Controlling Temporal Pseudo-Label Marginals for Stable Online Test-Time Adaptation

Rani Amar
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel
rani.amar@biu.ac.il

Ofir Lindenbaum
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel
ofir.lindenbaum@biu.ac.il

Abstract

Continual Test-Time Adaptation (CTTA) methods update models online using a stream of unlabeled data to overcome distribution shifts encountered at test time. However, CTTA is often unstable, as it can reinforce biases in early predictions and increasingly predict only a few classes, resulting in drift or even collapse. In this paper, we identify narrowing pseudo-label class usage as an indicator of CTTA instability and quantify this behavior. Most existing methods have attempted to address this issue by either selecting reliable adaptation samples or re-weighting the loss to encourage diverse pseudo-labels, but they do not explicitly enforce a class marginal over a temporal window. Instead, we propose Sinkhorn Class-Marginal Regularization (SCALAR), a simple plug-and-play module that controls the pseudo-label distribution over a short temporal window. SCALAR works by maintaining recent predictions in a buffer, balancing their class distribution via the Sinkhorn algorithm to match a specified target distribution, and using the balanced soft targets for adaptation. Without changing the model architecture, SCALAR maintains the per-step runtime and increases the peak allocated GPU memory by only 10 MB on ImageNet-C. Our method is particularly effective on skewed, non-stationary data with severe distribution shifts, where SCALAR significantly alleviates drift and collapse when combined with existing stabilization methods based on entropy or diversity. Code is available at `https://github.com/r-isr/SCALAR`.

1 Introduction

Continual test-time adaptation (CTTA) assumes a deployment setting where a model must continuously adapt at test time under non-stationary conditions.

In CTTA, models must adapt online to a stream of unlabeled test samples that may differ in distribution from the training data and may change over time, without access to the source data [28]. Typically, CTTA methods adopt a teacher–student framework where an exponential moving average (EMA) of the student parameters defines the *teacher* model, which generates pseudo-labels for each batch. The *student* is updated online using these pseudo-labels as supervision [28, 10, 30, 20, 19]. Other methods such as TENT [27] minimize an entropy objective to directly update the BatchNorm affine parameters of a single model.

In both settings, the adaptation signal is driven by the model's own predictions. Small biases can cause the adaptation signal to focus on fewer classes even in approximately balanced data streams due to the self-training effect. This can cause the pseudo-label distribution to become less diverse and eventually collapse to a small subset of classes. The CTTA community has observed two failure modes: (1) error accumulation during test-time adaptation, ranging from *drift* (gradual accumulation of errors) to *collapse* (rapid accumulation of errors), and (2) *forgetting* of the model's accuracy on

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

clean data after adapting on corrupted data [28, 30, 12]. Existing CTTA methods attempt to mitigate temporal instabilities by modifying local model updates. These include filtering out potentially unreliable predictions [22, 23], promoting pseudo-label diversity [29], adding regularization to prevent the test-time model from deviating too far from the source model [22, 21], and reducing temporal correlation in adaptation batches using a memory bank [12, 30]. However, they do not explicitly project window-level pseudo-label class marginals toward a specified target.

We aim to control the temporal class marginals of the pseudo-labels, which form the supervision signal in CTTA. We propose Sinkhorn Class-Marginal Regularization (SCALAR), a lightweight intervention to mitigate the marginal contraction that can contribute to test-time collapse.

Concretely, SCALAR maintains a short first-in-first-out (FIFO) queue of recent teacher logits and balances their class marginals over this window using the Sinkhorn–Knopp algorithm. This generates soft labels by rescaling teacher predictions so their class marginals over the window are close to a user-specified target prior (uniform by default). This procedure approximates a constrained KL projection of teacher predictions to satisfy a desired temporal class marginal [24, 9].

SCALAR is not intended to recover the test-time class prior but to stabilize the supervision signal observed by the student. While providing the correct class prior can reduce corruption errors, SCALAR with a uniform prior still provides substantial stabilization even in skewed settings. However, SCALAR with a uniform prior can slightly hinder the model's ability to specialize to target classes if the target distribution consists of only a subset of source classes and the model is already stably adapting to them.

We find SCALAR's stabilization benefit is highly state-dependent: when class usage is stable, SCALAR often has a smaller effect. But as class usage contracts and errors increase, SCALAR can counteract this contraction. We evaluate this behavior across balanced and skewed streams, fixed and dynamic skew schedules, different adaptation speeds, both teacher–student and entropy minimization settings, and backbone architectures like ViT-B and ResNet-50.

Since SCALAR is a window-level intervention, it complements other methods that stabilize updates on a per-sample (e.g., filtering, loss weighting) or per-parameter level (regularization). We demonstrate SCALAR's complementarity to the EATA method [22], which uses an entropy- and diversity-aware, history-dependent approach to select samples for updating. Despite adaptive sample selection, EATA does not explicitly constrain class marginals of the supervision signal. Applying SCALAR on top of EATA further reduces drift and mitigates collapse.

Contributions. Our contributions are threefold:

- **Temporal control of pseudo-labels.** We propose SCALAR, which uses Sinkhorn balancing over a short queue of recent predictions to bring their class marginals close to a specified target prior.
- **When SCALAR helps.** We track the effective number of pseudo-label classes to study when marginal balancing reduces drift and collapse. We also show that a uniform prior can hinder specialization when only a subset of source classes is present, and adaptation is already stable.
- **Stabilization across adaptation methods.** We show that SCALAR reduces drift and collapse under skewed streams and aggressive adaptation in both teacher–student and entropy-minimization settings, including EATA. Our ImageNet-C measurements show no measurable increase in runtime and a 10 MB increase in peak allocated GPU memory.

2 Related Work

Test-time and continual test-time adaptation. Test-time adaptation (TTA) updates a model at inference using unlabeled test samples [26, 18]. TENT [27] does this by minimizing prediction entropy with respect to the BatchNorm affine parameters. Continual TTA (CTTA) considers streams whose distribution changes over time [28]. Many CTTA methods use a teacher–student framework, where an EMA teacher generates pseudo-labels, and the student learns from them online [28, 10, 30, 20, 19]. Self-supervised learning methods, including contrastive learning, teacher–student learning, and masked image modeling [8, 15, 13, 6, 16, 2], have also informed auxiliary objectives for adaptation without labels [7, 19].

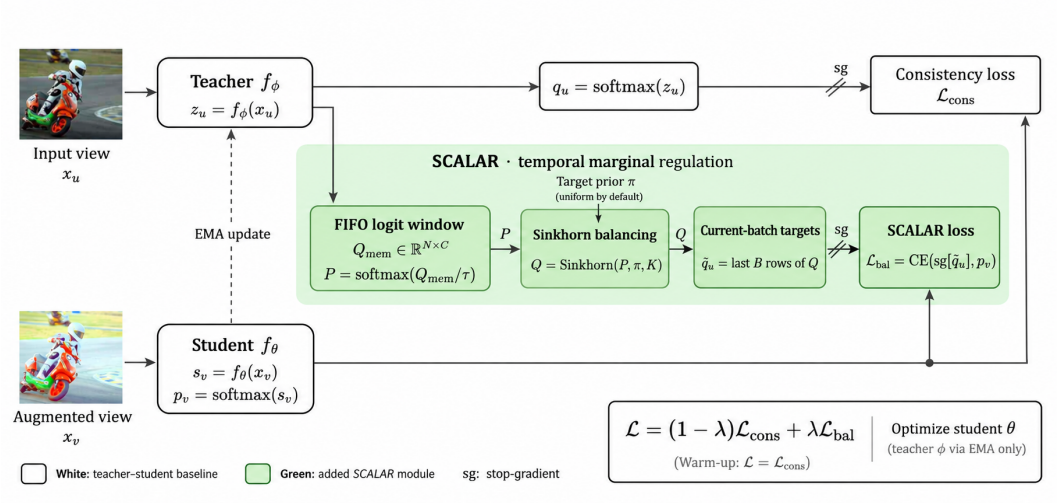


Figure 1: **Overview of the SCALAR framework.** White blocks denote the baseline teacher–student CTTA pipeline; green blocks denote the added SCALAR module. Given an input view x_u for the teacher branch and an augmented view x_v for the student branch, the teacher produces logits z_u and probabilities $q_u = \mathrm{softmax}(z_u)$. Recent teacher logits are stored in a FIFO window and converted to probabilities P, which are balanced toward a target prior π via Sinkhorn to produce balanced targets $\tilde{q}_u$. The student produces logits s_v and probabilities p_v, and is optimized using a mixture of the baseline consistency loss and the SCALAR loss based on $\tilde{q}_u$. Gradients are stopped through teacher-derived targets, marked by sg, and teacher parameters are updated via EMA. Here $\mathcal{L}_{\mathrm{cons}} \equiv \mathcal{L}_{\mathrm{consistency}}$ and $\mathcal{L}_{\mathrm{bal}} = \mathrm{CE}(\mathrm{sg}[\tilde{q}_u], p_v)$.

SeLa [1] and SwAV [5] use Sinkhorn balancing to avoid degenerate cluster assignments. ReMixMatch [3] aligns pseudo-labels with the labeled-data class distribution using a moving average of predictions over recent batches. SCALAR instead applies a joint Sinkhorn projection over a short prediction window during CTTA, using a specified target prior without source data.

Class-prior shift and temporally correlated streams. Temporally correlated inputs and class-ordered streams can increase error accumulation and cause prediction collapse in CTTA [30, 10, 21, 12]. Universal TTA [21] examines these failures under class imbalance and class-ordered streams, while NOTE [12] studies adaptation to temporally correlated samples.

