CTSC: CROSS-TASK SCORE CALIBRATION FOR CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-incremental learning asks a model to acquire new classes over a sequence of tasks while recognizing all seen classes without task identity at test time. This global prediction setting depends on comparable scores across task blocks, yet sequential training can leave those blocks on incompatible scales. We study this cross-task score problem in hybrid ETF/classifier models and propose CTSC, which combines progressive equiangular tight frame (ETF) anchors and feature preservation with classifier standardization before score fusion and additive task-bias fitting afterward. Pre-fusion standardization controls the classifier contribution relative to ETF evidence, whereas post-fusion task offsets adjust cross-task competition while preserving within-task rankings. Across Seq-CIFAR-10, Seq-CIFAR-100, and Seq-TinyImageNet with replay budgets of 200 and 500, CTSC improves Class-IL recognition and reduces forgetting under matched local evaluation. Fixed-checkpoint interventions show that task offsets recover cross-task errors without changing within-task predictions, and that pre-fusion standardization and post-fusion correction are not interchangeable. Calibration-pool and task-prior interventions characterize when replay-fitted offsets transfer to the test distribution.

1 INTRODUCTION

Class-incremental learning requires recognizing all classes encountered in a stream with limited access to earlier data. New learning can disrupt representations and classifier decisions, causing catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999; Kirkpatrick et al., 2017). Yet an example can remain correctly classified within its original task while losing to a class introduced later. This gap points to cross-task score competition as a distinct source of error alongside representation instability (Rebuffi et al., 2017; Hou et al., 2019; Lanzillotta et al., 2026).

Cross-task score comparability is central to Class-IL inference: even when old-class features remain useful, the scores assigned to different task blocks can drift onto incompatible scales. Hyperspherical continual learning makes this issue visible. Equiangular tight frame (ETF) targets organize class directions, drawing on Neural Collapse (Papyan et al., 2020), and anchors can remain fixed or expand as classes arrive (Yang et al., 2023; Seo et al., 2024; Zhou et al., 2025; Wang et al., 2026). This geometry improves directional structure, but separated anchors alone do not make all class scores comparable once anchor scores are fused with a learned classifier.

Existing approaches correct classifier bias through learned correction layers, weight normalization or alignment, and separated softmax competition (Wu et al., 2019; Hou et al., 2019; Zhao et al., 2020; Ahn et al., 2021). In hybrid ETF/classifier models, score operations before and after fusion are not interchangeable. Pre-fusion standardization controls the classifier branch before it mixes with ETF evidence and can change the final within-task ordering. Post-fusion task correction shifts a whole task block after hybrid scores are formed, preserving within-task rankings while changing cross-task competition.

CTSC is built around this pre-fusion versus post-fusion distinction. Figure 1 summarizes the failure mode and the resulting design: representation geometry is preserved, and score calibration is applied at the stages where cross-task competition is formed.

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

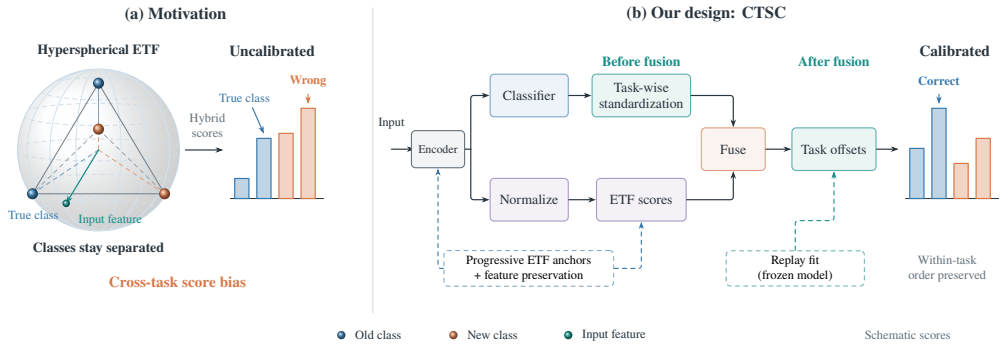


Figure 1: Motivation and design of CTSC. (a) Separated ETF class directions can coexist with a higher competing new-task score. (b) CTSC preserves feature geometry through progressive ETF anchors and feature preservation, standardizes classifier scores before ETF fusion, and fits task offsets on replay examples with the model frozen. The post-fusion offsets adjust cross-task competition while preserving within-task rankings. Bar heights are schematic.

We propose CTSC, a calibrated hyperspherical continual learning method that separates representation constraints from score calibration. Progressive ETF anchors and feature preservation maintain hyperspherical structure. Input-dependent classifier standardization sets the classifier contribution before ETF fusion, and additive task correction calibrates cross-task competition after fusion with the model fixed. Inference compares all seen classes using known class-to-task assignments and does not require the input's task identity. We further analyze how fitted offsets transfer across calibration distributions.

On Seq-CIFAR-10, Seq-CIFAR-100, and Seq-TinyImageNet, CTSC improves Class-IL final average accuracy over locally reproduced ProNC (Wang et al., 2026), a strong matched baseline, by 1.81 and 1.45 points on average with replay budgets of 200 and 500, while reducing Class-IL forgetting in all matched dataset–buffer settings. Fixed-checkpoint recovery and placement analyses connect these gains to cross-task score competition. Calibration-pool and task-prior interventions identify when replay-fitted offsets transfer to the test distribution.

The paper makes three contributions:

- We isolate bias-correctable cross-task errors in a frozen hyperspherical model, separating score correction from representation changes.
- We introduce CTSC, which separates pre-fusion classifier standardization from post-fusion task correction in hybrid ETF/classifier scores across three datasets and two replay budgets.
- We characterize how fusion placement, calibration sample source, and task frequencies affect recognition through fixed-checkpoint interventions.

2 Related Works

2.1 Continual Learning and Representation Forgetting

Continual learning methods address forgetting through regularization, replay, or architectural isolation (De Lange et al., 2022; Masana et al., 2023). Regularization constrains changes to parameters or functions (Kirkpatrick et al., 2017; Li & Hoiem, 2018; Aljundi et al., 2018), while replay methods preserve past information through exemplars or logit distillation, including iCaRL, GEM, ER, and DER++ (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Buzzega et al., 2020). These methods reduce information loss during training, but Class-IL recognition also depends on whether scores from different tasks remain comparable in a single softmax. A model can retain within-task discrimination and still assign systematically higher scores to classes from recent tasks. Recent analyses separate representation shift from classifier-head failure (Kim et al.,

2025; Lanzillotta et al., 2026). CTSC targets this score-comparability bottleneck in a setting where hyperspherical features preserve within-task structure but cross-task logits remain misaligned.

2.2 Neural Collapse and ETF Geometry

Neural Collapse and hyperspherical methods make class geometry explicit through concentrated features, aligned classifier weights, and simplex equiangular tight frame (ETF) prototypes (Papyan et al., 2020; Zhu et al., 2021). ETF-based continual-learning methods use this geometry to provide uniformly separated class anchors, including fixed or progressively expanded targets as classes arrive (Yang et al., 2023; Seo et al., 2024; Wang et al., 2026). ETF geometry stabilizes representation learning and gives classes separated directions, but separated anchors alone do not guarantee calibrated hybrid scores after sequential training. CTSC couples ETF geometry with classifier standardization before score fusion and task-offset calibration afterward, jointly controlling representation geometry and cross-task score competition.

2.3 Classifier Bias and Logit Calibration

Classifier-bias methods adjust scores through class statistics, bias correction, normalization, or separated softmax (Belouadah & Popescu, 2019; Wu et al., 2019; Hou et al., 2019; Zhao et al., 2020; Ahn et al., 2021). BiC fits an affine correction for new-class logits on validation data (Wu et al., 2019); WA aligns new-class weight norms with old-class norms (Zhao et al., 2020); SS-IL separates training-time softmax competition and uses task-wise distillation (Ahn et al., 2021). These methods show that score bias is central to class-incremental recognition, with most corrections acting on old-new imbalance, class statistics, or training-time softmax structure.

Confidence calibration adjusts predicted probabilities (Guo et al., 2017), while class-dependent logit adjustment changes decision boundaries under class-prior shifts (Menon et al., 2021; Ren et al., 2020). Here, calibration denotes score correction for cross-task recognition rather than probability reliability alone. CTSC targets hybrid ETF/classifier scores by normalizing the classifier branch before ETF fusion and fitting offsets for all seen tasks after fusion. Unlike shared affine old-new correction or weight-norm alignment, the pre-fusion operation changes the classifier/ETF tradeoff, whereas additive offsets preserve the resulting within-task ordering. CTSC treats representation geometry and cross-task score calibration as complementary parts of Class-IL inference.

3 Methodology

CTSC couples progressive hyperspherical representation learning with cross-task score calibration. Backbone features feed a learned classifier and a normalized ETF branch (Figure 2). The method separates representation constraints from score calibration: anchors and feature preservation stabilize the hyperspherical representation, while pre-fusion classifier standardization controls the classifier contribution to the hybrid score and post-fusion offsets adjust competition between task blocks without changing within-task rankings.

3.1 Problem Setting

The model receives a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$. Task $\mathcal{T}_t$ introduces a disjoint class set $\mathcal{C}_t$ and training data $\mathcal{D}_t = \{(x_i, y_i)\}$ with $y_i \in \mathcal{C}_t$. After task t, prediction is over $\mathcal{C}_{\leq t} = \bigcup_{u=1}^{t} \mathcal{C}_u$, with $K_t = |\mathcal{C}_{\leq t}|$. A buffer $\mathcal{M}_t$ stores up to M exemplars from the seen classes. We write $\tau(c)$ for the known task assignment of class c. Class-IL inference evaluates all seen classes and requires no task identity for the input. The evaluated streams introduce equal-sized class blocks, which form the default task-frequency setting for calibration.

3.2 Normalized Features and ETF Anchors

Given an input x, the backbone produces a feature vector h_t, which is normalized for anchor scoring and alignment:

$$h_t = f_\theta(x), \qquad \bar{h}_t = \frac{h_t}{\max(\|h_t\|_2, \epsilon)}. \tag{1}$$

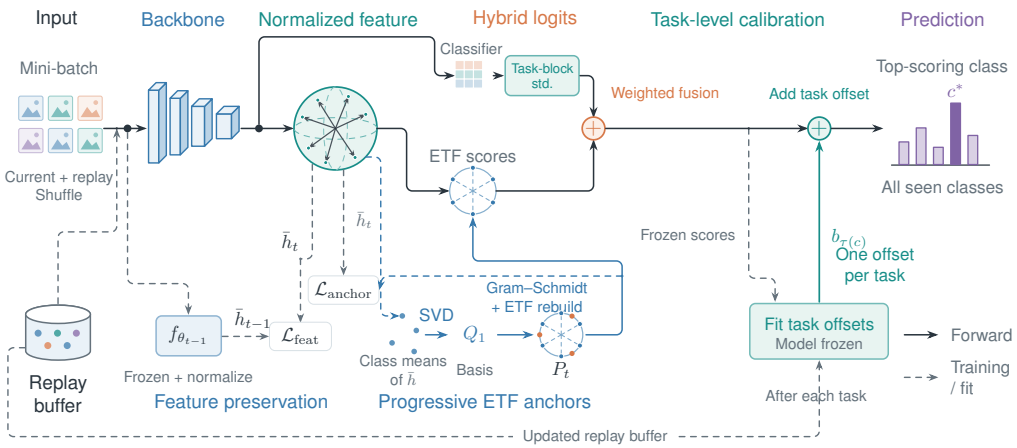


Figure 2: Overview of CTSC. Backbone features are normalized on a hypersphere and aligned with an ETF frame initialized by SVD and expanded by Gram–Schmidt. Current-task data are naturally concatenated with replay exemplars, hybrid classifier/anchor logits provide class scores, and replay-buffer task calibration adjusts cross-task offsets without changing within-task class rankings.

Normalization makes the ETF score depend on feature direction rather than magnitude. Each seen class has an anchor $p_c \in \mathbb{R}^d$ in a progressively expanded ETF (Wang et al., 2026). With an orthonormal basis and $2 \le K_t \le d$, the simplex ETF satisfies, up to numerical precision,

$$\|p_c\|_2 = 1, \qquad p_i^\top p_j = -\frac{1}{K_t - 1} \quad (i \neq j). \tag{2}$$

The first task trains without an ETF alignment loss or ETF scores. At its end, CTSC averages normalized backbone features within each class, centers and normalizes these class means, and obtains an orthonormal basis Q_1 through an SVD fit (Appendix A.2). Before training task $t > 1$, Gaussian candidate directions are added to the existing basis $Q_{t-1} \in \mathbb{R}^{d \times K_{t-1}}$. Each candidate v_j is orthogonalized against the old basis and the new directions already accepted:

$$\hat{q}_j = v_j - Q_{t-1} Q_{t-1}^\top v_j - \sum_{\ell < j} q_\ell q_\ell^\top v_j, \qquad q_j = \frac{\hat{q}_j}{\|\hat{q}_j\|_2}. \tag{3}$$

Appending the accepted directions gives Q_t, from which we rebuild the active K_t-class ETF:

$$P_t = Q_t \left(I - \frac{1}{K_t} \mathbf{1}\mathbf{1}^\top \right) \sqrt{\frac{K_t}{K_t - 1}}, \tag{4}$$

followed by column normalization. Rebuilding the centered frame changes old anchors as well as adding new ones, even though the old basis directions are retained. The resulting P_t is fixed during task t; it supplies the geometric branch whose scores are later fused with the standardized classifier branch and calibrated across tasks.

3.3 Hybrid Logits

The learned affine classifier operates on the unnormalized backbone feature:

$$a_c(x) = w_c^\top h_t + \beta_c. \tag{5}$$

To control its contribution to the hybrid score, classifier logits are standardized separately within each seen task block. For $u = \tau(c)$, define

$$\mu_u(x) = \frac{1}{|\mathcal{C}_u|} \sum_{j \in \mathcal{C}_u} a_j(x), \tag{6}$$

$$\sigma_u(x) = \sqrt{\frac{1}{|\mathcal{C}_u|} \sum_{j \in \mathcal{C}_u} (a_j(x) - \mu_u(x))^2}, \tag{7}$$

$$\hat{a}_c(x) = \frac{a_c(x) - \mu_{\tau(c)}(x)}{\max(\sigma_{\tau(c)}(x), \epsilon_{\text{cls}})}. \tag{8}$$

The implementation uses $\epsilon_{\text{cls}} = 10^{-6}$. These statistics are computed separately for each input across the classes within a task block. The anchor branch scores the same input by cosine alignment:

$$e_c(x) = \bar{h}_t^\top p_c. \tag{9}$$

The pre-bias hybrid score combines the standardized classifier and ETF branches:

$$r_c(x) = \lambda_{\text{cls}} \hat{a}_c(x) + \lambda_{\text{etf}} e_c(x), \qquad c \in \mathcal{C}_{\leq t}. \tag{10}$$

All six main configurations use $\lambda_{\text{cls}} = 0.2$ and $\lambda_{\text{etf}} = 1$. Standardization preserves classifier-block rankings but changes the classifier/ETF tradeoff and may alter their combined ranking. Task bias and post-fusion block standardization preserve the existing hybrid within-task ranking, so they cannot reproduce this effect (Appendix A.5). No further standardization is applied to r_c. Appendix A.6 analyzes the information retained by classifier standardization.

