AT-LPC: RELIABILITY-TEMPERED LOG-PARTITION CALIBRATION FOR CONTINUAL LOCAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual local learning provides a modular alternative to end-to-end optimization by assigning learning objectives to individual components. However, the class-score channels produced by these components can lose a consistent comparison reference as learning progresses through a nonstationary stream. We identify this failure mode as *temporal score incomparability*: within-task discrimination can remain useful while scores learned at different stages become unreliable under global all-class comparison. We diagnose this phenomenon in score space and introduce Automatic Tempered Log-Partition Calibration (AT-LPC), a replay-based post-update calibration method that restores a shared score reference without retraining the classifier. AT-LPC evaluates every class-score channel on a common replay memory, estimates class-wise tempered log partitions, and selects a shared temperature using robust score scale and effective sample size (ESS). This reliability-aware tempering suppresses poorly supported score-tail effects while retaining stronger distributional correction when replay evidence is sufficient. In five-seed one-pass experiments, AT-LPC improves final accuracy from 15.64% to 26.90% on CIFAR-10 and from 4.57% to 8.48% on CIFAR-100, while remaining compatible with fixed-schedule updates, blurry arrivals, recent-prior adjustment, and alternative local score models. These results establish score comparability and replay reliability as complementary factors in the stability–plasticity behavior of continual local learning.

1 INTRODUCTION

Local learning provides a modular alternative to end-to-end optimization by assigning learning objectives to individual network components while confining gradients within each block. The Forward-Forward (FF) algorithm, for example, trains local goodness functions using positive and negative examples rather than end-to-end credit assignment (Hinton, 2022). Subsequent work has extended this paradigm through class-wise competition, deeper architectures, and alternative local objectives (Papachristodoulou et al., 2024; Sun et al., 2025; Yao et al., 2026; Xu et al., 2026). Because prediction ultimately compares class evidence produced by locally optimized components, these scores must remain meaningfully comparable on a common scale.

Maintaining such comparability becomes difficult in a continual class-incremental stream. New classes arrive over time while prediction covers all classes observed so far (Rebuffi et al., 2017; van de Ven & Tolias, 2019). Incoming classes receive dense updates from newly observed examples, whereas earlier classes are maintained mainly through limited replay. As a result, class-score channels can evolve under different update histories and gradually drift in offset, spread, or distributional shape. A newer class may therefore obtain systematically larger scores than an older one, altering the global prediction even when the older class remains distinguishable from its original competitors. We refer to this failure mode as *temporal score incomparability*.

This phenomenon separates local discrimination from global score comparison. Useful class information need not disappear for all-class prediction to fail; instead, evidence learned at different stages may no longer share a reliable numerical reference. Figure 1 illustrates this distinction. A simple CIFAR-10 diagnostic makes it concrete: all-class accuracy is 17.78%, while restricting prediction to the correct two-class task raises accuracy to 62.66%. With the backbone and candidate set fixed, a class-wise additive score correction raises all-class accuracy to 24.19%. The former changes which

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

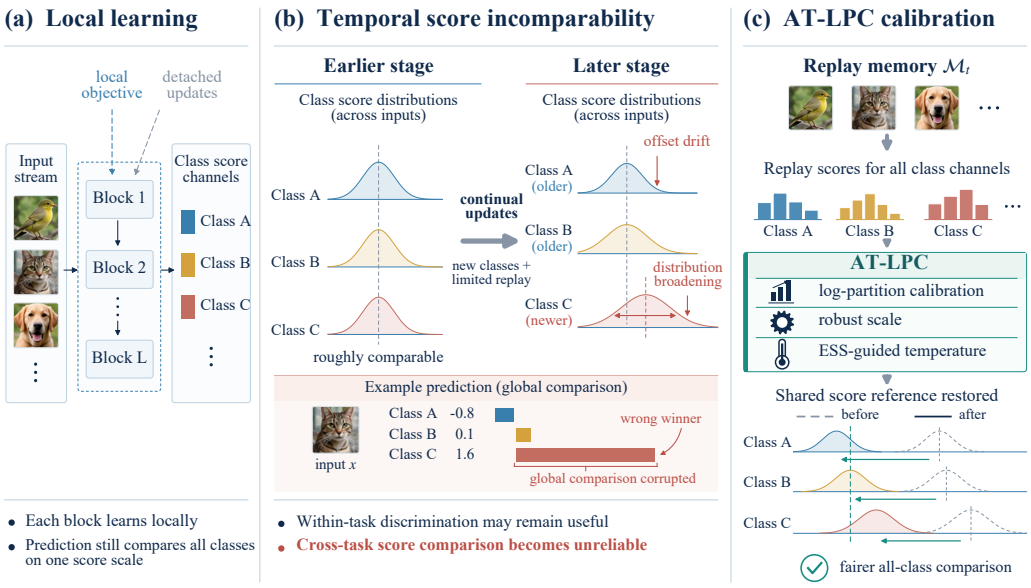

Figure 1. Temporal score incomparability and AT-LPC in continual local learning. (a) Local learning optimizes detached blocks with local objectives, while prediction compares their class evidence on a shared score scale. (b) Continual updates and limited replay can shift or broaden class-score distributions, preserving useful within-task discrimination while degrading cross-task comparison. (c) AT-LPC evaluates all class-score channels on a shared replay reference and uses robust scale and ESS-guided tempering to obtain reliability-aware class-wise corrections.

classes compete, whereas the latter changes only their score reference. Together, these observations indicate that score comparison itself accounts for a nontrivial part of continual prediction error.

We study this mismatch through *class-wise replay partitions*. At each calibration step, every class-score channel is evaluated on the same replay examples, giving all channels a common empirical input reference. For each class, a log partition summarizes its score distribution under this reference, and subtracting the partition yields *Log-Partition Calibration* (LPC). Unlike simple mean-centering, the log partition also reflects distributional shape. However, its exponential average can become dominated by a small number of large replay scores, making full partition correction sensitive to poorly supported score tails.

To address this limitation, we introduce *Automatic Tempered Log-Partition Calibration* (AT-LPC). AT-LPC selects a shared temperature from the current replay statistics rather than always using the full unit-temperature partition. A robust score-scale statistic captures the typical spread of the class channels, while normalized effective sample size (ESS) measures how concentrated the exponential weights are across replay examples. When replay support is broad, the method retains more of the distributional correction provided by LPC. When the weights concentrate on a few examples, tempering moves the partition toward mean-centering and reduces the influence of weakly supported tails. The resulting class-wise biases are cached after calibration, so inference requires only an additive score correction and no classifier retraining.

We evaluate AT-LPC under strict one-pass class-incremental learning and examine whether its behavior persists across different stream structures, local score models, and complementary score-adjustment strategies. The results consistently support the view that score comparability is a distinct component of the stability–plasticity problem in continual local learning.

Our main contributions are:

- We identify temporal score incomparability in continual local learning, where class-score channels can remain locally discriminative while losing a reliable reference for global all-class comparison.

- We introduce AT-LPC, a replay-based post-update calibration method that combines class-wise tempered log-partition correction with robust score-scale estimation and ESS-guided shared temperature selection.
- We characterize how replay-based partition correction and reliability-aware tempering affect the stability–plasticity behavior of continual local learning across multiple stream and score-model settings.

2 Related Work

Local learning and goodness scores. Forward-Forward trains layers with local goodness objectives using positive and negative examples (Hinton, 2022). DeeperForward extends this idea to deeper architectures and studies how normalization and architectural choices affect forward-only training (Sun et al., 2025). SFFA introduces excitatory and inhibitory channels within a symmetric local objective (Terres-Escudero et al., 2025). Local optimization produces class-related scores whose distributions may evolve independently across blocks. AT-LPC operates on these scores after local updates and calibrates their cross-class reference.

Learning from a stream. Replay methods retain selected past examples, while response-consistency methods preserve earlier model outputs (Buzzega et al., 2020). ER-ACE restricts the loss on incoming examples to current classes and reduces abrupt representation changes (Caccia et al., 2022). Layerwise Proximal Replay constrains changes in intermediate representations during replay updates (Yoo et al., 2024). Repeated or overlapping arrivals further show that stream structure affects continual evaluation (Hemati et al., 2023). These methods mainly shape what the learner retains during training. We study how these locally learned scores remain comparable as the stream evolves.

Classifier correction. Classifier correction adjusts outputs during or after incremental training. BiC learns an affine correction from held-out data (Wu et al., 2019). Weight Alignment and ScaIL rescale classifier weights to reduce class imbalance in the output layer (Zhao et al., 2020; Belouadah & Popescu, 2020). IL2M uses stored activation statistics to correct incremental predictions (Belouadah & Popescu, 2019). AT-LPC instead estimates a tempered partition for each class-score channel from a shared replay reference. Its correction depends on both the channel level and the support of its exponential tail.

Label priors and score distributions. Logit adjustment and Balanced Softmax compensate for class-frequency imbalance through label priors (Menon et al., 2021; Ren et al., 2020). LAS applies recent-prior adjustment during online continual training and therefore changes the training gradients (Huang et al., 2024). AT-LPC uses a different signal: the distribution of current replay scores after each update. Even class-balanced replay can produce unequal score partitions. Prior adjustment and partition calibration act on different quantities. Both can be applied within the same local model.

3 Temporal Score Incomparability

3.1 Local scores in a class-incremental stream

Let $\mathcal{C}_t$ contain the classes observed by time t. The replay memory is $\mathcal{M}_t$. At each selected local block ℓ, the learner produces a goodness score $g_{t,\ell,c}(x)$ for class c. We aggregate these local scores as

$$s_{t,c}(x) = \sum_{\ell \in \mathcal{L}} g_{t,\ell,c}(x), \qquad \hat{y}_t(x) = \arg\max_{c \in \mathcal{C}_t} s_{t,c}(x). \tag{1}$$

Here, $\mathcal{L}$ contains the blocks used for prediction. Representations passed between blocks are detached during local training. Each block can therefore change its score distribution through its own updates. Prediction compares the aggregated scores of all seen classes on one scale.

Class-wise normalization changes as the stream progresses. We call this *temporal partition drift*. It includes offsets and distribution-shape changes that affect exponential averages. The two introductory diagnostics probe candidate-set sensitivity and recoverable offset effects.

An old class may outscore its original competitor but lose to an upward-shifted later class. Calibration adjusts the reference for this global comparison.

Figure 2 diagnoses temporal score incomparability at the decision level. Many examples remain correct within their task but fail under all-class competition; with the backbone fixed, score-reference correction raises all-class accuracy from 17.78% to 24.19% while retaining all seen classes as candidates.

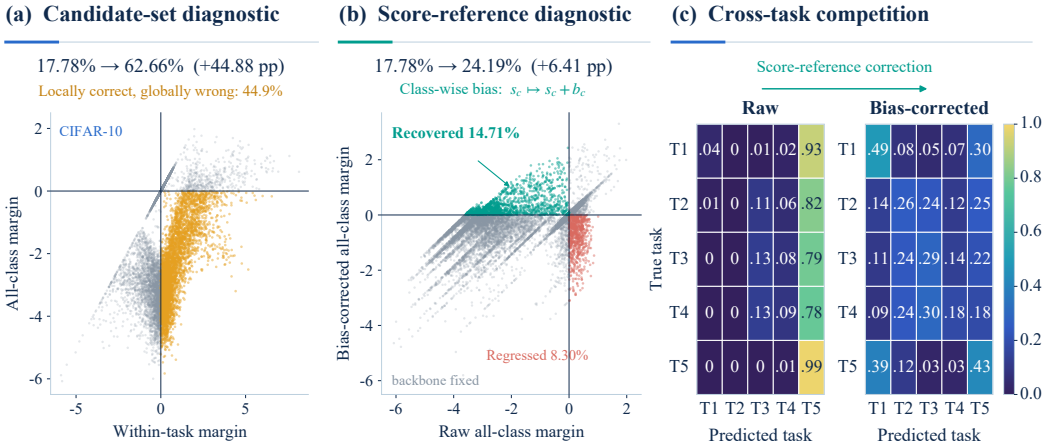

Figure 2. Decision-level diagnosis of temporal score incomparability on CIFAR-10. (a) Candidate-set restriction exposes the gap between local discrimination and all-class prediction. (b) Frozen-backbone score-reference correction partially restores global decisions. (c) Task-level predictions show reduced cross-task bias while retaining all seen classes as candidates.

3.2 A SHARED REPLAY REFERENCE

For each class c, define its replay scores as $S_{t,c} = \{s_{t,c}(x_i) : x_i \in \mathcal{M}_t\}$, with $m = |\mathcal{M}_t|$. Every class channel is evaluated on every image in the replay memory. Using the same replay set keeps the empirical input reference fixed across classes. For temperature λ, define

$$\widehat{A}_{t,c}(\lambda) = \begin{cases} \dfrac{1}{\lambda} \log \left[\dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \exp(\lambda s_{t,c}(x_i)) \right], & \lambda > 0, \\ \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} s_{t,c}(x_i), & \lambda = 0. \end{cases} \tag{2}$$

This partition averages one class channel over replay inputs. A softmax denominator instead compares classes for one input. At $\lambda = 1$, subtracting the partition normalizes each channel relative to its empirical replay distribution. We refer to this unit-temperature case as fixed log-partition calibration (LPC).