Existing methods address these failures by changing either the predictions or the samples used for adaptation. Universal TTA [21] reweights predictions using a class-prior estimate from the current batch. LAME [4] adjusts predictions using similarities between samples in feature space, without updating the model weights. EATA [22] updates the model only on samples selected for reliability and diversity. DELTA [31] addresses class bias through online loss reweighting, while TRIBE [25] uses balanced normalization and an anchored self-training loss.

SCALAR acts on the pseudo-labels used to update the model. It collects recent predictions in a short queue and balances their class marginals toward a specified prior. The balanced targets then provide supervision for adaptation, limiting the concentration of pseudo-label mass on a few classes.

3 Method

We first describe the teacher–student baseline, then introduce SCALAR as a step that balances the pseudo-labels used to update the student. SCALAR collects recent predictions in a short queue so that class usage is controlled across several batches. The architecture, optimizer, and update schedule remain the same. Fig. 1 shows the full procedure.

Let $f_{\theta_0} : \mathcal{X} \to \mathbb{R}^C$ be a C-class classifier. In the CTTA setting, the model adapts online on an unlabeled stream $\{x_t\}_{t\geq 1}$ drawn from a potentially shifting distribution, without access to source

data. We denote logits by $f_\theta(x) \in \mathbb{R}^C$ and probabilities by $\mathrm{softmax}(f_\theta(x))$. At each step, the model makes predictions and may use them to perform one adaptation update.

3.1 Teacher–Student CTTA Baseline

Our main experiments compare the same teacher–student baseline with and without SCALAR. The student f_θ learns from the teacher's predictions through gradient descent. The teacher f_ϕ follows the student through an exponential moving average (EMA) of its parameters. We describe these two updates below; augmentations and hyperparameters are given in App. A.

Two-view consistency. For each unlabeled input x_u, we form an augmented view $x_v = \mathcal{T}(x_u)$, where $\mathcal{T}$ denotes a stochastic data augmentation operator. The teacher predicts from the input without further augmentation, while the student predicts from the augmented view:

$$z_u = f_\phi(x_u), \qquad q_u = \mathrm{softmax}(z_u),$$
$$s_v = f_\theta(x_v), \qquad p_v = \mathrm{softmax}(s_v).$$

The student minimizes a symmetric consistency loss:

$$\mathcal{L}_{\text{consistency}} = \tfrac{1}{2}[\mathrm{CE}(\mathrm{sg}[q_u], p_v) + \mathrm{CE}(p_v, \mathrm{sg}[q_u])]\,,$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient, and $\mathrm{CE}(a, b) = -\sum_{c=1}^{C} a_c \log b_c$ denotes the cross-entropy between distributions a and b. This loss encourages the student to agree with the teacher despite the augmentation. The teacher predictions are held fixed during the student update.

EMA teacher. After each student update, the teacher's parameters are updated as an exponential moving average with momentum coefficient $\alpha \in [0, 1)$:

$$\phi \leftarrow \alpha\,\phi + (1 - \alpha)\,\theta.$$

A larger α makes the teacher follow the student more slowly, reducing the effect of any single update. However, averaging the parameters does not directly control which classes receive pseudo-label mass. SCALAR adds this control over a window of recent predictions. In our ViT baseline, attention and the remaining parameters use separate EMA rates (App. E.1); α denotes the rate for the corresponding parameters.

3.2 SCALAR: Class-Marginal Balancing via a Logit Queue

SCALAR balances predictions over a window of N samples, including the current batch of size B. We use a temperature $\tau > 0$ to convert logits to probabilities and K Sinkhorn iterations to balance them toward a target prior $\pi \in \Delta^{C-1}$, which is uniform by default. The weight $\lambda \in [0, 1]$ controls the contribution of the balanced targets to the student loss.

Logit queue. We maintain a FIFO queue $Q_{\mathrm{mem}} \in \mathbb{R}^{N \times C}$ containing the most recent teacher logits. At each step, the oldest B logits are removed, and the current logits z_u are appended. The queue is converted to a probability matrix via a row-wise softmax:

$$P = \mathrm{softmax}(Q_{\mathrm{mem}}/\tau) \in \mathbb{R}^{N \times C}, \qquad P_{i,j} = \frac{\exp((Q_{\mathrm{mem}})_{i,j}/\tau)}{\sum_{c=1}^{C} \exp((Q_{\mathrm{mem}})_{i,c}/\tau)}.$$

The column sums $P^\top \mathbf{1}_N$ give the total probability mass assigned to each class in the window, where $\mathbf{1}_N$ is a vector of ones. Dividing by N gives the average class distribution. Using several recent batches allows us to balance this distribution without requiring each small batch to represent all classes.

Sinkhorn–Knopp balancing. Given the target prior π, we seek a matrix Q whose rows sum to one and whose column sums match $N\pi$. Starting from $Q^{(0)} = P$, we iterate:

$$\tilde{Q}^{(k)} = \mathrm{Diag}(Q^{(k)}\mathbf{1}_C)^{-1} Q^{(k)}, \qquad Q^{(k+1)} = \tilde{Q}^{(k)}\,\mathrm{Diag}\left(\frac{N\pi}{\tilde{Q}^{(k)\top}\mathbf{1}_N}\right), \tag{1}$$

followed by a final row normalization. Here $\mathrm{Diag}(\cdot)$ forms a diagonal matrix from a vector. Row normalization makes each row a probability distribution, while column scaling adjusts the total mass

Algorithm 1 SCALAR Online Update.

Input: batch x_u, teacher f_ϕ, student f_θ, queue Q_{mem} (capacity N), temperature τ, iterations K, prior π, weight λ
$x_v \leftarrow \mathcal{T}(x_u)$; $z_u \leftarrow f_\phi(x_u)$; $q_u \leftarrow \text{softmax}(z_u)$; $p_v \leftarrow \text{softmax}(f_\theta(x_v))$
if $|Q_{\text{mem}}| = N$ **then**
    Update queue (dequeue oldest B, enqueue z_u)
    $P \leftarrow \text{softmax}(Q_{\text{mem}}/\tau)$
    $Q \leftarrow \text{SinkhornBalance}(P, \pi, K)$
    $\tilde{q}_u \leftarrow$ last B rows of Q
    $\mathcal{L} \leftarrow (1-\lambda)\mathcal{L}_{\text{consistency}} + \lambda\,\text{CE}(\text{sg}[\tilde{q}_u], p_v)$
else
    $Q_{\text{mem}} \leftarrow \text{Enqueue}(Q_{\text{mem}}, z_u)$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{consistency}}$ ▷ warm-up
$\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$; $\phi \leftarrow \alpha\,\phi + (1-\alpha)\,\theta$

Table 1: **Balanced corruption benchmarks (severity 5).** Average top-1 error (%) over all 15 corruptions on CIFAR-10-C, CIFAR-100-C, and ImageNet-C with ViT-B/16 backbones. *SCALAR* denotes our teacher–student CTTA pipeline with the FIFO logit queue and Sinkhorn balancing; *Baseline w/o SCALAR* is the same pipeline without this module. SCALAR uses a uniform class prior. For SCALAR and the teacher–student baseline, results are averaged over 5 seeds (std. ≈ 0.2); prior-method numbers use their published settings, which can differ in initialization and batch size.

Method	CIFAR-10-C	CIFAR-100-C	ImageNet-C
Source-only	28.1	35.4	55.8
TENT [27]	23.5	32.1	51.0
CoTTA [28]	24.6	34.8	54.8
VDP [11]	24.1	32.0	50.0
VIDA [20]	20.7	27.3	43.4
Versatile [29]	8.9	24.0	42.7
Universal [21]	–	–	45.0
ADMA [19]	12.6	26.4	42.5
REM [14]	9.4	23.4	39.2
SCALAR (ours)	**8.1**	**23.2**	**36.1**
Baseline (w/o SCALAR)	10.3	24.7	38.1

assigned to each class. After a finite number of iterations, the column sums approximately match $N\pi$. This procedure approximates the constrained KL projection of P [24, 9]. Since classes are rescaled by different factors, their ordering within a sample can change. We use the last B rows of Q as the balanced targets $\tilde{q}_u$ for the current batch; past predictions provide context for balancing.

Combined objective. We update the student using a mixture of the consistency loss and the loss on balanced targets $\tilde{q}_u$:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\text{consistency}} + \lambda\,\text{CE}(\text{sg}[\tilde{q}_u],\, p_v), \tag{2}$$

where λ determines how strongly the balanced targets influence the student relative to the original consistency loss. As above, we stop gradients through the targets and update the teacher only via EMA. While the queue fills, we train with $\mathcal{L}_{\text{consistency}}$ alone.

Algorithm 1 summarizes the online update procedure.

Computational cost and wall-clock overhead. SCALAR adds a Sinkhorn balancing step over the logit queue. Each update costs $\mathcal{O}(KNC)$ time and $\mathcal{O}(NC)$ memory, where N is the queue length, C the number of classes, and K the number of Sinkhorn iterations. For ImageNet-C (C=1000, N=1024, K=5), this is about 0.02 GFLOPs and a ∼4 MB fp32 queue. Measured per-step time is unchanged within noise (0.1063s → 0.1062s), and peak GPU memory increases by 10 MB allocated / 20 MB reserved. Full overhead details are provided in App. F.