3.4 Replay-Buffer Task Calibration

Classifier standardization controls the branch before fusion, but the resulting hybrid scores can still be misaligned across task blocks. After training task t, CTSC updates $\mathcal{M}_t$ to include current and earlier classes, freezes the model, and fits one bias per seen task on the replay-buffer calibration distribution. For a model input x, the final logit is

$$s_c(x) = r_c(x) + b_{\tau(c)}. \tag{11}$$

Let P_{raw} map stored buffer tensors to model inputs scaled to $[0, 1]$ without dataset normalization or random augmentation. We fit the empirical calibration objective with representations and classifier weights fixed:

$$\min_{b: \sum_s b_s = 0} L_{\text{raw}}(b), \qquad L_{\text{raw}}(b) = \frac{1}{|\mathcal{M}_t|} \sum_{(x,y) \in \mathcal{M}_t} -\log \frac{\exp(r_y(P_{\text{raw}}x) + b_{\tau(y)})}{\sum_{c \in \mathcal{C}_{\leq t}} \exp(r_c(P_{\text{raw}}x) + b_{\tau(c)})}. \tag{12}$$

We initialize the biases to zero and fit them with 80 Adam steps at learning rate 0.2. Subtracting their mean after optimization removes the common-offset ambiguity without changing softmax probabilities. At test time, scores are $r_c(P_{\text{norm}}x) + b_{\tau(c)}$, where P_{norm} applies the deterministic test preprocessing including dataset normalization. Section 4.5 compares this fitting target with normalized and independent calibration variants.

For fixed hybrid scores, a shared task bias adjusts cross-task competition while preserving within-task rankings and Task-IL predictions. The fitted offsets depend jointly on frozen scores and empirical task frequencies. At a finite optimum, average predicted task probability matches those frequencies, not hard prediction counts. The final calibration pools in our reliability runs have equal task counts; Appendix B.6 tests departures from this setting. Appendix A.4 gives the probability decomposition, optimization properties, and proofs, and Appendix A.1 specifies the fitting procedure.

3.5 Replay and Training Objective

Before training task $t > 1$, CTSC concatenates the replay buffer with the current-task dataset,

$$\widetilde{\mathcal{D}}_t = \mathcal{D}_t \parallel \mathcal{M}_{t-1}, \tag{13}$$

and shuffles $\widetilde{\mathcal{D}}_t$ in a single data loader, without task-stratified sampling or a separate replay mini-batch. Within a sampled mini-batch $\mathcal{B}$, we distinguish current-class and replay-class subsets, $\mathcal{B}^{\text{new}}$ and $\mathcal{B}^{\text{rep}}$. To decouple loss weighting from replay sampling frequency, we use an age-balanced empirical average for the CE terms. When both subsets are nonempty, the average of per-example losses $\{\ell_i\}_{i\in\mathcal{B}}$ is

$$\text{Bal}_\rho(\ell;\mathcal{B}) = \frac{\frac{1}{|\mathcal{B}^{\text{new}}|}\sum_{i\in\mathcal{B}^{\text{new}}}\ell_i + \rho_{\text{rep}}\frac{1}{|\mathcal{B}^{\text{rep}}|}\sum_{i\in\mathcal{B}^{\text{rep}}}\ell_i}{1+\rho_{\text{rep}}}. \tag{14}$$

If a batch contains only one subset, we use its empirical mean. The weighting controls each group's contribution to the loss; replay frequency is determined by sampling from the concatenated dataset.

CTSC applies this operator to the hybrid-logit and learned-classifier cross-entropies:

$$\mathcal{L}_{\text{cls}} = \lambda_{\text{hyb}}\,\text{Bal}_\rho(\ell_{\text{ce}}(s(x),y);\mathcal{B}) + \lambda_{\text{head}}\,\text{Bal}_\rho(\ell_{\text{ce}}(a(x),y);\mathcal{B})\,. \tag{15}$$

The classifier CE uses raw logits $a(x)$ over the full dataset-sized output head, whereas hybrid CE uses seen-class scores. All main configurations use $\lambda_{\text{head}} = 1$. The auxiliary hybrid CE weight λ_{hyb} is 0.1 except for Seq-TinyImageNet with buffer 500, where it is 0.5. When this term is active, $s(x)$ uses the stored biases from the previous task, with zero bias for newly introduced classes; these offsets stay fixed during training and are refitted after the task. Run-specific weights are listed in Appendix B.

When a previous checkpoint is available, feature preservation constrains feature direction on the hypersphere using the sampled augmented batch:

$$\mathcal{L}_{\text{feat}} = \frac{1}{2|\mathcal{B}|}\sum_{x\in\mathcal{B}}\left(\bar{h}_t(x)^\top \bar{h}_{t-1}(x) - 1\right)^2. \tag{16}$$

The anchor loss pulls normalized features toward their class anchors:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{2|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}}\left(\bar{h}_t(x)^\top p_y - 1\right)^2. \tag{17}$$

The full training loss is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}}. \tag{18}$$

The feature loss uses a frozen previous-task backbone on the same augmented inputs and averages over both current and replay examples. The anchor loss also uses an ordinary sample mean; only the CE terms use Eq. 14, with $\rho_{\text{rep}} = 1$. Feature preservation and anchor alignment are inactive during first-task training because the previous backbone and ETF are not yet available. All main configurations use $\lambda_{\text{anchor}} = 20$ and disable logit distillation. After each task, a frozen backbone copy is retained, the buffer keeps up to $\lfloor M/K_t \rfloor$ exemplars per seen class with random selection for new classes, and task biases are fitted on the updated buffer. The training loss therefore preserves feature directions without requiring old logit targets.

4 Experiments

We compare CTSC with locally reproduced ProNC under the same streams and replay budgets, use the remaining ProNC baselines as benchmark context, and analyze score calibration through ablations and fixed-checkpoint diagnostics.

4.1 Experimental Setup

Datasets and tasks Following the ProNC benchmark protocol (Wang et al., 2026), we evaluate Seq-CIFAR-10, Seq-CIFAR-100 (Krizhevsky, 2009), and Seq-TinyImageNet (Le & Yang, 2015; Deng et al., 2009) with 5, 10, and 10 disjoint tasks containing 2, 10, and 20 classes each, respectively. The replay buffer stores 200 or 500 exemplars in total. Class-IL predicts among all seen classes without a task identifier; Task-IL restricts prediction to the classes of the true task.

Table 1: Main comparison. Each cell reports multi-run average FAA (FF), and Avg. averages the three datasets. ProNC and CTSC are locally reproduced; other baseline values follow ProNC Table 1 (Wang et al., 2026). Error terms appear in Appendix B.1–B.2.

Buffer	Method	Seq-CIFAR-10		Seq-CIFAR-100		Seq-TinyImageNet		Avg.	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
200	ER	44.79 (59.30)	91.19 (6.07)	21.78 (75.06)	60.19 (27.38)	8.49 (76.53)	38.17 (40.47)	25.02 (70.30)	63.18 (24.64)
	iCaRL	49.02 (23.52)	88.99 (25.34)	28.00 (47.20)	51.43 (36.20)	7.53 (31.06)	28.19 (42.47)	28.18 (33.93)	56.20 (34.67)
	GEM	25.54 (80.36)	90.44 (9.57)	20.75 (77.40)	58.84 (29.59)	–	–	–	–
	GSS	39.07 (72.48)	88.80 (8.49)	19.42 (77.62)	55.38 (32.81)	–	–	–	–
	DER	61.93 (35.79)	91.40 (6.08)	31.23 (62.72)	63.09 (25.98)	11.87 (64.83)	40.22 (40.43)	35.01 (54.45)	64.90 (24.16)
	DER++	64.88 (32.59)	91.92 (5.16)	28.13 (60.99)	66.80 (23.91)	11.34 (73.47)	43.06 (39.02)	34.78 (55.68)	67.26 (22.70)
	LODE	68.01 (24.63)	93.11 (4.75)	26.65 (44.29)	71.23 (18.75)	15.13 (64.00)	51.42 (29.66)	36.60 (44.31)	71.92 (17.72)
	Co2L	51.27 (30.17)	84.69 (2.91)	18.09 (64.14)	49.19 (27.83)	12.95 (62.04)	38.40 (40.75)	27.44 (52.12)	57.43 (23.83)
	CILA	59.68 (37.52)	91.36 (5.89)	19.49 (64.01)	53.93 (33.07)	12.98 (63.11)	37.32 (41.40)	30.72 (54.88)	60.87 (26.79)
	MNC3L	51.09 (33.74)	85.07 (4.90)	15.81 (62.51)	43.91 (39.79)	10.57 (59.68)	32.78 (45.10)	25.82 (51.98)	53.92 (29.93)
	STAR	65.94 (15.99)	95.12 (2.06)	38.15 (42.17)	79.53 (17.32)	13.64 (68.51)	43.01 (39.16)	39.24 (42.22)	72.55 (19.51)
	CSReL	37.46 (26.34)	69.22 (17.16)	29.06 (58.23)	66.99 (23.20)	18.14 (49.77)	45.04 (34.12)	28.22 (44.78)	60.42 (24.83)
	NCT	51.59 (22.48)	80.63 (1.41)	26.38 (27.40)	75.75 (4.79)	10.95 (49.33)	52.71 (15.88)	29.64 (33.07)	69.70 (7.36)
	ProNC	65.47 (32.98)	96.82 (0.72)	42.46 (36.59)	85.13 (4.58)	27.48 (42.85)	68.89 (9.85)	45.14 (37.47)	83.61 (5.05)
	CTSC	**66.86 (24.83)**	**96.82 (0.70)**	**44.42 (27.64)**	**85.71 (3.89)**	**29.57 (39.92)**	**71.51 (6.83)**	**46.95 (30.80)**	**84.68 (3.81)**
500	ER	57.74 (43.22)	93.61 (3.50)	22.35 (73.08)	73.98 (16.23)	9.99 (75.21)	48.64 (30.73)	30.03 (63.84)	72.08 (16.82)
	iCaRL	47.55 (28.20)	88.22 (22.61)	33.25 (40.99)	58.16 (27.90)	9.38 (37.30)	31.55 (39.44)	30.06 (35.50)	59.31 (29.98)
	GEM	26.20 (78.93)	92.16 (5.60)	25.54 (71.34)	66.31 (20.44)	–	–	–	–
	GSS	49.73 (59.18)	91.02 (6.37)	21.92 (74.12)	60.28 (26.57)	–	–	–	–
	DER	70.51 (24.02)	93.40 (3.72)	41.36 (49.07)	71.73 (25.98)	17.75 (59.95)	51.78 (28.21)	43.21 (44.35)	72.30 (19.30)
	DER++	72.70 (22.38)	93.88 (4.66)	38.20 (49.18)	74.77 (15.75)	19.38 (58.75)	51.91 (25.47)	43.43 (43.44)	73.52 (15.29)
	LODE	75.91 (18.18)	94.19 (3.94)	40.01 (32.58)	80.06 (8.96)	20.50 (56.51)	61.49 (18.61)	45.47 (35.76)	78.58 (10.50)
	Co2L	61.78 (17.79)	89.51 (2.65)	26.64 (48.60)	62.32 (23.47)	18.71 (49.64)	50.74 (13.80)	35.71 (38.68)	67.52 (13.31)
	CILA	67.82 (18.22)	93.29 (0.65)	31.27 (45.67)	68.29 (16.98)	18.09 (64.14)	49.19 (27.83)	39.06 (42.68)	70.26 (15.15)
	MNC3L	52.20 (27.88)	85.94 (3.15)	22.29 (46.09)	56.43 (24.29)	11.52 (44.96)	36.32 (39.00)	28.67 (39.64)	59.56 (22.15)
	STAR	69.19 (12.21)	95.36 (2.38)	47.56 (28.68)	78.28 (15.43)	21.31 (57.20)	59.32 (19.49)	46.02 (32.70)	77.65 (12.43)
	CSReL	40.82 (29.18)	73.30 (16.89)	39.22 (47.03)	75.09 (15.55)	22.13 (50.01)	51.94 (28.20)	34.06 (42.07)	66.78 (20.21)
	NCT	60.93 (13.82)	81.27 (3.84)	33.77 (27.85)	75.87 (5.06)	18.24 (50.90)	62.30 (6.84)	37.65 (30.86)	73.15 (5.25)
	ProNC	73.72 (26.97)	96.38 (0.38)	48.15 (24.78)	85.36 (4.79)	29.10 (38.64)	69.98 (9.82)	50.32 (30.13)	83.91 (5.00)
	CTSC	**75.02 (24.35)**	**96.27 (0.93)**	**49.38 (22.84)**	**86.62 (3.91)**	**30.91 (38.01)**	**71.23 (7.59)**	**51.77 (28.40)**	**84.71 (4.14)**

Training CTSC uses ResNet-18 (He et al., 2016) and SGD with batch size 32. Training inputs use random cropping, horizontal flipping, and dataset-specific normalization; post-task bias fitting uses raw replay tensors without augmentation. Seq-CIFAR-10 and Seq-CIFAR-100 are trained for 50 epochs per task, and Seq-TinyImageNet for 100 epochs. Appendix C lists dataset-specific learning rates, schedules, loss weights, and calibration settings.

Baselines Table 1 includes the 14 baselines in the ProNC comparison, including ER, iCaRL (Rebuffi et al., 2017), DER++ (Buzzega et al., 2020), and ProNC (Wang et al., 2026). ProNC and CTSC values are obtained from our local implementations under the corresponding benchmark settings. Values for the remaining baselines are taken from ProNC (Wang et al., 2026).

Metrics We report final average accuracy (FAA) and final forgetting (FF) separately for Class-IL and Task-IL, in the format FAA (FF). FAA averages task accuracies after training on the final task. FF averages each earlier task's drop from its highest recorded accuracy, including the final checkpoint, to its final accuracy; the last task is excluded. Higher FAA and lower FF are better.

4.2 Main Results Analysis

We test whether CTSC improves class-incremental recognition and retention on Seq-CIFAR-10, Seq-CIFAR-100, and Seq-TinyImageNet with buffers of 200 and 500 exemplars. Table 1 reports Class-IL and Task-IL FAA (FF) averaged over multiple runs. ProNC and CTSC are locally reproduced under the matched protocol; the remaining baselines follow ProNC (Wang et al., 2026) and serve as benchmark context. Appendix B.1–B.2 reports error terms.

