MT-CC: Multi-Group Temperature Scaling for Asymmetric Calibration Behavior in Class-Incremental Learning

Sungjun Yun **Yonghee Choi** **Dong-Jun Han**
Yonsei University
{sungjun.yun, 2025321337, djh}@yonsei.ac.kr

Abstract

As deep neural networks are required to continuously adapt to evolving data, class-incremental learning (CIL) has become an active research area. However, most existing works primarily focus on improving accuracy while overlooking confidence calibration, which measures the trustworthiness of the model's predictions. Although temperature scaling (TS) is effective for static calibration, it has rarely been explored in continual settings where data distributions evolve over time. In this paper, we introduce Multi-Group Temperature Scaling for Continual Calibration (MT-CC), a novel post-hoc calibration framework that addresses the asymmetric calibration behavior in CIL, where the rate of confidence changes does not align with the rate of accuracy degradation across tasks, leading to different levels of miscalibration. Our key contribution is to design multiple temperature parameters through a lightweight grouping network that separates samples by confidence statistics and task-aware signals in CIL. Furthermore, a Groupwise Difference-between-Confidence-and-Accuracy (GDCA) regularization is incorporated to promote intra-group consistency. Experiments demonstrate that MT-CC significantly improves calibration performance while maintaining accuracy, offering a principled framework for reliable continual learning.

1 Introduction

In modern deep learning applications, it has become increasingly important for a model to provide reliable indications of whether its predictions are likely to be correct. In particular, in a well-calibrated model, the predicted confidence should align with the empirical likelihood that the prediction is correct. This allows users to trust the model's outputs when it exhibits high confidence, while rejecting predictions made with low confidence. However, in practice, neural networks are often miscalibrated, producing confidence scores that do not align with their actual accuracy, frequently being overconfident in wrong predictions.

In this context, confidence calibration has attracted increasing attention, aiming to align the model's predicted confidence with the true likelihood of correctness. Various methods have been proposed, including temperature scaling (TS) [12], which performs a post-hoc adjustment of logits to improve calibration without altering the model's predictions. In addition to such post-hoc approaches [7, 12, 16, 22, 37, 45–47, 50], train-time calibration methods have also been explored [8, 14, 23, 27, 28, 36, 41, 49], where the model is explicitly trained to produce well-calibrated confidence scores. Compared to train-time schemes, post-hoc methods such as TS offer a clear advantage, as they can be directly applied to any pre-trained model without sacrificing accuracy.

Code available at https://github.com/DaeZZang/MT-CC

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

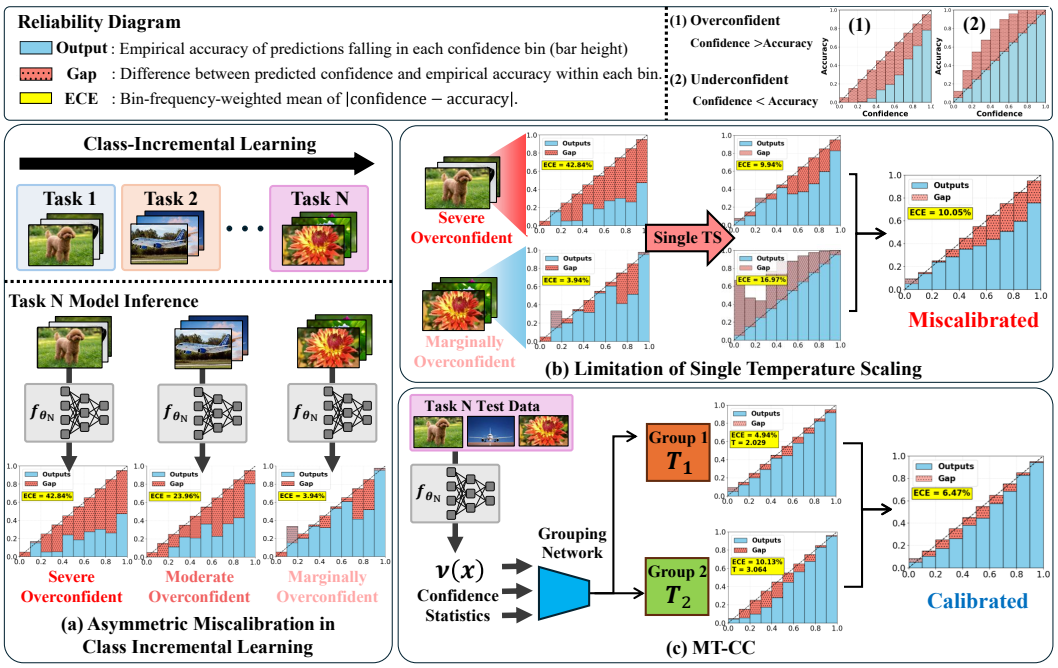


Figure 1: The reliability diagram visualizes calibration by comparing empirical accuracy and predicted confidence within each confidence bin, while a lower ECE indicates better calibration. **(a)** In class-incremental learning, accuracy and confidence evolve at *different rates* across tasks, producing **asymmetric calibration behaviors**: the gap between confidence and accuracy varies across tasks due to catastrophic forgetting. **(b)** As a result, a single global temperature cannot simultaneously calibrate these heterogeneous groups. **(c)** Our MT-CC routes each sample to a group-specific temperature to capture asymmetric calibration behaviors in class-incremental learning.

Fundamental challenges. However, most prior works on calibration have focused on static scenarios, where the target task and data distribution remain fixed. In many practical settings, however, models are required to learn continuously from a stream of tasks where new classes are introduced over time without revisiting all previous data. In such dynamic environments, maintaining well-calibrated confidence becomes particularly challenging, as the distribution shifts induced by new tasks often lead to varying levels of miscalibration across tasks.

Concretely, as illustrated in Figure 1a, models in class-incremental learning (CIL) [9, 35, 43] exhibit asymmetric calibration behaviors, where accuracy and confidence evolve at different rates across tasks under continual updates (see Figure 2). As a result, a single temperature value T fails to calibrate all classes effectively, since different task groups exhibit distinct confidence characteristics and require differentiated scaling.

Although T-CIL [16] recently proposed a TS strategy tailored to CIL, it still relies on a single temperature value and thus cannot effectively address the asymmetric calibration behaviors of CIL, as illustrated in Figure 1b. Joy et al. [17], Tomani et al. [42] propose sample-dependent temperature scaling, but they are developed under static, fixed-distribution settings and do not explicitly capture the CIL-specific calibration heterogeneity induced by continual task shifts. To the best of our knowledge, these aspects have been largely overlooked in existing calibration research and remain unresolved. Our work seeks to bridge that gap by addressing the following question:

How can we ensure reliable model calibration in continual learning, where asymmetric miscalibration behaviors across tasks pose persistent challenges?

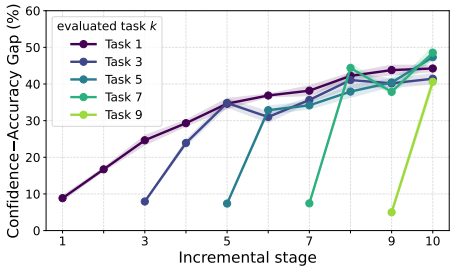


Figure 2: Confidence-accuracy gap of selected tasks over incremental stages of CIFAR-100. The results reveal **asymmetric calibration behaviors** in continual learning, where the discrepancy between confidence and accuracy varies across tasks due to catastrophic forgetting. Results for all tasks are provided in Appendix D.

Contributions. In this paper, we propose multi-group temperature scaling for continual calibration (MT-CC), a novel calibration strategy that introduces multiple temperature parameters determined by a grouping network, effectively tackling the asymmetric calibration behavior in continual learning. To this end, we train the grouping network using CIL-specific confidence statistics, including the probability mass assigned to old and new classes and the logit margin, which together capture heterogeneous calibration needs. This enables the grouping network to assign an appropriate temperature to each group, thereby achieving calibration both within each group and across the entire dataset, as shown in Figure 1c. In addition, we propose a Groupwise Difference-between-Confidence-and-Accuracy (GDCA) loss to balance confidence and accuracy within each group, thereby further improving calibration.

Our main contributions are summarized as follows:

- We analyze asymmetric calibration behaviors in class-incremental learning, where different tasks exhibit varying degrees of miscalibration, and exploit these behaviors to design a CIL-aware calibration framework.
- We propose multi-group temperature scaling for continual calibration (MT-CC), a novel calibration framework that introduces multiple temperature parameters determined by a grouping network, enabling input-dependent calibration. The grouping network is conditioned on old/new confidence statistics together with a logit margin signal, allowing it to capture asymmetric calibration behaviors and assign an appropriate temperature to each group.
- We conduct extensive experiments using various benchmark datasets, demonstrating the superiority of MT-CC in class-incremental settings.

2 Related Work

Confidence calibration. Compared to train-time calibration methods [8, 14, 23, 27, 28, 36, 41, 49] that incorporate calibration objectives during the training process, post-hoc methods [7, 12, 16, 22, 37, 45–47, 50] calibrate the confidence by adjusting the output logits of an already trained model. Post-hoc methods offer a clear advantage in that they preserve accuracy and can be directly applied to pre-trained models. A representative approach is temperature scaling (TS), which rescales logits with a single scalar temperature [12]. Subsequent extensions, including ETS, IRM [50], PTS [42], AdaTS [17], and grouping calibration [45], relax the single-temperature assumption of TS. However, they are primarily designed for static settings and thus struggle to handle continual task shifts in CIL. As shown in Section 5, directly applying these methods to CIL yields limited performance because they overlook the asymmetric calibration behavior induced by continual updates and catastrophic forgetting. MT-CC instead learns a grouping function from CIL-aware signals, including old/new confidence and logit margin: while the former directly captures the old/new class structure unique to CIL, the latter becomes CIL-relevant because catastrophic forgetting alters its relationship with miscalibration relative to static settings.

Class-incremental learning. The central challenge in CIL is catastrophic forgetting, where learning new classes degrades performance on previously learned ones [9, 31, 35, 43]. To address this, prior work has proposed regularization, parameter-isolation, and replay-based methods [1, 2, 4–6, 18, 20, 26, 29, 30, 38–40, 44, 48, 51], all aiming to preserve past knowledge and maintain accuracy across learned classes. However, most CIL studies focus primarily on classification accuracy, while largely overlooking confidence calibration.

Confidence calibration in continual learning. Li et al. [25] recently studied confidence calibration in continual learning. T-CIL [16], a recent state-of-the-art calibration method tailored to CIL, estimates a temperature after each task using augmented replay data from validation samples. However, its single-temperature formulation struggles to capture the asymmetric calibration behaviors that emerge across tasks under continual updates. MT-CC addresses this limitation by assigning multiple temperatures through a grouping network for task-aware calibration.

3 Background and Problem Setup

3.1 Background: Confidence Calibration and Temperature Scaling

Confidence calibration. In a K-way classification problem with label space $\mathcal{Y} = \{1, \dots, K\}$, a model parameterized by θ outputs a probability vector $s = \mathrm{softmax}(f_\theta(x)) \in [0,1]^K$ for an input x,

where $f_\theta(x) \in \mathbb{R}^K$ denotes the logit vector. The predicted label is $\hat{y} = \arg\max_{y \in \mathcal{Y}} s[y]$, where $s[y]$ denotes the predicted probability for class $y \in \mathcal{Y}$, and its associated confidence is $\hat{s} = \max_{y \in \mathcal{Y}} s[y]$. A model is perfectly calibrated if $\mathbb{P}(y = \hat{y} \mid \hat{s} = p) = p$ for all $p \in [0, 1]$, meaning that predictions with confidence p are correct with probability p.

Temperature scaling. Temperature scaling (TS) [12] is a post-hoc method that achieves calibration by rescaling the model's output logit z using a single scalar temperature $T > 0$. Given a sample x and its corresponding logit $z = f_\theta(x) \in \mathbb{R}^K$, the calibrated probability vector for sample x is computed as $s(z, T) = \mathrm{softmax}\left(\frac{z}{T}\right)$. The temperature T is learned on a validation set $\mathcal{D}^{\mathrm{val}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{val}}}$ by minimizing the negative log-likelihood (NLL) loss.