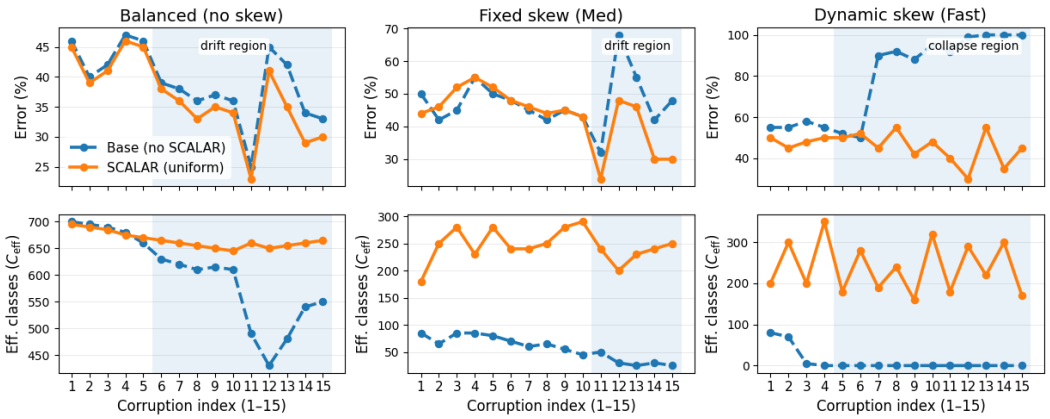


Figure 2: **Temporal contraction across adaptation regimes.** Temporal evolution of error (top) and effective pseudo-label classes C_{eff} (bottom) in a teacher–student continual test-time adaptation setup with a ViT-B/16 backbone (ImageNet-C, severity 5, single-pass protocol). Here, C_{eff} denotes the effective number of classes in the teacher pseudo-label distribution. **Left:** Balanced stream (no skew) exhibits gradual contraction and late-stage drift. **Middle:** Fixed skew with rotating dominant classes (medium adaptation) results in stronger drift. **Right:** Dynamic skew with fast adaptation induces rapid contraction leading to collapse. Across regimes, SCALAR stabilizes class usage and mitigates the associated error growth.

Table 2: **Fixed-skew continual streams across adaptation speeds (ImageNet-C, severity 5).** Each cell reports average corruption error and post-stream clean error (%), mean ± SD over three runs. Adaptation speed is controlled by the learning rate and EMA momentum (App. A.3). *SCALAR-U* uses a uniform class prior; *SCALAR-C* uses the class prior of each corruption segment (oracle reference). At high speed, the baseline collapsed in 2/3 runs; all SCALAR runs remained stable.

Method	High speed		Medium speed		Low speed	
	Avg Err	Clean Err	Avg Err	Clean Err	Avg Err	Clean Err
Baseline (w/o SCALAR)	86.8 ±14.4	76.4 ±40.6	48.3 ±4.4	25.2 ±5.9	**47.5** ±0.6	17.4 ±0.3
SCALAR-U	48.1 ±1.1	**21.1** ±0.5	42.9 ±0.7	18.2 ±0.4	49.1 ±0.4	17.3 ±0.1
SCALAR-C	**46.1** ±1.0	22.2 ±0.4	**42.5** ±0.4	**17.7** ±0.3	48.2 ±0.8	**17.1** ±0.1

4 Experiments

Our experiments address three questions:

- **(Q1) State-dependent stabilization.** When does SCALAR reduce drift and collapse? How does its benefit change as pseudo-label class usage contracts? Does it also improve predictive calibration?
- **(Q2) Choice of target prior.** When the test-time class prior is unknown, does a uniform target stabilize adaptation? How much does an accurate prior help, and when does a uniform target hinder specialization?
- **(Q3) Use with other adaptation methods.** Does SCALAR improve methods that already select samples for reliability and diversity? Does it also help TENT with a ResNet backbone?

4.1 Setup

Protocol. Unless otherwise stated, experiments follow the standard continual test-time adaptation (CTTA) protocol [28]: a pretrained model adapts online to a single-pass stream of unlabeled test data, with no access to source data or ground-truth labels during adaptation. We apply this protocol to both standard corruption streams and controlled label-skewed streams constructed from ImageNet-C.

Datasets and streams. We use ImageNet-C and CIFAR-10/100-C [17] at severity 5. These benchmarks contain corrupted versions of the original validation images, grouped into 15 corruption types such as noise, blur, weather, and digital artifacts, which are processed sequentially to form a continual test stream. For the standard balanced benchmarks, we use the original corruption streams and report average top-1 error across all corruptions.

To study sustained prior shift, we construct fixed- and dynamic-skew ImageNet-C streams by resampling the class composition within each corruption segment; full details are provided in App. G. In both cases, classes are divided into head and tail groups, and skew is induced by oversampling head-class examples while subsampling tail-class examples. In the fixed-skew setting, each segment contains 50 head classes and 950 tail classes; each head class is sampled to 100 examples, while each tail class contributes one example. Thus, the per-class head–tail ratio is 100:1, with roughly 84% of the segment mass assigned to head classes. The skew strength is fixed across the stream, while the dominant class identities are deterministically reshuffled across corruption segments. In the dynamic-skew setting, the head-set size and per-class oversampling strength both vary across segments, producing head–tail ratios ranging from 25:1 to 250:1 while keeping the aggregate head mass roughly comparable. All methods are evaluated on streams with the same skew configuration. Long-horizon persistent-skew and true-subset variants are defined with the corresponding experiments; full construction details are provided in App. G.

Models. Our primary experiments use a ViT-B/16 backbone within a teacher–student framework. We also add SCALAR to EATA [22] with ViT-B/16 and to TENT [27] with a ResNet-50 backbone.

SCALAR settings. Unless otherwise stated, we use $N = 1024$, $\tau = 0.1$, and $\lambda = 0.5$; for EATA, whose entropy filter retains only sharply peaked predictions, we use $\tau = 0.7$ and $N = 4096$. Learning rates and EMA momenta are listed in App. A.3.

Metrics. We report: (i) top-1 corruption error (lower is better), (ii) post-stream clean error, measured on clean data after adaptation to quantify forgetting, and (iii) the effective number of pseudo-label classes, $C_{\text{eff}}(t)$, which we use only as a label-free diagnostic of temporal class-marginal contraction, not as an accuracy measure:

$$C_{\text{eff}}(t) = \exp\Big(-\sum_{y=1}^{C} \bar{p}_t(y)\, \log \bar{p}_t(y) \Big),$$

where $\bar{p}_t$ is the window-averaged pseudo-label distribution (before Sinkhorn balancing). $C_{\text{eff}}(t) = C$ under uniform class usage and decreases as probability mass concentrates on fewer classes. Concentration can be appropriate when the stream contains only a few classes, and diverse predictions can still be wrong. We therefore examine $C_{\text{eff}}(t)$ alongside error when studying drift and collapse.

Baselines and variants. Baseline denotes our teacher–student CTTA pipeline without SCALAR's Sinkhorn marginal-balancing module. **SCALAR** or **SCALAR full framework** denotes the same teacher–student pipeline with the SCALAR module. When SCALAR is added to an existing method, we write **Method+SCALAR**, e.g., EATA+SCALAR or TENT+SCALAR, meaning that only the SCALAR marginal-regulation module is added to that method. Unless otherwise stated, SCALAR uses a uniform class prior. We write **SCALAR-U** for this variant and **SCALAR-C** for the oracle-prior variant; the target is the segment prior or aggregate stream prior, as specified for each experiment.

4.2 State-Dependent Stabilization

We first examine how pseudo-label class usage changes during adaptation. We track $C_{\text{eff}}(t)$, defined in Sec. 4.1, together with error to see when class contraction accompanies a loss of accuracy.

Fig. 2 shows the evolution of corruption error and $C_{\text{eff}}(t)$ across balanced, fixed-skew, and dynamic-skew regimes. Error tends to increase as $C_{\text{eff}}(t)$ decreases. Balanced streams show mild gradual contraction, fixed skew strengthens this effect, and dynamic skew with rapid adaptation leads to sharp contraction and collapse.

On standard balanced corruption benchmarks (Tab. 1), where contraction is mild, SCALAR gives moderate improvements (e.g., $38.1\% \rightarrow 36.1\%$ on ImageNet-C). Under stronger drift or collapse, SCALAR reduces the decline in $C_{\text{eff}}(t)$ and the associated rise in error. Its benefit is larger in these unstable regimes.

Table 3: **Calibration under aggressive adaptation** (ViT-B/16, fixed-skew ImageNet-C, high speed as in Tab. 2, single pass, averaged over three runs). The baseline collapsed in 2/3 runs.

Method	ECE ↓	NLL ↓
Baseline	0.73	13.37
SCALAR	**0.07**	**2.15**

Table 4: **EATA under dynamic label skew** (ImageNet-C, severity 5, ViT-B/16). Corruption / post-stream clean error (%), mean ± SD over three runs (stable) and eight runs (aggressive). Under aggressive adaptation, EATA collapses in 4/8 runs, whereas all SCALAR runs remain stable.

Method	Avg Err	Clean Err	Coll.
Stable			
EATA	50.8±2.4	18.8±0.6	0/3
+SCALAR-U	**49.6**±2.1	**18.0**±0.4	0/3
Aggressive			
EATA	77.0±11.3	67.0±37.0	4/8
+SCALAR-U	**57.0**±4.6	**21.1**±1.0	**0/8**

4.3 Uniform Targets under Class Skew

A uniform target may differ substantially from the class distribution in a skewed stream. We test whether it can still stabilize adaptation, then compare it with the correct prior to measure the cost of this mismatch.