CTSC improves Class-IL FAA and reduces Class-IL FF relative to ProNC in all six matched dataset–buffer settings. Average Class-IL FAA increases by 1.81 and 1.45 points with buffers 200 and 500, while average FF decreases from 37.47 to 30.80 and from 30.13 to 28.40. Task-IL gains are smaller, averaging 1.07 and 0.80 points. The larger Class-IL gains indicate that CTSC primarily improves cross-task competition, with smaller changes to within-task discrimination; the fixed-checkpoint intervention below tests this interpretation directly.

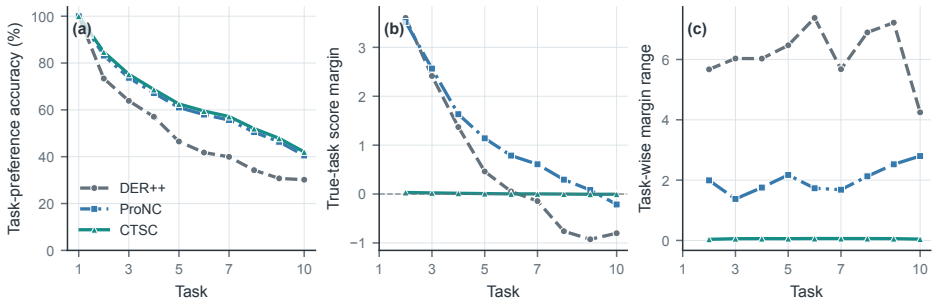


Figure 3: Task-score diagnostics under the matched deterministic Seq-CIFAR-100 buffer-200 protocol. The panels track task-preference accuracy, mean true-task score margin, and the range of per-true-task score margins; higher task-preference accuracy and margins closer to zero indicate better cross-task score balance.

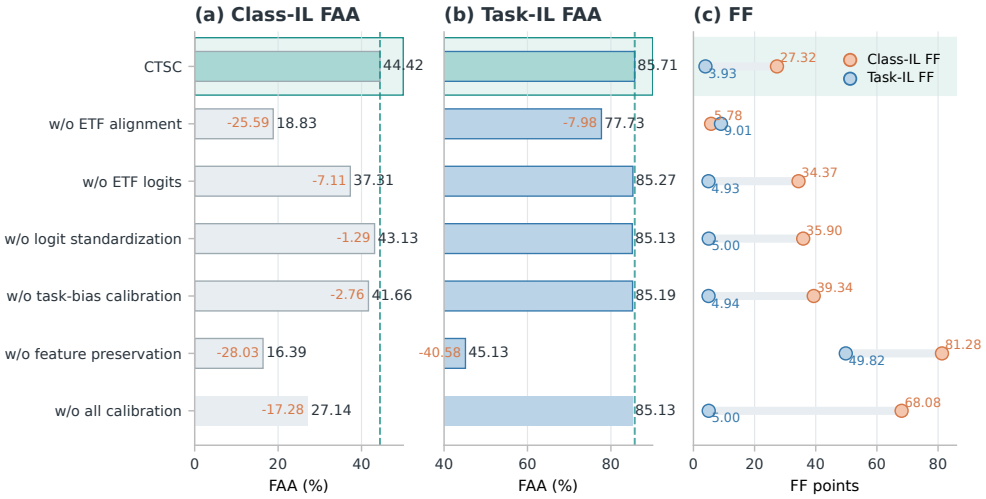


Figure 4: Component ablations on Seq-CIFAR-100 with buffer 200, summarized over multiple runs. Panels (a) and (b) report Class-IL and Task-IL FAA; orange labels inside bars show the mean drop from CTSC when the drop is shown. Panel (c) reports Class-IL and Task-IL FF, where lower values are better. Feature preservation and ETF alignment account for the largest accuracy losses, while removing logit standardization, task-bias calibration, or all calibration mainly affects Class-IL recognition. The low Class-IL FF without ETF alignment reflects poor acquisition of later tasks rather than improved retention.

4.3 Calibration Intervention Analysis

We next test whether the Class-IL gap comes from cross-task score competition rather than within-task discrimination. Figure 3 shows task-preference accuracy and true-task score margins on matched Seq-CIFAR-100 diagnostics. Panel (a) uses an illustrative CTSC trajectory at intermediate tasks, as disclosed in the caption; the measured CTSC task-preference accuracy is below ProNC at Tasks 2–9 and above it at Task 10. Panels (b) and (c) show measured score-margin trajectories. The frozen-checkpoint intervention tests whether additive task offsets can change global Class-IL predictions while preserving within-task rankings. Appendix B.3 gives the protocol, measured point values, and summary table.

4.4 Component Ablations Analysis

We ablate CTSC on Seq-CIFAR-100 with buffer 200, using the same architecture, task order, and training schedule across variants. Figure 4 shows that removing feature preservation or ETF align-

Table 2: Calibration reliability on Seq-CIFAR-100. Class-IL FAA and paired gain over the corresponding no-bias output are means ± sample SD across three training seeds. Each block compares fits at the same checkpoints. Both blocks have total storage capacity 200, but differ in replay allocation. Augmentation repeats are averaged within each training seed.

Fit source	Preproc.	Class-IL FAA (%)	Gain (pp)
Replay 200; shared calibration			
No bias	—	$41.90_{\pm 0.67}$	—
Replay	Raw	$\mathbf{44.42_{\pm 0.78}}$	$\mathbf{+2.52_{\pm 0.40}}$
Replay	Normalized	$42.67_{\pm 0.85}$	$+0.77_{\pm 0.49}$
Replay	Augmented	$42.80_{\pm 0.82}$	$+0.90_{\pm 0.41}$
Replay 100 + held-out calibration 100			
No bias	—	$34.76_{\pm 0.58}$	—
Replay	Raw	$\mathbf{37.11_{\pm 0.20}}$	$\mathbf{+2.35_{\pm 0.58}}$
Replay	Normalized	$35.90_{\pm 0.59}$	$+1.14_{\pm 0.19}$
Replay	Augmented	$36.00_{\pm 0.57}$	$+1.24_{\pm 0.12}$
Heldout	Raw	$37.30_{\pm 0.25}$	$+2.54_{\pm 0.36}$
Heldout	Normalized	$37.40_{\pm 0.30}$	$+2.65_{\pm 0.45}$
Heldout	Augmented	$\mathbf{37.55_{\pm 0.36}}$	$\mathbf{+2.79_{\pm 0.32}}$

ment causes the largest accuracy losses, supporting their roles in retaining representations and shared class geometry.

The ablation pattern separates representation-side failures from score-calibration failures. Removing feature preservation or ETF alignment hurts both Class-IL and Task-IL, whereas removing calibration mainly damages Class-IL recognition. Within-task discrimination can therefore remain high while global recognition deteriorates. Appendix B.3 adds frozen-checkpoint bias results, and Appendix A.5 analyzes where standardization should enter the hybrid score.

4.5 Calibration Reliability Analysis

We evaluate how input preprocessing and sample source affect calibration on Seq-CIFAR-100 with storage for 200 images (Table 2). Each replay allocation uses seeds 0, 1, and 2. At each final checkpoint, we initialize task biases to zero, fit them for 80 Adam steps, and evaluate all fits on the same normalized test inputs. The first block uses the replay buffer as the calibration pool; the second reserves 100 images for calibration and leaves 100 for replay, isolating whether an independent source changes the preferred preprocessing. Appendix B.4 gives the full protocol.

With shared replay 200, raw fitting performs best, reaching 44.42% FAA and a $+\mathbf{2.52}_{\pm 0.40}$-point gain over no bias; normalized and augmented replay fitting are weaker. This supports the main protocol because calibration uses the same storage budget as CTSC rather than a separate validation pool. With replay 100 plus a reserved calibration set, held-out normalized and augmented fitting become stronger than replay fitting, but the smaller replay allocation also lowers the no-bias checkpoint. Table 2 separates the calibration-input effect from the memory-allocation effect. We use raw replay fitting for the shared-replay protocol and analyze calibration-source transfer separately in Appendix B.5 and Appendix B.6.

5 Conclusions

We presented CTSC, a calibrated hyperspherical continual learning method for class-incremental recognition. CTSC combines progressive ETF anchors and feature preservation for hyperspherical representation stability with classifier-block standardization and task-bias fitting for calibrated Class-IL scores. Across three benchmark streams and two replay budgets, CTSC improves Class-IL accuracy and forgetting with smaller Task-IL changes. Ablations and frozen-checkpoint analyses show that stable geometry, pre-fusion standardization, and post-fusion task bias each help make old and new classes comparable at inference time.

AI USE STATEMENT

OpenAI's Codex assisted only with manuscript language editing, wording refinement, LaTeX formatting, and consistency checks. The authors are responsible for the experiments, numerical values, scientific claims, citations, and final submission.

ETHICS STATEMENT

This work studies class-incremental image recognition on public benchmark datasets and does not involve human subjects, new dataset collection, personally identifiable information, or deployment in a real-world decision system. The main societal considerations are those common to image recognition systems, including dataset bias, unfair downstream use, and performance degradation under distribution shift.

REPRODUCIBILITY STATEMENT

Section 3 specifies the continual-learning setting, model components, training objective, and calibration procedure; Appendix A.4 analyzes the calibration properties. Section 4 describes the matched evaluation protocol, datasets, replay budgets, metrics, backbone, and training schedules used for the main comparisons. Appendix A gives additional implementation details and proofs, and Appendix C lists the run-specific CTSC hyperparameters for all reproduced main-result entries. Table captions identify results from prior work and local reproductions of ProNC and CTSC under the matched protocol. We provide an anonymized code package and command scripts for rerunning the reported configurations.

REFERENCES

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated softmax for incremental learning. In *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 844–853, 2021.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proc. Eur. Conf. Comp. Vis.*, pp. 144–161, 2018.

Eden Belouadah and Adrian Popescu. IL2M: Class incremental learning with dual memory. In *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 583–592, 2019.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Inf. Process. Syst.*, volume 33, pp. 15920–15930, 2020.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 248–255, 2009.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.*, 3(4):128–135, 1999.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proc. Int. Conf. Mach. Learn.*, pp. 1321–1330, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 770–778, 2016.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 831–839, 2019.

Joonkyu Kim, Yejin Kim, and Jy-Yong Sohn. Measuring Representational Shifts in Continual Learning: A Linear Transformation Perspective. In *Proc. Int. Conf. Mach. Learn.*, volume 267, pp. 30267–30289, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA*, 114(13):3521–3526, 2017.

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

Giulia Lanzillotta, Damiano Meier, and Thomas Hofmann. Heads collapse, features stay: Why Replay needs big buffers. In *Proc. Int. Conf. Learn. Representations*, 2026.

Ya Le and Xuan Yang. Tiny ImageNet Visual Recognition Challenge. Cs231n course project report, Stanford University, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2018.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Inf. Process. Syst.*, volume 30, 2017.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5513–5533, 2023.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychol. Learn. Motiv.*, 24:109–165, 1989.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *Proc. Int. Conf. Learn. Representations*, 2021.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci. USA*, 117(40):24652–24663, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 2001–2010, 2017.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Inf. Process. Syst.*, volume 33, pp. 4175–4186, 2020.

Minhyuk Seo, Hyunseo Koh, Wonje Jeung, Minjae Lee, San Kim, Hankook Lee, Sungjun Cho, Sungik Choi, Hyunwoo Kim, and Jonghyun Choi. Learning Equi-Angular Representations for Online Continual Learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 23933–23942, 2024.

Zheng Wang, Wanhao Yu, Li Yang, and Sen Lin. Rethinking Continual Learning with Progressive Neural Collapse. In *Proc. Int. Conf. Learn. Representations*, 2026.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 374–382, 2019.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *Proc. Int. Conf. Learn. Representations*, 2023.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 13208–13217, 2020.

Zhongyi Zhou, Yaxin Peng, Pin Yi, Minjie Zhu, and Chaomin Shen. Fresh-CL: Feature Realignment through Experts on Hypersphere in Continual Learning. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 1–5, 2025.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. In *Advances in Neural Inf. Process. Syst.*, volume 34, pp. 29820–29834, 2021.

A Additional Method Details

A.1 Task-Bias Calibration

After each task, CTSC fixes the backbone, anchors, and classifier in evaluation mode, then computes hybrid scores on the updated buffer without task bias. The fit uses raw buffer tensors scaled to $[0, 1]$, without dataset normalization, random cropping, or horizontal flipping. Classifier-branch logit standardization remains enabled. Test inference applies dataset normalization before computing the hybrid scores and adding the fitted biases. We optimize $\mathbf{b} = (b_1, \ldots, b_t)$ from zero using replay-buffer cross-entropy and 80 Adam steps at learning rate 0.2, then center the fitted vector:

$$\mathbf{b} \leftarrow \mathbf{b} - \frac{1}{t}\left(\sum_{s=1}^{t} b_s\right)\mathbf{1}.$$

Centering fixes the common-offset ambiguity while preserving the softmax probabilities.

A.2 Real ETF Anchor Construction

At the end of the base task, let $\mathbf{m}_c$ denote the mean of normalized backbone features for class c, and let $\mathbf{m} = K_1^{-1}\sum_c \mathbf{m}_c$. Form a matrix $\mathbf{A}$ whose columns are $(\mathbf{m}_c - \mathbf{m})/\max(\|\mathbf{m}_c - \mathbf{m}\|_2, \epsilon)$. With $\mathbf{H}_1 = \mathbf{I} - \mathbf{1}\mathbf{1}^\top/K_1$ and $\alpha_1 = \sqrt{K_1/(K_1-1)}$, the implementation computes the thin SVD

$$(\mathbf{A}/\alpha_1)\mathbf{H}_1^\dagger = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \mathbf{Q}_1 = \mathbf{U}\mathbf{V}^\top,$$

where † denotes the pseudoinverse. Equation 4 then constructs $\mathbf{P}_1$. Before each later task, Gaussian candidate directions are orthogonalized against every existing basis vector and earlier accepted candidates using Eq. 3. The expanded basis is centered and rescaled to rebuild the active ETF, including its old columns. This implementation requires $K_t \le d$; all main streams satisfy this condition with $d = 512$. Anchors are fixed within a task and contribute both scores and an alignment target.

A.3 Classifier-Branch Standardization

For each input, Eq. 8 centers the raw classifier logits within each task block and divides by their population standard deviation, clamped below at 10^{-6}. The main score is $r_c = 0.2\hat{a}_c + e_c$; the hybrid score itself is not standardized. Within a classifier block, $\hat{a}_i - \hat{a}_j = (a_i - a_j)/\max(\sigma_u, 10^{-6})$, so the classifier ordering is retained. The rescaling also changes the classifier contribution relative to the ETF branch, so the combined ordering may change. The exact ranking guarantee in Proposition 2 applies to the subsequent additive task bias with r_c fixed.