3.2 Problem Setup: Calibration in Class-Incremental Learning

Class-incremental learning setup. We consider the standard class-incremental learning (CIL) setting, consisting of a sequence of B tasks $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^B\}$, where each task $\mathcal{D}^b = \{(x_i^b, y_i^b)\}_{i=1}^{n_b}$ contains n_b training samples. Here, x_i^b denotes an input sample and $y_i^b \in \mathcal{Y}_b$ represents its corresponding label. Each task has a distinct label space, i.e., $\mathcal{Y}_b \cap \mathcal{Y}_{b'} = \emptyset$ for $b \neq b'$. After completing task b, the model is evaluated on all classes encountered so far, denoted as $\mathcal{Y}_{1:b} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_b$. Following Hwang et al. [16], we maintain a small memory buffer $\mathcal{M}_{b-1}$ storing representative samples from previous tasks to mitigate catastrophic forgetting. During training on task b, the model learns jointly from the current task data $\mathcal{D}^b$ and the memory buffer $\mathcal{M}_{b-1}$. After training, the buffer is updated to $\mathcal{M}_b$ by retaining a subset of samples from $\mathcal{D}^b \cup \mathcal{M}_{b-1}$.

Goal and problem formulation. Unlike prior works that primarily focus on improving accuracy, we aim to address confidence calibration in CIL explicitly. To this end, we integrate TS into the CIL framework by optimizing the temperature after each task b and applying it to calibrate model confidence. Specifically, each task b consists of two steps:

- **Training step.** The model θ learns on $\mathcal{D}^b \cup \mathcal{M}_{b-1}$ to acquire new knowledge while preserving performance on previous tasks.
- **Calibration step.** After training is completed, the model is recalibrated using validation data $\mathcal{D}_b^{\mathrm{val}}$, adjusting its confidence mapping to reflect the new data distribution.

Our goal is to build a well-calibrated model that remains reliable as tasks evolve, thereby reducing miscalibration arising from distributional and label-space shifts.

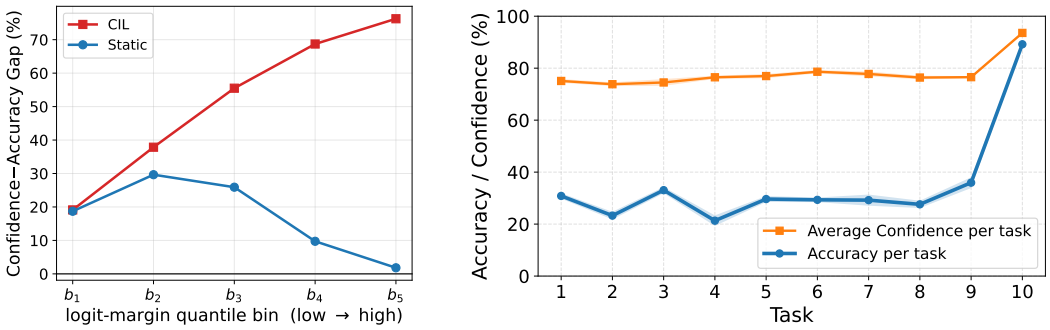
3.3 Aren't the Existing Solutions Good Enough?

In CIL, we identify a key challenge that prior works in static settings have overlooked: the confidence-accuracy gap generally widens under continual updates, with different rates and trajectories across tasks, resulting in *different levels of miscalibration* (Figure 2). Consequently, conventional TS methods that employ a single global temperature are unable to simultaneously achieve good calibration across heterogeneous confidence distributions. As illustrated in Figure 1b, the single-temperature approach in T-CIL [16], the current state-of-the-art calibration method for CIL, fails to maintain consistent calibration, highlighting its limitation in handling asymmetric miscalibration behaviors. Moreover, while several adaptive per-sample temperature scaling methods have been proposed [17, 42], they typically assume a static data distribution and do not explicitly account for continual task evolution. We later show that these prior works exhibit significant miscalibration in CIL settings. These observations underscore the need for a more flexible, group-aware calibration strategy tailored to CIL.

4 Proposed MT-CC Algorithm

We propose MT-CC (*Multi-Group Temperature Scaling for Continual Calibration*), a new calibration framework designed for class-incremental learning.

Overview of approach. At its core, MT-CC introduces a lightweight grouping module (Sections 4.1 and 4.2) that learns to partition samples into multiple groups based on confidence- and margin-based indicators, reflecting heterogeneous miscalibration behaviors in CIL. Each group is then equipped with its own temperature parameter, allowing calibration to adapt locally rather than relying on a single global scaling factor. In addition, a Groupwise Difference-between-Confidence-and-Accuracy (GDCA) loss (Section 4.3) is employed to align predicted confidence with correctness within each group, further ensuring intra-group reliability and enhancing overall calibration.



(a) Confidence–accuracy gap across logit-margin quantile bins on CIFAR-100.

(b) Task-wise accuracy and confidence on CIFAR-100 after learning all 10 tasks.

Figure 3: **Calibration behavior in CIL.** (a) The final-stage CIL model is compared with a static model jointly trained on all classes using five equal-frequency logit-margin bins from low to high margin. The confidence–accuracy gap increases with margin only under CIL. (b) Task-wise accuracy and confidence are measured after learning all 10 tasks, showing larger gaps for earlier tasks than for newly introduced tasks.

4.1 Grouping Network Design

We define a set of group-specific temperature parameters $\{T_1, T_2, \ldots, T_M\}$, where M denotes the total number of groups. We then introduce a grouping network g_ϕ, whose goal is to map each sample to the most appropriate temperature according to its confidence-related characteristics.

Grouping network mechanism (Inference). The grouping network g_ϕ is applied after the classifier, and takes the model's logits along with derived confidence statistics as inputs. For each sample x with classifier output $z = f_\theta(x) \in \mathbb{R}^K$, we construct a calibration feature vector $v(x)$ as:

$$v(x) = [\delta,\ p_{\text{old}},\ p_{\text{new}},\ p\,], \tag{1}$$

where $v(x)$ concatenates informative signals capturing different aspects of model confidence and uncertainty. Specifically, $v(x)$ consists of the softmax probability vector $p = \mathrm{softmax}(z)$, the logit margin δ, the old-task confidence p_{old}, and the new-task confidence p_{new}. Then, the concatenated vector $v(x)$ passes through the grouping network g_ϕ as follows: $g_\phi(x) = \mathrm{softmax}(W_\phi v(x) + b_\phi)$, where $W_\phi \in \mathbb{R}^{M\times d}$ and $b_\phi \in \mathbb{R}^M$ are learnable parameters of the grouping network, M is the total number of calibration groups, and $v(x) \in \mathbb{R}^d$ denotes the calibration feature vector in Eq. 1. The output of the grouping function, i.e., $g_\phi(x)$, produces a probability distribution over the M calibration groups. Consequently, the sample x is assigned to the group with the highest probability (e.g., group m), and the final calibrated output is given by $p_{\text{cal}}(x) = \mathrm{softmax}\left(\frac{f_\theta(x)}{T_m}\right)$. Here, T_m denotes the temperature associated with the group to which sample x is assigned by $g_\phi(x)$.

Design of grouping network inputs. One of our key contributions is the CIL-aware input design in Eq. 1. The input features are designed to capture complementary aspects of prediction confidence under continual learning: the logit margin tracks overconfidence caused by forgetting under CIL, the old/new-task confidences explicitly summarize the old-versus-new class structure that is otherwise implicit in the full softmax vector, while the softmax vector preserves class-wise detail. Together, these signals allow the grouping network to distinguish samples with different calibration needs.

• **Logit margin (δ):** We define the logit margin as the difference between the two largest logits, $\delta(x) = z_{(1)} - z_{(2)}$. To analyze calibration behavior across logit margins, we compare the final-stage CIL model with a static model jointly trained on all CIFAR-100 classes. Using the uncalibrated outputs of each model on the same test set, we divide samples into five equal-frequency logit-margin bins, $b_1, \ldots, b_5$, ordered from low to high margin. We report the confidence–accuracy gap in each bin, averaged over five seeds. As shown in Figure 3a, unlike in the static setting, overconfidence under CIL increases systematically with the logit margin. Consistently, the NLL-optimal temperature increases with δ under CIL, while remaining nearly constant in the static model (Appendix B.10), indicating that high-margin predictions require stronger calibration correction under CIL. We exploit this relationship to predict calibration needs, using δ as a CIL-aware input signal.

• **Old/new-task confidence ($p_{\text{old}}, p_{\text{new}}$):** For a given sample x, we define the old-task confidence as the total probability mass assigned to previously learned classes, $p_{\text{old}}(x) = \sum_{y \in \mathcal{C}_{\text{old}}} p(y \mid x)$, where $p(y \mid x)$ denotes the softmax probability for class y. If the current task is b, the old-class set is given by $\mathcal{C}_{\text{old}} = \mathcal{Y}_{1:b-1}$. Similarly, we define the new-task confidence as $p_{\text{new}}(x) = \sum_{y \in \mathcal{C}_{\text{new}}} p(y \mid x)$, where $\mathcal{C}_{\text{new}} = \mathcal{Y}_b$ denotes the set of classes introduced in the current task. As shown in Figure 3b, after all 10 tasks, old tasks generally exhibit larger confidence–accuracy gaps than new tasks, revealing distinct calibration behavior. We therefore use their respective probability masses as CIL-aware signals to capture sample-specific calibration needs. These quantities explicitly summarize the old-versus-new class structure contained in the full softmax vector, directly exposing this CIL-relevant structure to the grouping network. Appendix B.12 shows consistent improvements over using only $[\delta, p]$ across routing-training data sizes, whereas random class-mass summaries yield no comparable benefit.

• **Softmax probability vector (p):** We use the full softmax probability vector, $p = \mathrm{softmax}(f_\theta(x))$, to capture fine-grained class-wise confidence information. Even when samples have similar logit margins, their probability distributions across classes can differ, providing additional information for identifying different calibration needs. We further analyze this complementarity in Appendix B.11.

4.2 Training Grouping Network

At the beginning of each task, the validation set is divided into two disjoint subsets of equal size: $\mathcal{S}_{\text{grp}}$ is used to train the grouping network g_ϕ, while $\mathcal{S}_{\text{cal}}$ is the set reserved for groupwise temperature scaling in the subsequent stage. This split prevents overfitting and ensures that temperature parameters are tuned on unseen calibration samples.

Grouping network training. During optimization, g_ϕ and the group temperatures $\{T_1, T_2, \ldots, T_M\}$ are trained jointly on $\mathcal{S}_{\text{grp}}$ under the following objective:

$$\mathcal{L}_{\text{MT-CC}} = -\frac{1}{|\mathcal{S}_{\text{grp}}|} \sum_{(x,y) \in \mathcal{S}_{\text{grp}}} \log \sum_{m=1}^{M} g_\phi(x)_m \cdot s(f_\theta(x), T_m)[y]. \tag{2}$$

This objective is designed as a differentiable relaxation of hard groupwise temperature scaling. Ideally, each sample should be assigned to a single calibration group, but directly optimizing such discrete assignments is non-differentiable. We therefore use the soft routing weights $g_\phi(x)_m$ during this stage so that the grouping network and temperatures can be optimized jointly.

Groupwise temperature scaling. After training, the grouping network g_ϕ partitions $\mathcal{S}_{\text{cal}}$ into M disjoint groups according to the most probable assignment. Each sample x is thereby assigned to a single group and temperature scaling is applied independently within each group. Following the standard temperature scaling, the optimal temperature T_m^* for group m is obtained by minimizing the NLL loss over its corresponding subset. Each group's initial temperature T_m is initialized with the value learned during the joint MT-CC optimization, which provides a well-informed starting point for fine-tuning on its local calibration subset. The resulting groupwise temperatures $\{T_m^*\}$ are then stored and reused as initialization in the next task's MT-CC stage.

