy (%), mean over 15 corruptions and five draws, with the gate recalibrated at each batch size on the same calibration images and the settings of Table 2.

B	Method	Shuffled	Temporal $\delta = 0.1$	Class order	Persistent $\alpha = 0.03$	Recovery	Mean
B=16	Source	59.53	59.53	59.53	62.41	59.53	60.11
	BN-Adapt	77.51 → 75.65	52.61 → 63.04	21.40 → 59.55	22.68 → 61.53	43.88 → 66.00	43.62 → 65.16
	TENT	75.00 → 74.41	38.31 → 62.21	15.35 → 59.55	23.11 → 61.50	33.14 → 65.84	36.98 → 64.70
B=8	Source	59.53	59.53	59.53	62.43	59.53	60.11
	BN-Adapt	74.09 → 72.08	50.86 → 62.47	20.80 → 59.38	22.20 → 60.61	42.14 → 64.50	42.02 → 63.81
	TENT	62.12 → 68.27	26.19 → 61.23	12.75 → 59.43	23.62 → 60.89	27.54 → 63.62	30.45 → 62.69