The calibrated prediction score is

$$\widetilde{s}_{t,c}(x) = s_{t,c}(x) - \beta \widehat{A}_{t,c}(\lambda_t). \tag{3}$$

The strength β controls how much correction is applied, while the temperature determines which parts of the score distribution influence the partition. Mean-centering appears as the $\lambda \to 0$ endpoint, whereas AT-LPC selects one shared temperature λ_t from the replay statistics of all seen classes.

Effect on decision margins. For two classes c and d,

$$\widetilde{s}_{t,c}(x) - \widetilde{s}_{t,d}(x) = s_{t,c}(x) - s_{t,d}(x) - \beta \left[\widehat{A}_{t,c}(\lambda_t) - \widehat{A}_{t,d}(\lambda_t) \right]. \tag{4}$$

Calibration shifts the pairwise decision threshold by a replay-derived constant. Differences between inputs within each class channel remain unchanged.

Figure 3 follows the computation from replay inputs to prediction. Scale and ESS determine the shared temperature and cached class partitions.

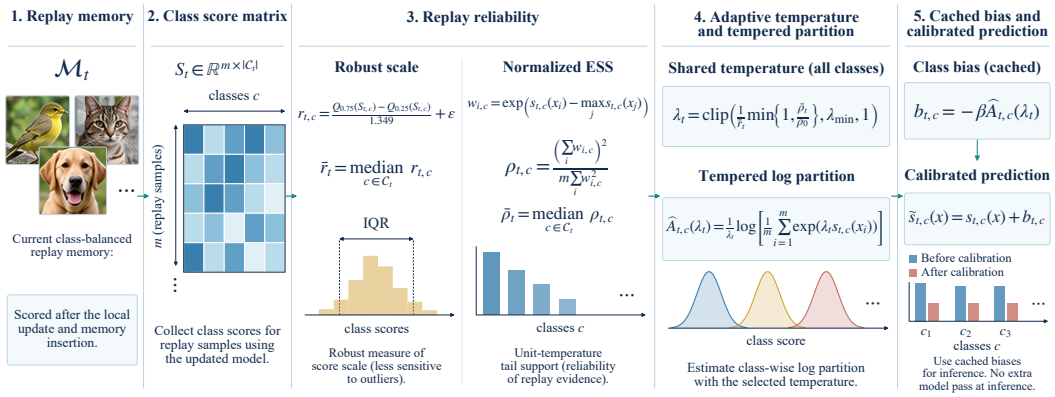

Figure 3. AT-LPC computation after a local update. The updated model scores every replay image for all seen classes. Robust scale and normalized ESS select one shared temperature. Tempered class partitions produce a cached bias vector, which is added to query scores. Distributions and score bars are schematic.

4 Automatic Reliability Tempering

4.1 Scale and Tail Support

AT-LPC selects its temperature from two replay statistics: score scale and tail support. For each class, we estimate a robust score scale as

$$r_{t,c} = \frac{Q_{0.75}(S_{t,c}) - Q_{0.25}(S_{t,c})}{1.349} + \epsilon, \qquad \bar{r}_t = \text{median}_{c \in \mathcal{C}_t}\, r_{t,c}. \tag{5}$$

The interquartile range measures central spread with less sensitivity to isolated extremes. Dividing by 1.349 gives standard-deviation units for a Gaussian distribution. The class median supplies one replay-scale estimate.

Tail support is measured from the unit-temperature exponential weights,

$$w_{i,c} = \exp\left(s_{t,c}(x_i) - \max_j s_{t,c}(x_j)\right), \qquad \rho_{t,c} = \frac{(\sum_i w_{i,c})^2}{m \sum_i w_{i,c}^2}, \qquad \bar{\rho}_t = \text{median}_{c \in \mathcal{C}_t}\, \rho_{t,c}. \tag{6}$$

Here $m\rho_{t,c}$ is the effective sample size (ESS). The normalized value $\rho_{t,c}$ lies in $[1/m, 1]$. It reaches one for uniform weights and approaches $1/m$ when one replay example dominates. Max subtraction improves numerical stability without changing the ESS.

The shared temperature is

$$\lambda_t = \text{clip}\left(\frac{1}{\bar{r}_t} \min\left\{1, \frac{\bar{\rho}_t}{\rho_0}\right\}, \lambda_{\min}, 1\right), \qquad \rho_0 = 0.08, \quad \lambda_{\min} = 0.005. \tag{7}$$

With broad replay support, score scale largely determines the temperature. As the exponential weights become concentrated, the ESS factor reduces the temperature further. The upper bound $\lambda_t = 1$ recovers fixed LPC, while the lower bound keeps the computation away from zero.

4.2 How Tempering Changes the Partition

Temperature controls how strongly the partition depends on the shape of the score distribution. Let S_c denote the score of class c under a fixed reference distribution. With mean μ_c, variance v_c, and third cumulant $\kappa_{3,c}$,

$$A_c(\lambda) = \frac{1}{\lambda} \log \mathbb{E}\left[e^{\lambda S_c}\right] = \mu_c + \frac{\lambda}{2} v_c + \frac{\lambda^2}{6} \kappa_{3,c} + O(\lambda^3). \tag{8}$$

Lower temperatures reduce the contribution of variance, skew, and higher-order tail terms. As $\lambda \to 0$, the partition approaches the mean. Tempering therefore changes which parts of the replay-score distribution contribute to calibration.

ESS measures how much replay support is available for the exponential average. For $Y_i = e^{\lambda S_i}$, let $\widehat{\mathrm{CV}}^2(Y)$ denote the empirical squared coefficient of variation. Then

$$\frac{\widehat{\mathrm{CV}}^2(Y)}{m} = \frac{1}{\widehat{\mathrm{ESS}}_\lambda} - \frac{1}{m}. \tag{9}$$

This identity links concentrated exponential weights to variation in the empirical exponential mean. Under iid sampling, this quantity enters the delta-method variance estimate with a factor of λ^{-2}.

The weights also describe the influence of individual replay examples. Let

$$p_{i,c}(\lambda) = \frac{e^{\lambda S_{i,c}}}{\sum_j e^{\lambda S_{j,c}}}.$$

Then

$$\frac{\partial \widehat{A}_c(\lambda)}{\partial S_{i,c}} = p_{i,c}(\lambda), \qquad \sum_i p_{i,c}(\lambda)^2 = \frac{1}{\widehat{\mathrm{ESS}}_\lambda}. \tag{10}$$

Diffuse weights spread influence across replay examples; concentrated weights place it on a few examples. AT-LPC lowers the temperature as support concentrates, moving the partition toward mean-centering. All classes share this temperature, selected from class medians to reflect the typical replay state. Broad support keeps the estimate closer to full LPC.

Offsets and scale. Adding a constant b_c to one class channel adds the same constant to its partition. Full correction therefore removes the offset exactly. With strength β, the residual shift is $(1-\beta)b_c$. A common positive rescaling also changes the appropriate temperature. The inverse-scale term compensates for this effect before clipping, while the unit-temperature ESS can change with the rescaled scores.

4.3 Online execution and cost

AT-LPC runs after the local training update and memory insertion. Replay examples join the incoming minibatch when available. Every ten minibatches, the model scores the current replay memory for all seen classes. These scores are used to compute the robust scale, normalized ESS, temperature, and class partitions.

The resulting calibration bias is $b_{t,c} = -\beta \widehat{A}_{t,c}(\lambda_t)$, and is cached until the next refresh. Prediction uses $\widetilde{s}_{t,c}(x) = s_{t,c}(x) + b_{t,c}$, followed by the maximum over all seen classes.

Each refresh requires one replay scoring pass, $O(m|\mathcal{C}_t|)$ reductions, and quantile estimation for the IQR. Between refreshes, inference adds $O(|\mathcal{C}_t|)$ biases without another model pass. Persistent storage holds the bias vector and scalar statistics; the temporary score matrix uses $O(m|\mathcal{C}_t|)$ space.

5 Experimental Protocol

Streams and memory. We evaluate strict one-pass class-incremental streams on CIFAR-10, CIFAR-100, and Tiny-ImageNet-200. CIFAR-10 contains five two-class phases, CIFAR-100 contains ten ten-class phases, and Tiny-ImageNet contains twenty ten-class phases. Each incoming example is used once as stream data, while stored examples may be replayed. The main replay memory stores 16 images per class. Prediction is performed over all classes observed so far, without access to task identity. We also evaluate a blurry CIFAR-10 stream with overlap width 0.5, where neither training nor calibration receives task IDs.

Models and baselines. Our primary model contains nine detached convolutional goodness blocks with channel sizes $(20, 20, 20, 40, 40, 40, 80, 80, 80)$. Matched local baselines include experience replay (ER), fixed LPC, Weight Alignment-style correction, and a memory-matched BiC-style correction. We also evaluate independent linear classifiers attached to local blocks and a continual adaptation of DeeperForward TinyCNN. BP ResNet-18 with ER, ER-ACE, joint ER, and ER-LAS provides end-to-end continual-learning references.

Calibration settings. The main correction strength is $\beta = 0.9$, selected from $\{0.5, 0.75, 0.9, 1\}$ using one CIFAR-10 development seed. using one CIFAR-10 development seed. The controller constants are fixed at $\rho_0 = 0.08$ and $\lambda_{\min} = 0.005$. DeeperForward TinyCNN uses an architecture-specific $\beta = 0.5$, selected on seed zero and then transferred to CIFAR-100. Online calibration refreshes every ten minibatches. The deployed calibration statistics are computed directly from replay scores.

Metrics and uncertainty. We report final average accuracy, anytime accuracy, average forgetting, and mean current-task accuracy. They measure end-of-stream performance, performance during training, retention, and learning of new classes. Main CIFAR experiments use five matched seeds. We report mean and standard deviation where available. Paired confidence intervals use nonparametric bootstrap resampling over matched seeds, with 20,000 resamples for the main comparisons and 100,000 for the blurry-stream analysis. Single-seed Tiny-ImageNet and SFFA experiments are reported separately.

6 RESULTS

Table 1 reports the four primary metrics. Figure 4 summarizes the gains and their relation to current-task learning.

Table 1. Strict one-pass class-incremental results (mean $\pm$ standard deviation over five seeds). AT denotes AT-LPC. Entries without a dispersion term report means. Higher is better except forgetting.

Method	Final ↑	Anytime ↑	Forget. ↓	Current ↑
CIFAR-10				
Local ER	15.64 ± 0.55	39.77 ± 0.66	67.31 ± 1.76	69.48 ± 1.00
+ fixed LPC ($\beta = .9$)	25.61 ± 2.51	44.03 ± 1.70	29.97 ± 4.31	49.59 ± 3.11
+ AT	26.90 ± 2.05	44.81 ± 1.64	32.89 ± 3.23	53.22 ± 3.69
Local ER-LAS	23.08 ± 1.66	45.37 ± 0.42	53.20 ± 4.48	65.64 ± 1.96
Local ER-LAS + fixed LPC	26.42 ± 1.86	45.20 ± 1.67	25.87 ± 5.22	46.58 ± 4.23
Local ER-LAS + AT	27.39 ± 1.32	45.92 ± 1.29	31.28 ± 3.77	51.93 ± 2.39
Linear-head ER	20.64	42.60	54.92	64.58
Linear-head ER + AT	24.56	44.18	30.26	48.77
CIFAR-100				
Local ER	4.57 ± 0.55	14.65 ± 0.48	31.10 ± 0.80	32.56 ± 0.99
+ fixed LPC ($\beta = .9$)	8.36 ± 0.37	19.57 ± 0.44	14.54 ± 0.76	21.45 ± 0.56
+ AT	8.48 ± 0.32	19.59 ± 0.42	14.53 ± 0.65	21.56 ± 0.48
Local ER-LAS	7.93	16.91	30.18	35.09
Local ER-LAS + AT	11.52	21.05	17.88	27.62

6.1 PARTITION CORRECTION AND AUTOMATIC TEMPERING

Partition calibration substantially improves long-horizon performance in both primary streams. On CIFAR-10, AT-LPC raises final accuracy from 15.64% to 26.90% and anytime accuracy from 39.77% to 44.81%. Forgetting drops by 34.42 percentage points. The gain comes with lower current-task accuracy, indicating a clear shift toward retention.

Automatic tempering improves the operating point reached by fixed LPC at the same correction strength. Relative to fixed LPC, AT-LPC gains 1.29 final-accuracy points and 0.79 anytime points. Current-task accuracy recovers by 3.63 points, while forgetting increases by 2.92 points. Thus, tempering retains most of the long-term benefit of partition correction while reducing its effect on newly arriving classes.

CIFAR-100 shows a similar gain from partition correction. AT-LPC improves final accuracy by 3.91 points and anytime accuracy by 4.94 points over local ER. Fixed and automatic LPC have paired gaps of 0.112 final points and 0.018 anytime points. Partition correction accounts for most of this gain.