4.3 Groupwise DCA Loss

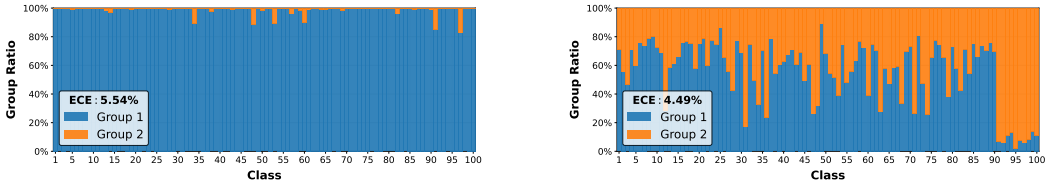
We introduce the *Groupwise Difference-between-Confidence-and-Accuracy (GDCA)* loss as an auxiliary objective. The standard DCA loss penalizes the gap between the average confidence and empirical accuracy of a model [27]. We extend this idea to the group level by enforcing such alignment within each calibration group. Let G_m denote the subset of samples assigned to the m-th calibration group. The GDCA loss is defined as

$$\mathcal{L}_{\text{GDCA}} = \sum_{m=1}^{M} w_m \left| \mathrm{Acc}(G_m) - \mathrm{Conf}(G_m) \right|, \qquad w_m = \frac{\sqrt{|G_m|}}{\sum_{j=1}^{M} \sqrt{|G_j|}}, \tag{3}$$

where $\mathrm{Acc}(G_m)$ and $\mathrm{Conf}(G_m)$ denote the empirical accuracy and average calibrated confidence within group G_m, respectively. Here, $\mathrm{Conf}(G_m)$ is computed using the calibrated mixture confidence $a(x, \hat{y}) \triangleq \sum_{k=1}^{M} g_\phi(x)_k \, s(z, T_k)[\hat{y}]$ for the predicted class $\hat{y}$. GDCA gradients are carried by the confidence term, updating the temperature parameters. Through these shared temperatures, GDCA stabilizes the coupled routing-temperature optimization and promotes groupwise calibration consistency (see Appendix C for a detailed mechanism-level discussion). The overall training objective of MT-CC is written as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MT-CC}} + \lambda \mathcal{L}_{\text{GDCA}}$, where λ controls the strength of GDCA.

Table 1: Comparison with various baseline methods. Since all evaluated methods are post-hoc calibration techniques, they preserve the original classification accuracy of the base models (65.24% for CIFAR-10, 56.47% for CIFAR-100, and 29.42% for Tiny-ImageNet). Each performance value is averaged over all sequential tasks in CIL. While maintaining identical classification performance, our method robustly outperforms the SOTA T-CIL in every calibration metric across diverse setups.

Dataset	CIFAR-10			CIFAR-100			Tiny-ImageNet		
Method	**ECE (↓)**	**AECE (↓)**	**SCE (↓)**	**ECE (↓)**	**AECE (↓)**	**SCE (↓)**	**ECE (↓)**	**AECE (↓)**	**SCE (↓)**
Vanilla	28.51±0.85	28.49±0.86	8.83±0.31	27.59±0.11	27.57±0.11	1.35±0.02	45.72±0.16	45.72±0.16	1.28±0.01
TS [12]	24.27±0.84	24.24±0.85	8.16±0.30	16.13±0.19	16.09±0.23	0.99±0.01	27.28±0.58	27.30±0.60	0.93±0.01
ETS [50]	22.75±0.83	22.65±0.81	5.92±0.20	16.70±0.39	17.62±0.25	0.56±0.01	28.93±0.59	28.90±0.63	0.52±0.01
IRM [50]	22.16±0.69	22.01±0.65	5.85±0.16	17.64±0.37	17.51±0.39	0.56±0.01	28.35±0.84	28.32±0.80	0.52±0.01
PTS [42]	25.24±2.01	25.10±1.99	6.08±0.42	18.48±0.53	18.37±0.53	0.57±0.01	36.39±1.08	36.32±1.08	0.56±0.01
AdaTS [17]	22.45±0.96	22.33±0.97	5.93±0.20	17.54±0.51	17.48±0.53	0.56±0.01	28.77±0.66	28.74±0.70	0.53±0.00
MT-CC w/o aug	**21.25**±1.42	**21.22**±1.39	**5.52**±0.23	**15.96**±1.05	**15.88**±1.09	**0.54**±0.01	**26.00**±0.92	**25.97**±0.90	**0.51**±0.01
T-CIL [16]	15.34±1.34	15.32±1.32	7.14±0.38	6.03±0.42	6.04±0.35	0.94±0.03	18.96±0.35	18.99±0.40	0.82±0.01
T-CIL + ETS	13.94±1.35	14.04±1.30	5.38±0.36	5.63±0.52	5.84±0.46	0.53±0.01	22.14±0.22	22.20±0.22	0.51±0.01
T-CIL + IRM	14.69±1.16	14.94±1.01	5.37±0.37	6.78±0.60	6.81±0.61	**0.51±0.01**	17.87±0.50	18.01±0.36	**0.49±0.01**
T-CIL + PTS	15.00±1.67	14.93±1.55	5.32±0.38	9.87±0.77	10.14±0.67	0.55±0.01	21.81±0.47	22.04±0.50	0.51±0.01
T-CIL + AdaTS	14.03±1.59	14.00±1.55	5.41±0.37	7.67±0.52	7.66±0.49	0.53±0.01	20.25±0.58	20.23±0.59	0.58±0.01
MT-CC	**10.18**±1.48	**10.16**±1.54	**4.64**±0.25	**4.49**±0.53	**4.44**±0.49	**0.51**±0.01	**15.82**±0.29	**15.80**±0.32	0.52±0.01



(a) Naive grouping (b) MT-CC grouping

Figure 4: Portion of samples from each class allocated to each group. Results are shown after training on the 10th task of CIFAR-100. Compared to (a) naive confidence-based grouping, which yields imbalanced groups, (b) MT-CC achieves a more effective group allocation, leading to lower ECE.

5 Experimental Results

We evaluate the proposed MT-CC framework by measuring the calibration performance under the CIL setting. All results are reported over five seeds for CIFAR-10, CIFAR-100, Tiny-ImageNet, and ViT-based CIL experiments, and over three seeds for ImageNet.

Datasets and CIL setup. We conduct experiments on the widely used benchmarks for CIL: CIFAR-10 [21], CIFAR-100 [21], Tiny-ImageNet [24] and ImageNet [10]. Following the training protocol of T-CIL [16], for CIFAR-10 and CIFAR-100, we use a ResNet-32 backbone [13] with 5 tasks (2 classes per task) and 10 tasks (10 classes per task), retaining memory sizes of 200 and 2,000 samples, respectively. For Tiny-ImageNet, we adopt ResNet-18 and construct 10 tasks of 20 classes each, using 2,000 memory samples. For ImageNet, we employ ResNet-18 and partition the 1,000 classes into 10 tasks, forming a challenging setup with memory sizes of 5,000, 10,000, and 20,000 samples. We also evaluate ViT-B/16 [11] with ER on CIFAR-100 and ImageNet-R [15], using 10 tasks, a 2,000-sample replay buffer, and 500 validation samples.

Calibration metrics. We evaluate calibration using three standard metrics: Expected Calibration Error (ECE) [33], Adaptive Expected Calibration Error (AECE) [32], and Static Calibration Error (SCE) [34]. ECE measures the gap between predicted confidence and empirical accuracy using fixed confidence bins, while AECE uses adaptive binning based on the confidence distribution of the test data. SCE extends calibration evaluation to the multi-class setting by considering class-wise probabilities. In our experiments, we use 10 bins for all metrics.

Baselines. We compare MT-CC with representative post-hoc calibration methods. (i) Vanilla denotes the continual learning model without calibration. (ii) TS [12] learns a single global temperature. (iii) ETS [50] extends TS through an ensemble of probability transformations. (iv) IRM [50] performs non-parametric calibration via multi-class isotonic regression. (v) PTS [42] learns prediction-specific temperatures using a lightweight neural network. (vi) AdaTS [17] estimates instance-wise temperatures using a class-conditional VAE and a lightweight neural network. (vii) T-CIL [16] optimizes TS using adversarially perturbed exemplars for class-incremental learning. (viii) For stronger baselines, we apply ETS, IRM, PTS, and AdaTS to T-CIL [16], denoted as T-CIL+[method]. To ensure a fair comparison in CIL, we report calibration results under two validation protocols: the raw new-task

Table 2: ImageNet results with varying replay buffer sizes. Since all evaluated methods are post-hoc, they share the same accuracies of the base models: for **ER**, 32.86% (5K), 41.23% (10K), 49.52% (20K); for **iCaRL**, 44.08% (5K), 49.29% (10K), 54.24% (20K). Reflecting that SCE values vary with the number of labels, per mille (‰) is used.

Replay buffer		5K			10K			20K		
CL	**CA**	**ECE(↓)**	**AECE(↓)**	**SCE(↓)**	**ECE(↓)**	**AECE(↓)**	**SCE(↓)**	**ECE(↓)**	**AECE(↓)**	**SCE(↓)**
ER	Vanilla	29.95±0.30	29.95±0.30	2.18±0.02	22.46±0.19	22.46±0.19	1.80±0.02	16.39±0.23	16.38±0.23	1.45±0.02
	TS [12]	29.66±0.33	29.63±0.33	2.16±0.02	22.67±0.17	22.65±0.18	1.79±0.02	16.83±0.18	16.80±0.20	1.44±0.01
	ETS [50]	30.68±0.25	30.65±0.26	1.09±0.01	23.53±0.16	23.53±0.16	1.09±0.01	17.62±0.22	17.58±0.23	0.91±0.00
	PTS [42]	30.32±0.44	30.32±0.44	1.09±0.01	22.79±0.36	22.80±0.36	1.89±0.01	16.52±0.27	16.51±0.28	0.91±0.01
	T-CIL [16]	13.45±0.15	13.43±0.13	1.79±0.01	7.74±0.23	7.72±0.24	1.49±0.01	10.57±0.22	10.52±0.24	1.27±0.00
	MT-CC	**12.59**±0.56	**12.62**±0.56	**1.05**±0.01	**7.25**±0.56	**7.26**±0.56	**0.91**±0.01	**8.44**±0.69	**8.44**±0.69	**0.82**±0.02
iCaRL	Vanilla	22.85±0.18	22.85±0.19	1.93±0.02	17.91±0.23	17.90±0.23	1.68±0.02	12.94±0.83	12.93±0.82	1.44±0.05
	TS [12]	22.24±0.31	22.22±0.29	1.93±0.02	18.08±0.38	18.07±0.37	1.68±0.02	16.18±2.88	16.16±2.88	1.47±0.10
	ETS [50]	22.79±0.12	22.78±0.12	1.16±0.01	18.67±0.43	18.64±0.44	1.03±0.01	15.10±0.11	15.08±0.11	0.90±0.01
	PTS [42]	21.82±0.05	21.83±0.04	**0.96**±0.00	17.63±0.16	17.63±0.17	**0.83**±0.01	14.07±0.09	14.09±0.10	0.99±0.01
	T-CIL [16]	5.99±0.30	5.98±0.31	1.80±0.02	6.84±0.01	6.80±0.04	1.57±0.01	9.69±1.48	9.64±1.47	1.36±0.03
	MT-CC	**4.87**±0.26	**4.88**±0.25	1.05±0.01	**5.16**±0.14	**5.12**±0.12	0.95±0.01	**6.88**±0.79	**6.80**±0.79	**0.85**±0.01

Table 3: ViT-based CIL post-hoc calibration results over five seeds. Base accuracy is preserved at 90.99% on CIFAR-100 and 79.23% on ImageNet-R. Metrics are in 10^{-2} scale.

Method	**CIFAR-100**			**ImageNet-R**		
	ECE	**AECE**	**SCE**	**ECE**	**AECE**	**SCE**
Vanilla	6.23±0.19	0.41±0.02	0.34±0.02	10.68±0.31	0.38±0.01	0.37±0.03
TS [12]	6.44±0.20	0.30±0.02	0.34±0.03	11.80±0.29	0.32±0.01	0.38±0.03
ETS [50]	9.82±0.37	0.29±0.02	0.34±0.03	17.30±0.93	0.29±0.02	0.39±0.03
IRM [50]	8.39±0.34	0.31±0.01	0.36±0.03	15.51±0.99	0.28±0.02	0.41±0.03
PTS [42]	9.66±0.44	0.29±0.02	0.34±0.03	18.43±1.03	0.27±0.02	0.38±0.03
AdaTS [17]	6.00±0.68	0.28±0.02	0.34±0.02	8.93±0.66	0.28±0.01	0.37±0.02
MT-CC w/o aug	**5.10**±0.49	**0.27**±0.01	**0.33**±0.02	**7.99**±0.63	**0.28**±0.02	**0.36**±0.03
T-CIL [16]	5.23±0.54	0.30±0.02	0.34±0.03	8.04±0.71	0.29±0.02	0.36±0.03
T-CIL + ETS	5.49±0.53	0.29±0.01	0.35±0.03	8.13±0.72	0.28±0.01	0.37±0.02
T-CIL + IRM	5.51±0.57	0.29±0.02	0.34±0.03	7.95±0.64	0.28±0.01	0.37±0.03
T-CIL + PTS	6.03±0.66	0.29±0.02	0.34±0.03	10.00±0.84	0.29±0.01	0.38±0.02
T-CIL + AdaTS	6.04±0.61	0.27±0.02	0.33±0.02	9.03±0.85	0.28±0.01	0.37±0.02
MT-CC	**4.57**±0.28	**0.26**±0.02	**0.33**±0.03	**7.55**±0.29	**0.26**±0.01	**0.30**±0.03

validation set and the T-CIL-augmented validation set. Within each protocol, all baselines use the same number of validation samples for temperature tuning. This allows us to compare standard post-hoc methods and CIL-specific methods under consistent validation settings. Importantly, all methods preserve the same accuracy since they operate in a post-hoc manner.