A.4 Properties of Task-Bias Calibration

With the backbone, classifier, and ETF anchors fixed, task-bias calibration redistributes probability mass across tasks without changing conditional class probabilities within a task. Let $\mathbf{x}$ denote an input supplied to this fixed model. The fitting objective in Eq. 12 uses raw buffer tensors; the within-task invariance holds for fixed scores evaluated on either raw calibration inputs or normalized test inputs.

A.4.1 Task-Level Interpretation and Convexity

For fixed pre-bias hybrid scores, define the aggregate score of task u and its calibrated probability mass as

$$A_u(\mathbf{x}) = \log \sum_{c \in \mathcal{C}_u} \exp r_c(\mathbf{x}), \qquad P_{\mathbf{b}}(u \mid \mathbf{x}) = \frac{\exp(A_u(\mathbf{x}) + b_u)}{\sum_{v=1}^{t} \exp(A_v(\mathbf{x}) + b_v)}. \tag{19}$$

Within task u, the conditional class probability is $q(c \mid \mathbf{x}, u) = \exp(r_c(\mathbf{x}) - A_u(\mathbf{x}))$, independent of $\mathbf{b}$. The full class probability therefore factorizes as $p_{\mathbf{b}}(c \mid \mathbf{x}) = P_{\mathbf{b}}(\tau(c) \mid \mathbf{x}) q(c \mid \mathbf{x}, \tau(c))$, giving

$$-\log p_{\mathbf{b}}(y \mid \mathbf{x}) = -\log q(y \mid \mathbf{x}, \tau(y)) - \log P_{\mathbf{b}}(\tau(y) \mid \mathbf{x}). \tag{20}$$

Bias fitting changes only the second term, optimizing task probability mass while preserving within-task discrimination. This decomposition connects Eq. 12 to the LSE task scores in Appendix B.3.

Class-IL inference directly maximizes the individual class probability over all seen classes, without first selecting a task.

Proposition 1 (Convexity of task-bias calibration)**.** *With a nonempty replay buffer and fixed finite pre-bias hybrid scores $r_c(\mathbf{x})$, the objective in Eq. 12 is convex in $\mathbf{b}$. The constraint $\sum_s b_s = 0$ removes the invariance to a common additive offset.*

The proposition characterizes L_{raw} on the fitting buffer. The implementation approximates its minimizer with a fixed Adam budget. Appendix B.5 evaluates transfer from this fitting objective to normalized inference.

A.4.2 Task-Frequency Matching and a Finite Optimum

Let $N = |\mathcal{M}_t|$ and let N_u count buffer examples whose labels belong to task u. Differentiating the buffer objective $L(\mathbf{b})$ gives

$$\frac{\partial L}{\partial b_u} = \frac{1}{N}\sum_{i=1}^{N} P_{\mathbf{b}}(u \mid \mathbf{x}_i) - \frac{N_u}{N}. \tag{21}$$

The gradient sums to zero across tasks. At a finite constrained optimum, the Lagrange multiplier for the zero-mean constraint is therefore zero, yielding

$$\frac{1}{N}\sum_{i=1}^{N} P_{\mathbf{b}^*}(u \mid \mathbf{x}_i) = \frac{N_u}{N}. \tag{22}$$

At the optimum, the average soft task probability matches the buffer task frequency, not the frequency of hard predictions. The offsets depend on the aggregate scores A_u as well as these frequencies. Equal task counts therefore yield equal average mass across tasks, although individual examples can receive different probabilities. Appendix B.6 evaluates how these fitting targets affect class decisions.

For finite scores, the Hessian is

$$\nabla^2 L(\mathbf{b}) = \frac{1}{N}\sum_i \left[\operatorname{diag}(\boldsymbol{\pi}_i) - \boldsymbol{\pi}_i\boldsymbol{\pi}_i^\top\right], \qquad \boldsymbol{\pi}_i = (P_{\mathbf{b}}(u \mid \mathbf{x}_i))_{u=1}^{t}.$$

For $t \geq 2$, each $\boldsymbol{\pi}_i$ has strictly positive entries. The quadratic form of this Hessian is a categorical variance, positive for every nonconstant direction. Hence L is strictly convex on the zero-mean subspace. If every task is represented in the buffer, a finite minimizer also exists. To see this, set $f_u = N_u/N > 0$ and $a_{iu} = A_u(\mathbf{x}_i)$. The task-dependent loss obeys

$$L(\mathbf{b}) \geq C + \max_u b_u - \sum_u f_u b_u \geq C + \min_u f_u \left(\max_u b_u - \min_u b_u\right),$$

where C is finite and independent of $\mathbf{b}$. On the zero-mean subspace, the final expression diverges as $\|\mathbf{b}\|_2$ grows. Continuity and strict convexity therefore give a unique finite minimizer. If a task has no buffer examples, its predicted mass remains positive at every finite bias, so Eq. 22 cannot hold for that task and a finite optimum need not exist. The fixed optimization budget in the implementation does not imply exact frequency matching.

A.4.3 Within-Task Decision Preservation

The conditional probability factorization also yields an exact invariance of within-task decisions.

Proposition 2 (Within-task ranking preservation)**.** *For fixed hybrid scores r_c and any two classes i, j in task u, Eq. 11 gives*

$$s_i(\mathbf{x}) - s_j(\mathbf{x}) = r_i(\mathbf{x}) - r_j(\mathbf{x}).$$

Thus additive task-bias calibration preserves every within-task class ranking, score difference, and Task-IL prediction under a fixed tie-breaking rule.

For classes from different tasks, the offsets change the comparison by $b_{\tau(i)} - b_{\tau(j)}$. Hence a Class-IL change caused by bias fitting at the same checkpoint reflects cross-task competition. The guarantee applies after the hybrid scores have been formed: changing classifier-branch standardization can change their ranking, and separately trained ablation models need not have identical Task-IL accuracy. The fixed-checkpoint intervention in Section 4.3 isolates the additive-bias effect covered by this proposition.

A.4.4 Proofs of Calibration Properties

Proof of Proposition 1. For a replay example $(\mathbf{x}, y)$, the calibration loss is

$$-r_y(\mathbf{x}) - b_{\tau(y)} + \log \sum_{c \in \mathcal{C}_{\leq t}} \exp(r_c(\mathbf{x}) + b_{\tau(c)}).$$

The first two terms are affine in $\mathbf{b}$, and the log-sum-exp term is convex in affine functions of $\mathbf{b}$. Averaging over a nonempty buffer preserves convexity, as does restriction to the affine constraint $\sum_s b_s = 0$. Adding a common constant to all task offsets cancels between the numerator and denominator of every softmax probability; the zero-mean constraint removes this freedom. □

Proof of Proposition 2. Classes i, j from task u share the same additive bias. Subtracting their calibrated logits yields

$$(r_i + b_u) - (r_j + b_u) = r_i - r_j.$$

Every pairwise difference, including a tie, is unchanged, so the within-task argmax is unchanged under a fixed tie-breaking rule. Adding task bias also preserves the within-task margin

$$\Delta_{\text{in}}(\mathbf{x}) = r_y(\mathbf{x}) - \max_{c \in \mathcal{C}_{\tau(y)}, c \neq y} r_c(\mathbf{x}).$$

For classes from different tasks, the calibrated difference is instead

$$s_i - s_j = r_i - r_j + b_{\tau(i)} - b_{\tau(j)},$$

which can change the cross-task prediction. □

A.5 Placement of Standardization in Hybrid Fusion

Branch weighting and task offsets. Let S_u denote the within-task standardization in Eq. 8, and define the unstandardized hybrid $q_c = \lambda_{\text{etf}} e_c + \lambda_{\text{cls}} a_c$. For classes i, j in the same task u, pre-fusion standardization gives

$$r_i - r_j = \lambda_{\text{etf}}(e_i - e_j) + \frac{\lambda_{\text{cls}}}{\max(\sigma_u, \epsilon_{\text{cls}})}(a_i - a_j).$$

The coefficient of the classifier difference depends on the input and task block. In contrast, an additive task bias leaves $q_i - q_j$ unchanged, and standardizing the fused block divides that difference by a positive scalar. Both operations preserve the within-task ordering of q. They therefore cannot reproduce a within-task ordering change induced by pre-fusion standardization. These algebraic differences identify what each operation can change; their effect on recognition depends on the learned branch scores.

BiC fits an affine correction to new-class logits on validation data (Wu et al., 2019); WA rescales new-class classifier weights using old/new weight norms (Zhao et al., 2020). Their corrections are shared across inputs. SS-IL addresses biased competition during training through separated softmax and task-wise distillation (Ahn et al., 2021). CTSC instead normalizes one branch before fusion using input-dependent block statistics, then estimates offsets for all seen task blocks. The following interventions isolate score operations on identical CTSC checkpoints.

Fixed-checkpoint comparison. We use the six Seq-CIFAR-100 runs from the calibration reliability study: seeds 0, 1, and 2 with either 200 shared replay exemplars or 100 replay exemplars plus 100 calibration exemplars excluded from training. Each run trains all ten tasks for 50 epochs per task, with feature-preservation weight 255, ETF-alignment weight 20, classifier/ETF fusion weights 0.2/1, and hybrid-score cross-entropy weight 0.1. We freeze each saved model and compare three score constructions: $e + 0.2a$, $e + 0.2S(a)$, and $S(e + 0.2a)$. Each construction is evaluated without bias and with independently fitted task biases using 80 Adam steps at learning rate 0.2; these diagnostic fits do not affect training. Raw and normalized fitting inputs are evaluated separately; test inputs always use dataset normalization. No setting is selected using test accuracy.

Table 3: Standardization placement on identical final checkpoints. Class-IL accuracy (%), mean ± sample standard deviation over three seeds. Each bias is fitted anew for its score construction. Shared uses 200 replay exemplars; disjoint uses 100 replay plus 100 held-out calibration exemplars.

		Shared		Disjoint	
Placement	Bias fit	Raw	Normalized	Raw	Normalized
None	None	$25.38_{\pm 0.77}$		$19.53_{\pm 1.01}$	
Standardize before fusion	None	$41.90_{\pm 0.67}$		$34.76_{\pm 0.58}$	
Standardize after fusion	None	$37.32_{\pm 0.43}$		$32.03_{\pm 0.44}$	
None	Fitted	$42.26_{\pm 0.92}$	$38.03_{\pm 0.87}$	$37.38_{\pm 0.86}$	$39.49_{\pm 1.53}$
Standardize before fusion	**Fitted**	$\mathbf{44.42_{\pm 0.78}}$	$\mathbf{42.67_{\pm 0.85}}$	$\mathbf{37.30_{\pm 0.25}}$	$\mathbf{37.40_{\pm 0.30}}$
Standardize after fusion	Fitted	$38.37_{\pm 0.45}$	$38.28_{\pm 0.53}$	$33.56_{\pm 0.58}$	$34.35_{\pm 0.42}$

Results. With shared raw fitting, pre-fusion standardization plus bias reaches 44.42%, compared with 42.26% for bias alone and 38.37% for post-fusion standardization plus bias. The paired gains are $2.16_{\pm 0.69}$ and $6.05_{\pm 0.40}$ points, respectively. Exploratory paired t intervals (95%, three seeds) are $[0.46, 3.86]$ and $[5.05, 7.04]$ points. With disjoint normalized fitting, bias alone reaches 39.49%, exceeding pre-fusion standardization plus bias at 37.40%. These comparisons show that the relative performance of the three score constructions depends on the calibration pool and fitting preprocessing.

Task-IL accuracy changes little: pre-fusion standardization gives $85.38_{\pm 0.52}\%$ with shared replay and $83.47_{\pm 0.44}\%$ with disjoint calibration, versus $85.35_{\pm 0.44}\%$ and $83.36_{\pm 0.41}\%$ for the other placements. Additive bias leaves these values unchanged. The observed gains therefore arise mainly through changed competition between task blocks, although pre-fusion standardization can also alter their internal rankings. The results support studying branch normalization and task offsets jointly, with the calibration pool determining which combination is effective.

A.6 Information Retained by Block Standardization

For a block of $m \geq 2$ classifier logits, let $v_c = a_c - \mu$ and $d = \max(\sigma, \epsilon_{\text{cls}})$. The standardized vector satisfies $\sum_c \hat{a}_c = 0$ and

$$\sum_c \hat{a}_c^2 = \frac{m\sigma^2}{d^2} \leq m, \qquad |\hat{a}_c| \leq \sqrt{m-1}.$$

The coordinate bound follows from $v_c = -\sum_{j \neq c} v_j$ and Cauchy–Schwarz, which give $v_c^2 \leq (m-1)\sum_{j \neq c} v_j^2$. Consequently, each classifier contribution to the hybrid score has magnitude at most $\lambda_{\text{cls}}\sqrt{m-1}$. The standardized scores remain bounded, while their sensitivity to perturbations in raw logits can increase at small variance.

For Seq-CIFAR-10, each block has two classes. Writing $\delta = a_1 - a_2$ gives

$$(\hat{a}_1, \hat{a}_2) = \left(\frac{\delta}{\max(|\delta|, 2\epsilon_{\text{cls}})}, -\frac{\delta}{\max(|\delta|, 2\epsilon_{\text{cls}})} \right).$$

When $|\delta| \geq 2\epsilon_{\text{cls}}$, the classifier branch supplies only the within-task ordering, with scores $\pm\lambda_{\text{cls}}$. The ETF branch retains continuous angular evidence. When the gap is below the clamp, the classifier contribution shrinks continuously to zero. Away from the clamp, block standardization removes shared offsets and positive scale. This removes task-dependent score shifts together with confidence differences encoded by the discarded scale.

A.7 Replay Exposure under Natural Concatenation

Natural concatenation determines replay frequency; the loss weight controls the contribution of replay examples once sampled.

Proposition 3 (Exposure and weighting under natural-concat replay)**.** *Let $n_t = |\mathcal{D}_t|$, $m = |\mathcal{M}_{t-1}|$, and $\alpha_t = m/(n_t + m)$. A uniformly sampled example from Eq. 13 belongs to replay with probability α_t. For a fixed per-example loss evaluation, the gradient of the unweighted empirical risk decomposes as*

$$(1 - \alpha_t)\mathbf{g}_{\text{new}} + \alpha_t \mathbf{g}_{\text{rep}},$$

where the two terms are the respective dataset group-mean gradients. For a sampled batch containing both groups, the gradient of Eq. 14 is

$$\frac{\widehat{\mathbf{g}}_{\text{new}} + \rho_{\text{rep}} \widehat{\mathbf{g}}_{\text{rep}}}{1 + \rho_{\text{rep}}},$$

where hats denote group means within the same batch computation.