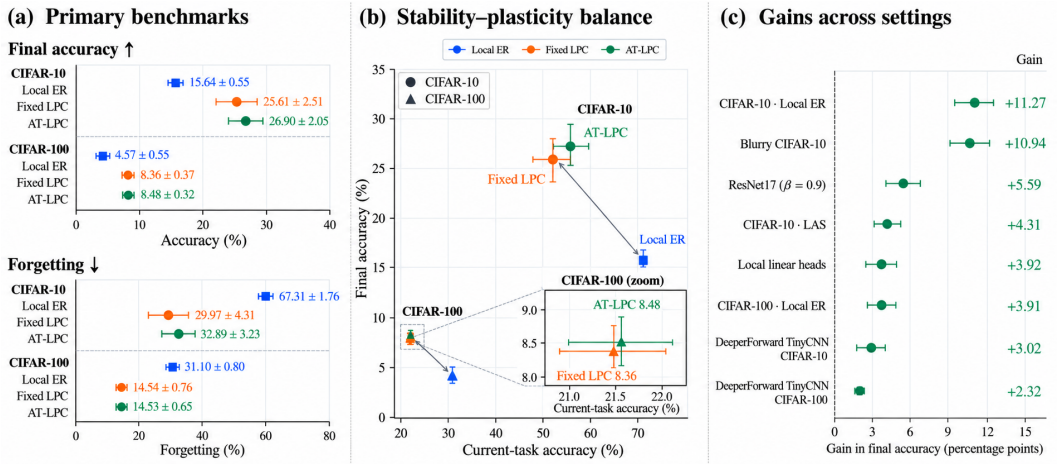

Figure 4. Performance and operating points across local-learning settings. (a) Primary CIFAR results. (b) Final and current-task accuracy, with a local view of the CIFAR-100 calibration results. Error bars in (a,b) show standard deviations over five seeds. (c) Paired final-accuracy gains over each stated baseline; bars show 95% bootstrap intervals. The ResNet17 row uses the recorded shared-checkpoint variant described in Appendix J.

6.2 Fixed-schedule updates and blurry arrivals

AT-LPC does not require task boundaries to determine when calibration is updated. On CIFAR-10, refreshing the calibration statistics every ten minibatches closely matches boundary-triggered calibration. Final and anytime differences are -0.052 and -0.048 percentage points, respectively.

The same schedule remains effective when neighboring classes overlap in time. On blurry CIFAR-10, online AT-LPC raises final seen-class accuracy from 16.53% to 27.47%, a paired gain of 10.94 points. Mean checkpoint accuracy improves by 4.39 points. The selected temperature stays near one early in the stream and decreases later as the replay weights become more concentrated. The controller follows the evolving replay statistics throughout the stream.

Figure A7 in Appendix K aligns the accuracy and temperature trajectories on a shared training axis. It shows when tempering becomes active during overlapping arrivals.

6.3 Composition with recent-prior adjustment

Recent-prior adjustment and partition calibration operate on different signals. LAS modifies training through recent class-frequency estimates, whereas AT-LPC acts on the replay-score distributions after the update. We therefore evaluate the two corrections within the same nine-block local model.

On CIFAR-10, Local ER-LAS reaches 23.08% final accuracy. Adding AT-LPC increases this value to 27.39%, with a paired gain of 4.31 points. Forgetting decreases by 21.92 points. Relative to LAS with fixed LPC, automatic tempering adds 0.97 final points and 0.72 anytime points, while recovering 5.35 points of current-task accuracy.

The combination also improves CIFAR-100. Final accuracy increases from 7.93% to 11.52%, and anytime accuracy rises from 16.91% to 21.05%. Score-distribution calibration thus remains useful after recent-prior correction shapes the training objective.

6.4 Transfer across local score models

Calibration also helps independent linear classifiers attached to detached blocks. Final CIFAR-10 accuracy rises from 20.64% to 24.56%, a paired gain of 3.92 points.

We also evaluate a continual adaptation of DeeperForward TinyCNN. With $\beta = 0.5$, AT-LPC improves final CIFAR-10 accuracy from 28.69% to 31.71%. Using the same strength on CIFAR-100 raises final accuracy from 5.55% to 7.87%. The gain transfers across both datasets.

Appendix J extends the analysis to a recorded ResNet17 shared-checkpoint variant, including task-wise prediction changes, component ablations, calibration-reference sensitivity, and temperature–strength trade-offs.

6.5 Controller behavior and correction strength

Tempering is most useful when replay poorly supports full-tail LPC. Across 20 matched post-initial CIFAR-10 LAS checkpoints, activation $1 - \lambda$ correlates with the accuracy advantage over fixed LPC (Spearman 0.769). Tail unreliability gives a correlation of 0.672. Stronger tempering thus aligns with greater gains over full-tail calibration.

Figure A8 shows these checkpoint associations with seed-specific markers. A discrete temperature grid also tests the controller. The final mean gap to the test-label grid oracle is 0.212 points on CIFAR-10 and 0.022 on CIFAR-100. Across $\rho_0 \in [0.04, 0.16]$, final accuracy spans 0.882 points on CIFAR-10 and remains unchanged on CIFAR-100.

Correction strength controls a different aspect of the method. On CIFAR-10 LAS, increasing β from 0 to 1 raises final accuracy from 23.08% to 27.61% and reduces forgetting from 53.20% to 27.80%. Current-task accuracy decreases from 65.64% to 49.14%, while anytime accuracy peaks at $\beta = 0.5$. The primary setting $\beta = 0.9$ therefore emphasizes long-horizon retention, whereas the temperature controls how strongly each correction relies on the replay-score tail.

Figure A9 shows the full strength curve and its retention–plasticity trade-off. Replay-capacity results appear in Figure A10.

7 Conclusion

Class-wise goodness scores need a common reference as a local learner moves through a stream. We identified temporal score incomparability as a failure mode in which locally useful class evidence is no longer reliably comparable across stages of learning. Replay partitions provide a shared reference for this comparison, while their exponential weights indicate how broadly the memory supports the resulting estimate. AT-LPC combines this reference with robust score scale and ESS-guided tempering to reduce the influence of weakly supported score tails. Across the evaluated local-learning settings, partition calibration improves long-horizon class prediction and alters the balance between retention and current-class learning, while automatic tempering moderates this trade-off when full-tail calibration is poorly supported. The same formulation remains applicable under fixed-schedule updates, blurry arrivals, LAS composition, and alternative local score models. These results suggest that score comparability should be treated as a distinct part of continual local learning, alongside representation learning and replay.

Reproducibility Statement

The appendices provide the full AT-LPC formulation and evaluation protocol, including online calibration, temperature selection, numerical implementation, parameter selection, evaluation metrics, and paired statistical analysis. The supplementary code implements the model-agnostic AT-LPC calibration layer, its fixed-schedule online refresh, principal controller ablations, and numerical tests of the core equations. Post-hoc diagnostic analyses are explicitly separated from the deployed calibration rule.

AI Use Statement

Generative AI tools assisted with code organization, artifact indexing, figure illustration, language editing, mathematical exposition, and document preparation. The authors are responsible for the scientific claims, sources, and final submission.

REFERENCES

Eden Belouadah and Adrian Popescu. IL2M: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 583–592, 2019.

Eden Belouadah and Adrian Popescu. ScaIL: Classifier weights scaling for class incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1266–1275, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022.

Hamed Hemati, Andrea Cossu, Antonio Carta, Julio Hurtado, Lorenzo Pellegrini, Davide Bacciu, Vincenzo Lomonaco, and Damian Borth. Class-incremental learning with repetition. In *Proceedings of the 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pp. 437–455. PMLR, 2023.

Geoffrey E. Hinton. The Forward-Forward algorithm: Some preliminary investigations. *CoRR*, abs/2212.13345, 2022.

Zhehao Huang, Tao Li, Chenhe Yuan, Yingwen Wu, and Xiaolin Huang. Online continual learning via logit adjusted softmax. *Transactions on Machine Learning Research*, 2024.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Andreas Papachristodoulou, Christos Kyrkou, Stelios Timotheou, and Theocharis Theocharides. Convolutional channel-wise competitive learning for the Forward-Forward algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14536–14544, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Liang Sun, Yang Zhang, Weizhao He, Jiajun Wen, Linlin Shen, and Weicheng Xie. DeeperForward: Enhanced Forward-Forward training for deeper and better performance. In *International Conference on Learning Representations*, 2025.

Erik B. Terres-Escudero, Javier Del Ser, and Pablo Garcia-Bringas. A contrastive symmetric Forward-Forward algorithm (SFFA) for continual learning tasks. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 49–69. PMLR, 2025.

Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Siyuan Xu, Yujie Wu, Jibin Wu, Lei Deng, Mingkun Xu, Qinghao Wen, and Guoqi Li. Advancing the Forward-Forward algorithm towards high-performance deep local learning. *Neural Networks*, 200:108765, 2026. doi: 10.1016/j.neunet.2026.108765.

Jie-En Yao, Hong-En Chen, and C.-C. Jay Kuo. HCL-FF: Hierarchical and contrastive learning for Forward-Forward algorithm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27429–27438, 2026.

Jinsoo Yoo, Yunpeng Liu, Frank Wood, and Geoff Pleiss. Layerwise proximal replay: A proximal point method for online continual learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57199–57216. PMLR, 2024.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13208–13217, 2020.

A NOTATION AND THE REPLAY REFERENCE

The calibration layer receives a matrix of scores and returns a class-bias vector. Table 2 collects the quantities used in this transformation. Time subscripts are omitted when a statement concerns one fixed checkpoint.

Table 2. Notation for replay partition calibration.

Symbol	Meaning
$\mathcal{C}_t$	Set of classes observed by stream time t.
$\mathcal{M}_t$, m	Replay memory and its current number of stored images.
$\mathcal{L}$	Local blocks selected for additive prediction.
$s_{t,c}(x)$	Additive score of class c for input x.
$S_{t,c}$	Scores of channel c on every image in $\mathcal{M}_t$.
$\widehat{A}_{t,c}(\lambda)$	Tempered empirical log partition of channel c.
$r_{t,c}$, $\bar{r}_t$	Robust class-wise scale and its median across seen classes.
$\rho_{t,c}$, $\bar{\rho}_t$	Normalized unit-temperature ESS and its class median.
ρ_0	Reference normalized ESS, fixed at 0.08.
λ_t	Shared temperature, clipped to $[0.005, 1]$.
β	Correction strength; 0.9 for the primary backbone.
$b_{t,c}$	Cached additive bias $-\beta\widehat{A}_{t,c}(\lambda_t)$.
v_c, $\kappa_{3,c}$	Variance and third cumulant of a score distribution.

A.1 NORMALIZATION OVER INPUTS

The memory defines the empirical input measure

$$\widehat{P}_{\mathcal{M}_t} = \frac{1}{m}\sum_{i=1}^{m}\delta_{x_i}. \tag{11}$$

For each class, the partition is computed under this same measure:

$$\widehat{A}_{t,c}(\lambda) = \lambda^{-1}\log \mathbb{E}_{\widehat{P}_{\mathcal{M}_t}}[e^{\lambda s_{t,c}(X)}]. \tag{12}$$

The partition normalizes a class channel across replay inputs. Labels determine class-balanced memory construction; the resulting score matrix determines calibration. An image can contribute to every channel's partition, regardless of its own label.

At full correction, define $q_{t,c}(x) = \exp\{\lambda[s_{t,c}(x) - \widehat{A}_{t,c}(\lambda)]\}$. Its memory average is exactly one:

$$\frac{1}{m}\sum_{i=1}^{m} q_{t,c}(x_i) = e^{-\lambda\widehat{A}_{t,c}(\lambda)}\frac{1}{m}\sum_{i=1}^{m} e^{\lambda s_{t,c}(x_i)} = 1. \tag{13}$$

The identity explains the shared reference imposed by LPC. The corrected channel expresses how strongly an input scores relative to that channel's replay average in exponential space. These values are channel-wise normalized scores, rather than posterior class probabilities.

A.2 WHY THE MEMORY IS SHARED ACROSS CHANNELS

Class-conditioned references would mix input-distribution changes with score-channel changes. For example, one channel might be evaluated on easy within-class examples and another on a broader set. Their partitions would then describe different reference measures. Evaluating all channels on all stored images holds the empirical input measure fixed within a refresh. The remaining differences come from the scores produced on that measure.

The reference still evolves as the stream and memory change. At each refresh, the current model scores the current memory. Stored images therefore need not retain the scores they had when first observed. The bias tracks the present score distributions of retained examples. This is why the update order places calibration after training and memory insertion.

B Analytical Properties of Tempered Partitions

B.1 Mean-centering as a continuous endpoint

Consider a finite score vector $S_1, \ldots, S_m$. Define

$$K(\lambda) = \log\left(\frac{1}{m}\sum_{i=1}^{m} e^{\lambda S_i}\right), \qquad A(\lambda) = K(\lambda)/\lambda. \tag{14}$$

Since $K(0) = 0$, l'Hôpital's rule gives

$$\lim_{\lambda \to 0} A(\lambda) = K'(0) = \frac{1}{m}\sum_{i=1}^{m} S_i = \mu. \tag{15}$$

The finite empirical moment-generating function is smooth near zero. Its derivatives give the empirical cumulants. Expanding K and dividing by λ yields

$$A(\lambda) = \mu + \frac{\lambda}{2}v + \frac{\lambda^2}{6}\kappa_3 + \frac{\lambda^3}{24}\kappa_4 + O(\lambda^4). \tag{16}$$

Here $v = m^{-1}\sum_i (S_i - \mu)^2$, and κ_4 is the fourth cumulant. This expansion describes how temperature introduces distribution shape beyond the mean. A broad or skewed channel can receive a different correction even when its mean matches another channel.