Uniform prior in skewed streams. Tab. 2 reports results on fixed-skew ImageNet-C streams across adaptation speeds, controlled by the learning rate and EMA momentum. In high-speed regimes, where the baseline collapses in 2/3 runs, SCALAR-U reduces the corruption error from 86.8% to 48.1% and the post-stream clean error from 76.4% to 21.1%. It also improves corruption and clean performance at medium speed. At low speed, however, corruption error increases from 47.5% to 49.1%, while clean error changes from 17.4% to 17.3%, consistent with over-regularization when adaptation is already stable. Thus, the benefit depends on whether adaptation is becoming unstable, even though the class skew is the same at all three speeds. A target-sharpening baseline also collapses in 2/3 runs at high speed (App. E.4).

Benefit of an accurate prior. To quantify the benefit of accurate prior knowledge, we compare against **SCALAR-C**, which uses the empirical prior of each corruption segment as an oracle reference. Across the fixed-skew regimes, the correct prior further reduces corruption error relative to the uniform prior. At high speed, both targets prevent collapse in these runs. Knowing the prior improves accuracy further, but is not required for the observed stabilization.

4.4 Calibration

We also examine whether SCALAR improves the probabilities assigned to predictions. In a separate single-pass evaluation under aggressive adaptation, SCALAR reduces expected calibration error (ECE) from 0.73 to 0.07 and negative log-likelihood (NLL) from 13.37 to 2.15 (Tab. 3). These results show improved calibration in this setting, in addition to the reductions in top-1 error reported above.

4.5 Adding SCALAR to EATA

We next test whether SCALAR helps when the adaptation method already selects samples for reliability and diversity. EATA uses prediction entropy and a running estimate of past predictions to select samples for updating. We add SCALAR to control the class marginals of the supervision signal and evaluate both methods under dynamic skew (Tab. 4). In all EATA experiments, with or without SCALAR, the Fisher regularizer is disabled.

At EATA's stable operating point, adding SCALAR reduces average error from 50.8% to 49.6%. However, EATA is sensitive to larger learning rates: under aggressive adaptation its average error rises to 77.0% and it collapses in four of eight runs. SCALAR prevents collapse in all runs, reducing average error to 57.0% and keeping clean error at 21.1%, close to its stable-regime level. Sample selection alone does not prevent collapse at this learning rate; balancing the targets provides additional stabilization.

Table 5: **Long-horizon persistent skew** (ImageNet-C, five passes, >400K images, three runs). The same 50 classes dominate throughout (100 samples each, one per remaining class, per corruption). Error (%), mean ± SD. SCALAR-C uses the aggregate class prior of the stream (gray, oracle reference). Bold: best non-oracle result.

Method	Pass 1	Pass 5	Overall	Final clean	Coll.
EATA	58.1	78.6	66.5±12.9	52.3±41.3	1/3
+SCALAR-U	**49.3**	**48.7**	**50.0**±2.7	**21.5**±1.4	**0/3**
+SCALAR-C	47.9	43.3	43.9±1.7	21.4±0.8	0/3

Table 6: **Long-horizon class absence** (ImageNet-C, five passes, two runs). Only the same 50 classes appear; the remaining 950 are absent. Error (%), mean over runs; SCALAR-C in gray (oracle reference). Bold: best non-oracle result.

Method	Pass 1	Pass 5	Final clean	Coll.
Conservative				
EATA	**51.2**	**44.2**	18.3	0/2
+SCALAR-U	51.6	48.7	**17.6**	0/2
+SCALAR-C	51.8	42.6	17.4	0/2
Aggressive				
EATA	70.7	76.9	64.1	1/2
+SCALAR-U	**50.0**	**52.9**	**22.2**	**0/2**
+SCALAR-C	54.6	37.3	20.6	0/2

Table 7: **Transfer to TENT** (ResNet-50, standard ImageNet-C stream, severity 5). Corruption / post-stream clean error (%), mean ± SD over three random stream orderings, paired across methods. At medium and high speeds, TENT collapses in 3/3 runs; all TENT+SCALAR runs remain stable.

	Optimal speed		Medium speed		High speed	
Method	Avg Err	Clean Err	Avg Err	Clean Err	Avg Err	Clean Err
TENT-R50 [27]	67.5 ±0.4	32.9 ±0.3	88.3 ±1.3	97.1 ±1.1	95.9 ±0.4	99.6 ±0.2
TENT + SCALAR	66.5 ±0.3	30.9 ±0.2	**67.4** ±0.5	**37.1** ±0.8	**69.8** ±1.1	**43.3** ±1.2

Long-horizon persistent skew. The dynamic-skew stream changes which classes dominate. To test whether the benefit persists when the imbalance lasts longer, we keep the same 50 dominant classes throughout five passes (Tab. 5). EATA progressively degrades, from 58.1% error in the first pass to 78.6% in the fifth, and collapses in one of three runs. EATA+SCALAR-U remains stable in all runs, reaching 48.7% in the fifth pass and reducing final clean error from 52.3% to 21.5%. Using the aggregate stream prior further reduces overall error from 50.0% to 43.9%, whereas the stability benefit is already obtained with a uniform target.

Boundary case: true class absence. In the persistent-skew experiment, all classes still appear. We now remove the tail classes entirely: only the same 50 classes appear over five passes, and the remaining 950 are absent (Tab. 6). A uniform target assigns mass to these absent classes, so it may hinder specialization to the observed subset. Under conservative adaptation, EATA specializes to the observed classes, whereas EATA+SCALAR-U improves less, indicating mild over-regularization; the aggregate stream prior (SCALAR-C) recovers this specialization. Under aggressive adaptation, EATA collapses in one of two runs, whereas EATA+SCALAR-U remains stable but does not improve across passes. SCALAR-C again improves steadily. The uniform target prevents collapse in this setting, but limits specialization to the classes that are present.

4.6 Transfer Beyond ViT Teacher–Student Adaptation

We also add SCALAR to TENT with a ResNet-50 backbone (Tab. 7). This tests whether balancing helps a single model trained by entropy minimization, without an EMA teacher. SCALAR gives a small improvement at the stable operating point and larger gains when faster adaptation causes drift and forgetting. At high adaptation speed, where TENT collapses in all three runs, SCALAR reduces corruption error from 95.9% to 69.8% and clean error from 99.6% to 43.3%.

4.7 Ablation: Temporal Window and Hyperparameters

We examine the effect of the temporal window size N, temperature τ, loss weight λ, and the number of Sinkhorn iterations K on ImageNet-C. A longer window includes more predictions when balancing class usage, but also retains older predictions for longer.

We evaluate $N \in \{256, 512, 1024, 4096\}$, $\tau \in [0.01, 1.0]$, and $\lambda \in [0.1, 0.9]$. Corruption error is 36.1% at both $N = 1024$ and 4096, compared with 38.2% at $N = 512$. It varies by 1.1 percentage points over $\tau \in [0.01, 1.0]$ and by 0.5 percentage points over $\lambda \in [0.3, 0.7]$. Additional Sinkhorn iterations beyond $K = 5$ did not yield measurable improvements. Detailed sensitivity curves are provided in App. C.

5 Conclusion

We proposed SCALAR to control the class marginals of the pseudo-labels used for continual test-time adaptation. SCALAR balances recent predictions over a short window before they are used to update the model. Its benefit is largest in our experiments when class usage contracts and adaptation becomes unstable. It also reduces drift and collapse when added to EATA and TENT, and improves calibration under aggressive adaptation.

A uniform target provides stabilization without knowledge of the test-time prior. The correct prior can improve accuracy further, but the choice of target also matters when adaptation is already stable. Under persistent class absence, uniform balancing can limit useful specialization to the classes that remain.

Limitations and future work. Our experiments focus on image classification and controlled skewed streams constructed from corruption benchmarks. Further work is needed to test SCALAR on naturally occurring non-i.i.d. streams and on tasks such as detection and segmentation. A further question is how to adapt the target prior and balancing strength so that they allow specialization while still preventing collapse.

Broader impact. Reducing adaptation-induced errors could improve the reliability of deployed image classifiers under changing conditions. However, forcing a uniform target can worsen performance under genuine class imbalance, so deployment requires validation on the intended population and monitoring for harmful adaptation.

Acknowledgments and Disclosure of Funding

Ofir Lindenbaum was supported by the MOST grant No. 0007341. The authors declare no competing interests.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020.

[4] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, 2022.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[7] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, 2022.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

[10] Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *CVPR*, 2023.

[11] Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *AAAI*, 2023.

[12] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: Robust continual test-time adaptation against temporal correlation. In *NeurIPS*, 2022.

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[14] Jisu Han, Jaemin Na, and Wonjun Hwang. Ranked entropy minimization for continual test-time adaptation. In *ICML*, 2025.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[18] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. doi: 10.1016/j.patcog.2018.03.005.

[19] Jiaming Liu, Ran Xu, Senqiao Yang, Renrui Zhang, Qizhe Zhang, Zehui Chen, Yandong Guo, and Shanghang Zhang. Continual-MAE: Adaptive distribution masked autoencoders for continual test-time adaptation. In *CVPR*, 2024.

[20] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. ViDA: Homeostatic visual domain adapter for continual test time adaptation. In *ICLR*, 2024.

[21] Robert A. Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *WACV*, 2024.

[22] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, 2022.

[23] 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 *ICLR*, 2023.

[24] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964. doi: 10.1214/aoms/1177703591.

[25] Yongyi Su, Xun Xu, and Kui Jia. Towards real-world test-time adaptation: Tri-net self-training with balanced normalization. In *AAAI*, 2024.