Implementation details. For all datasets, the ResNet-based main experiments use a single fully connected layer for the grouping network $g_\phi(x)$ with a learning rate of 0.01. The GDCA regularization weight is set to 1.0 for ResNet and 0.1 for ViT experiments. The initial temperature values are set to $[1.5, 1.5]$, and the validation set is generated using the T-CIL framework and divided into $\mathcal{S}_{\text{grp}}$ and $\mathcal{S}_{\text{cal}}$ with equal sizes. The module is optimized using the Adam optimizer [19] with a batch size of 16 for 50 epochs. The number of groups is set to $M = 2$ in our experiments, as the two-group configuration is sufficient to provide reliable calibration performance, as shown in Appendix B.1. The main experiments are conducted based on ER [6]. To further verify the applicability of MT-CC, we additionally compare the performance of different approaches using representative CIL methods. ViT-based experiments and their implementation details are provided in Appendix A.

5.1 Main Experimental Results

Comparison with baselines. Table 1 compares MT-CC with representative accuracy-preserving post-hoc calibration methods. MT-CC achieves the lowest ECE and AECE across CIFAR-10, CIFAR-100, and Tiny-ImageNet, while remaining competitive in SCE. We further report *MT-CC w/o aug*, which denotes MT-CC trained on the raw new-task validation set without T-CIL's augmented validation samples. It remains competitive among raw-validation baselines, suggesting that the multi-group temperature design drives the calibration gains.

Analysis of group partitioning behavior. Figure 4 compares the group allocation results of naive grouping and MT-CC. Naive grouping, which only takes confidence values as input, leads to a highly imbalanced class-group distribution in the final task. MT-CC maintains a balanced distribution across groups and effectively distinguishes samples with different calibration behaviors. The grouping network in MT-CC leads to a higher temperature in Group 2, which contains not only the majority

Table 4: Ablation on grouping and input signals of MT-CC.

Exp	Ablated Components			CIFAR-10		CIFAR-100		Tiny-ImageNet	
	Logit Margin	Old/New Confidence	GDCA	ECE (↓)	AECE (↓)	ECE (↓)	AECE (↓)	ECE (↓)	AECE (↓)
$\mathcal{A}$ (All)	✓	✓	✓	10.18±1.48	10.16±1.54	**4.49±0.53**	**4.44±0.49**	**15.82±0.29**	**15.80±0.32**
$\mathcal{B}$	×	✓	✓	**10.12±1.52**	**10.12±1.53**	4.95±0.37	4.93±0.34	15.90±0.36	15.90±0.41
$\mathcal{C}$	✓	✓	×	11.81±1.41	11.81±1.40	4.78±0.42	4.79±0.40	15.93±0.37	15.94±0.42
$\mathcal{D}$	✓	×	✓	10.91±1.26	10.91±1.26	4.63±0.30	4.65±0.26	19.48±0.40	19.47±0.44
$\mathcal{E}$	×	×	×	12.45±1.23	12.42±1.20	5.54±0.42	5.54±0.41	19.69±0.83	19.68±0.86

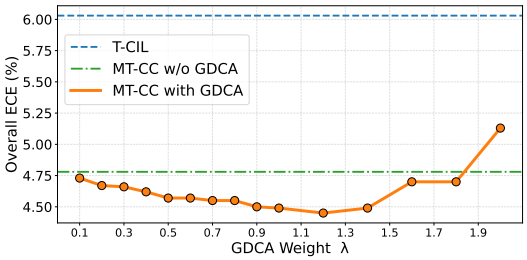


Figure 5: Effect of GDCA weight λ on overall calibration. Across a broad range of λ values, both MT-CC variants (with and without GDCA) consistently outperform the baseline.

Table 5: Versatility with other CIL methods.

Method	ECE	AECE	SCE
EEIL + TS	14.29±0.63	14.29±0.64	1.05±0.02
EEIL + T-CIL	4.28±0.35	4.34±0.28	1.01±0.03
EEIL + MT-CC	**3.67±0.20**	**3.71±0.18**	**0.58±0.01**
BiC + TS	34.19±1.54	34.13±1.55	1.55±0.05
BiC + T-CIL	27.52±0.85	27.44±0.83	1.41±0.04
BiC + MT-CC	**26.73±0.99**	**26.65±0.97**	**0.84±0.03**
iCaRL + TS	17.89±0.41	17.83±0.42	1.24±0.03
iCaRL + T-CIL	8.64±1.12	8.62±1.14	1.13±0.04
iCaRL + MT-CC	**6.00±1.02**	**6.10±0.99**	**0.68±0.02**
IL2M + TS	10.15±0.43	10.10±0.41	0.88±0.02
IL2M + T-CIL	8.54±1.37	8.62±1.33	0.90±0.02
IL2M + MT-CC	**7.72±0.89**	**7.75±0.86**	**0.47±0.02**
DER + TS	17.68±0.18	17.63±0.20	0.90±0.01
DER + T-CIL	6.18±0.31	6.12±0.30	0.79±0.02
DER + MT-CC	**4.10±0.39**	**4.06±0.41**	**0.69±0.01**

Table 6: Comparison of task-wise grouping, per-task TS, and the proposed MT-CC.

Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Overall
GT Old/New TS	**2.18±0.63**	11.50±1.77	5.43±2.05	4.81±1.11	3.47±1.28	3.76±0.46	4.35±1.97	6.00±2.34	7.88±2.22	10.37±2.02	5.97±0.32
GT Per-task TS	**2.18±0.63**	10.79±1.60	5.42±2.03	4.80±1.11	3.47±1.27	3.77±0.46	**4.34±1.96**	6.00±2.33	7.88±2.22	10.42±1.93	5.91±0.31
MT-CC	**2.18±0.63**	**6.99±2.03**	**3.22±0.94**	**2.32±0.50**	**3.16±0.65**	**2.49±0.36**	4.75±2.39	**4.72±3.27**	**7.11±2.75**	**7.97±2.13**	**4.49±0.53**

of new-task samples (classes 91 – 100) but also many of the old-task samples (classes 1 – 90). As a result, this adaptive grouping and temperature assignment lead to improved calibration performance.

5.2 Further Studies

Scalability and backbone generality (ImageNet and ViT). We further evaluate MT-CC on ImageNet and ViT-based CIL settings. Table 2 shows that MT-CC remains effective on ImageNet with 1,000 classes across different memory budgets and replay-based CIL methods. Table 3 further demonstrates that MT-CC generalizes beyond CNN backbones, achieving strong calibration performance with ViT on CIFAR-100 and ImageNet-R while preserving the same base accuracy. Overall, these results indicate that MT-CC is robust to larger output spaces and different backbone architectures.

Ablation on grouping and input signals. Table 4 analyzes the contribution of MT-CC components. Removing either the logit margin or old/new confidence feature generally increases ECE and AECE, indicating that both contribute useful calibration signals. Eliminating GDCA further degrades performance, confirming its role in improving groupwise calibration.

Effect of GDCA regularization. Figure 5 illustrates the effect of the GDCA weight, evaluated on the CIFAR-100 dataset. GDCA improves ECE across a wide range of λ values, confirming its effectiveness. Notably, both MT-CC variants, with and without GDCA, consistently outperform the T-CIL baseline, further demonstrating the robustness of our method.

Compatibility with other CIL methods. To further evaluate the compatibility of MT-CC, we conduct experiments by integrating it with several milestone replay-based CIL methods, including EEIL [4], iCaRL [38], BiC [43], IL2M [3], and DER [44]. CIFAR-100 is adopted for experiments. As shown in Table 5, MT-CC consistently achieves the best calibration performance across all frameworks, demonstrating its plug-and-play applicability to diverse CIL methods.

Comparison with ground-truth grouping baselines. Table 6 compares MT-CC with two oracle ground-truth grouping baselines on CIFAR-100. The first baseline, *GT Old/New TS*, splits samples into two groups, old-task classes $\mathcal{Y}_{1:b-1}$ and the new-task classes $\mathcal{Y}_b$, and applies temperature scaling to each group. The second baseline, *GT Per-task TS*, uses a more fine-grained oracle partition by assigning samples to their corresponding task groups and learning a separate temperature for each task. Since each per-task group contains fewer calibration samples, we use a smaller learning rate for GT Per-task TS to ensure stable and fair optimization. Despite this oracle advantage, MT-CC

achieves lower ECE than both grouping baselines in most tasks. These results show that MT-CC learns a more effective calibration boundary than rule-based task grouping.

6 Conclusion

In this work, we proposed MT-CC, a multi-group temperature scaling strategy for handling asymmetric calibration behavior in CIL. MT-CC incorporates a strategically designed grouping network that leverages old/new-task confidence and logit margin as inputs. The GDCA loss further stabilizes this network and promotes intra-group consistency without sacrificing accuracy. Extensive experiments demonstrate that MT-CC achieves state-of-the-art calibration performance in CIL, providing a promising direction for building reliable models in continual learning settings.

Limitations and future work. A deeper theoretical understanding of MT-CC remains to be developed. Extending MT-CC to different continual learning domains (e.g., large language models) is also an important direction for future work.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25470983), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00457882, AI Research Hub Project, IITP-2026-RS-2026-25544188, Leading Generative AI Human Resources Development), and the Technology Development Program of MSS (RS-2025-25464351). Dong-Jun Han is the corresponding author.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019.

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[4] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

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

[7] Wonjeong Choi, Jungwuk Park, Dong-Jun Han, Younghyun Park, and Jaekyun Moon. Consistency-guided temperature scaling using style and content information for out-of-domain calibration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11588–11596, 2024. doi: 10.1609/aaai.v38i10.29041.

[8] Yun-Wei Chu, Dong-Jun Han, Seyyedali Hosseinalipour, and Christopher Brinton. Unlocking the potential of model calibration in federated learning. In *International Conference on Learning Representations*, 2025.

[9] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in

classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385, 2021.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16081–16090, 2022.

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[16] Seong-Hyeon Hwang, Minsu Kim, and Steven Euijong Whang. T-cil: Temperature scaling using adversarial perturbation for calibration in class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15339–15348, 2025.

[17] Tom Joy, Francesco Pinto, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14919–14926, 2023.

[18] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[19] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[24] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[25] Lanpei Li, Elia Piccoli, Andrea Cossu, Davide Bacciu, and Vincenzo Lomonaco. Calibration of continual learning models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2024.

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[27] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. *arXiv preprint arXiv:2009.04057*, 2020.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[30] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–82, 2018.

[31] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

[32] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in neural information processing systems*, 33:15288–15299, 2020.

[33] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[34] Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2020.

[35] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[36] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[37] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[39] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[40] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[42] Christian Tomani, Daniel Cremers, and Florian Buettner. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European conference on computer vision*, pages 555–569. Springer, 2022.

[43] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019.

[44] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021.

[45] Jia-Qi Yang, De-Chuan Zhan, and Le Gan. Beyond probability partitions: Calibrating neural networks with semantic aware grouping. *Advances in Neural Information Processing Systems*, 36:58448–58460, 2023.

[46] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, 2001.

[47] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[48] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[50] Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR, 2020.

[51] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020.