The population expansion has the same form when its moment-generating function exists near zero. The empirical statement requires only finite scores. This distinction allows the local expansion to explain the implemented estimator directly, without a sampling model for memory construction.

B.2 Bounds and monotonicity

For $\lambda > 0$, Jensen's inequality gives a lower bound, while the largest score gives an upper bound:

$$\mu \le A(\lambda) \le S_{\max}, \qquad S_{\max} = \max_i S_i. \tag{17}$$

The lower bound follows from $\log \mathbb{E}[e^{\lambda S}] \ge \lambda \mathbb{E}[S]$. The upper bound follows because each exponential is at most $e^{\lambda S_{\max}}$. Tempering therefore moves within an interpretable interval between the mean and the maximum.

Monotonicity follows from an information identity. Set

$$p_i(\lambda) = \frac{e^{\lambda S_i}}{\sum_j e^{\lambda S_j}}, \qquad u_i = 1/m. \tag{18}$$

Since $K'(\lambda) = \sum_i p_i(\lambda) S_i$,

$$A'(\lambda) = \frac{\lambda K'(\lambda) - K(\lambda)}{\lambda^2} \tag{19}$$

$$= \frac{D_{\mathrm{KL}}(p(\lambda) \| u)}{\lambda^2} \ge 0. \tag{20}$$

The equality uses $\log[p_i(\lambda)/u_i] = \lambda S_i - K(\lambda)$. Thus increasing temperature raises the partition unless all scores coincide. Different channels can rise at different rates, which changes their relative correction.

B.3 Sensitivity to individual replay scores

The derivative of the partition with respect to a replay score is

$$\frac{\partial A(\lambda)}{\partial S_i} = p_i(\lambda). \tag{21}$$

This gives the exponential weights a direct influence interpretation. A memory image with a large weight has a large local effect on the partition. Under uniform weights, the derivative is $1/m$ for

every image. With concentrated weights, a few images account for most of the estimator's local sensitivity.

For a small perturbation vector Δ, the first-order change is $p(\lambda)^\top \Delta$. Cauchy–Schwarz gives

$$|p(\lambda)^\top \Delta| \le \|p(\lambda)\|_2 \|\Delta\|_2 = \frac{\|\Delta\|_2}{\sqrt{\widehat{\mathrm{ESS}}_\lambda}}. \tag{22}$$

This deterministic local relation complements the variance interpretation below. ESS describes both the spread of influence across memory and a plug-in relative-variance statistic.

B.4 An exact ESS identity

Let $Y_i = e^{\lambda S_i} > 0$ and $\bar{Y} = m^{-1} \sum_i Y_i$. Define

$$\widehat{\mathrm{ESS}}_\lambda = \frac{(\sum_i Y_i)^2}{\sum_i Y_i^2}, \qquad \widehat{\mathrm{CV}}^2(Y) = \frac{m^{-1} \sum_i (Y_i - \bar{Y})^2}{\bar{Y}^2}. \tag{23}$$

For every positive vector Y, the following identity holds:

$$\frac{\widehat{\mathrm{CV}}^2(Y)}{m} = \frac{1}{\widehat{\mathrm{ESS}}_\lambda} - \frac{1}{m}. \tag{24}$$

To prove it, expand the centered second moment:

$$\widehat{\mathrm{CV}}^2(Y) = \frac{m^{-1} \sum_i Y_i^2}{\bar{Y}^2} - 1, \tag{25}$$

$$\frac{\widehat{\mathrm{CV}}^2(Y)}{m} = \frac{\sum_i Y_i^2}{m^2 \bar{Y}^2} - \frac{1}{m} = \frac{\sum_i Y_i^2}{(\sum_i Y_i)^2} - \frac{1}{m}. \tag{26}$$

The last expression is Eq. 24. This algebra uses the empirical variance with divisor m. It holds independently of whether the scores are iid.

For iid scores with finite exponential second moment, the delta method supplies a sampling interpretation. With $\mu_Y = \mathbb{E}[Y]$ and $g(y) = \lambda^{-1} \log y$,

$$\mathrm{Var}[g(\bar{Y})] \approx \frac{\mathrm{Var}(Y)}{m \lambda^2 \mu_Y^2}. \tag{27}$$

Substituting the empirical relative variance gives

$$\widehat{\mathrm{Var}}[\widehat{A}(\lambda)] \approx \frac{1}{\lambda^2} \left(\frac{1}{\widehat{\mathrm{ESS}}_\lambda} - \frac{1}{m} \right). \tag{28}$$

The empirical identity is exact. The delta-method variance approximates iid sampling variability. The controller evaluates ESS at $\lambda = 1$, where the question is support for full-tail LPC. Variance assessment at another temperature uses ESS recomputed at that temperature.

B.5 Concentration for bounded iid scores

Fix a score function, a temperature $\lambda > 0$, and an iid reference sample of size m. Assume $S \in [L, U]$ almost surely. Set $a = e^{\lambda L}$, $b = e^{\lambda U}$, and $\mu_Y = \mathbb{E}[e^{\lambda S}]$. For $\delta \in (0, 1)$, define

$$\eta_m = (b - a) \sqrt{\frac{\log(2/\delta)}{2m}}. \tag{29}$$

If $\eta_m < \mu_Y$, then with probability at least $1 - \delta$,

$$|\widehat{A}_m(\lambda) - A(\lambda)| \le -\frac{1}{\lambda} \log \left(1 - \frac{\eta_m}{\mu_Y} \right). \tag{30}$$

Hoeffding's inequality gives $|\bar{Y} - \mu_Y| \le \eta_m$ on an event of probability at least $1 - \delta$. Let $r = \eta_m / \mu_Y < 1$. On that event, $\bar{Y}/\mu_Y \in [1 - r, 1 + r]$. Since $-\log(1 - r) \ge \log(1 + r)$, taking logarithms bounds the absolute log ratio by $-\log(1 - r)$. Division by λ proves Eq. 30.

The exponential range $b - a$ makes score range relevant to partition uncertainty. Lower temperature contracts this range. The factor $1/\lambda$ also enters the bound, so its full expression matters. For an evolving replay learner, the score function and memory can depend on earlier training. The result above describes a fixed-score iid reference experiment; the empirical identities remain valid for the actual memory.

B.6 From partition error to a decision margin

A deterministic argument links partition estimation to class prediction. Fix $\beta \geq 0$ and a common temperature. Suppose a reference partition A_c and its estimate satisfy $|\widehat{A}_c - A_c| \leq e$ for every seen class. Each corrected score then differs from its reference value by at most βe. A pairwise margin differs by at most $2\beta e$.

If the reference winner has margin greater than $2\beta e$ over every competitor, the estimated correction preserves that winner. This statement follows by applying the error bound to the winner and competitor separately. It identifies the margin needed to absorb a known partition error. The premise is a partition-error bound at the selected temperature.

C Offsets, Scale, and the Controller

C.1 Exact class-offset behavior

Add a constant d_c to class channel c on all inputs. For any fixed $\lambda > 0$,

$$\widehat{A}'_c(\lambda) = \lambda^{-1} \log \left[\frac{1}{m} \sum_i e^{\lambda(S_{i,c} + d_c)} \right] \tag{31}$$

$$= d_c + \widehat{A}_c(\lambda). \tag{32}$$

The same identity holds at $\lambda = 0$ by the mean definition. Interquartile range is unchanged by a constant shift. Max-subtracted exponential weights are also unchanged. The controller therefore selects the same temperature after arbitrary class-wise constant offsets.

At prediction, the transformed corrected score is

$$\widetilde{s}'_c(x) = s_c(x) + d_c - \beta[\widehat{A}_c(\lambda) + d_c] = \widetilde{s}_c(x) + (1 - \beta)d_c. \tag{33}$$

Full correction cancels the offset. The primary $\beta = 0.9$ leaves one tenth of an added channel offset. The controller statistics remain unchanged under the same offset transformation.

C.2 What inverse scale compensates

For a positive constant a, the partition satisfies

$$\widehat{A}_{aS}(\lambda/a) = a\widehat{A}_S(\lambda). \tag{34}$$

The proof substitutes $(\lambda/a)(aS_i) = \lambda S_i$ in the exponential average. Thus reciprocal temperature scaling preserves the exponent and scales the resulting partition with the score.

The IQR part of r_c scales by a. If the reliability factor $\kappa = \min(1, \bar{\rho}/\rho_0)$ remains fixed, the unclipped controller approximates this reciprocal response. It is exact for the IQR scale without the additive ϵ. Clipping introduces the stated endpoints.

C.3 Why ESS also responds to score scaling

Unit-temperature ESS changes under general score rescaling. Its weights become $e^{aS_i - aS_{\max}}$, which usually concentrate more as a grows. A two-score example makes this explicit. For scores 0 and d,

$$\rho(d) = \frac{(1 + e^d)^2}{2(1 + e^{2d})}. \tag{35}$$

The value is one at $d = 0$ and tends to $1/2$ as $d \to \infty$. Replacing d with ad therefore changes the reliability measurement. The full controller therefore changes both the score units and the reliability response.

Robust scale measures the spread of central scores. ESS responds to the concentration created by their exponentials. Both depend on scores, but they emphasize different aspects of the replay distribution. Their combination is evaluated through the fixed-LPC, sensitivity, and oracle comparisons.

C.4 Strength and temperature act differently

Strength multiplies the whole partition, while temperature changes the partition's shape terms. Near zero temperature,

$$\beta A_c(\lambda) = \beta\mu_c + \frac{\beta\lambda}{2}v_c + \frac{\beta\lambda^2}{6}\kappa_{3,c} + O(\lambda^3). \tag{36}$$

Reducing β weakens mean correction and all higher terms together. Reducing λ retains the mean term while reducing the higher terms at different rates. The two operations can therefore move different decision boundaries, even at a similar total bias magnitude. This explains the separate strength and temperature analyses in the experiments.

D Calibration Algorithm and Numerical Evaluation

D.1 Online schedule

Table 3 gives the fixed-schedule form used for the online study. It assumes the underlying learner supplies local updates, memory insertion, and additive seen-class scores. The calibration procedure operates after these existing training steps.

Table 3. Online replay calibration. The refresh period is $K = 10$ minibatches.

Inputs: local learner, class-balanced memory, strength β, constants ρ_0 and $\lambda_{\min}$.
State: seen-class set, minibatch counter, cached bias vector initialized to zero.

1. Read the incoming minibatch and update the seen-class set.
2. Sample replay examples when memory is available.
3. Perform detached block-local updates on incoming and replay data.
4. Insert incoming examples through the learner's class-balanced memory rule.
5. On every Kth minibatch, score every stored image for all seen classes.
6. Compute class-wise IQR scales and unit-temperature normalized ESS.
7. Take class medians and apply Eq. 7.
8. Compute the partitions at the selected shared temperature. Cache their negatives multiplied by β.
9. For prediction, add the cached biases and maximize over seen classes.

The cache remains fixed between refreshes. Unseen-class biases are zero, and unseen classes are excluded from prediction. Before a channel receives a refreshed bias, its initialized value is used. The schedule depends on minibatch count, so an evaluation call does not itself define a calibration event.

D.2 Stable log-mean-exp

For one class, let $z_i = \lambda S_i$ and $z_{\max} = \max_i z_i$. Evaluate

$$\widehat{A}(\lambda) = \frac{z_{\max} + \log \sum_i e^{z_i - z_{\max}} - \log m}{\lambda}. \tag{37}$$

Every exponential lies in $(0, 1]$, and at least one equals one. This prevents overflow from large positive scores. Subtracting $\log m$ preserves the empirical mean rather than the unnormalized sum. All class channels use the same m in a given refresh.

Unit-temperature ESS uses its own max subtraction, before multiplying scores by the selected temperature. This preserves the controller's definition. The support diagnostic uses unit-temperature weights; partition evaluation uses the selected temperature.

The analytic mean limit explains small-temperature behavior. The evaluated controller uses the positive lower bound 0.005. A separate exact mean-centering control can use the explicit $\lambda = 0$ definition.

D.3 Persistent and temporary resources

Let $C = |\mathcal{C}_t|$. The cached vector requires C scalar values. The replay memory remains the memory already used for rehearsal. A full score matrix contains mC values during calibration. Reductions for partitions and ESS are linear in this matrix size. Exact quantile computation depends on whether an implementation sorts values or uses selection.

If the model scores memory in batches, model activations can be released between batches. Exact IQR computation still needs the relevant score values or an equivalent exact selection strategy. Model activations and calibration statistics occupy separate storage. Cached inference adds the stored bias vector directly.