[26] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

[27] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[28] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, 2022.

[29] Xu Yang, Xuan Chen, Moqi Li, Kun Wei, and Cheng Deng. A versatile framework for continual test-time domain adaptation: Balancing discriminability and generalizability. In *CVPR*, 2024.

[30] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *CVPR*, 2023.

[31] Bowen Zhao, Chen Chen, and Shu-Tao Xia. DELTA: Degradation-free fully test-time adaptation. In *ICLR*, 2023.

Table 8: Reproduction checks against published benchmark results. All values are average top-1 corruption error (%). Published results are retained in Table 1.

Method / Dataset	Reported	Reproduced
Source-only / CIFAR-10-C	28.1	27.9
Source-only / CIFAR-100-C	35.4	36.0
Source-only / ImageNet-C	55.8	56.2
VIDA / ImageNet-C	43.4	43.35
TENT / ImageNet-C	51.0	50.48

This appendix gives implementation details, hyperparameter analyses, per-corruption results, additional ablations, computational costs, and the construction of skewed streams.

A Implementation Details

A.1 Experimental Setup

Baselines and reproduction. For methods whose results are taken from prior CTTA papers, we use the reported numbers under the corresponding benchmark setting. As a check on our evaluation, we reproduce the source-only baseline using the ViT-B/16 checkpoint from our experiments. The reproduced source-only errors differ from the published values by at most 0.6 percentage points. Table 8 lists these reproduction checks. Our main controlled comparisons use paired runs with and without SCALAR under the same initialization.

Hyperparameters (balanced benchmarks). For CIFAR-10-C experiments, we use a batch size of 4, dual-rate EMA parameters $(\alpha_{\text{att}}, \alpha_{\text{rest}}) = (0.90, 0.99)$, and learning rate $\eta = 5 \times 10^{-5}$. CIFAR-100-C experiments use batch size 4, learning rate $\eta = 5 \times 10^{-5}$, and dual-rate EMA parameters $(\alpha_{\text{att}}, \alpha_{\text{rest}}) = (0.9999, 0.999)$. ImageNet-C experiments use batch size 8, learning rate $\eta = 10^{-5}$, and dual-rate EMA parameters $(\alpha_{\text{att}}, \alpha_{\text{rest}}) = (0.90, 0.99)$. We chose EMA rates for stable adaptation of the teacher–student baseline without SCALAR and fixed them for each dataset. The ablation in App. E.1 compares single- and dual-rate EMA settings with SCALAR; it does not compare EMA sensitivity with and without SCALAR.

A.2 Augmentation Pipeline

The teacher processes the incoming images without further augmentation, while the student receives strongly augmented views for the consistency objective. Table 9 lists the student augmentations. We use transformations intended to preserve the class label while changing appearance, such as color changes and small geometric distortions. These augmentations are used for the adaptation loss; predictions are made on inputs without these additional transformations.

A.3 Experiment-Specific Settings

Unless otherwise stated, all experiments use the SCALAR settings of Sec. 4.1 and the batch sizes of App. A.1, and collapse is defined as in App. G. The adaptation regimes in the main paper differ mainly in learning rate and, for the teacher–student pipeline, EMA momentum; we list these values below for each experiment.

Fixed-skew adaptation speeds (Tab. 2). High speed uses $\eta = 10^{-5}$ with $\alpha_{\text{att}} = \alpha_{\text{rest}} = 0.9$; medium speed uses $\eta = 5{\times}10^{-6}$ with $\alpha_{\text{att}} = \alpha_{\text{rest}} = 0.99$; and low speed uses $\eta = 5{\times}10^{-7}$ with $\alpha_{\text{att}} = \alpha_{\text{rest}} = 0.999$.

EATA under dynamic skew (Tab. 4). When SCALAR is added to EATA, the queue stores the logits of the samples that EATA selects for the update, and the loss combines EATA's weighted entropy with the cross-entropy to the balanced targets, with weights $1 - \lambda$ and λ. The queue retains the logits of the latest N selected samples; steps with no selected samples skip the update and leave the queue unchanged. In all EATA experiments, with or without SCALAR, the Fisher regularizer

is disabled. The stable operating point uses $\eta = 10^{-5}$, chosen as the learning rate at which EATA alone remains stable across runs; the aggressive regime uses $\eta = 6{\times}10^{-4}$. EATA uses batch size 8, entropy margin $E_0 = \ln(1000)/2 - 1$, and redundancy margin $\epsilon = 0.05$.

Long-horizon experiments (Tabs. 5–6). The persistent-skew stream uses $\eta = 3{\times}10^{-4}$. On the class-absence stream, the conservative regime uses $\eta = 5{\times}10^{-5}$ and the aggressive regime uses $\eta = 3{\times}10^{-4}$.

TENT (Tab. 7). Since TENT has no teacher, SCALAR stores the model's own logits in the queue, and the loss combines the entropy with the cross-entropy to the balanced targets, with weights $1 - \lambda$ and λ. TENT uses batch size 64; optimal, medium, and high speeds use $\eta = 2.5{\times}10^{-4}$, $8{\times}10^{-2}$, and $1.5{\times}10^{-1}$, respectively.

Temporal contraction (Fig. 2). The balanced stream uses the ImageNet-C settings of App. A; the fixed- and dynamic-skew streams use the medium and high speeds of Tab. 2, respectively.

Table 9: Student augmentation pipeline for consistency training. Strong augmentations are applied only to student inputs; teacher inputs are not augmented.

Augmentation	Parameters + note
RandomAffine	degrees 0° translate $(32/W,\ 32/H)$ scale 0.98–1.02, bilinear, fill=0
ColorJitterPro	brightness 0.6–1.4, contrast 0.7–1.3, saturation 0.5–1.5, hue ± 0.06, gamma 0.7–1.3
RandomGrayscale	$p = 0.2$

B Additional Details on Sinkhorn Balancing

B.1 KL-Projection Interpretation

The balancing step seeks targets that satisfy the desired class marginals while staying close to the teacher predictions in KL divergence [24, 9]. The constraints apply to the whole window: individual samples can still have different class distributions. Below, we give the window construction and normalization steps used to approximate this projection.

Window of teacher predictions. Let B be the current batch size and $M = N - B$ the number of past logits retained in the queue, so that the window has N rows as in Sec. 3. We stack the teacher logits from the current batch and the queue:

$$\text{logits}_{\text{eff}} \in \mathbb{R}^{(B+M)\times C}.$$

We convert logits to probabilities with a temperature $\tau > 0$:

$$P = \text{softmax}\big(\text{logits}_{\text{eff}}/\tau\big) \in \mathbb{R}^{N\times C}, \quad N = B + M.$$

Each row of P is a class distribution for one sample. The column sums give the total predicted mass per class; dividing them by N gives the average distribution over the window.

Sinkhorn balancing to a prior. We seek $Q \in \mathbb{R}^{N\times C}$ with row sums 1 and column marginals $N\pi \in \mathbb{R}^C$:

$$Q\mathbf{1} = \mathbf{1}, \qquad \mathbf{1}^\top Q = N\,\pi^\top,$$

where $\pi \in \Delta^{C-1}$ encodes the desired class usage (uniform $\pi_c = 1/C$ when no test-time prior is known, or a given prior when class skew is specified). We obtain Q by applying K rounds of Sinkhorn scaling. Starting from $Q^{(0)} = P$, each Sinkhorn iteration consists of:

$$\tilde{Q}^{(k)} \leftarrow Q^{(k)} \oslash \big(Q^{(k)}\mathbf{1}\big),$$

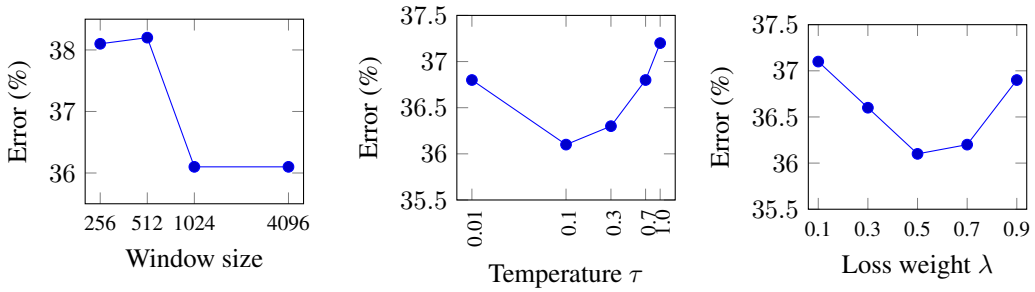


Figure 3: **SCALAR hyperparameter sensitivity on ImageNet-C (class-balanced corruption setting, severity 5; error %).**

$$Q^{(k+1)} \leftarrow \tilde{Q}^{(k)} \operatorname{Diag}\Big(\frac{N\,\pi}{\tilde{Q}^{(k)\top}\mathbf{1}_N}\Big), \qquad k = 0, \ldots, K-1,$$

and a final row normalization

$$Q \leftarrow Q^{(K)} \oslash \big(Q^{(K)}\mathbf{1}\big).$$

Here $\oslash$ denotes row-wise division by the current row sums and $\operatorname{Diag}(\cdot)$ forms a diagonal matrix, as in Eq. (1). The final normalization ensures that each row sums to one. Since we use a finite number of iterations, the column sums need only approximately match $N\pi$; Q therefore approximates the KL projection onto the specified marginals. In the implementation, target column masses are clamped to at least 10^{-12} and 10^{-12} is added to all row and column sums, which avoids division by zero for classes with zero prior mass.