Appendix

A Training Hyperparameters and Implementation Details **15**

A.1 Representative Replay-Based CIL Methods 16

A.2 Details of Calibration Metrics 16

B Additional Ablation Studies **17**

B.1 Ablation on the Number of Groups 17

B.2 Ablation on the Replay Memory Size 17

B.3 Comparison of ECE, AECE, SCE Across Sequential Tasks on CIFAR-100 18

B.4 Results on CIFAR-100: 60-5 Split 18

B.5 Robustness to ECE Binning and Brier Score 19

B.6 Applicability to Static Classification 19

B.7 Effect of Validation Set Size 19

B.8 Inference Time Analysis 20

B.9 Comparison with Feature-Matched and Grouped Calibration Methods 20

B.10 CIL Changes the Calibration Meaning of Logit Margin 20

B.11 Complementarity of Margin and Confidence 21

B.12 Benefit of Explicit Old/New Class Structure 22

B.13 Continual Update and Stability of the Grouping Network 23

C How GDCA Stabilizes Routing-Temperature Optimization **24**

D Catastrophic Forgetting and Asymmetric Calibration Behaviors **25**

E Training and Inference Procedure of MT-CC **26**

F Reliability Diagram **27**

F.1 Group-wise Reliability Diagrams at the Final Task 27

F.2 Reliability Diagrams across Sequential Tasks 27

Table 7: Hyperparameters for Group Calibration Experiments

	ResNet-based				ViT-based	
Hyperparameter	**CIFAR-10**	**CIFAR-100**	**Tiny-ImageNet**	**ImageNet**	**CIFAR-100**	**ImageNet-R**
Dataset Configuration						
Dataset	cifar10	cifar100	tiny-imagenet	imagenet	cifar100	ImageNet-R
Number of Classes	10	100	200	1000	100	200
New Classes per Task	2	10	20	100	10	20
Total Tasks	5	10	10	10	10	10
Batch Size	128	128	128	256	128	128
Validation Size	500	500	100	5000	500	500
Replay Buffer Size	0.2K	2K	2K	5K/10K/20K	2K	2K
Model Architecture						
Model	ResNet-32	ResNet-32	ResNet-18	ResNet-18	ViT-B/16	ViT-B/16
Image Size	32	32	64	224	224	224
Temperature Scaling						
Calibration Epochs	100	100	100	100	50	50
Calibration Batch Size	128	128	128	128	128	128
Temp Learning Rate	0.1	0.1	0.1	0.1	0.03	0.03
MT-CC						
Number of Groups	2	2	2	2	2	2
Group Data Ratio	0.5	0.5	0.5	0.5	0.5	0.5
Grouping Network						
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.01	0.01	0.01	0.01	0.001	0.001
Epochs	50	50	50	50	50	50
Temperature Initialization						
Init Values	[1.5, 1.5]	[1.5, 1.5]	[1.5, 1.5]	[1.5, 1.5]	[1.0, 1.0]	[1.0, 1.0]
Groupwise DCA						
GDCA Weight	1.0	1.0	1.0	1.0	0.1	0.1

A Training Hyperparameters and Implementation Details

Implementation details. All calibration hyperparameters used for MT-CC and the corresponding baselines are summarized in Table 7. Calibration is performed post-hoc and independently of backbone training. For the underlying CIL methods, we follow their original training protocols unless otherwise specified. For ER, CIFAR-10 and CIFAR-100 are trained for 200 epochs with an initial learning rate of 0.1 and step decay at epochs 100 and 150, using momentum 0.9 and an ℓ_2 weight decay of 2×10^{-4}. Tiny-ImageNet follows the same optimizer and learning-rate schedule with a batch size of 256. Following the original DER protocol, DER additionally uses a 50-epoch decoupled stage with an initial learning rate of 0.05 and decay at epochs 15 and 30, together with a distillation temperature of 5.0. Its pretraining stage is run for 200 epochs with an initial learning rate of 0.1 and decay at epochs 60, 120, and 160. For the remaining CIL baselines, we follow the hyperparameters reported in their original publications. Before being provided to the grouping network, the logit margin is min-max normalized to ensure a comparable input scale with the other calibration features. To ensure positive temperature values and stable optimization, each group-specific temperature is parameterized in log-space. For each group $m \in \{1, \dots, M\}$, we introduce an unconstrained parameter $\tau_m \in \mathbb{R}$ and define $T_m = \exp(\tau_m)$. Accordingly, $\frac{\partial \mathcal{L}}{\partial \tau_m} = \frac{\partial \mathcal{L}}{\partial T_m} T_m$. At the first task, τ_m is initialized from the predefined initial temperature, while at subsequent tasks it is initialized from the optimized temperature obtained at the previous task.

Grouping network input dimensionality. The grouping network takes the full softmax probability vector $\mathbf{p}$ as part of its input. To keep its input dimensionality fixed throughout continual learning, we represent $\mathbf{p}$ in the full class space and zero-pad the entries corresponding to unseen classes. Therefore, although the number of encountered classes increases over tasks, the dimensionality of the grouping-network input remains unchanged. Since the grouping network consists of only a lightweight fully connected layer, this fixed-dimensional representation introduces negligible additional overhead.

A.1 Representative Replay-Based CIL Methods

We provide brief descriptions of the replay-based CIL methods evaluated in our experiments.

EEIL performs end-to-end incremental learning by jointly optimizing the classification loss and a knowledge distillation objective over replayed exemplars, mitigating feature drift across tasks.

iCaRL combines exemplar replay with knowledge distillation and a nearest-mean-of-exemplar classifier, preserving old knowledge and stable decision boundaries across tasks.

BiC introduces a lightweight bias-correction layer to compensate for logit shifts from old/new class imbalance, trained on a held-out validation set to reduce bias accumulation.

IL2M addresses score decay in old classes by leveraging stored classifier norms and historical prediction statistics to rescale logits post-hoc, reducing imbalance-induced degradation.

DER adopts a dynamically expandable architecture that freezes previously learned feature extractors and allocates new branches for incoming tasks, preserving prior knowledge by structural isolation rather than distillation alone. All methods follow their original protocols, with MT-CC applied as a post-hoc calibration module on top of each trained backbone.

A.2 Details of Calibration Metrics

This section provides detailed definitions of the calibration metrics used in the main paper. We also introduce the multiclass Brier score, which evaluates probabilistic predictions without binning, and report the corresponding results later in Appendix B.5.

Expected Calibration Error (ECE) partitions predictions into Q confidence bins and computes

$$\mathrm{ECE} = \sum_{i=1}^{Q} \frac{|B_i|}{N} \, |A_i - C_i|,$$

where $|B_i|$ is the number of samples in bin i, N is the total number of samples, and A_i and C_i denote the average accuracy and confidence in bin i, respectively.

Adaptive Expected Calibration Error (AECE) employs adaptive binning based on the cumulative confidence distribution within each test domain:

$$\mathrm{AECE} = \sum_{i=1}^{Q} \frac{|\tilde{B}_i|}{N} \, |A_i - C_i|,$$

where $|\tilde{B}_i|$ denotes the number of samples in adaptive bin i.

Static Calibration Error (SCE) extends calibration evaluation to the multiclass setting:

$$\mathrm{SCE} = \frac{1}{K} \sum_{k=1}^{K} \sum_{b=1}^{Q} \frac{n_{bk}}{N} \left| \mathrm{acc}(b,k) - \mathrm{conf}(b,k) \right|,$$

where K is the number of classes, n_{bk} is the number of predictions in bin b for class k, and $\mathrm{acc}(b,k)$ and $\mathrm{conf}(b,k)$ are the accuracy and confidence of that bin.

Brier Score measures the mean squared difference between the predicted probability distribution and the one-hot encoded ground-truth label:

$$\mathrm{Brier} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(p_{n,k} - \mathbf{1}[y_n = k] \right)^2,$$

where $p_{n,k}$ is the predicted probability of class k for sample n, and $\mathbf{1}[y_n = k]$ is one if the ground-truth label y_n is class k and zero otherwise. A lower Brier score indicates better probabilistic predictions.

B Additional Ablation Studies

B.1 Ablation on the Number of Groups

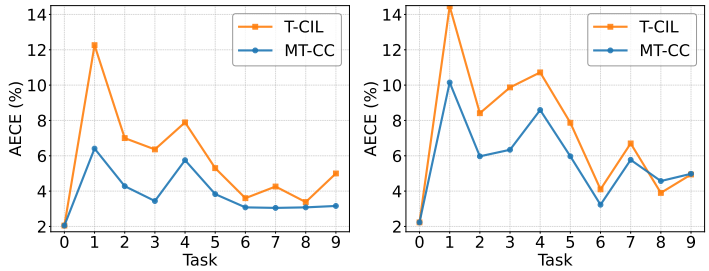
Table 8: Ablation on the number of groups M on CIFAR-100 over five seeds. $M = 1$ corresponds to single-temperature scaling (T-CIL).

M	ECE ↓	AECE ↓	Brier ↓
1 (T-CIL)	6.03±0.42	6.04±0.35	0.570±0.005
2	**4.49±0.53**	**4.44±0.49**	**0.556±0.006**
3	4.51±0.47	4.48±0.45	0.558±0.007
4	4.50±0.41	4.50±0.42	0.559±0.007
5	4.79±0.40	4.77±0.41	0.561±0.007

We evaluate the sensitivity of MT-CC to the number of groups M by varying M from 1 to 5, where $M = 1$ corresponds to single-temperature scaling (T-CIL). As shown in Table 8, nearly all calibration improvement occurs from $M = 1$ to $M = 2$ (ECE: $6.03 \rightarrow 4.49$); $M = 3$ and $M = 4$ provide no meaningful additional gains across metrics, while $M = 5$ degrades performance. Thus, our choice of $M = 2$ is based on the observed saturation of performance rather than evaluating only two groups.

One possible reason is that increasing M introduces more group-specific temperatures to optimize, making the joint routing-temperature optimization more complex. At the same time, the calibration data are partitioned among more groups, leaving fewer samples to reliably estimate each temperature, particularly for sparsely populated groups. Consistent with this explanation, when $M > 2$, the learned routing effectively collapses to two dominant groups, leaving the remaining groups sparsely populated and their temperatures poorly estimated.

B.2 Ablation on the Replay Memory Size

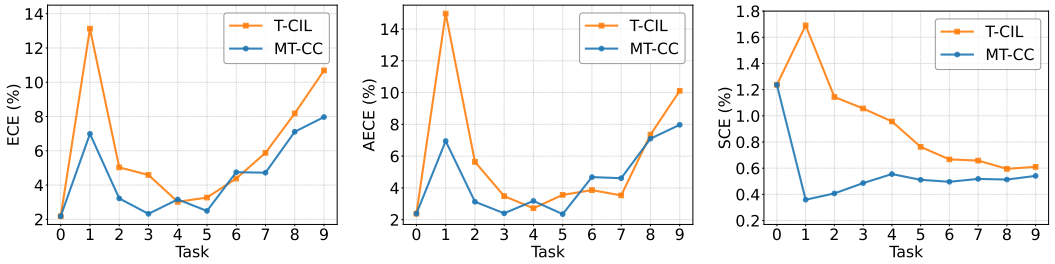


(a) Replay buffer 3K ECE (b) Replay buffer 4K ECE

Figure 6: Ablation study on replay buffer size on CIFAR-100. We report ECE for each task when using a 3K or 4K replay memory.

We vary the replay buffer size while keeping the same CIFAR-100 experimental setup as in the main paper. Across all buffer configurations, our method maintains consistently strong calibration performance. Moreover, MT-CC outperforms T-CIL on nearly all tasks, demonstrating its robustness to memory size changes.

B.3 Comparison of ECE, AECE, SCE Across Sequential Tasks on CIFAR-100



(a) ECE (b) AECE (c) SCE

Figure 7: Comparison of calibration metrics across sequential tasks using CIFAR-100.

Figure 7 presents a comparison of ECE, SCE, and AECE across sequential tasks on CIFAR-100. MT-CC consistently achieves lower calibration errors than T-CIL across all three metrics throughout the continual learning process.

B.4 Results on CIFAR-100: 60-5 Split

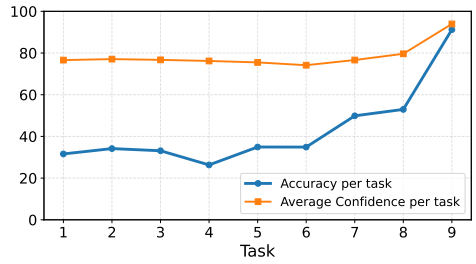


Figure 8: Per-task accuracy and confidence on the final step of the CIFAR-100 60-5 split, illustrating the discrepancy between confidence and accuracy across tasks.

Method	Acc. (↑)	ECE (↓)	AECE (↓)
Vanilla	54.84±0.41	32.45±0.35	32.45±0.36
TS	54.84±0.41	21.15±0.84	21.12±0.84
ETS	54.84±0.41	22.32±0.46	22.31±0.45
IRM	54.84±0.41	23.29±0.54	23.28±0.53
T-CIL	54.84±0.41	5.48±0.32	5.48±0.33
T-CIL + ETS	54.84±0.41	5.89±0.34	5.87±0.38
T-CIL + IRM	54.84±0.41	6.98±0.69	7.00±0.66
MT-CC	54.84±0.41	**4.79±0.31**	**4.80±0.30**

Table 9: Calibration performance on CIFAR-100 under the 60-5 split. All values are averaged over all nine steps.