At the main per-class budget, full memories contain 160 CIFAR-10 images, 1,600 CIFAR-100 images, and 3,200 Tiny-ImageNet images. Their score matrices contain 1,600, 160,000, and 640,000 values, respectively. These dimensions follow from 16 images per class and the final class count.

E Experimental Settings and Parameter Selection

E.1 Stream design

Table 4 summarizes the stream settings. A phase groups arrivals in the class-incremental evaluation protocol. The classifier still selects among all classes observed so far. The correct phase is supplied only in the explicitly labeled task-oracle diagnostic.

Table 4. Stream settings and the main replay budget. Full-memory sizes are derived from the per-class budget.

Dataset	Classes	Phases	Classes per phase	Full memory
CIFAR-10	10	5	2	160
CIFAR-100	100	10	10	1,600
Tiny-ImageNet-200	200	20	10	3,200

Strict one-pass training refers to incoming stream examples. Replay explicitly permits reuse of stored examples. The blurry CIFAR-10 setting introduces overlap width 0.5 between arrivals. Its training and calibration procedures receive no task IDs. Reported checkpoint averages use the evaluation schedule associated with that stream.

The primary model has nine detached convolutional blocks. Their channel counts are $20, 20, 20, 40, 40, 40, 80, 80, 80$. The linear-head study attaches an independent classifier to each detached local block. DeeperForward TinyCNN retains the released four-stage topology and uses the adapted one-pass training protocol.

E.2 Development and confirmation

The primary correction strength was chosen once on a CIFAR-10 development seed. The candidate set was $\{0.5, 0.75, 0.9, 1\}$, and the selected value was 0.9. The normalized-ESS reference 0.08 and clipping interval $[0.005, 1]$ were fixed before confirmation. The main cross-dataset comparisons use that frozen rule.

DeeperForward TinyCNN uses a separate architecture-specific choice. The initial $\beta = 0.9$ transfer did not retain anytime performance. A seed-zero development selection chose $\beta = 0.5$. That value was then used for the five-seed CIFAR-10 confirmation and frozen for CIFAR-100.

The test-label temperature grid is a later diagnostic. It compares the automatic final-checkpoint choice with a discrete set chosen using test accuracy. Its output is reported as oracle regret. The ρ_0 sensitivity study is also separate from the frozen confirmation rule. The deployed rule remains fixed during these diagnostics.

Table 5. Calibration settings and their roles.

Quantity	Value	Role
Primary β	0.9	Chosen on one CIFAR-10 development seed.
TinyCNN β	0.5	Separate seed-zero architecture selection.
ρ_0	0.08	Frozen normalized-ESS reference.
λ bounds	$[0.005, 1]$	Positive lower bound and full-tail upper endpoint.
Refresh interval	10 minibatches	Fixed-schedule online calibration.
Memory	16 images per class	Main rehearsal and calibration reference.
Temperature grid	$0.01, 0.025, 0.05,$ $0.1, 0.2, 0.35,$ $0.5, 0.75, 1$	Test-label final-checkpoint diagnostic.
Sensitivity range	$\rho_0 \in [0.04, 0.16]$	Post-confirmation sensitivity analysis.

F Metrics and Paired Statistical Analysis

F.1 Phase-based metrics

Let $a_{t,k}$ be percentage accuracy on task k after phase t, with $k \leq t$. For T phases, final average accuracy is

$$\text{Final} = \frac{1}{T}\sum_{k=1}^{T} a_{T,k}. \tag{38}$$

For evaluation at each phase, anytime accuracy is

$$\text{Anytime} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{t}\sum_{k=1}^{t} a_{t,k}\right). \tag{39}$$

Forgetting averages the drop from each non-final task's past peak:

$$\text{Forgetting} = \frac{1}{T-1}\sum_{k=1}^{T-1}\left[\max_{u\in\{k,\ldots,T-1\}} a_{u,k} - a_{T,k}\right]. \tag{40}$$

Mean current-task accuracy is

$$\text{Current} = \frac{1}{T}\sum_{t=1}^{T} a_{t,t}. \tag{41}$$

The four metrics summarize different portions of the same accuracy trajectory. Final accuracy measures the endpoint. Anytime accuracy gives weight to intermediate operation. Forgetting compares earlier peaks with the endpoint, while current-task accuracy measures performance when classes arrive.

F.2 Checkpoint averages in overlapping streams

For the blurry stream, let h_j denote seen-class accuracy at checkpoint j. Mean checkpoint accuracy is $J^{-1}\sum_{j=1}^{J} h_j$. This notation avoids assigning an artificial current task to overlapping arrivals. Final seen-class accuracy is the last checkpoint's value.

Checkpoint averages depend on the evaluation schedule. The paired blurry comparison uses the same schedule for the two methods. This makes the reported gain a comparison over matched observation times. It is distinct from an average over individual incoming examples.

F.3 Paired bootstrap intervals

For matched seeds $s = 1, \ldots, n$, define d_s as the metric for one method minus its paired reference. A bootstrap replicate samples n seed indices with replacement and averages their differences. The percentile interval is taken from the resulting empirical distribution. Primary intervals use 20,000 resamples, and the blurry analysis uses 100,000.

Pairing preserves common seed effects in each difference. The interval is therefore based on seed-level differences, not independently resampled method means. Main tables report standard deviations when those summaries are available. A standard deviation across runs and a bootstrap interval for a paired gain describe different quantities.

Published means and gains can differ slightly after rounding. For example, the higher-precision CIFAR-10 values are 15.636% and 26.904%. Their difference rounds to 11.27 points, while subtracting the displayed two-decimal means gives 11.26. Paired gains retain the precision and rounding of the paired analysis.

F.4 Checkpoint dependence in the mechanism analysis

The mechanism study contains 20 post-initial checkpoints grouped within five seeds. Correlation is computed between controller activation and the advantage over fixed LPC. The bootstrap resamples seed clusters to preserve dependence among checkpoints from the same run. Leave-one-seed-out analyses check whether the sign persists after removing each run in turn.

These observations share training histories within a seed. Treating all checkpoints as unrelated replicates would discard that structure. The reported interval therefore uses the seed-cluster procedure. The resulting association describes alignment between the controller statistic and observed calibration benefit.

G Paired Results and Classifier Controls

G.1 Primary paired effects

Table 6 collects the paired intervals. All entries are percentage-point differences in the stated metric. A positive forgetting difference means more forgetting; a positive accuracy difference means higher accuracy.

Table 6. Paired differences from the five-seed analyses.

Comparison	Metric	Difference	95% interval
CIFAR-10 AT minus ER	Final	11.27	[9.94, 12.47]
	Anytime	5.04	[3.86, 6.23]
CIFAR-10 AT minus fixed LPC	Final	1.29	[0.64, 1.94]
	Anytime	0.79	[0.60, 1.01]
	Current	3.63	[1.96, 5.15]
	Forgetting	2.92	[0.49, 5.25]
CIFAR-100 AT minus ER	Final	3.91	[3.45, 4.39]
	Anytime	4.94	[4.84, 5.06]
CIFAR-100 AT minus fixed LPC	Final	0.112	[0.048, 0.168]
	Anytime	0.018	[0.005, 0.032]
Online minus boundary AT, CIFAR-10	Final	−0.052	[−0.240, 0.094]
	Anytime	−0.048	[−0.160, 0.020]
Blurry AT minus ER, CIFAR-10	Final	10.94	[9.54, 12.59]
	Checkpoint mean	4.39	[3.53, 5.70]

The CIFAR-10 comparison separates partition estimation from temperature selection. Fixed LPC already supplies most of the gain over raw local ER. Automatic tempering then recovers current-class

accuracy while retaining a lower forgetting level than ER. On CIFAR-100, fixed and automatic estimates produce similar aggregate accuracy.

G.2 Scale and affine controls

The Weight Alignment-style control uses group-norm alignment within the local setting. It reaches 15.95% final and 39.53% anytime CIFAR-10 accuracy. The paired analysis does not show a clear anytime-accuracy difference from local ER. A memory-matched BiC-style affine control reaches 16.43% final accuracy. These local adaptations test classifier-scale and recency-affine explanations using the recorded memory setting.

These controls adapt classifier-scale and affine corrections to the local replay learner. Their results complement the optimized-bias diagnostic, which tests how much a frozen model can gain from class-offset changes.

Table 7. Additional CIFAR-10 score controls. N/a indicates metrics absent from the recorded summary.

Control	Final	Anytime
Group-norm Weight Alignment-style	15.95	39.53
Memory-matched BiC-style	16.43	*n/a*
Fixed LPC, $\beta = 1$	23.83	*n/a*
Fixed LPC, $\beta = 0.9$	25.61	44.03
AT-LPC, $\beta = 0.9$	26.90	44.81

Full-strength fixed LPC reaches 23.83% final accuracy, compared with 25.61% at $\beta = 0.9$. Automatic tempering at the latter strength reaches 26.90%. This comparison motivates separating correction strength from tail use. The two parameters act on different terms of the calibrated decision margin.

G.3 Reading the diagnostic accuracies

The frozen-model diagnostic uses a CIFAR-10 learner with 17.78% all-class accuracy. The primary five-seed ER experiment has mean accuracy 15.64%. Each comparison uses its own matched model state.

The task oracle supplies the correct two-class candidate set and reaches 62.66%. The frozen-backbone bias diagnostic reaches 24.19% while retaining all-class prediction. The former measures the value of a restricted comparison. The latter measures recoverable score-offset effects in the diagnostic model. Together, they separate candidate-set sensitivity from recoverable offset effects.

H Recent-Prior Adjustment and Partition Correction

H.1 Where the two operations enter

Let $z_c(x)$ be a local class score used during LAS training. With recent prior estimate $\widehat{\pi}_t(c)$, the adjusted training score has the form

$$z_c(x) + \tau \log \widehat{\pi}_t(c). \tag{42}$$

This prior term changes the loss and its gradients. The class partition instead averages exponentiated current scores over replay inputs after an update. It depends on the score distribution even when recent class counts remain the same.

Suppose LAS-trained scores retain additive channel drift d_c. The partition still shifts by d_c, as shown in Appendix C. Full partition correction removes that residual offset. Partial correction leaves $(1 - \beta)d_c$. This algebra is compatible with any prior estimator used earlier in training. The composition experiment measures whether useful residual correction remains in practice.

H.2 MATCHED LOCAL COMPOSITION

On CIFAR-10, LAS plus AT-LPC gains 4.31 final points over LAS. Anytime means are 45.37% and 45.92%; the paired analysis does not show a clear difference in anytime accuracy. Forgetting falls by 21.92 points. These measurements place the main composition benefit in end-of-stream retention.

Against LAS plus fixed LPC, automatic tempering gains 0.97 final and 0.72 anytime points. Current-task accuracy gains 5.35 points, while forgetting rises by 5.41 points. The correction thus moves away from fixed LPC's stronger stabilization within the LAS-trained model.

On CIFAR-100, adding AT-LPC to LAS gains 3.59 final and 4.14 anytime points. Forgetting falls by 12.30 points, and current-task accuracy falls by 7.48 points in the paired summary. These results extend the same training-time and post-update composition across datasets.

Table 8 gives the paired final and anytime effects.

Table 8. Paired statistical analysis for CIFAR-10 LAS composition. Differences are percentage points over five seeds.

Comparison	Metric	Difference	95% interval
LAS + AT minus LAS	Final	4.31	$[3.57, 5.09]$
LAS + AT minus LAS + fixed LPC	Final	0.97	$[0.25, 2.03]$
LAS + AT minus LAS + fixed LPC	Anytime	0.72	$[0.36, 1.04]$

H.3 END-TO-END REFERENCE PROTOCOLS

The BP ResNet-18 comparisons supply two recorded ER references. ER-ACE is paired with an ER value of 21.48% and reaches 28.36%. ER-LAS is paired with joint ER at 14.45% and reaches 26.75%. Each ER reference is matched to its associated training method.

The local LAS composition is the direct test of residual partition correction. It keeps the nine-block architecture and places the calibration on the same local learning setting. The end-to-end controls provide broader continual-learning context.

I TRANSFER TO OTHER LOCAL MODELS

I.1 INDEPENDENT LOCAL LINEAR HEADS

The linear-head model attaches an independent classifier to each detached local block. Its CIFAR-10 final accuracy rises from 20.64% to 24.56% with AT-LPC. The paired gain is 3.92 points. Anytime accuracy rises from 42.60% to 44.18%. Forgetting changes from 54.92% to 30.26%, while current-task accuracy changes from 64.58% to 48.77%.

These scores come from a distinct local classifier parameterization. Calibration still receives additive class evidence from detached blocks. The observed change therefore concerns the shared replay reference across two forms of local scores. The current-task and forgetting values show the same retention-oriented shift as the primary model.

Table 9 summarizes the matched transfer gains.

I.2 DEEPERFORWARD TINYCNN

The TinyCNN adaptation uses the released four-stage DeeperForward topology with a one-pass stream. Its $\beta = 0.5$ was selected on seed zero before the confirmation described in Appendix E. On CIFAR-10, final accuracy rises from 28.69% to 31.71%. The gain is 3.02 points. Forgetting falls by 11.36 points, current-task accuracy falls by 6.06 points, and no clear paired difference is observed in anytime accuracy.