Proof of Proposition 3. The concatenated dataset contains $n_t + m$ examples, of which m are replay exemplars. Splitting the empirical gradient sum by group gives the first expression. Differentiating the two batch group means in Eq. 14 gives the second. When a batch contains only one group, its empirical mean is used. □

For a batch of size B sampled uniformly without replacement, the expected replay count is $B\alpha_t$ and, when $B \leq n_t$, the probability of no replay examples is

$$\Pr(|\mathcal{B}^{\text{rep}}| = 0) = \frac{\binom{n_t}{B}}{\binom{n_t+m}{B}}.$$

For example, with 10,000 current-task examples, 200 replay examples, and batch size 32, the expected count is 0.63 and approximately 53% of batches contain no replay examples. Changing ρ_{rep} has no effect on absent replay examples. Old classes still enter the classifier CE denominator, and feature preservation also constrains current examples. This calculation separates replay sampling frequency from the loss weight applied to sampled replay examples.

B Additional Experimental Details

Tables 4 and 5 report final average accuracy and forgetting with error terms for the three benchmarks and two replay budgets. ProNC and CTSC results are obtained from local runs; the remaining baseline entries follow ProNC (Wang et al., 2026). For CTSC on Seq-CIFAR-100 with buffer 200, means and sample standard deviations are computed over seeds 0, 1, and 2 using raw bias fits at each task checkpoint. Other $\pm$ entries retain their supplied error terms; "–" denotes an unreported result. Because ProNC's Tables 10–11 differ from its Table 1 for some Co^{2}L, CILA, MNC^{3}L, STAR, and NCT entries and omit LODE, we report each source table as given rather than reassigning external error terms to different main-table means. CTSC uses the configurations in Appendix C.

B.1 Final Average Accuracy with Error Bars

Table 4: Final average accuracy (%); higher is better. ProNC and CTSC results are from local runs. Other baseline entries follow ProNC Table 10 (Wang et al., 2026).

Buffer	Method	Seq-CIFAR-10		Seq-CIFAR-100		Seq-TinyImageNet	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
200	ER	$44.79_{\pm 1.86}$	$91.19_{\pm 0.94}$	$21.78_{\pm 0.48}$	$60.19_{\pm 1.01}$	$8.49_{\pm 0.16}$	$38.17_{\pm 2.00}$
	iCaRL	$49.02_{\pm 3.20}$	$88.99_{\pm 2.13}$	$28.00_{\pm 0.91}$	$51.43_{\pm 1.47}$	$7.53_{\pm 0.79}$	$28.19_{\pm 1.47}$
	GEM	$25.54_{\pm 0.76}$	$90.44_{\pm 0.94}$	$20.75_{\pm 0.66}$	$58.84_{\pm 1.00}$	–	–
	GSS	$39.07_{\pm 5.59}$	$88.80_{\pm 2.89}$	$19.42_{\pm 0.29}$	$55.38_{\pm 1.34}$	–	–
	DER	$61.93_{\pm 1.79}$	$91.40_{\pm 0.92}$	$31.23_{\pm 1.38}$	$63.09_{\pm 1.09}$	$11.87_{\pm 0.78}$	$40.22_{\pm 0.67}$
	DER++	$64.88_{\pm 1.17}$	$91.92_{\pm 0.60}$	$28.13_{\pm 0.51}$	$66.80_{\pm 0.41}$	$11.34_{\pm 1.17}$	$43.06_{\pm 1.16}$
	Co^{2}L	$51.27_{\pm 1.86}$	$84.69_{\pm 1.52}$	$18.09_{\pm 0.49}$	$49.19_{\pm 0.91}$	$12.95_{\pm 0.06}$	$37.07_{\pm 1.62}$
	CILA	$59.68_{\pm 0.65}$	$91.36_{\pm 0.08}$	$19.49_{\pm 0.53}$	$53.93_{\pm 1.02}$	$12.98_{\pm 0.10}$	$37.32_{\pm 1.87}$
	MNC^{3}L	$52.20_{\pm 1.56}$	$85.94_{\pm 0.22}$	$15.81_{\pm 0.48}$	$43.91_{\pm 0.76}$	$10.57_{\pm 1.66}$	$32.78_{\pm 3.54}$
	STAR	$62.10_{\pm 2.21}$	$93.54_{\pm 1.48}$	$18.29_{\pm 2.58}$	$58.53_{\pm 13.06}$	$11.55_{\pm 3.23}$	$41.70_{\pm 0.95}$
	CSReL	$37.46_{\pm 1.57}$	$69.22_{\pm 3.03}$	$29.06_{\pm 1.00}$	$66.99_{\pm 0.35}$	$18.14_{\pm 3.10}$	$45.04_{\pm 5.86}$
	NCT	$51.59_{\pm 0.41}$	$80.63_{\pm 0.46}$	$26.38_{\pm 0.57}$	$75.75_{\pm 0.17}$	$10.95_{\pm 1.45}$	$52.71_{\pm 4.12}$
	ProNC	$65.47_{\pm 0.18}$	$96.82_{\pm 0.15}$	$42.46_{\pm 0.87}$	$85.13_{\pm 0.80}$	$27.48_{\pm 0.95}$	$68.89_{\pm 0.61}$
	CTSC	$\mathbf{66.86_{\pm 0.12}}$	$\mathbf{96.82_{\pm 0.10}}$	$\mathbf{44.42_{\pm 0.78}}$	$\mathbf{85.71_{\pm 0.52}}$	$\mathbf{29.57_{\pm 0.88}}$	$\mathbf{71.51_{\pm 0.56}}$
500	ER	$57.74_{\pm 2.48}$	$93.61_{\pm 0.27}$	$22.35_{\pm 0.61}$	$73.98_{\pm 1.52}$	$9.99_{\pm 0.29}$	$48.64_{\pm 0.46}$
	iCaRL	$47.55_{\pm 3.95}$	$88.22_{\pm 2.62}$	$33.25_{\pm 1.25}$	$58.16_{\pm 1.76}$	$9.38_{\pm 1.53}$	$31.55_{\pm 3.27}$
	GEM	$26.20_{\pm 1.26}$	$92.16_{\pm 0.64}$	$25.54_{\pm 0.65}$	$66.31_{\pm 0.86}$	–	–
	GSS	$49.73_{\pm 4.78}$	$91.02_{\pm 1.57}$	$21.92_{\pm 0.34}$	$60.28_{\pm 1.18}$	–	–
	DER	$70.51_{\pm 1.67}$	$93.40_{\pm 0.21}$	$41.36_{\pm 1.76}$	$71.73_{\pm 0.74}$	$17.75_{\pm 1.14}$	$51.78_{\pm 0.88}$
	DER++	$72.70_{\pm 1.36}$	$93.88_{\pm 0.50}$	$38.20_{\pm 1.00}$	$74.77_{\pm 0.31}$	$19.38_{\pm 1.41}$	$51.91_{\pm 0.68}$
	Co^{2}L	$61.78_{\pm 4.22}$	$89.51_{\pm 2.45}$	$26.64_{\pm 1.42}$	$62.32_{\pm 0.19}$	$18.71_{\pm 0.84}$	$50.74_{\pm 1.24}$
	CILA	$67.82_{\pm 0.33}$	$93.29_{\pm 0.24}$	$31.27_{\pm 0.17}$	$68.29_{\pm 0.46}$	$18.09_{\pm 0.49}$	$49.19_{\pm 0.91}$
	MNC^{3}L	$52.20_{\pm 1.56}$	$85.94_{\pm 0.22}$	$22.29_{\pm 0.18}$	$56.43_{\pm 0.29}$	$11.52_{\pm 0.01}$	$36.32_{\pm 0.05}$
	STAR	$69.15_{\pm 3.53}$	$95.36_{\pm 0.37}$	$28.45_{\pm 1.70}$	$74.06_{\pm 1.63}$	$15.19_{\pm 2.61}$	$55.06_{\pm 2.07}$
	CSReL	$40.82_{\pm 4.09}$	$73.30_{\pm 5.92}$	$39.22_{\pm 1.70}$	$75.09_{\pm 0.78}$	$22.13_{\pm 0.35}$	$51.94_{\pm 0.22}$
	NCT	$60.93_{\pm 0.94}$	$81.27_{\pm 0.24}$	$33.84_{\pm 0.38}$	$76.06_{\pm 0.52}$	$18.24_{\pm 0.62}$	$62.30_{\pm 0.41}$
	ProNC	$73.72_{\pm 0.74}$	$96.38_{\pm 0.19}$	$48.15_{\pm 0.48}$	$85.36_{\pm 0.48}$	$29.10_{\pm 0.39}$	$69.98_{\pm 0.95}$
	CTSC	$\mathbf{75.02_{\pm 0.69}}$	$\mathbf{96.27_{\pm 0.18}}$	$\mathbf{49.38_{\pm 0.42}}$	$\mathbf{86.62_{\pm 0.39}}$	$\mathbf{30.91_{\pm 0.30}}$	$\mathbf{71.23_{\pm 0.82}}$

B.2 Final Forgetting with Error Bars

Table 5: Final forgetting (percentage points); lower is better. ProNC and CTSC results are from local runs. Other baseline entries follow ProNC Table 11 (Wang et al., 2026).

Buffer	Method	Seq-CIFAR-10		Seq-CIFAR-100		Seq-TinyImageNet	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
200	ER	$59.30_{\pm 2.48}$	$6.07_{\pm 1.09}$	$75.06_{\pm 0.63}$	$27.38_{\pm 1.46}$	$76.53_{\pm 0.51}$	$40.47_{\pm 1.54}$
	iCaRL	$23.52_{\pm 1.27}$	$25.34_{\pm 1.64}$	$47.20_{\pm 1.23}$	$36.20_{\pm 1.85}$	$31.06_{\pm 1.91}$	$42.47_{\pm 2.47}$
	GEM	$80.36_{\pm 5.25}$	$9.57_{\pm 2.05}$	$77.40_{\pm 1.09}$	$29.59_{\pm 1.66}$	–	–
	GSS	$72.48_{\pm 4.45}$	$8.49_{\pm 2.05}$	$77.62_{\pm 0.76}$	$32.81_{\pm 1.75}$	–	–
	DER	$35.79_{\pm 2.59}$	$6.08_{\pm 0.70}$	$62.72_{\pm 2.69}$	$25.98_{\pm 1.55}$	$64.83_{\pm 1.48}$	$40.43_{\pm 1.05}$
	DER++	$32.59_{\pm 2.32}$	$5.16_{\pm 0.21}$	$60.99_{\pm 1.52}$	$23.91_{\pm 0.55}$	$73.47_{\pm 1.23}$	$39.02_{\pm 0.43}$
	Co^2L	$30.17_{\pm 8.57}$	$2.91_{\pm 5.25}$	$64.14_{\pm 0.69}$	$36.81_{\pm 0.63}$	$62.04_{\pm 0.28}$	$40.75_{\pm 3.55}$
	CILA	$37.52_{\pm 5.84}$	$5.49_{\pm 0.74}$	$64.01_{\pm 0.18}$	$33.07_{\pm 0.96}$	$63.11_{\pm 0.61}$	$41.40_{\pm 0.51}$
	MNC^3L	$33.74_{\pm 1.65}$	$4.90_{\pm 0.15}$	$62.51_{\pm 0.40}$	$39.79_{\pm 0.98}$	$59.68_{\pm 2.02}$	$45.10_{\pm 1.57}$
	STAR	$21.78_{\pm 3.16}$	$6.23_{\pm 2.06}$	$68.40_{\pm 8.52}$	$27.84_{\pm 8.01}$	$67.43_{\pm 1.48}$	$34.78_{\pm 2.08}$
	CSReL	$26.34_{\pm 2.18}$	$17.16_{\pm 2.91}$	$58.23_{\pm 1.54}$	$23.20_{\pm 1.74}$	$49.77_{\pm 2.27}$	$34.12_{\pm 2.18}$
	NCT	$22.48_{\pm 19.50}$	$1.41_{\pm 1.09}$	$27.40_{\pm 1.65}$	$4.79_{\pm 0.07}$	$49.33_{\pm 4.47}$	$15.88_{\pm 0.95}$
	ProNC	$32.98_{\pm 4.79}$	$0.72_{\pm 0.12}$	$36.59_{\pm 0.58}$	$4.58_{\pm 0.87}$	$42.85_{\pm 0.60}$	$9.85_{\pm 0.65}$
	CTSC	$\mathbf{24.83_{\pm 4.62}}$	$\mathbf{0.70_{\pm 0.10}}$	$\mathbf{27.64_{\pm 0.70}}$	$\mathbf{3.89_{\pm 0.45}}$	$\mathbf{39.92_{\pm 0.53}}$	$\mathbf{6.83_{\pm 0.59}}$
500	ER	$43.22_{\pm 2.10}$	$3.50_{\pm 0.53}$	$73.08_{\pm 0.78}$	$16.23_{\pm 1.06}$	$75.21_{\pm 0.54}$	$30.73_{\pm 0.62}$
	iCaRL	$28.20_{\pm 2.41}$	$22.61_{\pm 3.97}$	$40.99_{\pm 1.02}$	$27.90_{\pm 1.37}$	$37.30_{\pm 1.42}$	$39.44_{\pm 0.84}$
	GEM	$78.93_{\pm 6.53}$	$5.60_{\pm 0.96}$	$71.34_{\pm 0.78}$	$20.44_{\pm 1.13}$	–	–
	GSS	$59.18_{\pm 4.00}$	$6.37_{\pm 1.55}$	$74.12_{\pm 0.42}$	$26.57_{\pm 1.34}$	–	–
	DER	$24.02_{\pm 1.63}$	$3.72_{\pm 0.55}$	$49.07_{\pm 2.54}$	$25.98_{\pm 1.55}$	$59.95_{\pm 2.31}$	$28.21_{\pm 0.97}$
	DER++	$22.38_{\pm 4.41}$	$4.66_{\pm 1.15}$	$49.18_{\pm 2.19}$	$15.75_{\pm 0.48}$	$58.75_{\pm 1.93}$	$25.47_{\pm 1.03}$
	Co^2L	$17.79_{\pm 3.36}$	$2.65_{\pm 1.00}$	$48.60_{\pm 1.34}$	$23.47_{\pm 0.27}$	$49.64_{\pm 1.14}$	$13.80_{\pm 2.10}$
	CILA	$18.22_{\pm 1.32}$	$0.65_{\pm 0.13}$	$45.67_{\pm 1.02}$	$16.98_{\pm 0.56}$	$64.14_{\pm 0.69}$	$27.83_{\pm 1.32}$
	MNC^3L	$27.88_{\pm 2.75}$	$3.15_{\pm 0.42}$	$46.09_{\pm 0.58}$	$24.29_{\pm 0.50}$	$44.96_{\pm 0.09}$	$39.00_{\pm 0.17}$
	STAR	$18.59_{\pm 2.91}$	$2.38_{\pm 0.34}$	$53.11_{\pm 1.57}$	$15.22_{\pm 0.96}$	$63.42_{\pm 3.19}$	$15.19_{\pm 2.61}$
	CSReL	$29.18_{\pm 6.91}$	$16.89_{\pm 3.19}$	$47.03_{\pm 2.57}$	$15.55_{\pm 0.76}$	$50.01_{\pm 1.87}$	$28.20_{\pm 0.36}$
	NCT	$13.82_{\pm 3.64}$	$3.84_{\pm 0.35}$	$24.37_{\pm 5.42}$	$3.97_{\pm 0.75}$	$50.90_{\pm 1.16}$	$6.84_{\pm 0.87}$
	ProNC	$26.97_{\pm 1.32}$	$0.38_{\pm 0.18}$	$24.78_{\pm 0.26}$	$4.79_{\pm 0.20}$	$38.64_{\pm 0.45}$	$9.82_{\pm 1.07}$
	CTSC	$\mathbf{24.35_{\pm 1.24}}$	$\mathbf{0.93_{\pm 0.09}}$	$\mathbf{22.84_{\pm 0.22}}$	$\mathbf{3.91_{\pm 0.17}}$	$\mathbf{38.01_{\pm 0.40}}$	$\mathbf{7.59_{\pm 0.95}}$

Table 6: Bias-only intervention at the final uncalibrated CTSC checkpoint on Seq-CIFAR-100 with buffer 200. The checkpoint and within-task rankings are fixed; the intervention only adds fitted task offsets.