We further evaluate our method on the CIFAR-100 60-5 split, where the model is first trained on 60 base classes and then updated over eight incremental steps of five new classes each. This setting enlarges the initial task and induces a more imbalanced class distribution than the standard 10-task protocol, allowing us to verify that the proposed calibration framework remains effective under a different and more challenging CIL configuration. Except for the change in the number of classes per step, experimental settings follow the standard CIFAR-100 protocol.

Figure 8 compares per-task accuracy and confidence after the final incremental step. The results show pronounced discrepancies across tasks, where confidence remains consistently high while accuracy varies substantially, revealing stronger asymmetric miscalibration than in the standard 10-task configuration. This imbalance increases calibration difficulty because the model assigns similar confidence levels to tasks that differ widely in accuracy.

Despite this more challenging regime, MT-CC achieves the lowest ECE and AECE among all baselines in Table 9. MT-CC surpasses T-CIL as well as T-CIL combined with ETS or IRM. This shows that learned grouping and multi-temperature calibration generalize beyond the standard protocol and remain effective under enlarged and imbalanced task structures.

B.5 Robustness to ECE Binning and Brier Score

Table 10: Calibration performance on CIFAR-10, CIFAR-100, and Tiny-ImageNet under different numbers of ECE bins and the multiclass Brier score. Lower values indicate better calibration. Results are averaged over five seeds.

	CIFAR-10			CIFAR-100			Tiny-ImageNet		
Method	ECE@15 ↓	ECE@20 ↓	Brier ↓	ECE@15 ↓	ECE@20 ↓	Brier ↓	ECE@15 ↓	ECE@20 ↓	Brier ↓
T-CIL	15.36±1.39	15.37±1.40	0.498±0.019	6.19±0.39	6.27±0.38	0.570±0.005	19.04±0.42	19.13±0.41	0.883±0.004
T-CIL+ETS	14.01±1.31	14.05±1.30	0.497±0.019	5.80±0.45	5.87±0.44	0.571±0.004	22.21±0.21	22.25±0.21	0.901±0.003
T-CIL+IRM	14.90±0.95	14.83±1.11	0.506±0.019	6.88±0.66	6.97±0.67	0.571±0.006	18.08±0.42	18.05±0.51	0.884±0.005
T-CIL+AdaTS	14.31±1.44	14.34±1.44	0.501±0.023	7.76±0.50	7.86±0.51	0.586±0.004	21.75±1.25	21.80±1.28	0.905±0.012
MT-CC	**10.22**±1.51	**10.27**±1.48	**0.447**±0.015	**4.62**±0.51	**4.72**±0.54	**0.556**±0.006	**15.87**±0.37	**15.93**±0.36	**0.839**±0.003

To ensure that the observed calibration gains are not artifacts of specific metric configurations, we investigate the binning sensitivity of our evaluation. Specifically, we re-evaluate all methods on CIFAR-100 using 15 and 20 bins for ECE. Furthermore, to provide a strictly binning-free assessment, we incorporate the multiclass Brier score. We use the same trained models and calibrated probabilities across all evaluations, and report the mean and standard deviation over five seeds.

As shown in Table 10, MT-CC consistently achieves the lowest ECE across all three bin configurations, and the relative ranking of the methods remains unchanged as the number of bins increases. Crucially, MT-CC also obtains the lowest multiclass Brier score, confirming its overall robustness.

B.6 Applicability to Static Classification

Table 11: Comparison of single-temperature scaling and MT-CC in static and class-incremental learning settings on CIFAR-100 over five seeds.

Setting	Single-T ECE ↓	MT-CC ECE ↓
Static joint training	0.87 ± 0.14	0.87 ± 0.14
CIL (task average)	6.03 ± 0.42	$\mathbf{4.49 \pm 0.53}$

Although groupwise temperature scaling is applicable to static classification, its benefit in MT-CC arises from modeling the asymmetric calibration behavior induced by continual learning. To verify that the improvement does not simply result from using multiple temperatures, we apply the same calibration pipeline to a jointly trained static CIFAR-100 model.

As shown in Table 11, MT-CC provides no additional improvement over single-temperature scaling in the static setting. The learned router effectively collapses to a single group. Specifically, the fraction of samples assigned to the smaller group is 0.000 ± 0.000, and the active-group temperature differs from the single-temperature solution by less than 0.003 in every seed. MT-CC therefore reduces to standard single-temperature scaling in this setting.

In contrast, under CIL, MT-CC reduces the task-averaged ECE from 6.03 to 4.49. This comparison confirms that MT-CC effectively targets the unique calibration shifts inherent to CIL, rather than merely benefiting from increased parameter capacity.

B.7 Effect of Validation Set Size

Table 12: Calibration performance under varying validation set sizes (lower is better).

Validation Size	**0**	**500**	**1000**	**1500**	**2000**
T-CIL	16.13±0.19	6.19±0.66	6.38±0.47	6.24±0.39	6.03±0.42
MT-CC	**15.96**±1.05	**5.92**±0.95	**6.04**±0.78	**5.89**±0.77	**4.49**±0.53

Table 12 reports the effect of validation set size. MT-CC consistently outperforms T-CIL, a state-of-the-art method tailored to temperature scaling in class-incremental learning. Even under limited validation data regimes, MT-CC consistently outperforms T-CIL, demonstrating robustness to calibration data scarcity. Notably, even when the validation set size is 0, MT-CC achieves an ECE of 15.96 compared to 16.13 for T-CIL, relying solely on samples from the current task for grouping network training and temperature scaling.

B.8 Inference Time Analysis

Table 13: The average per-sample inference time across all tasks on ImageNet over 3 seeds. Relative time is reported with TS as the baseline (1×). MT-CC requires negligible additional inference overhead compared to standard TS.

Method	Per-sample inference time (μs)	Relative
TS	1250.41	1.0000×
PTS	1250.65	1.0002×
AdaTS	1252.63	1.0018×
MT-CC	1250.80	1.0003×

Table 13 reports the average inference time across all tasks on ImageNet in microsecond scale, measured on a single NVIDIA RTX A6000 GPU. AdaTS incurs slightly higher inference cost than others due to the use of a class-conditional VAE. In contrast, MT-CC introduces negligible additional overhead compared to standard temperature scaling (TS). Since the grouping module is a lightweight network, the additional latency introduced by MT-CC is negligible in practice.

B.9 Comparison with Feature-Matched and Grouped Calibration Methods

MT-CC outperforms both feature-matched instance-wise temperature scaling and unsupervised group-wise temperature scaling, showing that its improvement cannot be attributed solely to the CIL-aware inputs or the use of multiple temperatures. We evaluate progressively stronger baselines that isolate the contributions of the input features, heterogeneous calibration groups, and learned calibration-aware routing.

Table 14: Comparison with feature-matched and grouped calibration methods on CIFAR-100 over 5 seeds. The CIL-aware setting uses the same feature vector as MT-CC.

Method	**Calibration / Routing**	**Feature**	**ECE** ↓	**AECE** ↓	**Brier** ↓
PTS	Instance-wise TS	Top-10 logits	9.87 ± 0.77	10.14 ± 0.67	0.596 ± 0.008
PTS	Instance-wise TS	CIL-aware features	6.88 ± 0.91	6.87 ± 0.91	0.575 ± 0.008
AdaTS	Instance-wise TS	Penultimate feature	7.67 ± 0.52	7.66 ± 0.49	0.586 ± 0.004
AdaTS	Instance-wise TS	CIL-aware features	6.55 ± 0.74	6.52 ± 0.75	0.574 ± 0.004
GC+TS [45]	Semantic-aware learned grouping + group-wise TS	Penultimate feature	7.23 ± 0.25	7.24 ± 0.26	0.576 ± 0.005
K-means Group-TS	K-means ($K = 2$) + group-wise TS	CIL-aware features	5.41 ± 0.51	5.37 ± 0.51	0.561 ± 0.005
MT-CC	Learned routing + group-wise TS	CIL-aware features	$\mathbf{4.49 \pm 0.53}$	$\mathbf{4.44 \pm 0.49}$	$\mathbf{0.556 \pm 0.006}$

Contribution of CIL-aware features. CIL-aware features provide more informative calibration signals under continual distribution shifts. We provide PTS and AdaTS with the same input vector used by MT-CC, $v(x) = [\delta, p_{\text{old}}, p_{\text{new}}, p]$. Both methods improve over their original feature settings, confirming the effectiveness of the proposed input design.

Contribution of asymmetric calibration groups. Explicitly modeling asymmetric calibration groups improves upon instance-wise temperature prediction. K-means Group-TS clusters the same CIL-aware feature vectors into two groups, optimizes a separate temperature for each cluster, and assigns each test sample the temperature of its nearest centroid. It outperforms the feature-matched PTS and AdaTS variants, supporting the use of group-specific calibration.

Contribution of learned calibration-aware routing. Learning groups according to the calibration objective is more effective than clustering the feature space independently of calibration. MT-CC outperforms K-means Group-TS even though both use the same input features and two group-specific temperatures.

Comparison with existing grouping calibration. We further compare MT-CC with GC+TS [45], a prior semantic-aware grouping calibration method developed for static classification. GC+TS learns a grouping function from deep semantic features and applies a separate temperature to each learned group. The comparison shows that CIL-aware grouping is more effective for capturing calibration heterogeneity under continual task shifts.

B.10 CIL Changes the Calibration Meaning of Logit Margin

The logit margin is strongly associated with the required calibration correction under CIL, whereas this relationship is nearly absent in static classification. For a jointly trained static CIFAR-100

model and the CIL model at the final task, we divide samples into 10 equal-count margin bins and independently fit the NLL-optimal temperature T^* in each bin.

Table 15: NLL-optimal temperature T^* across logit-margin bins on CIFAR-100 averaged over 5 seeds.

Setting	**b0 (low)**	**b1**	**b2**	**b3**	**b4**	**b5**	**b6**	**b7**	**b8**	**b9 (high)**
Static	1.98	1.99	2.00	1.95	2.00	1.94	1.96	1.92	2.04	2.10
CIL	1.58	1.68	1.77	1.84	2.00	2.13	2.36	2.59	2.99	3.82

In the static model, T^* remains nearly constant across margin bins even though accuracy increases with the margin. Under CIL, however, catastrophic forgetting degrades accuracy without a corresponding drop in prediction margin, so high-margin predictions become increasingly overconfident. Thus, the logit margin becomes a CIL-aware signal of the calibration correction required by different samples.

B.11 Complementarity of Margin and Confidence

Confidence provides additional calibration information even after the logit margin is controlled. We divide samples into two equal-count margin regions and further split each region into low- and high-confidence halves. We independently fit T^* to each subset and define ΔT^* as the difference between the optimal temperatures of the high- and low-confidence subsets.

Table 16: Optimal temperatures after controlling for the logit margin.

Setting	**Margin region**	**Low-conf.** T^*	**High-conf.** T^*	$\boldsymbol{\Delta T^*}$
Static	Low	1.98	1.99	0.01
	High	1.94	2.03	0.09
CIL	Low	1.60	1.95	0.34
	High	2.26	3.32	1.05

Confidence produces only small temperature differences in the static model but substantially larger differences under CIL. This result shows that margin and confidence capture complementary aspects of CIL-induced calibration heterogeneity.

Table 17: Benefit of explicitly providing the old/new class structure on CIFAR-100 across different grouping-set sizes. The optimization budget is matched across data sizes and S_{cal} is fixed. Results report ECE (%) and Brier score over five seeds.

$\lvert S_{\mathrm{grp}}\rvert$	$[\delta, p_{\mathrm{old}}, p_{\mathrm{new}}, p]$		$[\delta, p]$		$[\delta, p_{\mathrm{rand1}}, p_{\mathrm{rand2}}, p]$		$[\delta, p_{\mathrm{old}}, p_{\mathrm{new}}]$	
	ECE	Brier	ECE	Brier	ECE	Brier	ECE	Brier
50	**4.77±0.43**	**0.565±0.009**	4.92±0.50	0.573±0.006	5.00±0.43	0.573±0.006	5.98±0.48	0.572±0.004
100	**4.36±0.46**	**0.560±0.007**	4.69±0.40	0.571±0.005	4.81±0.45	0.571±0.006	5.78±0.30	0.571±0.005
250	**4.49±0.45**	**0.558±0.006**	4.70±0.27	0.572±0.005	4.73±0.34	0.572±0.006	5.74±0.33	0.570±0.005
500	**4.54±0.56**	**0.557±0.006**	4.72±0.41	0.571±0.006	4.71±0.37	0.571±0.006	5.83±0.30	0.571±0.005
1000	**4.49±0.53**	**0.556±0.006**	4.63±0.30	0.571±0.007	4.59±0.40	0.571±0.007	5.82±0.31	0.571±0.005

B.12 Benefit of Explicit Old/New Class Structure

Although p_{old} and p_{new} are linear functions of the full softmax vector p and therefore do not increase the expressiveness of the linear grouping network, explicitly providing them exposes the old-versus-new class structure directly to the router. We investigate whether explicitly exposing this structure benefits calibration despite the same information already being contained in p.