The same strength transfers to CIFAR-100. Final accuracy rises from 5.55% to 7.87%, a gain of 2.32 points. Anytime accuracy gains 2.53 points. Forgetting falls by 6.10 points, while current-task accuracy falls by 3.16 points. The paired intervals exclude zero for these four changes.

Table 9. Final-accuracy transfer comparisons over five matched seeds. TinyCNN uses DeeperForward.

Model	Dataset	Baseline	AT-LPC	Gain	95% interval
Local linear heads	CIFAR-10	20.64	24.56	3.92	$[2.68, 4.93]$
TinyCNN	CIFAR-10	28.69	31.71	3.02	$[2.10, 3.94]$
TinyCNN	CIFAR-100	5.55	7.87	2.32	$[2.18, 2.48]$

The TinyCNN study changes both architecture and its strength selection. The CIFAR-100 study then holds that chosen strength fixed. The resulting cross-dataset comparison uses the same architecture-specific strength.

J ResNet17 Shared-Checkpoint Study

J.1 Offline and continual protocols

An additional DeeperForward ResNet17 campaign evaluates a deeper local backbone. Offline seeds 101, 202, and 303 complete 150 epochs. Their CIFAR-10 test accuracies are 85.77%, 86.04%, and 86.12%, with mean 85.9767%. These values describe offline training. The continual experiment uses a separate one-pass protocol and five seeds, 1001 through 1005.

For each continual seed, one raw backbone is trained through five two-class phases and saved at phase checkpoints. Each checkpoint includes a memory snapshot with 16 images per class. All calibration variants share that model and memory state. The supplementary figures report checkpoint-refreshed diagnostics: calibration is recomputed from the saved checkpoints and replay snapshots, with $\beta = 0.9$ except in the strength sweep. The original online bias cache was not saved.

The recorded ResNet17 implementation computes controller statistics over all ten output channels, while prediction is restricted to classes seen at the checkpoint. Before the final phase, this channel set differs from the seen-class statistics in Eqs. 5 and 6. All ten classes are seen at the final checkpoint. Unless stated otherwise, markers and lines show means over seeds 1001–1005, and error bars show one sample standard deviation across seeds.

The recorded engineering settings include local CIFAR-10 binary loading and legacy-checkpoint compatibility. The campaign uses a constant-learning-rate scheduler setting. Engineering seed 1000 is excluded from the five-seed aggregate. The added records leave the objective and data order unchanged.

J.2 Accuracy and retention

Table 10 compares the calibration variants on shared checkpoints. Table 11 reports paired final-accuracy effects. At $\beta = 0.9$, AT-LPC gains 5.590 final points over raw scores and 2.618 points over fixed LPC. Mean current-task accuracy rises from 31.684% under fixed LPC to 62.020% under tempering. Forgetting rises from 23.030 to 54.940 points. The higher final accuracy accompanies a recovery in current-class performance at this strength.

Table 10. ResNet17 continual CIFAR-10 results, averaged over five shared-checkpoint seeds. All values are percentages.

Variant	Final	Anytime	Forgetting	Current
Raw	12.478	33.290	73.680	71.422
Fixed LPC, $\beta = 0.5$	14.810	32.755	71.308	71.856
AT-LPC, $\beta = 0.5$	14.386	34.055	74.765	74.198
Fixed LPC, $\beta = 0.9$	15.450	31.193	23.030	31.684
AT-LPC, $\beta = 0.9$	18.068	35.067	54.940	62.020

At $\beta = 0.5$, AT-LPC gains 1.908 final points over raw scores. Its final accuracy is 0.424 points below fixed LPC. Anytime accuracy is 34.055% with tempering and 32.755% with fixed LPC. The two strengths produce different balances of final and anytime accuracy.

Table 11. ResNet17 paired final-accuracy differences.

Comparison	Difference	95% interval
AT-LPC $\beta = 0.5$ minus raw	1.908	$[1.712, 2.160]$
AT-LPC $\beta = 0.5$ minus fixed LPC	-0.424	$[-0.810, -0.052]$
AT-LPC $\beta = 0.9$ minus raw	5.590	$[4.548, 6.828]$
AT-LPC $\beta = 0.9$ minus fixed LPC	2.618	$[1.068, 3.582]$

Figure A1 follows performance across the five phases. AT-LPC finishes above raw scores and fixed LPC in all-seen-task accuracy, while fixed LPC retains higher earlier-task accuracy at this strength.

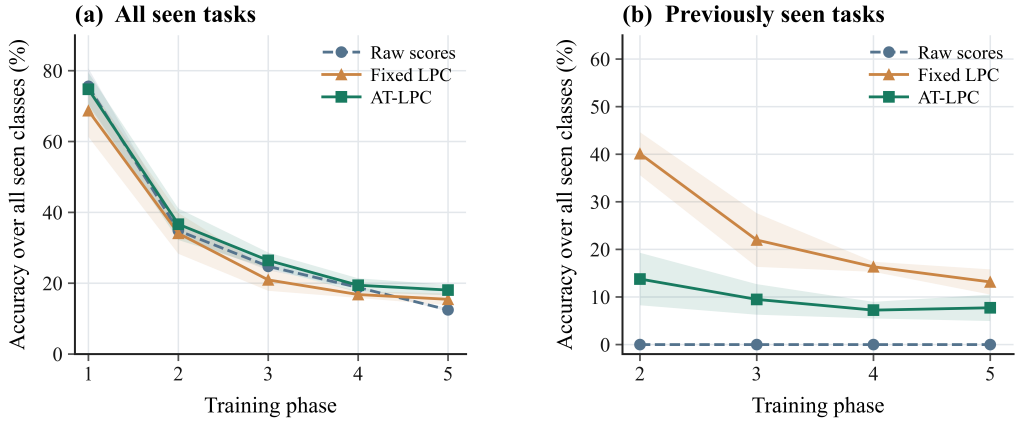

Figure A1. ResNet17 learning trajectories on CIFAR-10. (a) Accuracy averaged over all seen tasks. (b) Accuracy on previous tasks, excluding the current task. Both use all seen classes as prediction candidates. Calibration variants share each checkpoint and replay state, with $\beta = 0.9$. Lines show means over seeds 1001–1005; shading shows one standard deviation.

J.3 Task retention and prediction repair

Figure A2 resolves this trade-off by task and by prediction transition. Relative to fixed LPC, AT-LPC increases final current-task accuracy by 34.830 points while reducing the mean accuracy of the four earlier tasks by 5.435 points. Its final all-seen accuracy is 18.068%, compared with 15.450% for fixed LPC and 12.478% for raw scores. Replacing raw scores with AT-LPC repairs 9.158% of test predictions and regresses 3.568%, yielding the 5.590-point net gain. Relative to fixed LPC, the corresponding fractions are 11.074% and 8.456%, yielding 2.618 points. The gain therefore includes both recovered predictions and losses on previously correct examples.

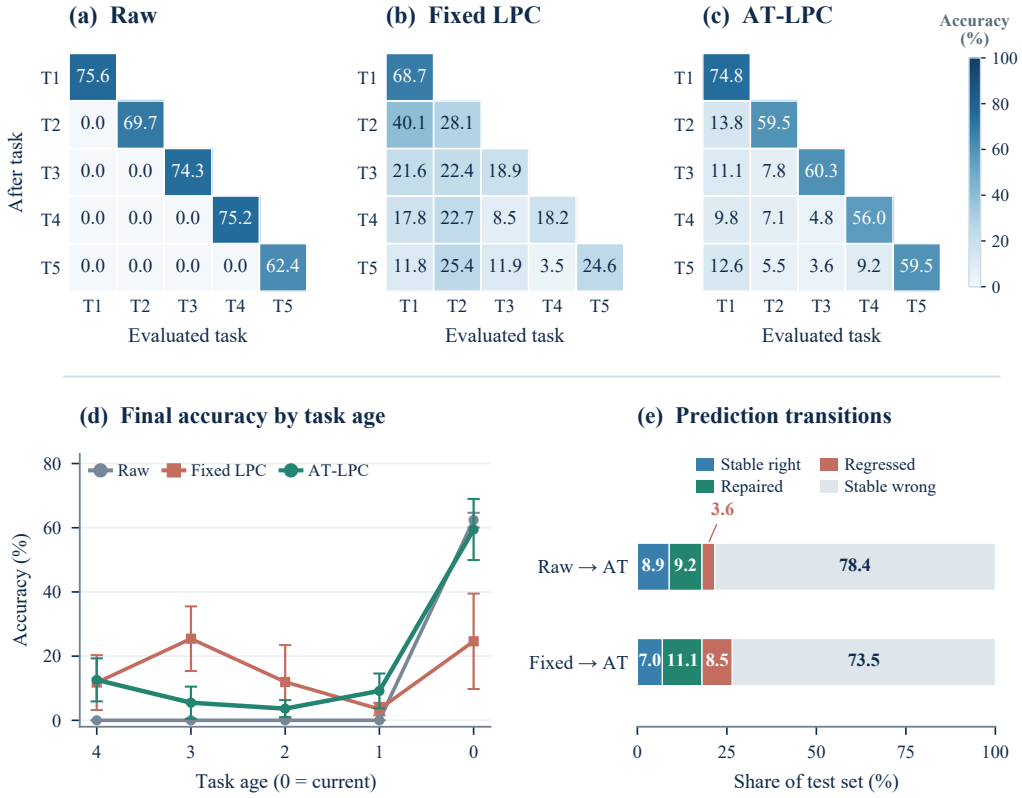

Figure A2. Task retention and prediction repair. (a–c) Mean task accuracy after each phase for raw scores, fixed LPC, and AT-LPC. Unseen tasks are masked; all panels share a 0–100% scale. (d) Final accuracy by task age, with age zero denoting the current task. (e) Fractions of the 10,000 final-phase test examples per seed that remain correct, are repaired, regress, or remain incorrect. Bars compare AT-LPC with each stated baseline. Results use $\beta = 0.9$ and five seeds; error bars in (d) show one sample standard deviation across seeds.

J.4 Decision-Level Score Diagnostics

Figure A3 repeats the decision-level diagnosis on the ResNet17 continual checkpoints. The same separation between within-task discrimination and all-class prediction remains visible in the deeper backbone. With the backbone fixed, AT-LPC partially repairs global decisions and redistributes the strong cross-task prediction bias.

J.5 Controller Components and Recorded Cost

The scale-only ablation uses $\lambda = \mathrm{clip}(1/\bar{r}, \lambda_{\min}, 1)$, and the ESS-only ablation uses $\lambda = \mathrm{clip}(\min\{1, \bar{\rho}/\rho_0\}, \lambda_{\min}, 1)$. Both retain the same partition estimator and correction strength.

Figure A4 shows increasing robust scale and decreasing normalized unit-temperature ESS as the stream progresses. The final median normalized ESS is about 0.00625. All five seeds reach $\lambda_{\min} = 0.005$ in phases 2–4, so the later checkpoints operate at the temperature floor. At the final checkpoint, scale-only calibration and AT-LPC reach similar all-seen accuracies, 17.954% and 18.068%, but their current-task accuracies differ: 43.890% and 59.450%. Mean centering gives 16.934% all-seen accuracy and 72.020% current-task accuracy. The component comparison thus exposes different allocations of accuracy between current and earlier tasks within this strongly tempered regime.

Calibration statistic computation averages 0.097 seconds per stored evaluation record. Recorded evaluation time averages 18.47 seconds, and peak GPU memory is identical across calibration variants.

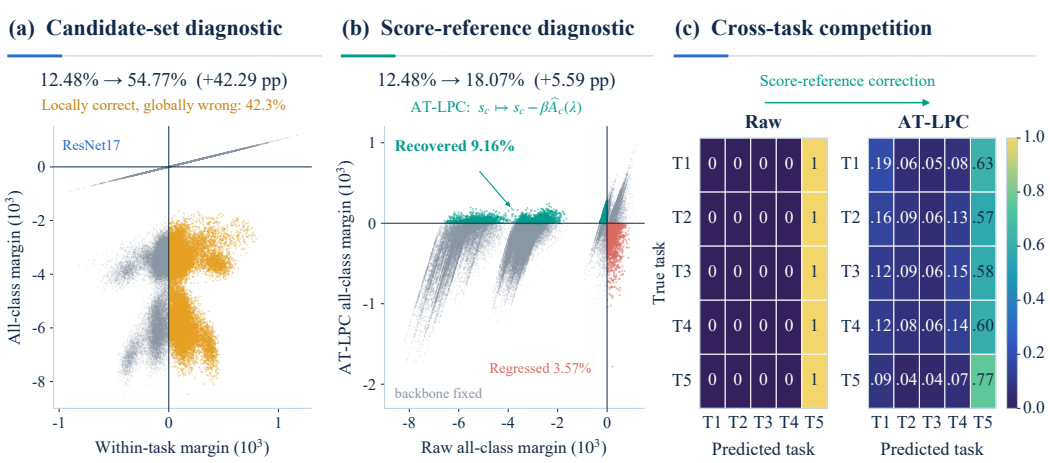

Figure A3. Decision-level diagnosis on ResNet17. (a) Candidate-set restriction exposes a large gap between within-task discrimination and all-class prediction. (b) With the backbone fixed, AT-LPC shifts global margins, recovering 9.16% and regressing 3.57% of predictions. (c) Task-level predictions show reduced cross-task concentration after calibration.