Objective and update. Let q_u denote the original teacher probabilities and $\tilde{q}_u$ the balanced targets for the current batch obtained by Sinkhorn projection toward the target prior π. The SCALAR-specific supervision term is

$$\mathcal{L}_{\text{bal}} = \text{CE}(\text{sg}[\tilde{q}_u],\, p_v).$$

The combined objective is

$$\mathcal{L} = (1-\lambda)\,\mathcal{L}_{\text{consistency}} + \lambda\,\mathcal{L}_{\text{bal}},$$

which is identical to Eq. (2). During warm-up, before marginal balancing is activated, the method reduces to $\mathcal{L}_{\text{consistency}}$.

We apply a single gradient step on the student parameters θ and then update the teacher via the dual-rate EMA. SCALAR modifies the teacher targets used in the loss without changing the backbone architecture.

C Hyperparameter Sensitivity

C.1 SCALAR module ablations (queue + Sinkhorn)

We vary the window size, temperature, and loss weight to examine how each affects SCALAR. All experiments in this section use a uniform target and K=5 Sinkhorn iterations.

Figure 3 summarizes the sensitivity of SCALAR to its three primary hyperparameters on ImageNet-C in the standard corruption setting: (i) the mixing weight λ controlling the strength of the SCALAR term, (ii) the temperature τ applied in the softmax over the queue window, and (iii) the window size N (queue length).

Windows of 256–512 samples give higher error than windows of 1024–4096 samples. Increasing the window from $N = 1024$ to 4096 gives no further reduction in this sweep. Thus, using more recent predictions helps up to a point, but the largest tested window offers no advantage over our default.

Error is lowest at $\tau = 0.1$ and $\lambda = 0.5$, with slightly higher values at either end of the tested ranges. The mixing-weight sweep under class skew (App. E.3) also gives lower corruption errors at $\lambda = 0.5$–0.75 than at 0.25.

Table 10: **CIFAR-10-C (severity 5) corruption results (error %, lower is better).** Per-corruption test error and average over all 15 corruptions. *Gain* is the improvement in average error relative to the original Source model, in percentage points (higher is better). **Boldface** indicates the best method for each corruption. SCALAR results are averaged over 5 seeds; the standard deviation of its average error is below 0.2 percentage points. Prior-method values are taken from the cited publications.

Time		$t \longrightarrow$																
Method	Ref	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contrast	Elastic	Pixelate	JPEG	Avg.	Gain
Source		60.1	53.2	38.3	19.9	35.5	22.6	18.6	12.1	12.7	22.8	5.3	49.7	23.6	24.7	23.1	28.1	0.0
TENT	[27]	57.7	56.3	29.4	16.2	35.3	16.2	12.4	11.0	11.6	14.9	4.7	22.5	15.9	29.1	19.5	23.5	4.6
CoTTA	[28]	58.7	51.3	33.0	20.1	34.8	20.0	15.2	11.1	11.3	18.5	4.0	34.7	18.8	19.0	17.9	24.6	3.5
VDP	[11]	57.5	49.5	31.7	21.3	35.1	19.6	15.1	10.8	10.3	18.1	4.0	27.5	18.4	22.5	19.9	24.1	4.0
VIDA	[20]	52.9	47.9	19.4	11.4	31.3	13.3	7.6	7.6	9.9	12.5	3.8	26.3	14.4	33.9	18.2	20.7	7.4
Versatile	[29]	**16.3**	11.1	9.6	8.4	14.6	8.6	5.5	6.3	5.7	7.1	**3.3**	5.4	10.9	7.7	12.8	8.9	19.2
REM	[14]	17.3	12.5	10.3	8.4	17.7	8.4	5.5	6.6	5.6	7.2	3.7	6.4	11.0	7.3	13.0	9.4	18.7
SCALAR	**Proposed**	20.32	**10.32**	**9.52**	**7.68**	**12.14**	**7.76**	**5.42**	**5.94**	**5.06**	**6.12**	3.48	**3.82**	**8.64**	**5.58**	**9.62**	**8.1**	**20.0**

Table 11: **CIFAR-100-C (severity 5) corruption results (error %, lower is better).** Per-corruption test error and average over all 15 corruptions. *Gain* is the improvement in average error relative to the original Source model, in percentage points (higher is better). **Boldface** indicates the best method for each corruption. SCALAR results are averaged over 5 seeds; the standard deviation of its average error is below 0.2 percentage points. Prior-method values are taken from the cited publications.

Time		$t \longrightarrow$																
Method	Ref	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contrast	Elastic	Pixelate	JPEG	Avg.	Gain
Source		55.0	51.5	26.9	24.0	60.5	29.0	21.4	21.1	25.0	35.2	11.8	34.8	43.2	56.0	35.9	35.4	0
TENT	[27]	53.0	47.0	24.6	22.3	58.5	26.5	19.0	21.0	23.0	30.1	11.8	25.2	39.0	47.1	33.3	32.1	3.3
CoTTA	[28]	55.0	51.3	25.8	24.1	59.2	28.9	21.4	21.0	24.7	34.9	11.7	31.7	40.4	55.7	35.6	34.8	0.6
VDP	[11]	54.8	51.2	25.6	24.2	59.1	28.8	21.2	20.5	23.3	33.8	**7.5**	**11.7**	32.0	51.7	35.2	32.0	3.4
VIDA	[20]	50.1	40.7	22.0	21.2	45.2	21.6	**16.5**	17.9	16.6	25.6	11.5	29.0	29.6	34.7	27.1	27.3	8.1
Versatile	[29]	38.2	31.8	18.2	20.8	34.3	**20.3**	17.5	**14.9**	**16.2**	22.9	11.5	27.5	28.2	32.5	**25.3**	24.0	11.4
ADMA	[19]	48.6	30.7	18.5	21.3	38.4	22.2	17.5	19.3	18.0	24.8	13.1	27.8	31.4	35.5	29.5	26.4	9.0
REM	[14]	**29.2**	**25.5**	**17.0**	**19.1**	35.2	21.2	18.3	19.5	18.7	22.8	15.5	17.6	31.6	26.2	33.0	23.4	12.0
SCALAR	**Proposed**	38.26	26.98	21.6	23.5	**28.78**	22.66	20.16	20.48	19.02	**21.2**	16.5	16.82	**25.6**	**21.06**	25.44	**23.2**	**12.2**

D Per-Corruption Results

The following tables report results for all 15 corruption types under the default configurations used in the main paper.

In Tables 10–12, boldface indicates the best method for each corruption. On CIFAR-10-C, SCALAR attains the lowest error for 13 of 15 corruptions, and on ImageNet-C, it is best on 13 of 15 as well, while also achieving the lowest average error and the largest gain. On CIFAR-100-C, SCALAR is best on 4 out of 15 corruptions, but still attains the best average error and gain.

E Additional Ablation Studies

E.1 Dual vs. Single EMA on CIFAR-100-C

Dual-rate EMA teacher. We use an EMA teacher with separate decay rates for attention blocks and the remaining layers. Let θ^{att} and θ^{rest} denote the student parameters for attention and for the rest of the network (MLP, convolution, embeddings), and ϕ^{att}, ϕ^{rest} the corresponding teacher parameters. After each student update, we apply

$$\phi^{\text{att}} \leftarrow \alpha_{\text{att}}\, \phi^{\text{att}} + (1 - \alpha_{\text{att}})\, \theta^{\text{att}},$$
$$\phi^{\text{rest}} \leftarrow \alpha_{\text{rest}}\, \phi^{\text{rest}} + (1 - \alpha_{\text{rest}})\, \theta^{\text{rest}},$$

with $(\alpha_{\text{att}}, \alpha_{\text{rest}})$ fixed by the teacher–student baseline configuration, independently of SCALAR. Separate rates allow attention and the remaining parameters to follow student updates at different speeds. We use dual-rate EMA in our ViT teacher–student experiments; SCALAR can also be used with other teacher–student designs or without an EMA teacher.

To study the effect of using separate EMA rates for different layers, we run a controlled ablation on CIFAR-100-C (severity 5) with SCALAR. We fix the learning rate and SCALAR hyperparameters (uniform prior, temperature $\tau = 0.1$, window size $N = 1024$, Sinkhorn iterations $K = 5$) and vary only the EMA configuration.

Table 12: **ImageNet-C (severity 5) corruption results (error %, lower is better).** Per-corruption test error and average over all 15 corruptions. *Gain* is the improvement in average error relative to the original Source model, in percentage points (higher is better). **Boldface** indicates the best method for each corruption. SCALAR results are averaged over 5 seeds; the standard deviation of its average error is below 0.2 percentage points. Prior-method values are taken from the cited publications. Versatile's reported mean of 42.7% is retained; its displayed entries average to 42.77%.