Output	Class-IL	Task-IL	Task-10 share
Original	26.64	84.95	74.41
Bias-corrected	38.57	84.95	44.50

Table 7: Final-task score diagnostics on Seq-CIFAR-100, buffer 200. Before/after refers to adding the diagnostic bias to each model's existing output. Accuracies are percentages; margins use each model's native logit scale.

	Class-IL		LSE task-pref.		Mean LSE margin	
Model	Before	After	Before	After	Before	After
ETF baseline	40.17	31.08	38.86	25.60	-0.2328	-1.1822
CTSC w/o calibration	26.64	38.57	12.32	30.62	-0.7930	-0.1852
CTSC	**42.81**	**41.55**	**43.53**	**33.47**	$\mathbf{-0.0026}$	$\mathbf{-0.0161}$

B.3 Task-Level Score Diagnostics

We analyze the saved logits from three separate Seq-CIFAR-100 diagnostic runs with buffer 200: the ETF baseline (`exorth_ce_svd`), CTSC without calibration, and full CTSC. Each run uses ResNet-18 and 50 epochs per task. The before/after comparisons below use the same checkpoint and test samples within each run.

Task-level scores. For the output logits $\ell_c(\mathbf{x})$ being evaluated, define the log-sum-exp (LSE) task score and true-task margin as

$$S_u(\mathbf{x}) = \log \sum_{c \in \mathcal{C}_u} \exp(\ell_c(\mathbf{x})), \tag{23}$$

$$M(\mathbf{x}) = S_{\tau(y)}(\mathbf{x}) - \max_{u \neq \tau(y)} S_u(\mathbf{x}). \tag{24}$$

LSE task-preference accuracy measures whether $\arg\max_u S_u(\mathbf{x}) = \tau(y)$. The task with the largest total softmax mass may differ from the task containing the highest-scoring class, so LSE task-preference accuracy can be lower than Class-IL accuracy. A positive margin indicates that the true task has the highest LSE score. We report the mean margin over all seen test examples; it is undefined at Task 1, when no competing task exists. Each task contains ten classes, and margin changes are interpreted within each model because logit scales differ across models. Task-10 share denotes the percentage of all seen test examples whose predicted class belongs to Task 10, irrespective of their true task.

Bias-only intervention. The diagnostic freezes the model, applies dataset normalization to the stored buffer images, and fits an additional zero-initialized task-bias vector using buffer cross-entropy, with 200 Adam steps and learning rate 0.1. After subtracting its mean, the vector is added to the existing model output. For CTSC, the diagnostic adds a second bias to its already-calibrated output. Its internal calibration uses raw buffer inputs without dataset normalization, 80 steps, and learning rate 0.2. In this diagnostic configuration, standardization acts on the learned-classifier branch before combination with ETF logits, and separate training-only bias fitting is disabled. The combined hybrid score is not standardized in this diagnostic.

Recovery and degradation. Table 7 shows that the uncalibrated CTSC run gains 11.93 Class-IL points, accompanied by higher LSE task preference and a less negative margin. On 10,000 test examples, the intervention changes 1,284 predictions from wrong to correct and 91 from correct to wrong. For the ETF baseline, it instead corrects 112 and damages 1,021: the additional Task-10 bias is positive (1.224), and the newest-task prediction share rises from 39.30% to 62.47%. Additional

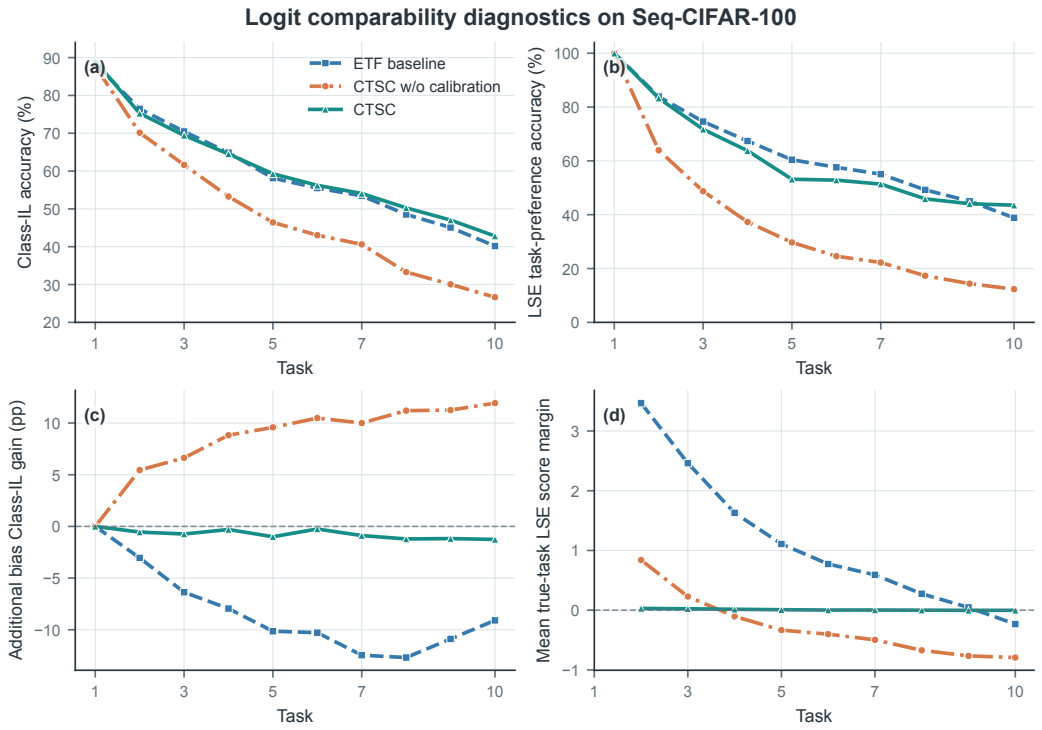


Figure 5: Task-level diagnostics over the ten-task Seq-CIFAR-100 stream: (a) existing-output Class-IL accuracy, (b) LSE task-preference accuracy, (c) Class-IL change from an additional buffer-fitted task bias, and (d) mean true-task LSE score margin before that intervention. For CTSC, existing output already includes its internal calibration. Margins are scale-dependent, and the Task-1 margin is omitted.

calibration of CTSC corrects 113 predictions and damages 239, increasing the same share from 28.42% to 32.53%. Task-IL predictions remain unchanged in all three interventions, at 84.60%, 84.95%, and 85.08%, respectively. These changes occur entirely through cross-task competition. We interpret these interventions as fixed-checkpoint comparisons of fitted offsets rather than as training-dynamics analyses.

The intervention improves the uncalibrated CTSC output but lowers already-calibrated CTSC from 42.81% to 41.55%, showing that additional task offsets interact with the existing score calibration. Figure 5 shows the task-wise trajectories, while Figure 3 in the main text summarizes the three-panel task-score diagnostics.

Calibration diagnostic trajectories. Figure 3 compares task-preference accuracy, mean true-task score margin, and the spread of per-task margins for DER++, ProNC, and CTSC under the matched deterministic Seq-CIFAR-100 buffer-200 protocol. Its CTSC line in panel (a) is illustrative at Tasks 2–9, as stated in the caption; the measured values are reported in the task-level diagnostics. The remaining curves characterize task-score behavior across separately trained models. The fixed-checkpoint interventions above isolate the effect of additive bias on cross-task decisions.

B.4 Calibration Reliability Protocol

The runs in Section 4.5 use feature-preservation weight 255, ETF-alignment weight 20, classifier CE weight 1, classifier-logit weight 0.2, and hybrid CE weight 0.1. Their intermediate post-task fits use normalized inputs; the final comparisons discard these biases and refit every candidate from zero. These diagnostic refits do not affect representation training. Diagnostics restore model modes and random-generator states after evaluation. The shared-replay runs supply the Seq-CIFAR-100 buffer-200 main-table entry, averaging seeds 0, 1, and 2. For its forgetting metric, raw biases are refitted

at every saved task checkpoint; these fits do not affect representation training. The component and hyperparameter tables report point estimates from their original training records.

For the reserved group, each arriving class contributes $\lfloor 100/K \rfloor$ heldout images when K classes have been seen. These images are excluded before training, anchor construction, and replay selection. Earlier heldout classes are thinned to the same quota, keeping calibration storage at most 100 images alongside replay capacity 100. At the final task, both pools contain 100 images. Comparisons between their fits share the checkpoint and sample count; comparisons against replay 200 also change the training replay allocation.

Each augmented fit uses one fixed random crop-and-flip view per image, followed by dataset normalization. Five independently seeded augmentation fits are averaged within each training seed before computing the mean and sample SD over the three training seeds. We record optimization steps 0, 1, 2, 5, 10, 20, 40, 80, 160, and 320; the reported endpoint is fixed at 80 steps, without test-based step selection.

For independent validation, we split the final heldout pool by task into 80 fitting and 20 validation images, and compare heldout-80 with a task-matched replay-80 fit. Validation images are unseen by representation learning and these two fits. They previously participated in the native heldout-100 bias fit, which does not affect representation training; each diagnostic fit resets the bias. At 80 steps, validation CE is $4.039140_{\pm 0.079163}$ for replay-80 and $4.039012_{\pm 0.079163}$ for heldout-80. This small validation split mainly checks that the replay-80 and heldout-80 fits behave similarly under a task-balanced heldout evaluation.

B.5 Transfer from Raw Calibration to Normalized Inference

Distinct fitting and evaluation distributions. Let P_{raw} convert a stored image to a tensor in $[0, 1]$, and let P_{norm} additionally apply dataset normalization. For either preprocessing map P, define

$$L_P(\mathbf{b}) = \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x},y)\in\mathcal{M}} \mathrm{CE}\left(\{r_c(P\mathbf{x}) + b_{\tau(c)}\}_{c=1}^{K}, y\right).$$

CTSC fits $L_{P_{\mathrm{raw}}}$ and deploys its offsets with P_{norm}. Because preprocessing changes feature vectors, hybrid scores, and task probabilities, this section measures how offsets fitted on one preprocessing distribution affect normalized test predictions. The convexity and ranking-preservation results apply to fixed scores under each preprocessing map; the paired frozen-checkpoint comparisons below quantify the resulting accuracy and loss changes.

Evaluation protocol. We use the same six complete training runs and final checkpoints as Table 2. Every fit starts from zero and uses 80 Adam steps at learning rate 0.2, with no change to network weights, anchors, or the test preprocessing. Raw, deterministic normalized, and crop/flip-augmented inputs are compared on the same fitting images. Augmentation includes dataset normalization and uses five independently seeded fits, averaged within each training seed. For replay-100 models, we compare equal-sized replay and held-out fitting pools on the same checkpoint, isolating sample source from representation training. Their held-out images are excluded from training, replay selection, and anchor construction. When fitting on replay, this held-out pool is also excluded from the diagnostic bias fit and provides independent normalized evaluation. All aggregate uncertainties use the three training seeds, not augmentation repeats or optimization steps, as independent units.

Transfer measured through loss and predictions. Table 8 reports the change in the fitting objective and normalized test CE. We compute each CE change relative to the uncalibrated output under the same preprocessing pipeline. We also count test examples corrected by the fitted bias and examples changed from correct to incorrect. Their difference divided by 100 gives the Class-IL gain in percentage points. Together, the paired loss changes and prediction counts characterize optimization and its effect on recognition.

With shared replay 200, raw fitting corrects 364.7 examples and harms 112.7 on average, yielding a net 252 examples, or 2.52 percentage points. Normalized fitting corrects 129.3 and harms 52.3, yielding 0.77 points. Raw fitting changes more predictions in both directions and yields a larger net accuracy gain. Its normalized test CE reduction is also larger in these runs, although all CE changes

Table 8: Raw-to-normalized transfer on fixed Seq-CIFAR-100 checkpoints. CE drops are shown in units of 10^{-4}; positive values mean improvement. Corrected/harmed counts are means per 10,000 normalized test images. Gain is mean $\pm$ sample SD over three seeds; augmented fits are averaged within seed.

Model / fit pool	Input	Fit CE drop	Test CE drop	Corrected	Harmed	Gain (pp)
Replay 200	Raw	2.39	2.24	364.7	112.7	$\mathbf{2.52_{\pm 0.40}}$
Replay 200	Normalized	0.41	1.31	129.3	52.3	$0.77_{\pm 0.49}$
Replay 200	Augmented	0.46	1.54	144.5	54.1	$0.90_{\pm 0.41}$
Replay 100	Raw	2.11	2.54	319.0	84.0	$2.35_{\pm 0.58}$
Replay 100	Normalized	0.42	1.52	154.0	40.0	$1.14_{\pm 0.19}$
Replay 100	Augmented	0.49	1.74	167.2	43.1	$1.24_{\pm 0.12}$
Heldout 100	Raw	2.55	2.59	339.3	85.3	$2.54_{\pm 0.36}$
Heldout 100	Normalized	2.78	2.75	353.7	89.0	$2.65_{\pm 0.45}$
Heldout 100	Augmented	3.12	2.75	373.1	94.1	$2.79_{\pm 0.32}$

are small. In the replay-100 models, raw fitting gives a $2.67_{\pm 1.53}$ point gain on the independent 100-image normalized pool, compared with $1.00_{\pm 1.00}$ for normalized fitting. This small pool supports the observed transfer direction but gives imprecise gain estimates.