The statistic time refers to the recorded calculation stage. A full online refresh also includes memory scoring.

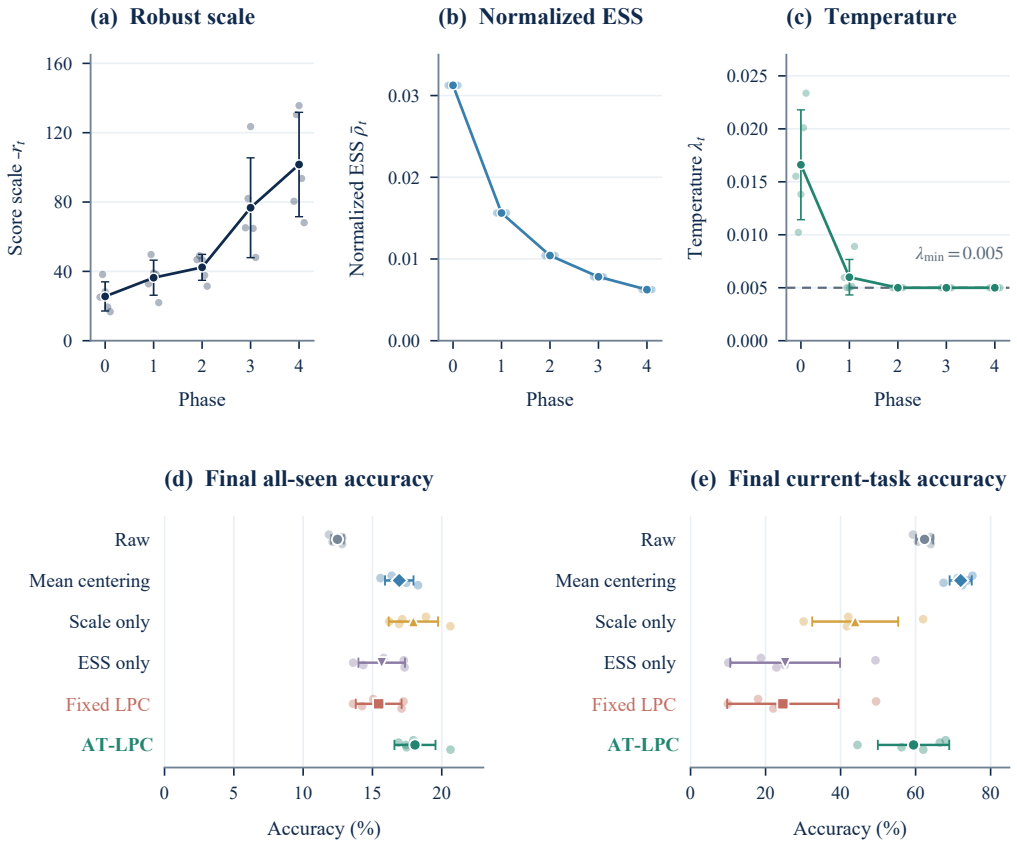

Figure A4. Controller statistics and component roles. (a–c) Robust scale, normalized unit-temperature ESS, and selected temperature across the five ResNet17 phase checkpoints. The dashed line marks $\lambda_{\min} = 0.005$; the numbers of seeds at this bound are $0, 2, 5, 5, 5$. (d,e) Final all-seen and current-task accuracy for raw scores, mean centering, scale-only, ESS-only, fixed LPC, and AT-LPC. Calibration variants use $\beta = 0.9$ and share each seed's model and replay snapshot. Large markers and error bars show five-seed means and sample standard deviations; faint markers show individual seeds.

J.6 Calibration-reference sensitivity

Figure A5 varies the reference used for calibration while holding the final trained model and its original replay budget fixed. We subsample $q \in \{1, 2, 4, 8, 16\}$ reference images per class and separately delete 1, 2, or 4 images per class at random. Each nontrivial perturbation has 50 repetitions per seed; the full-reference control has one evaluation. Repetitions are averaged within each seed before computing cross-seed means and sample standard deviations. Accuracy changes are paired with the same seed and method under the full reference.

The mean gain of AT-LPC over fixed LPC grows from 0.432 points at $q = 1$ to 2.618 points at $q = 16$. Under deletion, mean centering and AT-LPC incur absolute mean accuracy changes below 0.005 and 0.043 points, respectively, while fixed LPC improves by up to 0.282 points. These results describe sensitivity to the calibration reference at a fixed model state. Changing training replay capacity additionally changes representation learning, as studied in Figure A10.

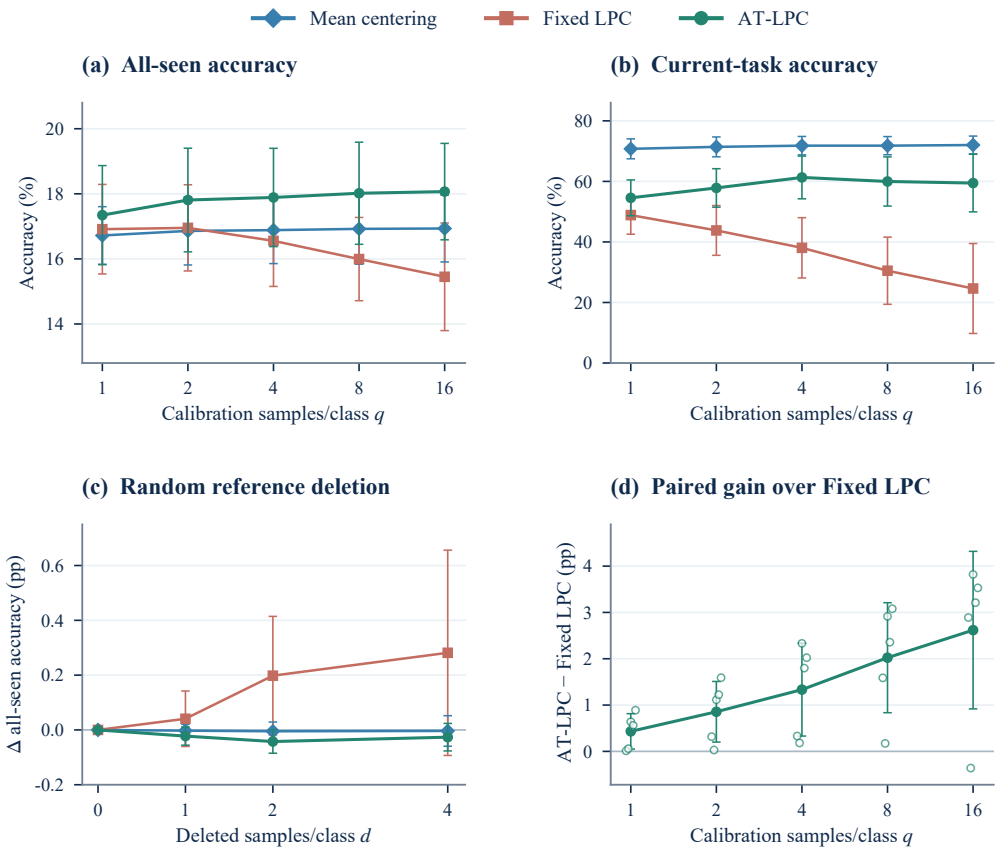

Figure A5. Calibration-reference sensitivity at final checkpoints. Trained models and the original replay budget of 16 images per class remain fixed. (a,b) All-seen and current-task accuracy under class-balanced reference subsampling. (c) Accuracy change after random deletion, relative to the same seed and method without deletion. (d) Paired gains of AT-LPC over fixed LPC; open circles show individual seed means. Each perturbation uses 50 repetitions, averaged within seed before aggregation. Full-reference controls use one evaluation. Results use five seeds and $\beta = 0.9$; error bars show sample standard deviations across seeds.

J.7 Temperature and correction strength

Figure A6 separates the effects of temperature and correction strength. The eleven-point temperature grid includes the mean-centering limit $\lambda = 0$ and the lower bound 0.005, with $\beta = 0.9$. For each seed and phase, the post hoc grid gap subtracts automatic-rule accuracy from the highest grid accuracy. Test labels define this diagnostic comparison and do not enter the deployed temperature rule. The mean gap is 4.460 points in phase 0 and 0.382 points at the final checkpoint. The final-phase strength sweep uses $\beta \in \{0, 0.25, 0.5, 0.75, 0.9, 1\}$. At $\beta = 1$, mean centering reaches 18.220% all-seen accuracy, compared with 16.434% for AT-LPC.

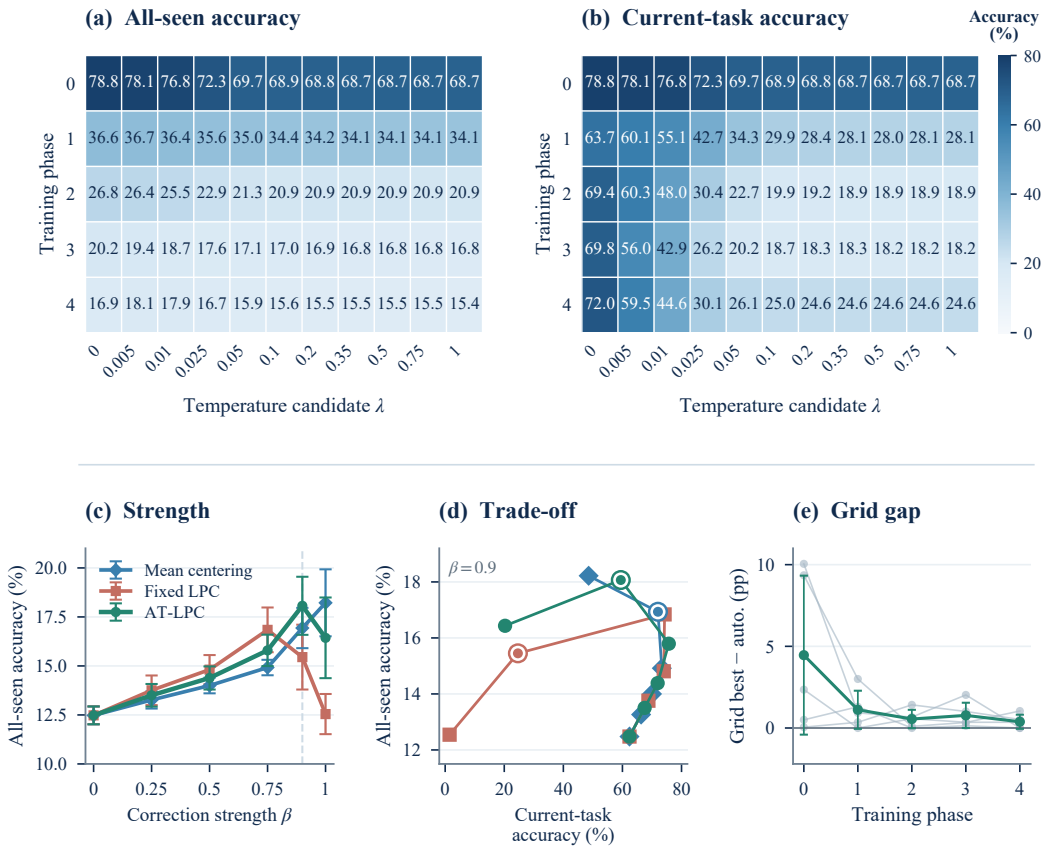

Figure A6. Temperature and correction-strength diagnostics. (a,b) Five-seed mean all-seen and current-task accuracy over an eleven-point temperature grid at $\beta = 0.9$, using a shared 0–80% scale. (c) Final all-seen accuracy across correction strengths. (d) Current-task and all-seen accuracy along the same sweep; outlined points mark $\beta = 0.9$. (e) Post hoc grid-best accuracy minus automatic-rule accuracy, computed within each seed and phase before aggregation. Thin lines show individual seeds. Error bars in (c,e) show one sample standard deviation across five seeds. Test labels are used for these post hoc temperature diagnostics.

K Online Calibration with Overlapping Arrivals

The blurry stream evaluates calibration when neighboring classes overlap in time. A fixed ten-minibatch schedule refreshes replay statistics throughout training. Table 6 reports a final gain of 10.94 points and a mean-checkpoint gain of 4.39 points over ER.

Figure A7 connects the selected temperature to the training trajectory. The temperature stays at one in the early checkpoints and then declines as the stream progresses. Seen-class accuracy ends at 27.47% with AT-LPC and 16.53% with ER. The shared horizontal axis links these changes to the same observation times.

L Temperature Diagnostics and Mechanism Evidence

L.1 Discrete-grid regret

The diagnostic temperature grid is

$$\Lambda = \{0.01, 0.025, 0.05, 0.1, 0.2, 0.35, 0.5, 0.75, 1\}. \tag{43}$$

For a final checkpoint, the grid oracle selects the temperature with highest test accuracy. The recorded regret compares that grid value with the automatic rule at the same checkpoint. The grid is discrete, so its best value is an analysis reference within Λ.