Time		$t \longrightarrow$																
Method	Ref	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contrast	Elastic	Pixelate	JPEG	Avg.	Gain
Source		53.0	51.8	52.1	68.5	78.8	58.5	63.3	49.9	54.2	57.7	26.4	91.4	57.5	38.0	36.2	55.8	0.0
TENT	[27]	52.2	48.9	49.2	65.8	73.0	54.5	58.4	44.0	47.7	50.3	23.9	72.8	55.7	34.4	33.9	51.0	4.8
CoTTA	[28]	52.9	51.6	51.4	68.3	78.1	57.1	62.0	48.2	52.7	55.3	25.9	90.0	56.4	36.4	35.2	54.8	1.0
VDP	[11]	52.7	51.6	50.1	58.1	70.2	56.1	58.1	42.1	46.1	45.8	23.6	70.4	54.9	34.5	36.1	50.0	5.8
VIDA	[20]	47.7	42.5	42.9	52.2	56.9	45.5	48.9	38.9	42.7	40.7	24.3	52.8	49.1	33.5	33.1	43.4	12.4
Versatile	[29]	47.5	42.1	41.6	55.5	55.4	44.5	47.9	38.8	37.8	39.6	23.6	57.0	44.4	33.5	32.3	42.7	13.1
ADMA	[19]	46.3	41.9	42.5	51.4	54.9	43.3	40.7	34.2	35.8	64.3	23.4	60.3	37.5	29.2	31.4	42.5	13.3
REM	[14]	**43.5**	**38.1**	39.2	53.2	49.0	43.5	42.8	37.5	35.2	35.4	23.2	46.8	41.6	28.9	30.2	39.2	16.6
SCALAR	**Proposed**	46.4	39.78	**38.88**	**46.24**	**44.28**	**38.2**	**37.02**	**33.28**	**33.22**	**32.42**	**21.74**	**40.9**	**34.36**	**26.76**	**28.54**	**36.1**	**19.7**

Table 13: **Single vs. dual EMA on CIFAR-100-C (severity 5).** Each row reports per-corruption error (%) on the CIFAR-100-C severity-5 stream, along with the average corruption error (*Avg.*) and the post-stream clean error on CIFAR-100 (*Clean*). Single-EMA rows use a shared rate for all parameters ($\alpha_{\text{att}} = \alpha_{\text{rest}}$), whereas the dual-EMA configuration decouples attention and the remaining weights.

LR	α_{rest}	α_{att}	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contrast	Elastic	Pixelate	JPEG	Avg.	Clean
5×10^{-5}	0.9990	0.9990	40.61	30.07	25.03	26.99	32.66	24.94	22.76	23.75	22.07	25.28	19.25	19.57	31.42	29.34	32.71	27.10	25.30
5×10^{-5}	0.9991	0.9991	40.00	**28.84**	24.43	26.45	33.75	25.66	23.52	25.63	23.19	26.36	21.08	20.98	32.70	29.66	32.57	27.70	26.66
5×10^{-5}	0.9992	0.9992	41.08	29.82	23.95	26.38	31.99	24.83	22.23	23.64	21.46	24.87	19.06	19.43	31.31	28.88	32.87	26.80	25.38
5×10^{-5}	0.9993	0.9993	41.84	30.24	25.31	27.02	34.87	25.77	22.24	24.21	22.60	27.14	19.88	19.97	31.95	28.05	31.20	27.50	24.66
5×10^{-5}	0.9994	0.9994	42.22	30.69	24.05	25.94	32.54	24.36	21.43	23.27	20.91	24.35	18.22	18.42	29.80	27.57	29.88	26.20	22.56
5×10^{-5}	0.9995	0.9995	43.48	31.34	24.91	27.29	33.43	24.87	21.60	23.17	20.85	24.69	18.13	18.71	30.59	31.42	29.85	27.00	22.63
5×10^{-5}	0.9996	0.9996	44.44	31.92	23.41	25.95	33.07	24.69	21.80	22.93	20.52	25.18	17.59	18.45	29.21	27.75	28.52	26.40	21.20
5×10^{-5}	0.9997	0.9997	47.44	36.02	23.51	25.30	33.43	23.65	19.61	20.66	18.99	23.34	15.56	16.65	27.82	29.03	27.84	25.90	19.49
5×10^{-5}	0.9998	0.9998	49.20	38.28	22.70	24.75	36.84	23.36	19.27	20.01	18.39	23.00	15.04	16.02	27.40	28.78	26.91	26.00	18.35
5×10^{-5}	0.99985	0.99985	50.17	40.31	22.52	24.94	36.12	23.69	18.94	19.26	17.72	22.66	13.70	15.84	26.41	28.27	25.67	25.70	17.14
5×10^{-5}	0.9999	0.9999	51.47	42.70	22.45	23.59	37.90	22.53	17.81	**17.73**	**16.36**	21.80	**12.34**	**14.78**	24.71	31.79	25.71	25.60	**15.92**
5×10^{-5}	**0.9990**	**0.9999**	**43.56**	32.51	**20.39**	**22.05**	**31.25**	**20.69**	**17.53**	18.32	16.67	**20.55**	13.66	15.43	**24.12**	**23.76**	**23.64**	**22.90**	16.73

Tab. 13 compares a *single-EMA* scheme, where a shared rate α is used for all parameters, and a *dual-EMA* scheme, where attention blocks and the remaining weights use different EMA rates $(\alpha_{\text{rest}}, \alpha_{\text{att}})$. For the single-EMA setting we sweep α in the range $[0.9990, 0.9999]$ in steps of 10^{-4} (with an additional point at $\alpha = 0.99985$), while the dual-EMA configuration uses $(\alpha_{\text{rest}}, \alpha_{\text{att}}) = (0.999, 0.9999)$.

The average corruption error is measured on the CIFAR-100-C severity-5 stream, and the clean-post error is measured on CIFAR-100 after completing the stream. Within this sweep, the dual-EMA configuration achieves the lowest observed corruption error (22.9%) and reduces the clean-post error below the initial clean error (from $\approx 17.3\%$ to 16.7%). The slowest single-EMA setting ($\alpha = 0.9999$) achieves the best clean-post error in this grid (15.9%), but at the cost of a higher average corruption error (25.6%), roughly 2–3 percentage points above the dual-EMA value. Even when sampling single-EMA rates as fine as $\alpha = 0.99985$, we do not find a single rate that matches the dual-EMA corruption–clean trade-off in this configuration. Separate EMA rates help in this CIFAR-100-C experiment. More datasets and hyperparameter settings are needed to test whether the benefit holds more generally.

E.2 Batch-Size Robustness

We examine the effect of batch size on EATA+SCALAR-U under persistent class skew over five consecutive passes (Tab. 14). The same 50 classes remain dominant throughout the stream. We vary the batch size while keeping the window size fixed at $N = 1024$, with learning rate $\eta = 10^{-5}$ and temperature $\tau = 0.3$. This is a separate configuration from the persistent-skew experiment in Tab. 5, which uses $N = 4096$, $\tau = 0.7$, and $\eta = 3 \times 10^{-4}$; the absolute errors are therefore not directly comparable.

SCALAR remains stable across the tested batch sizes, including batches of two samples. Larger batches show modestly lower corruption and post-stream clean errors in this setting.

Table 14: Batch-size sensitivity on ImageNet-C (severity 5) under five-pass persistent skew. Error (%), mean ± standard deviation over three runs.

Batch size	Avg. error	Final clean	Coll.
2	49.2 ± 1.2	20.3 ± 0.3	0/3
4	47.7 ± 0.6	19.9 ± 0.5	0/3
8	46.0 ± 0.8	19.3 ± 0.1	0/3

Table 15: Sensitivity to the mixing weight under class skew on ImageNet-C (severity 5). Each entry reports average corruption error / post-stream clean error (%).

Dominant classes	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 0.75$
25	50.7 / 21.6	43.9 / 18.5	45.8 / 18.0
50	49.6 / 19.4	48.4 / 18.8	48.7 / 18.1
200	48.9 / 20.3	46.7 / 19.2	46.6 / 18.3
Mean	49.7 / 20.4	46.3 / 18.9	47.0 / 18.1

E.3 Sensitivity to the Mixing Weight under Class Skew

We also examine the sensitivity to the SCALAR mixing weight λ under three levels of class skew, with 25, 50, and 200 dominant classes (Tab. 15). We use a window size of $N = 1024$ and report a single run per configuration.

The choice of mixing weight affects corruption and clean error differently. Averaged across the tested skew levels, $\lambda = 0.5$ gives the lowest corruption error, while $\lambda = 0.75$ gives the lowest post-stream clean error. Each configuration uses one run, so these comparisons do not establish a precise optimum.

E.4 Comparison with Target Sharpening

We compare SCALAR with a target-sharpening baseline in the fixed-skew, high-speed ImageNet-C setting of Tab. 2. The sharpened variant replaces the balanced targets $\tilde{q}_u$ in Eq. (2) with $\mathrm{softmax}(z_u/\tau)$, keeping the same warm-up length, temperature $\tau = 0.1$, and mixing weight $\lambda = 0.5$. The sharpening baseline is evaluated over three independent runs under matched adaptation conditions. The results are shown in Tab. 16.

Sharpening does not give a consistent improvement over the baseline, with both methods collapsing in two of the three runs. SCALAR-U gives lower corruption and final clean errors than sharpening in this comparison, suggesting that sharpening targets alone does not account for the benefit of balancing across the window.

F Computational Overhead

SCALAR converts an $N \times C$ FIFO queue of teacher logits to probabilities and applies K iterations of Sinkhorn row/column scaling. Ignoring the lower-order cost of the per-step softmax on the current mini-batch (typically negligible when $N \gg B$), each Sinkhorn iteration consists of two reductions and two elementwise normalization passes: *row normalization* $r = Q\mathbf{1}_C$, $Q \leftarrow Q \oslash r$, followed by *column scaling* $c = \mathbf{1}_N^\top Q$, $s = t \oslash c$, $Q \leftarrow Q \odot s$, where $\mathbf{1}_C$ and $\mathbf{1}_N$ are all-ones vectors, t encodes the target column marginals (e.g., $t = N\pi$ for a prior π), and $\oslash$ / $\odot$ denote broadcast division / multiplication.