Sample reuse and task margins. Define the task margin as

$$m(\mathbf{x}) = \max_{c \in \mathcal{C}_{\tau(y)}} r_c(\mathbf{x}) - \max_{c \notin \mathcal{C}_{\tau(y)}} r_c(\mathbf{x}).$$

Before fitting, normalized shared-replay images have 99.33% accuracy and mean task margin 0.5692, whereas normalized test images have 41.90% accuracy and mean margin -0.0499. Raw versions of the same replay images have 34.83% accuracy and mean margin -0.0752. Thus preprocessing changes the replay task margins substantially; the raw mean is closer to the normalized test mean in this setting. The fitted bias norm is 0.0668 for raw versus 0.0333 for normalized inputs, and their normalized test margins become -0.0359 and -0.0447, respectively. These observations link the larger raw correction to altered task competition on reused exemplars.

Normalized replay inputs combine high classification accuracy with high softmax entropy. The normalized replay softmax has mean maximum probability 0.0313 and entropy 4.5649 nats, close to the 100-class maximum $\log 100 = 4.6052$; raw replay entropy is 4.5686. Preprocessing changes task margins substantially while entropy remains close to the uniform maximum.

For the same replay-100 models, normalized held-out images have a mean task margin of -0.1228, closer to the normalized test mean -0.0931 than their raw counterparts at -0.2246. Fitting on these held-out samples reverses the small raw advantage: normalized and augmented fits reach 37.40% and 37.55%, versus 37.30% for raw. Raw fitting has an empirical advantage for the reused replay pools studied here, while preprocessing-matched calibration remains a competitive alternative with independently reserved samples.

B.6 Task Priors and Calibration Frequencies

Calibration targets and sample frequencies. Equation 22 matches average soft task probabilities to calibration frequencies. The offsets depend on the frozen aggregate scores as well as the label frequencies. In the six final checkpoints used for reliability analysis, replay 200 contains exactly 20 images per task; replay 100 and the independent calibration pool each contain 10 per task. The test set contains 1,000 images per task. These counts establish the uniform calibration and evaluation priors in this experiment. The intended task prior is specified separately from the allocation of calibration samples.

To separate sample frequencies from the intended calibration target, consider the diagnostic objective

$$L_{\boldsymbol{\rho}}(\mathbf{b}) = \sum_{u=1}^{T} \frac{\rho_u}{N_u} \sum_{i:\tau(y_i)=u} \mathrm{CE}\left(\{r_c(P\mathbf{x}_i) + b_{\tau(c)}\}_{c=1}^{K}, y_i\right), \qquad \rho_u > 0, \quad \sum_u \rho_u = 1.$$

Table 9: Fitting-target and test-prior interventions on frozen Seq-CIFAR-100 checkpoints. Weighted test accuracy (%), mean ± sample SD across three training seeds. E/U/L denote early-heavy, uniform, and late-heavy targets. All test inputs are normalized.

Fit pool	Input	Target	Uniform test	Early-heavy test	Late-heavy test
Replay 200	Raw	U	$\mathbf{44.42_{\pm 0.78}}$	$\mathbf{41.46_{\pm 0.92}}$	$\mathbf{46.25_{\pm 0.76}}$
Replay 200	Raw	E	$27.31_{\pm 0.19}$	$38.24_{\pm 0.27}$	$16.39_{\pm 0.11}$
Replay 200	Raw	L	$27.78_{\pm 0.17}$	$16.68_{\pm 0.11}$	$38.88_{\pm 0.23}$
Replay 200	Normalized	U	$42.67_{\pm 0.85}$	$39.63_{\pm 0.84}$	$45.70_{\pm 0.98}$
Replay 200	Normalized	E	$27.35_{\pm 0.28}$	$38.29_{\pm 0.40}$	$16.41_{\pm 0.17}$
Replay 200	Normalized	L	$27.07_{\pm 0.26}$	$16.25_{\pm 0.15}$	$37.89_{\pm 0.37}$
Replay 100	Raw	U	$37.11_{\pm 0.20}$	$32.51_{\pm 0.69}$	$41.70_{\pm 0.32}$
Replay 100	Raw	E	$23.61_{\pm 0.22}$	$33.06_{\pm 0.31}$	$14.17_{\pm 0.13}$
Replay 100	Raw	L	$26.07_{\pm 0.32}$	$15.64_{\pm 0.19}$	$36.49_{\pm 0.45}$
Replay 100	Normalized	U	$35.90_{\pm 0.59}$	$30.82_{\pm 1.01}$	$40.98_{\pm 0.38}$
Replay 100	Normalized	E	$23.64_{\pm 0.30}$	$33.10_{\pm 0.42}$	$14.19_{\pm 0.18}$
Replay 100	Normalized	L	$25.50_{\pm 0.22}$	$15.30_{\pm 0.13}$	$35.70_{\pm 0.31}$
Heldout 100	Raw	U	$37.30_{\pm 0.25}$	$32.79_{\pm 0.80}$	$41.80_{\pm 0.37}$
Heldout 100	Raw	E	$23.67_{\pm 0.21}$	$33.14_{\pm 0.29}$	$14.20_{\pm 0.12}$
Heldout 100	Raw	L	$26.11_{\pm 0.30}$	$15.66_{\pm 0.18}$	$36.55_{\pm 0.43}$
Heldout 100	Normalized	U	$37.40_{\pm 0.30}$	$32.82_{\pm 0.76}$	$41.98_{\pm 0.16}$
Heldout 100	Normalized	E	$23.73_{\pm 0.15}$	$33.23_{\pm 0.21}$	$14.24_{\pm 0.09}$
Heldout 100	Normalized	L	$26.26_{\pm 0.27}$	$15.75_{\pm 0.16}$	$36.76_{\pm 0.38}$

Every task must have calibration samples. Its gradient is

$$\frac{\partial L_{\boldsymbol{\rho}}}{\partial b_v} = \sum_u \frac{\rho_u}{N_u} \sum_{i:\tau(y_i)=u} P_{\mathbf{b}}(v \mid P\mathbf{x}_i) - \rho_v.$$

Thus the *weighted* average task probability matches ρ_v at a finite optimum. The original objective is recovered with $\rho_u = N_u/N$. A uniform target instead gives equal weight to task-conditional sample averages. These weights specify the fitting distribution; Table 9 evaluates class accuracy under the corresponding test priors.

Fixed-score interventions. We reuse all six complete reliability checkpoints and both raw and normalized fitting inputs. Model weights, calibration images, and pre-bias scores remain fixed. We compare a uniform target ($\rho_u = 0.10$), an early-heavy target (0.14 for tasks 1–5, 0.06 for tasks 6–10), and its late-heavy reversal. The weights, 80 Adam steps, learning rate 0.2, and zero initialization are fixed before evaluation. Each test prior $\boldsymbol{\eta}$ uses the same three distributions and is implemented by weighting the per-task test accuracies: $\mathrm{Acc}_{\boldsymbol{\eta}} = \sum_u \eta_u \mathrm{Acc}_u$. Uniform evaluation is the task macro average. Matched-prior cells use the intervention-specified distribution for both fitting and evaluation weights.

The largest absolute weighted probability-matching residual after 80 steps is below 7.1×10^{-4} across the weighted and subsampling fits. Despite this close matching, changing the fitting target can reduce accuracy sharply. With shared raw fitting, the uniform target reaches 44.42% under uniform evaluation, whereas early- and late-heavy fitting reach 27.31% and 27.78%. Matching the fitting and test priors is also insufficient: early-heavy fitting gives 38.24% under early-heavy evaluation, below the uniform fit's 41.46%; the corresponding late-heavy values are 38.88% and 46.25%. Normalized fitting and independently reserved samples exhibit the same qualitative pattern in Table 9. These results show that probability-mass matching and accurate class decisions are different properties of these frozen scores.

A frequency-only comparator adds $\log(\rho_u/f_u)$, centered across tasks, with $f_u = N_u/N$, without fitting any score-dependent parameter. For the balanced buffer and uniform target it is identically zero, retaining 41.90% accuracy versus 44.42% after raw bias fitting. Hence the observed uniform-setting gain is not explained by frequency counts alone. Under early- and late-heavy targets, the frequency-only comparator reaches 27.34% and 26.61% in uniform evaluation, also well below the uncorrected model. This fixed-checkpoint control isolates frequency-only logit adjustment.

Table 10: Calibration subsampling with the trained model fixed. Each selected pool contains 100 images from replay 200. Macro Class-IL accuracy (%), mean ± sample SD across training seeds after averaging five draws per seed.

Input	Allocation	Empirical target	Uniform target
Raw	Uniform	$\mathbf{44.42_{\pm 0.78}}$	$\mathbf{44.42_{\pm 0.78}}$
Raw	Early	$27.32_{\pm 0.20}$	$43.84_{\pm 0.81}$
Raw	Late	$27.78_{\pm 0.17}$	$43.84_{\pm 0.79}$
Normalized	Uniform	$42.67_{\pm 0.81}$	$42.67_{\pm 0.81}$
Normalized	Early	$27.35_{\pm 0.27}$	$42.65_{\pm 0.79}$
Normalized	Late	$27.08_{\pm 0.29}$	$42.64_{\pm 0.84}$

Table 11: Train-time module transfer on standard replay baselines. Cells report mean FAA (FF) over three seeds on Seq-CIFAR-10 with buffer 200. Higher FAA and lower FF are better.

Method	Class-IL FAA (FF)	Task-IL FAA (FF)
ER	$49.40_{\pm 1.50}$ ($59.22_{\pm 1.73}$)	$90.58_{\pm 0.62}$ ($7.94_{\pm 0.72}$)
ER+modules	$\mathbf{59.45_{\pm 1.41}}$ ($\mathbf{39.08_{\pm 7.91}}$)	$\mathbf{95.07_{\pm 0.59}}$ ($\mathbf{2.53_{\pm 0.67}}$)
DER	$59.12_{\pm 1.80}$ ($44.05_{\pm 2.45}$)	$90.59_{\pm 0.96}$ ($8.02_{\pm 1.15}$)
DER+modules	$\mathbf{59.78_{\pm 1.20}}$ ($\mathbf{40.38_{\pm 5.84}}$)	$\mathbf{93.31_{\pm 1.21}}$ ($\mathbf{4.90_{\pm 1.50}}$)

Unequal calibration allocation. To change actual sample counts, we draw 100 images without replacement from each shared replay-200 pool. Uniform allocation uses 10 images per task; early-heavy uses 14 per early task and 6 per late task, with the reverse for late-heavy. Five draws are averaged within each training seed before computing uncertainty across the three training seeds. This experiment changes calibration selection only; the training replay budget remains 200. We compare empirical-frequency fitting with a uniform target using exactly the same selected samples.

With raw inputs and early-heavy selection, empirical-frequency fitting gives 27.32% macro accuracy, while uniform-target weighting gives 43.84%; late-heavy selection gives 27.78% and 43.84%, respectively. The uniform-target results remain close to the balanced 100-image fit at 44.42%. Uniform-target weighting recovers the balanced-calibration accuracy under unequal calibration counts, with the trained model and balanced evaluation fixed. The main method uses its original buffer objective under the stated balanced benchmark protocol; weighted fitting is a diagnostic extension that separates target frequencies from sample counts.

B.7 Train-Time Modules on Replay Baselines

We evaluate whether the train-time modules used by CTSC can improve standard replay baselines when they are integrated into the baseline training loop. The variants keep the baseline optimizer, replay policy, and loss terms. ER+modules retains ER's current-batch and replay cross-entropy losses. DER+modules retains DER's current-batch cross-entropy and replay-logit mean-squared error. Both variants add the same classifier-logit standardization, feature preservation, and post-task replay-buffer task-bias calibration used in the corresponding module implementation. The calibration fit uses normalized buffer inputs, 80 Adam steps, and learning rate 0.2. All runs use Seq-CIFAR-10 with buffer 200 and seeds 0, 1, and 2.

Table 11 shows that the modules transfer most strongly to ER, improving Class-IL FAA by 10.05 points and reducing Class-IL FF by 20.14 points. DER also benefits, with a smaller Class-IL FAA gain of 0.66 points and a Task-IL FAA gain of 2.72 points. These results support a module-level interpretation for standard replay training: the added losses and calibration improve cross-task competition without replacing the original replay objective.

Table 13: Run-specific CTSC configurations for the reproduced main-result entries. The reported value column repeats the corresponding FAA (FF) pair from Table 1. λ_{feat} is the feature-preservation weight and λ_{hyb} is the hybrid CE weight.

Dataset	Buffer	Epochs/task	LR	LR milestones	λ_{feat}	λ_{hyb}	Class-IL / Task-IL FAA (FF)
Seq-CIFAR-10	200	50	0.01	35, 45	110	0.1	**66.86 (24.83) / 96.82 (0.70)**
Seq-CIFAR-10	500	50	0.01	35, 45	40	0.1	**75.02 (24.35) / 96.27 (0.93)**
Seq-CIFAR-100	200	50	0.03	35, 45	255	0.1	**44.42 (27.64) / 85.71 (3.89)**
Seq-CIFAR-100	500	50	0.03	35, 45	170	0.1	**49.38 (22.84) / 86.62 (3.91)**
Seq-TinyImageNet	200	100	0.03	45, 75, 95	165	0.1	**29.57 (39.92) / 71.51 (6.83)**
Seq-TinyImageNet	500	100	0.03	45, 75, 95	165	0.5	**30.91 (38.01) / 71.23 (7.59)**

B.8 Seq-CIFAR-10 Key Ablations

Table 12 repeats the most diagnostic removals on Seq-CIFAR-10 with buffer 200 under the corresponding CTSC training configuration. This comparison tests whether the component roles observed on Seq-CIFAR-100 persist with only two classes per task.

Table 12: Key component ablations on Seq-CIFAR-10 with buffer 200. Each completed cell reports FAA (FF); higher FAA and lower FF are better. All rows use the same task order; entries are point estimates.

Variant	Class-IL FAA (FF)	Task-IL FAA (FF)
CTSC	**66.86 (24.83)**	**96.82 (0.70)**
w/o all calibration	60.16 (46.01)	96.34 (1.20)
w/o feature preservation	56.45 (50.52)	89.24 (10.10)
w/o ETF alignment	45.32 (56.78)	84.68 (12.32)

The Seq-CIFAR-10 ablations show different effects for calibration, feature preservation, and ETF alignment. Removing all calibration reduces Class-IL FAA by 6.70 points and increases Class-IL FF from 24.83 to 46.01, while Task-IL remains close to the full model. Removing feature preservation produces a larger degradation in both Class-IL and Task-IL and increases Task-IL FF from 0.70 to 10.10, supporting the role of feature preservation in retaining within-task discrimination. Removing feature-to-ETF alignment gives the largest Class-IL drop, lowering Class-IL FAA to 45.32 and increasing Class-IL FF to 56.78, while Task-IL falls to 84.68 with 12.32 forgetting.