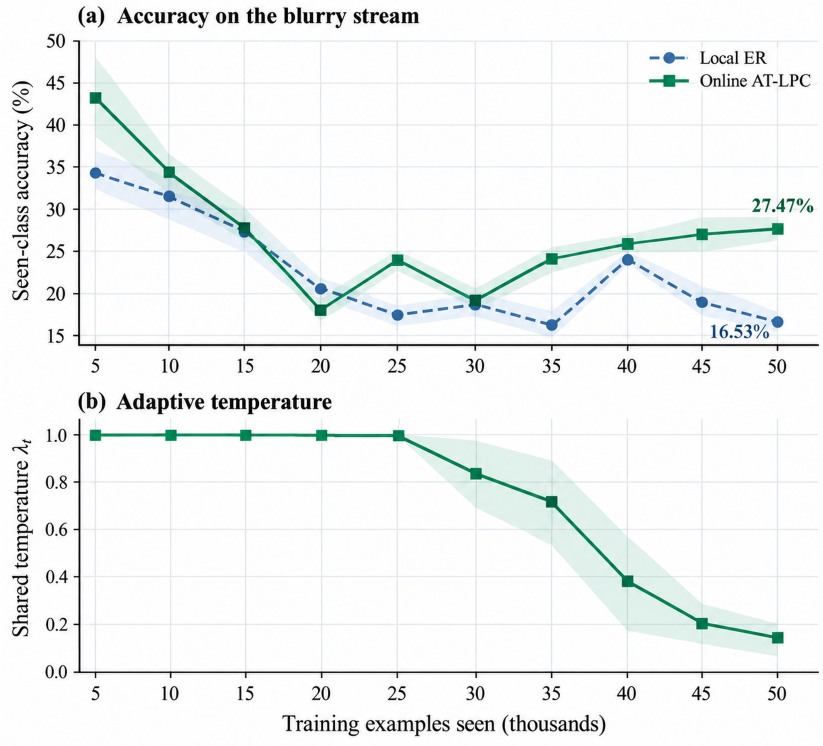

Figure A7. Online calibration on blurry CIFAR-10. (a) Seen-class accuracy in the one-pass stream with overlap width 0.5. (b) Shared temperature at the same checkpoints. Lines show five-seed means; shading shows one standard deviation. Calibration refreshes every ten minibatches. Training and calibration receive no task identities.

Across five seeds, mean regret is 0.212 points on CIFAR-10 and 0.022 on CIFAR-100. Maximum reported regret is 0.46 and 0.10 points, respectively. These final-checkpoint measurements quantify the gap between test-label grid selection and the replay-statistic rule.

L.2 Reference-ESS sensitivity

The sensitivity study varies ρ_0 over $[0.04, 0.16]$. CIFAR-10 final accuracy spans 0.882 points. The frozen value 0.08 is 0.128 points below the best value in that analysis. CIFAR-100 remains unchanged because $\min(1, \bar{\rho}/\rho_0)$ saturates at one over the tested range.

Saturation has a direct computational meaning. Once the reliability factor reaches one, changing its reference within that regime does not change the candidate temperature. The robust scale and clipping then determine the value. This saturation explains the unchanged CIFAR-100 result.

L.3 Activation and relative benefit

Table 12 reports the checkpoint associations. The checkpoint study measures tempering strength as $1-\lambda$. Its Spearman correlation with the accuracy advantage over fixed LPC is 0.769. Tail unreliability alone gives 0.672. All five leave-one-seed-out correlations for controller activation are positive.

These diagnostics connect the statistic to the comparison it is meant to improve. Fixed LPC trusts the unit-temperature exponential average. The controller reduces that trust when its replay support is concentrated. Positive checkpoint-level alignment shows that stronger activation accompanies larger relative benefit in the recorded LAS trajectories.

Figure A8 shows that activation and gain vary across checkpoints within each seed. The positive rank associations link stronger tempering to larger gains over full-tail LPC. The points retain their seed identities, matching the clustered statistical analysis.

Table 12. Statistical analysis of the checkpoint association. Intervals use the seed-cluster bootstrap over 20 post-initial checkpoints.

Predictor of the advantage over fixed LPC	Spearman correlation	95% interval
Controller activation $1 - \lambda$	0.769	[0.456, 0.909]
Tail unreliability	0.672	[0.341, 0.907]

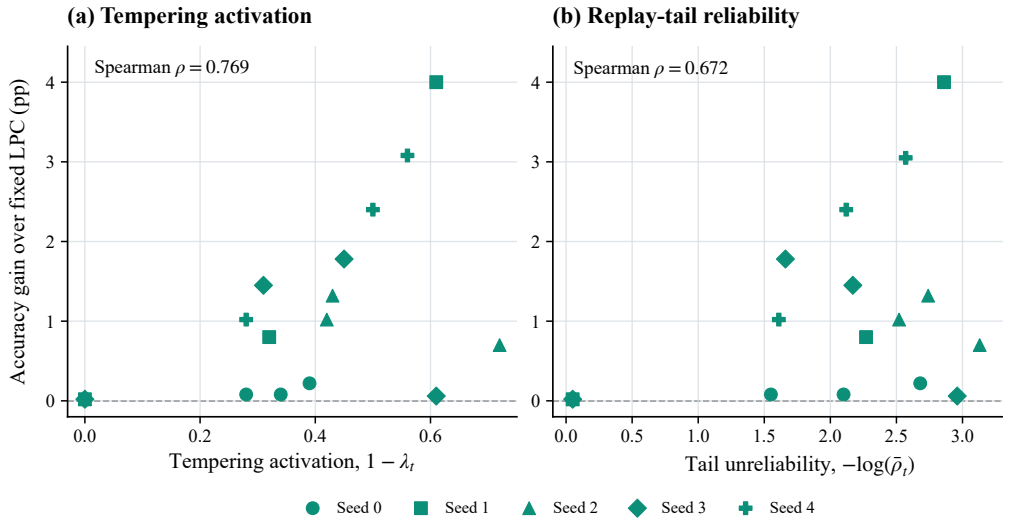

Figure A8. Replay reliability and the benefit of tempering. Each point is one of 20 matched post-initial CIFAR-10 LAS checkpoints. Shapes identify five seeds. The vertical axis is the accuracy gain over fixed LPC. (a) Tempering activation. (b) Tail unreliability. Dashed lines indicate zero gain; Table 12 gives the seed-cluster bootstrap intervals.

L.4 Strength Curve

The five-seed LAS strength analysis varies correction strength from zero to one. Final accuracy rises from 23.08% to 27.61%, while forgetting falls from 53.20% to 27.80%. Current-task accuracy falls from 65.64% to 49.14%. The paired analysis reports monotonic changes for these three quantities, with anytime accuracy peaking at $\beta = 0.5$.

The primary operating point uses $\beta = 0.9$, selected before confirmation. The strength curve explains what this choice emphasizes. It also complements the fixed-strength temperature comparisons: the strength study changes the amount of correction, while the temperature study changes its reliance on tails. Together, they describe how replay calibration affects continual prediction.

Figure A9(a) separates the best anytime operating point from the strongest final-accuracy setting. Panel (b) tracks the accompanying decrease in forgetting and current-task accuracy. The fixed-LPC point shows how temperature changes the operating point at a fixed strength.

M Additional Score Regimes and Memory Analysis

M.1 SFFA score regime

The SFFA adaptation uses a bounded ratio goodness construction. In the strict one-pass seed-zero experiment, replay accuracy is 9.95% and calibrated accuracy is 10.53%. Median normalized ESS is 0.99984. The selected temperature saturates at one, so the controller retains the full-tail endpoint in this setting.

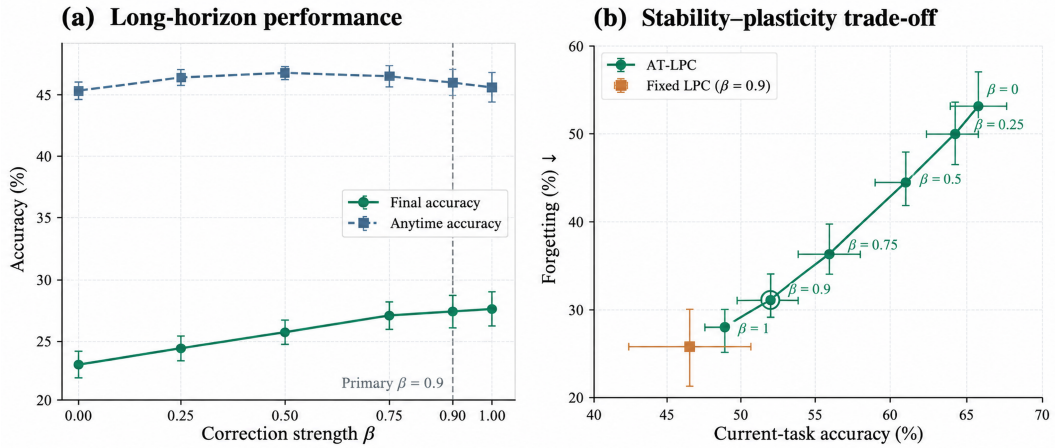

Figure A9. Correction strength and the stability–plasticity balance. Five-seed CIFAR-10 LAS results for $\beta \in \{0, 0.25, 0.5, 0.75, 0.9, 1\}$. (a) Final and anytime accuracy. (b) Forgetting versus current-task accuracy. The vertical line and outlined point mark $\beta = 0.9$. The square in (b) shows fixed LPC at the same strength. Error bars show one standard deviation.

Near-uniform exponential weights mean the replay average has broad support across stored inputs. This differs from the concentrated-weight regime that activates tempering. The SFFA result therefore provides a score-statistic contrast with the primary additive-goodness setting.

M.2 Tiny-ImageNet trajectory

Tiny-ImageNet-200 extends the stream to twenty phases in a single-seed analysis. Table 13 reports the three accuracy summaries. The controller moves from temperature one early in the stream to 0.014. At the later reported state, robust scale is 4.64 and median normalized ESS is 0.0052.

Table 13. Single-seed Tiny-ImageNet-200 results. All values are percentages.

Method	Final	Anytime	Current
Local ER	1.35	7.07	32.93
Fixed LPC	1.65	9.16	7.94
AT-LPC	**2.36**	**9.33**	10.41

The controller equation is consistent with these rounded statistics. Before clipping, $(1/4.64)(0.0052/0.08)$ is approximately 0.014. The reliability factor and scale both reduce the temperature. Relative to fixed LPC, current-task accuracy rises from 7.94% to 10.41%, and final accuracy rises from 1.65% to 2.36%.

M.3 Memory sensitivity

Figure A10 shows the memory-size analysis. The main setting uses 16 images per class. Memory size changes the reference sample and the potential support for score tails. The same stored examples also serve rehearsal, so changing this budget can affect both representation learning and the calibration statistics.

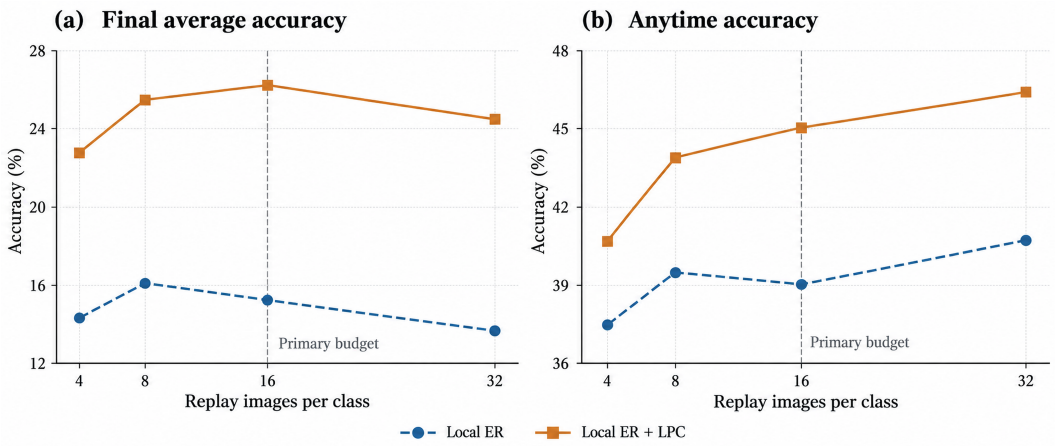

Figure A10. Replay-memory capacity in the nine-layer CIFAR-10 local learner. Seed-zero results compare Local ER and Local ER + LPC. Panels show (a) final average accuracy and (b) anytime accuracy. The vertical lines mark 16 replay images per class. Each point is one run; the same memory supports rehearsal and calibration.

Figure A10 shows higher accuracy with LPC at each tested budget. Final accuracy peaks at 16 images per class in this run, while anytime accuracy continues to rise at 32. The two metrics therefore describe different responses to memory capacity.

Normalized ESS describes the fraction of effective support. Raw ESS is that fraction multiplied by the current memory size. Two memories can therefore share the same normalized ESS while having different effective counts. Memory size thus complements normalized ESS when measuring replay support.