Each reduction and elementwise pass touches all NC entries, so one iteration costs $\Theta(NC)$ scalar operations. Over K iterations, the total complexity is

$$\mathcal{O}(KNC).$$

These operations act on the prediction matrix and do not require another forward or backward pass through the model.

Table 16: Comparison with target sharpening on fixed-skew ImageNet-C (severity 5, high speed). Average corruption error and final clean error (%), mean ± standard deviation over three runs. Coll. indicates collapsed runs. Baseline and SCALAR-U results are from Tab. 2.

Method	Avg. error	Final clean	Coll.
Baseline (w/o SCALAR)	86.8 ± 14.4	76.4 ± 40.6	2/3
Sharpening	90.7 ± 6.5	74.7 ± 43.3	2/3
SCALAR-U	48.1 ± 1.1	21.1 ± 0.5	0/3

Table 17: **Backbone vs. SCALAR module overhead.** Compute is per forward pass for ViT-B/16 and per Sinkhorn-based CTTA update for SCALAR. SCALAR assumes $C = 1000$ classes (ImageNet), window size $N = 1024$, and $K = 5$ Sinkhorn iterations, as used in our experiments. For CIFAR-10/100 ($C \in \{10, 100\}$) the overhead is another 10–100× smaller.

Component	Compute (GFLOPs)	Memory (MB)
ViT-B/16 backbone (forward)	≈ 17.5	≈ 330.3
SCALAR (ImageNet, FP32)	≈ 0.02	≈ 4 (fp32 queue)

Memory footprint. The memory cost is dominated by storing the logit queue Q_{mem}, i.e., NC floating-point entries, giving $\mathcal{O}(NC)$ memory. In our ImageNet experiments we use C=1000 classes and a window N≈1000 (e.g., N=1024), so Q_{mem} contains about 10^6 entries: ≈ 2 MB in fp16 (or ≈ 4 MB in fp32), plus negligible row/column buffers. For CIFAR-10/100, where $C \in \{10, 100\}$, both computation and memory scale linearly with C and are correspondingly smaller.

Compute relative to the backbone. For the window size and iteration count in Tab. 17, the estimated compute cost is below 1% of a ViT-B/16 forward pass. We report measured memory overhead below.

Measured overhead. On ImageNet-C (C=1000, window=1024) with micro-batch size B=8 on an NVIDIA RTX 3080, per-step runtime is unchanged within measurement noise (0.1063s → 0.1062s) and peak GPU memory increases by 10MB (`peak_alloc`) / 20MB (`peak_reserved`). These measurements use our FP32 implementation, which includes redundant softmax operations. Removing them may reduce the overhead further.

G Construction of Skewed Streams

We construct label-skewed ImageNet-C streams by controlling the class composition of each corruption segment. The stream stays skewed over extended intervals, allowing prediction biases to accumulate during adaptation.

For each corruption segment, we form a head–tail class split and resample examples according to this split. Head-class examples are oversampled with replacement to a fixed number of samples per head class, while tail classes are subsampled by keeping only a small number of examples per tail class. In our experiments, tail classes are subsampled to one example per class. This preserves the corruption sequence while making head classes more frequent. We generate the sample indices deterministically and reuse them across methods.

Fixed vs. dynamic label skew. In the *fixed-skew* setting, the skew strength and head-set size are fixed across corruption segments. Specifically, we use 50 head classes and 950 tail classes, oversampling each head class to 100 samples while keeping one example per tail class. The dominant class identities are selected deterministically for each segment, so all methods are evaluated on the same generated stream.

In the *dynamic-skew* setting, the number of head classes and samples per head class vary across corruption segments. We use a cyclic schedule in which the number of head classes follows

$$(30, 150, 20, 200, 40, 100),$$

and the corresponding number of samples per head class follows

$$(170, 33, 250, 25, 125, 50).$$

Tail classes are again subsampled to one example per class. This changes the per-class head–tail ratio over time, from 25:1 up to 250:1, while keeping the aggregate head mass roughly comparable across segments. Class identities are deterministically reshuffled across segments, and once the schedule is exhausted, it is repeated from the beginning.

Both streams change which classes dominate. The dynamic stream also changes how many classes dominate and how much more often they appear.

Collapse criterion. We count a run as collapsed when its post-stream clean error exceeds 95%.

Oracle prior. SCALAR-C uses the empirical class distribution of each corruption segment as its target prior π, recomputed for every segment from the generated stream. This oracle is used only as a reference; no sample-level labels enter the adaptation procedure.

Long-horizon persistent skew. For the five-pass experiment in Tab. 5, we use the same 50 dominant classes throughout the entire stream. Within each corruption segment, each dominant class contributes 100 samples and each of the remaining 950 classes contributes one sample. A pass consists of all 15 ImageNet-C corruptions processed sequentially, with no model reset between corruptions or passes; each pass contains approximately 89K samples and the full stream approximately 446K samples. Unlike the fixed-skew setting, where the dominant class identities change across segments, here the same dominant set persists throughout. The oracle prior in this experiment is the aggregate class distribution of the stream.

True-subset stream. To isolate persistent class absence, we construct a stricter variant in which only the same 50 ImageNet classes appear throughout the stream, and the remaining 950 classes are completely absent. The subset is held fixed across all corruption segments and five consecutive passes, again without model reset. The mismatch between the uniform target and the stream prior therefore persists across all five passes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main scope and claims: SCALAR regulates temporal pseudo-label class marginals for CTTA, is intended as a lightweight plug-in stabilization mechanism, and is primarily beneficial under skewed, non-stationary, and aggressive-adaptation regimes. The paper also clarifies that SCALAR is not designed to recover the true test-time class prior, and that standard balanced benchmarks mainly demonstrate competitive performance rather than the largest gains.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations and future directions in the conclusion. In particular, it notes that the evaluation focuses mainly on image classification and controlled skewed streams constructed from corruption benchmarks, and that broader validation on naturally occurring non-i.i.d. deployment streams remains important. It further identifies extension to dense prediction tasks such as detection and segmentation as future work. It also notes that a uniform target may mildly over-regularize when adaptation is already stable, particularly under persistent class absence or genuinely long-tailed distributions. Computational cost and wall-clock overhead are discussed in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results requiring assumptions or proofs. It includes the SCALAR algorithm, the Sinkhorn balancing equations, and the optimization objective, but the contributions are methodological and empirical rather than theorem-based.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the information needed to reproduce the main experiments, including datasets, corruption severity, CTTA protocol, model/backbone choices, adaptation settings, SCALAR hyperparameters, construction of fixed and dynamic skewed streams, evaluation metrics, and computational overhead. Additional implementation details and hyperparameter settings are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a code package with the SCALAR implementation, fixed/dynamic skew-stream construction utilities, configuration files, and example scripts for reproducing representative main experiments, including the standard ImageNet-C setting and the EATA+SCALAR dynamic-skew setting. The experiments use public datasets and pretrained checkpoints; the README describes the required environment, data/checkpoint placement, and example commands. Large datasets and pretrained checkpoints are not redistributed.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the CTTA protocol, datasets and corruption severity, standard and skewed stream construction, model backbones and initialization, evaluation metrics, baselines and variants, adaptation speeds, optimizer/update settings, and SCALAR hyperparameters. Additional implementation details, augmentation settings, and skew construction details are provided in the appendix.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For the standard balanced corruption benchmarks, SCALAR and the teacher–student baseline are averaged over 5 seeds (standard deviation of the average error $\approx$0.2 percentage points). For the skewed-stream adaptation-speed, long-horizon, and TENT experiments, we report mean $\pm$ SD over three runs (for EATA, three at the stable operating point and eight under aggressive adaptation), together with the number of collapsed runs; the calibration experiment uses three runs and the true-subset experiment uses two runs. For TENT, which is deterministic given the input stream, runs differ in the random ordering of samples within each corruption segment, and methods are paired on identical orderings. Error bars denote one sample standard deviation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the GPU used for wall-clock and memory measurements (NVIDIA RTX 3080), the micro-batch size, per-step runtime, and peak memory overhead for SCALAR. Together with the reported stream sizes and batch sizes, these measurements provide enough information to estimate per-run compute requirements. The paper also reports the algorithmic compute and memory complexity of the SCALAR module and compares its overhead to the ViT-B/16 backbone in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses standard public image classification benchmarks and pre-trained vision backbones, does not involve human subjects, private data, or high-risk data collection, and is conducted as methodological work on test-time adaptation. To the best of our knowledge, it conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader-impact paragraph discusses potential reliability benefits of reducing adaptation-induced errors and the risk of worse performance under genuine class imbalance, together with the need for validation on the intended population and monitoring during deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work does not release high-risk models or datasets. It uses standard public image classification benchmarks and pretrained vision backbones, so special safeguards are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original works for the datasets, benchmarks, pretrained vision backbones, and baseline methods used in the experiments. The released code package and README specify that the experiments rely on public datasets and pretrained checkpoints, and users are instructed to obtain these assets from their original sources and follow their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides a code package as a new asset. The package includes the SCALAR implementation, skew-stream construction utilities, configuration files, and example scripts for representative experimental settings. The accompanying README documents the required environment, public datasets and checkpoints, expected file locations, and example commands.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, data collection from human participants, or research with human subjects. The experiments use standard public image classification benchmarks and pretrained vision models.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, data collection from human participants, or research with human subjects. The experiments use standard public image classification benchmarks and pretrained vision models, so IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The research does not use LLMs as part of the core method, experiments, or scientific claims.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.