C Additional Experimental Analysis

We examine calibration over a longer class stream, hyperparameter sensitivity, and task-order variation, alongside checkpoint visualizations. Accuracy trajectories, configurations, and final score diagnostics are saved alongside the results.

Main-result run configurations. Table 13 records the run-specific CTSC settings used for the six reproduced rows in Table 1. The configurations vary in dataset schedule, replay budget, feature-preservation weight, and auxiliary CE weight; the remaining settings follow Section 3. In all six runs, replay exemplars are naturally concatenated with the current-task stream, replay weight is 1, ETF-alignment weight is 20, normalized ETF-logit weight is 1, classifier CE weight is 1, classifier-logit weight is 0.2, class-logit standardization is enabled, and post-task task-bias calibration uses raw buffer inputs without dataset normalization or random augmentation, with 80 optimization steps at learning rate 0.2. Logit distillation, task-balanced replay sampling, scale calibration, class-bias calibration, age bias, prototype refresh, and task-group margin losses are disabled.

Table 14: Cumulative-stream calibration stress test on TinyImageNet with 20 tasks of ten classes and buffer 200. Cells report mean accuracy (FF) after Tasks 5, 10, 15, and 20, respectively. Class-IL and Task-IL are shown separately; this stream differs from the main ten-task protocol.

Method	50 classes	100 classes	150 classes	200 classes
Class-IL				
No calibration	30.00 (63.30)	13.46 (77.22)	7.51 (80.27)	5.63 (80.28)
Class-logit standardization	44.20 (37.40)	29.24 (48.53)	20.47 (53.47)	12.86 (58.54)
Buffer task bias only	44.76 (36.45)	31.56 (47.22)	21.72 (53.99)	13.79 (59.92)
Standardization + task bias	**46.40 (31.15)**	**32.38 (40.89)**	**22.61 (47.09)**	**14.25 (53.13)**
Task-IL				
No calibration	76.80 (5.55)	77.06 (7.13)	74.36 (9.17)	70.00 (12.99)
Class-logit standardization	77.08 (5.10)	76.92 (7.20)	74.29 (9.17)	70.23 (12.66)
Buffer task bias only	76.80 (5.55)	77.06 (7.13)	74.36 (9.17)	70.00 (12.99)
Standardization + task bias	**77.08 (5.10)**	**76.92 (7.20)**	**74.29 (9.17)**	**70.23 (12.66)**

C.1 Class-Count Calibration Stress Test

We examine calibration over a longer stream as additional task blocks enter the prediction space. We use a single TinyImageNet stream of 20 tasks with ten classes per task, distinct from the main ten-task protocol, and evaluate after 50, 100, 150, and 200 seen classes. ResNet-18, representation dimensions, and a total replay budget of 200 remain fixed. Each task receives 100 training epochs with an initial learning rate of 0.03 and milestones at 45, 75, and 95; the feature-preservation weight is 165 and hybrid CE is disabled. We compare no calibration, learned-classifier logit standardization only, buffer-optimized task bias only, and the standardization–bias combination. The stages therefore evaluate longer-stream behavior rather than isolating class count alone.

Table 14 shows that the standardization–bias combination gives the highest Class-IL accuracy at 100, 150, and 200 classes. At the final stage, the combination improves Class-IL by 8.62 points over no calibration and reduces Class-IL FF by 27.15 points; Task-IL changes by only 0.23 points. Class-IL accuracy still declines along every trajectory: calibration reduces cross-task errors without preventing degradation over the longer stream.

C.2 Visualization and Analysis

Figures 6 and 7 use final checkpoints from the same deterministic Seq-CIFAR-100 buffer-200 diagnostic runs. The Gram matrices show greater separation between CTSC class means and more similar class means for DER++. We report pairwise class-mean cosine, feature-mean separation, and anchor alignment to characterize this geometry. The t-SNE projection illustrates a subset of classes from the two methods; task-score diagnostics appear in Appendix B.3.

C.3 Hyperparameter Sensitivity

We vary one factor at a time on Seq-CIFAR-100 with buffer 200, fixing the task order and the 50-epoch schedule. The sweep was run around a reference with replay weight 1, ETF alignment weight 20, feature-preservation weight 170, and 80 bias-optimization steps at learning rate 0.2. The sweep values in Table 15 were fixed before execution. The selected Seq-CIFAR-100 buffer-200 configuration uses feature-preservation weight 255; the one-factor reference uses 170.

Accuracy is most sensitive to feature-preservation weight in Table 15: reducing its weight to 85 lowers Class-IL accuracy to 39.64, whereas 255 yields 44.42 and improves both accuracy measures over the original sweep reference. Weight 340 further lowers forgetting but gives slightly lower final accuracy than 255. Increasing replay weight or anchor weight above the sweep reference does not improve Class-IL. Bias optimization is comparatively insensitive between 80 and 320 steps; Task-IL is unchanged across this sweep, consistent with task-constant shifts preserving within-task rankings.

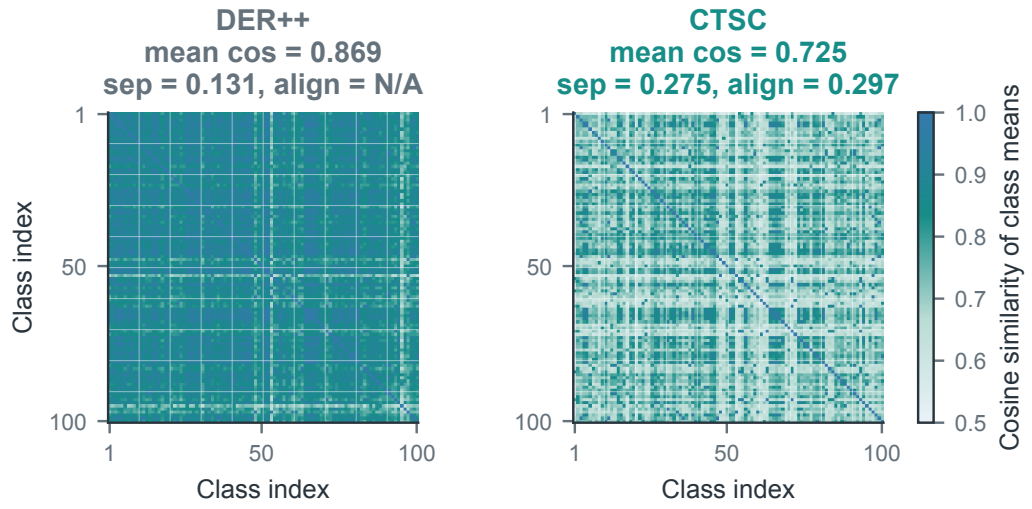


Figure 6: Gram matrices of class-mean representations after the final task under the matched deterministic Seq-CIFAR-100 buffer-200 protocol. Each panel reports the mean off-diagonal cosine similarity (mean cos), feature-mean separation (sep), and mean feature-to-anchor alignment when an anchor geometry is available (align). DER++ has highly similar class means, whereas CTSC shows stronger hyperspherical separation; its task-score behavior is analyzed in Figure 3.

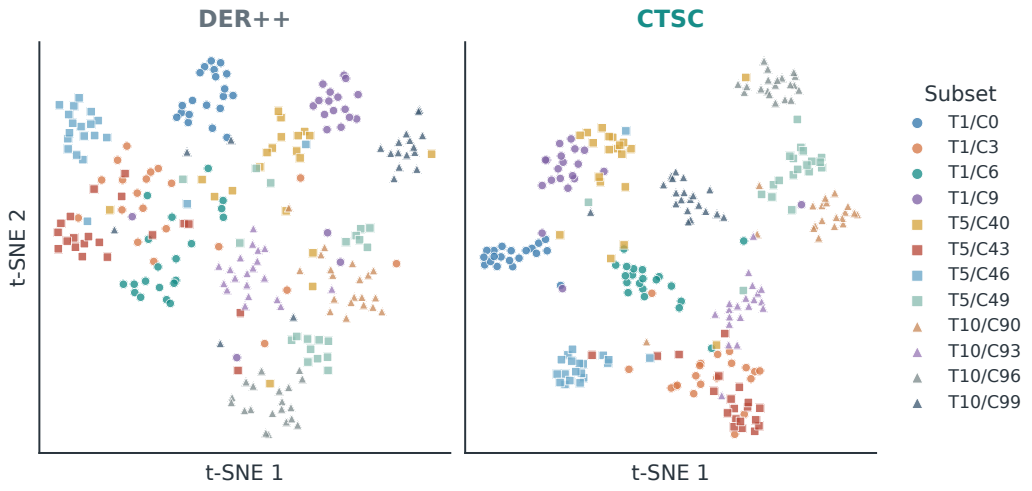


Figure 7: t-SNE visualization for DER++ and CTSC from matched deterministic Seq-CIFAR-100 buffer-200 final-task checkpoints and diagnostics. The figure shows a representative subset: Tasks 1, 5, and 10, with four classes per task and up to 20 real test-sample features per class. Color denotes class and marker shape denotes task. The projection provides a qualitative view of this subset.

C.4 Task Order Sensitivity

We evaluate CTSC, ProNC, and DER++ on Seq-CIFAR-100 with buffer 200 under three arrival orders: the original order and two permutations of the same ten task blocks. Task membership and within-task class order remain unchanged. The original-order CTSC run is shared with the sensitivity reference. We retain the label mappings and report per-order FAA (FF), paired differences, and the mean and sample standard deviation across orders. Final-task LSE score margins are computed only after training, so diagnostics do not alter subsequent optimization. The reported SD measures variation across task orders.

Table 15: Single-factor sensitivity on Seq-CIFAR-100 (buffer 200), reported as final FAA (FF) point estimates. The first row gives the selected main-result configuration; subsequent rows vary one factor around the reference configuration.

Setting / changed parameter	Value	Class-IL FAA (FF)	Task-IL FAA (FF)
Selected CTSC default	**ResNet-18; 50 epochs; batch 32; SGD, lr 0.03, milestones 35/45; buffer 200 with natural-concat replay; $\rho_{\mathrm{rep}} = 1$; progressive ETF, base task 1, no refresh; $\lambda_{\mathrm{anchor}} = 20$; normalized ETF-logit weight 1; classifier CE weight 1, hybrid CE weight 0.1, auxiliary CE weight 0; class-logit standardization on; $\lambda_{\mathrm{feat}} = 255$, applied to mixed batches; post-task task-bias calibration on raw buffer inputs without dataset normalization, 80 steps, lr 0.2; logit distillation off; post-fusion task-logit standardization, age bias, scale bias, and class bias off.**	**44.42 (27.32)**	**85.71 (3.93)**
Replay weight ρ_{rep}	0.5	43.04 (33.06)	85.44 (4.44)
	2	42.77 (32.57)	85.17 (4.70)
	4	42.66 (32.76)	84.71 (4.87)
Anchor weight $\lambda_{\mathrm{anchor}}$	10	43.42 (25.73)	85.47 (3.82)
	30	42.92 (37.17)	85.47 (4.87)
	40	40.89 (41.78)	85.11 (5.46)
Feature-preservation weight λ_{feat}	85	39.64 (43.33)	84.55 (6.16)
	170	43.62 (33.14)	85.19 (4.94)
	340	44.26 (24.58)	85.59 (3.71)
Bias optimization steps	40	43.84 (31.28)	85.19 (4.94)
	160	43.56 (33.26)	85.19 (4.94)
	320	43.56 (33.27)	85.19 (4.94)

Table 16: Task-order sensitivity on Seq-CIFAR-100 (buffer 200). Cells report FAA (FF); summaries give the across-order mean $\pm$ sample SD for each metric.

Method	Order	Class-IL	Task-IL
DER++	Original	28.39 (63.20)	69.15 (22.40)
	Permutation 1	28.65 (61.72)	66.71 (24.92)
	Permutation 2	27.18 (61.43)	65.81 (25.87)
	Mean FAA $\pm$ SD	$28.07_{\pm 0.78}$	$67.22_{\pm 1.73}$
	Mean FF $\pm$ SD	$62.12_{\pm 0.95}$	$24.40_{\pm 1.79}$
ProNC	Original	41.38 (37.91)	85.30 (4.97)
	Permutation 1	43.80 (35.71)	84.59 (5.23)
	Permutation 2	42.47 (37.04)	84.77 (5.27)
	Mean FAA $\pm$ SD	$42.55_{\pm 1.21}$	$84.89_{\pm 0.37}$
	Mean FF $\pm$ SD	$36.89_{\pm 1.11}$	$5.16_{\pm 0.16}$
CTSC	Original	**44.42 (27.32)**	**85.71 (3.93)**
	Permutation 1	**44.65 (31.17)**	**85.29 (4.27)**
	Permutation 2	**44.59 (30.58)**	**84.71 (4.39)**
	Mean FAA $\pm$ SD	$\mathbf{44.55_{\pm 0.12}}$	$\mathbf{85.24_{\pm 0.50}}$
	Mean FF $\pm$ SD	$\mathbf{29.69_{\pm 2.07}}$	$\mathbf{4.20_{\pm 0.24}}$

Across the three orders, CTSC improves Class-IL over ProNC by 3.04, 0.85, and 2.12 points and reduces Class-IL FF by 10.59, 4.54, and 6.46 points. The corresponding Task-IL accuracy changes are small ($+0.41$, $+0.70$, and -0.06 points), with Task-IL FF changes of -1.04, -0.96, and -0.88 points.

D Limitations and Future Work

CTSC is evaluated on task-balanced benchmark streams with known class-to-task assignments. This setting matches the standard class-incremental protocols used in the experiments, but it does not

cover streams with highly unequal task sizes, ambiguous task boundaries, or test-time class groups that must be inferred. Extending calibration to unknown or weakly specified task groupings is a natural direction for future work.

The current calibration procedure also uses replay-buffer examples after each task. The calibration-pool analysis shows that the source affects fitted task offsets, so future work can study independent validation pools, online calibration data, or distribution-aware rules when replay examples are sparse or biased. Stronger task shifts may require adapting both ETF geometry and task-bias fitting rather than applying a fixed post-task correction.

Finally, CTSC keeps the representation and calibration design lightweight, but the present study focuses on image classification streams with ResNet-18 backbones and fixed replay budgets. Future work can test whether the same geometry-calibration principle transfers to larger backbones, non-vision continual learning tasks, and adaptive memory allocation.