Input variants. To isolate the effect of explicitly providing the old/new class structure, we compare the following input configurations:

- $[\delta, p_{\mathrm{old}}, p_{\mathrm{new}}, p]$ **:** The full MT-CC input, which combines the logit margin, explicit old/new probability masses, and the full softmax vector.
- $[\delta, p]$ **:** Removes the explicit old/new probability masses while retaining the logit margin and full softmax vector.
- $[\delta, p_{\mathrm{rand1}}, p_{\mathrm{rand2}}, p]$ **:** Replaces the old/new probability masses with probability masses computed from a size-matched random partition of the classes. Since these random class-mass features are also linear functions of p, this variant controls for the effect of simply adding redundant aggregate features.
- $[\delta, p_{\mathrm{old}}, p_{\mathrm{new}}]$ **:** Retains the logit margin and explicit old/new probability masses while removing the full softmax vector.

Experimental setup. To verify that the benefit of the explicit old/new structure is not specific to a particular grouping-data regime, we vary the number of distinct samples used to train the grouping network, $|S_{\mathrm{grp}}| \in \{50, 100, 250, 500, 1000\}$, while keeping the temperature-fitting subset S_{cal} fixed. The same grouping samples are used across input variants at each data size. We match the optimization budget across grouping-set sizes so that each setting receives approximately the same number of gradient updates, isolating the effect of the number of distinct grouping samples. All other hyperparameters are kept identical.

Benefit of explicit old/new structure. As shown in Table 17, explicitly providing the old/new class structure consistently improves both ECE and Brier score over using only $[\delta, p]$ across grouping-data budgets. This indicates that the grouping network learns more effective calibration routing when the CIL-specific old/new structure is directly exposed.

CIL-specificity of the improvement. Replacing the true old/new probability masses with size-matched random class-mass summaries does not reproduce the same improvement and performs similarly to $[\delta, p]$. Thus, the gain cannot be explained merely by adding redundant linear aggregate features. Instead, it depends on explicitly exposing the old-versus-new structure aligned with continual learning.

Complementarity with the full softmax vector. Using only $[\delta, p_{\mathrm{old}}, p_{\mathrm{new}}]$ yields substantially higher ECE, showing that the old/new summaries do not replace the fine-grained class-wise information in p. Instead, they provide complementary CIL-aware structure.

Sample efficiency. The full MT-CC input achieves an ECE of 4.36% with only 100 grouping samples, comparable to 4.49% with 1,000 samples. This indicates that effective routing can be learned from a relatively small number of distinct grouping samples when the optimization budget is controlled.

B.13 Continual Update and Stability of the Grouping Network

We continually update the grouping network so that it can adapt to the evolving classifier across tasks. Continually updating the grouping network improves calibration while preserving the routing structure learned across tasks. We validate this design choice by comparing alternative update strategies and measuring group-assignment consistency across adjacent tasks.

Table 18: Average calibration performance of different grouping-network update strategies on CIFAR-100 over five seeds.

Grouping-network strategy	ECE ↓	AECE ↓	SCE ($\times 10^{-3}$) ↓	Brier ↓
MT-CC	$\mathbf{4.49 \pm 0.53}$	$\mathbf{4.44 \pm 0.49}$	$\mathbf{5.12 \pm 0.11}$	$\mathbf{0.556 \pm 0.006}$
Lag-1	4.82 ± 0.36	4.82 ± 0.36	5.30 ± 0.12	0.558 ± 0.006
Fresh each task	4.98 ± 0.35	4.96 ± 0.34	5.38 ± 0.14	0.561 ± 0.007

Effectiveness of continual router updates. Updating the previous router is more effective than either freezing it or discarding its previously learned routing structure. We compare the following three grouping-network update strategies on CIFAR-100:

- **MT-CC :** The router learned at task $t-1$ is carried forward and updated using the calibration data available at task t.
- **Lag-1 :** The router learned at task $t-1$ is used without further updates to calibrate the model at task t, while only the group temperatures are re-fitted.
- **Fresh each task :** The router is reinitialized and trained independently at every task.

As shown in Table 18, continually updating the router outperforms both freezing and reinitializing it across all calibration metrics. This validates the MT-CC design of retaining previously learned routing information while adapting it to the evolving classifier.

Stability of continual router updates. Continually updating the router causes little routing drift. When the classifier outputs are held fixed, both groups retain more than 90% of their assignments across adjacent tasks. We measure group retention over all adjacent task transitions and five seeds. Groups are ordered by their learned temperatures, with G1 denoting the lower-temperature group and G2 the higher-temperature group. For each group, retention is defined as the fraction of samples assigned to that group by the task $t-1$ router that remain in the same group under the task t router.

We evaluate retention under two conditions. In the actual transition, both the classifier and router change from task $t-1$ to task t. In the router-only condition, the task t classifier outputs are held fixed, and the task $t-1$ and task t routers are evaluated using the same outputs.

Table 19: Group retention across adjacent tasks on CIFAR-100, averaged over five seeds. G1 and G2 denote the lower- and higher-temperature groups, respectively.

Condition	G1 retention (%) ↑	G2 retention (%) ↑
Classifier and router change	71.8 ± 7.5	42.0 ± 12.4
Router change only	94.3 ± 6.3	90.3 ± 9.6

As shown in Table 19, the task $t-1$ and task t routers produce highly consistent assignments when evaluated on the same classifier outputs. The lower G2 retention in the actual transition is expected because the introduction of new classes substantially changes the composition of the high-temperature group.

C How GDCA Stabilizes Routing-Temperature Optimization

We investigate how GDCA stabilizes the coupled routing–temperature optimization. By complementing the sample-level NLL with groupwise confidence–accuracy alignment, GDCA helps the joint optimizer distinguish routing solutions with similar likelihood, guides the estimation of group-specific temperatures, and mitigates routing collapse into a single dominant group. Together, these effects explain the calibration improvement reported in the main paper. We first show that NLL alone is insufficient for calibration-relevant routing, then explain the optimization mechanism of GDCA, and finally analyze routing degeneracy with and without GDCA.

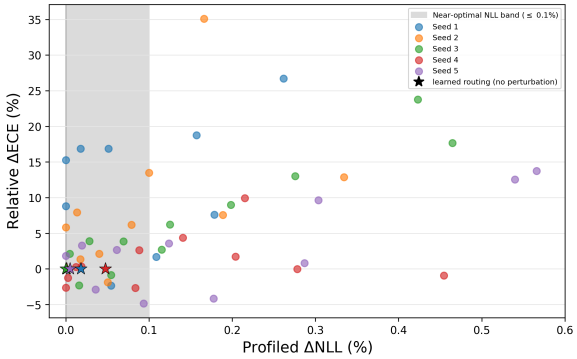


Figure 9: Profile-likelihood analysis of perturbed routing solutions at the final task on CIFAR-100 over five seeds. The group temperatures are re-optimized from scratch for each routing perturbation. For each seed, profiled $\Delta\mathrm{NLL}$ is measured relative to the minimum profiled NLL across all perturbations, while relative $\Delta\mathrm{ECE}$ is measured with respect to the learned routing. The shaded region indicates routing solutions whose profiled NLL lies within 0.1% of the seed-wise minimum.

NLL alone is insufficient for calibration-relevant routing. Optimizing the routing with NLL alone may not distinguish routing solutions with different calibration quality. Even after re-optimizing the temperatures for each routing configuration, substantially different routing solutions can achieve nearly identical NLL values. To examine this limitation, we perturb the learned routing and re-optimize the temperatures from scratch at each perturbation point.

As shown in Figure 9, some routing solutions within the near-optimal NLL band produce a relative increase of more than 15% in test ECE compared with the learned routing. Thus, NLL can be nearly indifferent among routing solutions with materially different calibration quality, making it insufficient as the sole objective for learning calibration-relevant routing.

Optimization mechanism. Each group G_m is constructed using a hard routing assignment, and thus gradients do not pass through the group membership or the empirical accuracy $\mathrm{Acc}(G_m)$. Instead, they are propagated through $\mathrm{Conf}(G_m)$, which is computed from the calibrated mixture confidence using the soft routing weights. Consequently, GDCA directly adjusts the group temperatures toward groupwise confidence–accuracy alignment. Once the temperatures become differentiated, the mixture confidence also becomes sensitive to the routing weights, allowing GDCA to update the router indirectly through the mixture confidence while jointly optimizing the temperatures.

Table 20: Routing-group sizes with and without GDCA over five seeds. The minority-group fraction denotes the proportion of test samples assigned to the less populated of the two groups.

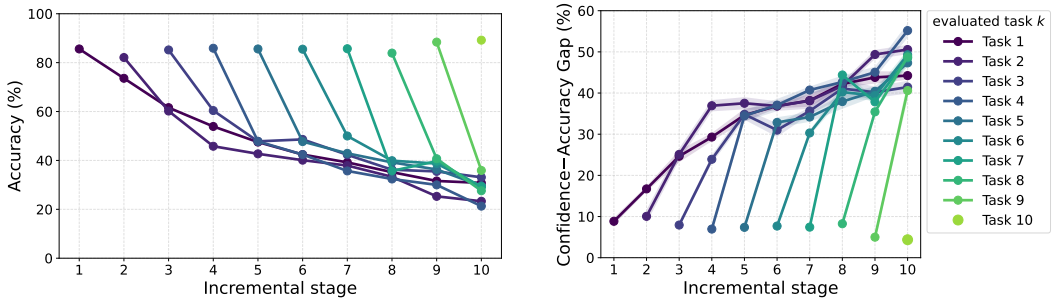
Routing behavior	**w/o GDCA**	**with GDCA**
Frequency of minority-group fraction below 20%	20.0%	4.4%
Mean minority-group fraction	29.7%	37.5%

Routing degeneracy with and without GDCA. During training, the router may assign most samples to one dominant group, causing the other group to become sparsely populated. As shown in Table 20, GDCA reduces the frequency with which the minority-group fraction falls below 20% from 20.0% to 4.4%, while increasing its mean fraction from 29.7% to 37.5%. These results indicate that GDCA mitigates routing degeneracy and maintains sufficiently populated groups for more reliable group-specific temperature scaling.

D Catastrophic Forgetting and Asymmetric Calibration Behaviors

Table 21: Average first-task performance degradation over 5 runs under the CIL setup.

Dataset	After First Task		After Middle Task		After Final Task	
	Acc (%)	Conf (%)	Acc (%)	Conf (%)	Acc (%)	Conf (%)
CIFAR-10	98.79	96.06	72.69	92.09	20.37	91.38
CIFAR-100	85.62	92.60	42.38	79.14	30.84	75.08
Tiny-ImageNet	70.60	88.47	9.10	68.69	4.70	69.82



(a) Per-task accuracy across incremental stages. Accuracy degradation follows different trajectories across tasks, indicating asymmetric forgetting dynamics.

(b) Per-task confidence–accuracy gaps. The gaps generally widen at different rates across tasks, revealing asymmetric calibration behaviors.

Figure 10: Catastrophic forgetting and asymmetric calibration behaviors on CIFAR-100. Each curve tracks a task from its introduction through subsequent incremental stages.

Table 21 shows that catastrophic forgetting is accompanied by a growing mismatch between confidence and accuracy. For the first task of CIFAR-10, accuracy decreases from 98.79% to 20.37%, whereas confidence decreases only from 96.06% to 91.38%. Consequently, the confidence–accuracy gap reaches 71.01 percentage points after the final task. CIFAR-100 and Tiny-ImageNet exhibit the same pattern: accuracy declines substantially more than confidence. These results show how forgetting can contribute to increasing overconfidence when confidence does not decrease in proportion to the loss of predictive accuracy.

Figure 10a further shows that the trajectories of accuracy degradation differ across tasks. Figure 10b complements this analysis by directly tracking the resulting confidence–accuracy gaps for all tasks on CIFAR-100. The gaps generally increase after subsequent tasks are introduced, but the increase is neither uniform nor strictly monotonic: different tasks exhibit distinct rates of increase and trajectories. Thus, the mismatch between confidence and accuracy evolves differently across tasks under continual updates, revealing asymmetric calibration behaviors.

Taken together, these results connect catastrophic forgetting with asymmetric calibration: predictive accuracy deteriorates without a proportional decrease in confidence, and the resulting gap evolves differently across tasks. This motivates the use of multiple temperatures in MT-CC to accommodate heterogeneous calibration needs.

E Training and Inference Procedure of MT-CC

Algorithm 1 Training Procedure of MT-CC

Input: Task sequence $\{\mathcal{D}^1, \ldots, \mathcal{D}^B\}$; calibration sets $\{\mathcal{C}^1, \ldots, \mathcal{C}^B\}$; number of groups M
Output: Grouping network g_ϕ and group temperatures $\{T_m^*\}_{m=1}^M$

```
Initialize grouping network g_φ and group temperatures {T_m^0}_{m=1}^M
for b = 1 to B do
    Split C^b into two disjoint subsets C^b → (S^b_grp, S^b_cal)
    Initialize {T_m^b}_{m=1}^M ← {T_m^{b-1}}_{m=1}^M
    For each x ∈ S^b_grp, compute v(x) = [δ, p_old, p_new, p]
    Update g_φ and {T_m^b}_{m=1}^M by minimizing L_total = L_MT-CC + λL_GDCA
    Assign each x ∈ S^b_cal to m*(x) = arg max_m g_φ(x)_m
    For each group m, refine T_m^b on its assigned samples in S^b_cal by minimizing NLL
end for
return g_φ and {T_m^B}_{m=1}^M
```

Training. Algorithm 1 summarizes MT-CC training. For each task, calibration data are split into grouping and calibration subsets. We jointly optimize the grouping network and group temperatures using $\mathcal{L}_{\text{total}}$, then freeze g_ϕ and refine each group temperature with NLL. The resulting temperatures initialize the next task.

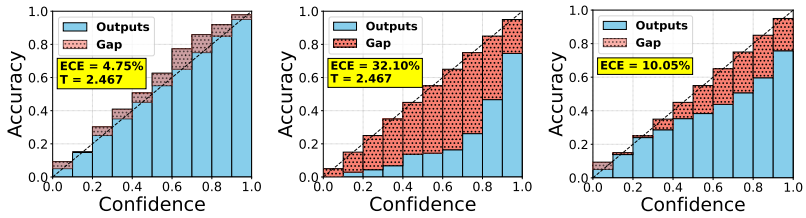
Algorithm 2 Inference Procedure of MT-CC

Input: Grouping network g_ϕ, optimized group temperatures $\{T_m^*\}_{m=1}^M$, test sample x and global temperature T_{glob}^*
Output: Calibrated probability vector $p_{\text{cal}}(x)$

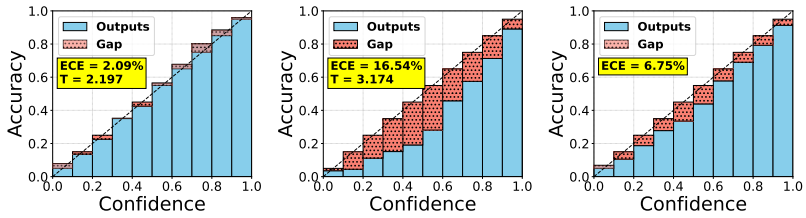
```
if b = 1 then
    p_cal(x) ← softmax( f_θ(x) / T*_glob )
else
    g*(x) ← arg max_m g_φ(x)_m
    p_cal(x) ← softmax( f_θ(x) / T*_{g*(x)} )
end if
```

Inference. Algorithm 2 summarizes the inference procedure of MT-CC. For the first task, the logits $f_\theta(x)$ are calibrated using a single optimized global temperature T_{glob}^* through standard temperature scaling. For subsequent tasks, the grouping network g_ϕ assigns each test sample to the group with the highest predicted probability. The calibrated confidence is then obtained by applying the corresponding optimized group-specific temperature to the logits before the softmax operation.

F Reliability Diagram



(a) Single T (group 1) (b) Single T (group 2) (c) Single T (overall)



(d) MT-CC (group 1) (e) MT-CC (group 2) (f) MT-CC (overall)

Figure 11: **(a), (b), (c):** Reliability diagrams of T-CIL [16] using a single temperature T. **(d), (e), (f):** Reliability diagrams of the proposed MT-CC with optimized grouping. Results are shown after training on the 10th task of CIFAR-100, comparing group 1, group 2, and all classes as determined by the proposed grouping function.

F.1 Group-wise Reliability Diagrams at the Final Task

Figure 11 compares the reliability diagrams of T-CIL and MT-CC after training on the 10th (final) task of CIFAR-100, separately for group 1, group 2, and the overall distribution determined by the proposed grouping network. With a single global temperature, T-CIL exhibits clearly distinct miscalibration patterns between the two groups: one group remains overconfident while the other becomes underconfident, so a single T cannot satisfy both simultaneously, leaving residual gaps in the overall reliability diagram. In contrast, MT-CC assigns a group-specific temperature to each subset of samples, substantially reducing the gap between confidence and accuracy within both groups and yielding a tighter alignment with the diagonal in the overall reliability diagram. This visually confirms our central claim that addressing the asymmetric calibration behaviors of CIL with multiple temperatures leads to more effective calibration than the single-temperature strategy.

F.2 Reliability Diagrams across Sequential Tasks

Figure 12 presents the reliability diagrams for CIFAR-100 from Task 2 to Task 10, comparing T-CIL and MT-CC across Group 1, Group 2, and the overall distribution. The diagrams illustrate how closely the model's predicted confidence aligns with the empirical accuracy, where curves closer to the diagonal indicate better calibration. Across most tasks, MT-CC exhibits consistently stronger overall calibration than T-CIL, confirming its advantage.

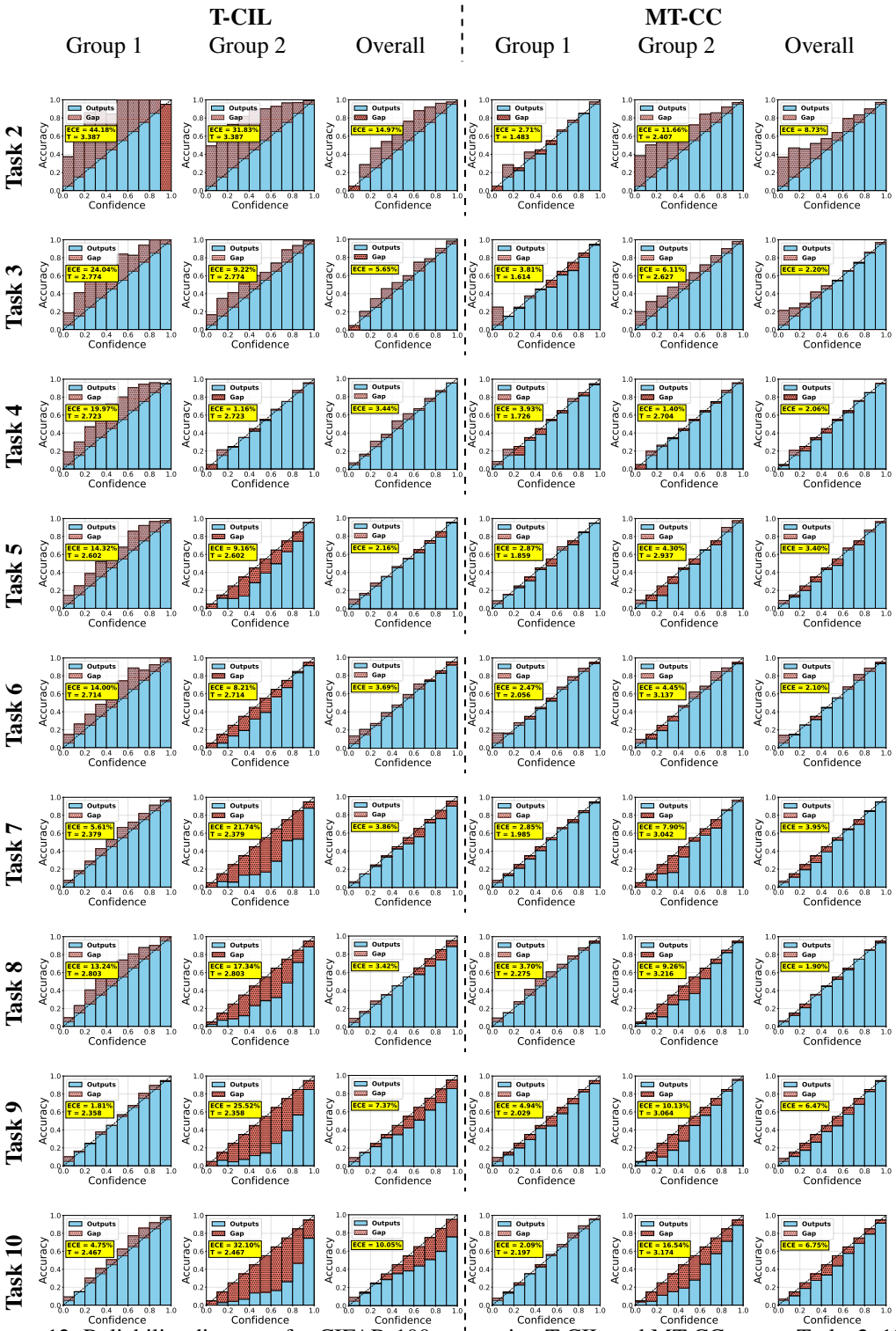


Figure 12: Reliability diagrams for CIFAR-100 comparing T-CIL and MT-CC across Tasks 2–10. Overall calibration performance shows consistently strong reliability across most tasks.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 explicitly state our three contributions: (i) identifying asymmetric calibration behaviors in CIL, (ii) MT-CC with multiple group-specific temperatures determined by a lightweight grouping network, and (iii) the Groupwise DCA (GDCA) regularization. Each claim is empirically supported in Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations and future directions at the end of the Conclusion (e.g., the lack of theoretical analysis of MT-CC and the extension to other continual learning domains such as large language models).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results requiring proofs; MT-CC is an empirical post-hoc calibration framework, and all claims are validated experimentally.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 4 describes the full MT-CC algorithm (grouping network architecture, input features, training procedure, GDCA loss), and Appendix A reports the complete set of dataset configurations, optimizer settings, and hyperparameters for both the backbone CIL methods and the calibration optimization, sufficient to reproduce all main results.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The implementation is publicly available on GitHub. The algorithm and training/calibration hyperparameters are described in Section 4 and Appendix A. All experiments use publicly available benchmarks, including CIFAR-10/100, Tiny-ImageNet, and ImageNet.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 describes datasets, CIL splits, baselines, and evaluation metrics, and Appendix A (Table of training hyperparameters) reports the complete optimizer, batch size, learning rate, replay buffer size, validation split, and calibration-specific hyperparameters for every dataset and method.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Most main calibration results are averaged over 5 random seeds, and the result tables report mean $\pm$ standard deviation across seeds. The only exception is the inference-time experiment, which is averaged over 3 random seeds due to its computational cost. Even in this 3-seed setting, the reported standard deviations are very small, indicating that the measurements are stable. The variability captured here corresponds to randomness in the calibration optimization, such as grouping-network initialization and validation-batch ordering; the underlying CIL backbones are held fixed.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments were conducted on a single NVIDIA RTX A6000 GPU. Because MT-CC is a post-hoc calibration method that does not retrain the backbone, each calibration run (per seed, per configuration) completes in roughly 5 minutes given precomputed logits, the trained backbone, and the validation set; the 5-seed total compute therefore remains modest. The inference-time analysis on ImageNet is reported in Appendix G.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: The work uses standard public image classification benchmarks (CIFAR-10/100, Tiny-ImageNet, ImageNet) and involves no human subjects, sensitive personal data, or deployed systems. We have reviewed the NeurIPS Code of Ethics and the research conforms with it.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The paper proposes a post-hoc confidence-calibration framework for class-incremental learning evaluated on standard image-classification benchmarks. It is foundational methodological research with no direct deployment scenario; the only foreseeable societal effect is the indirect positive contribution of more trustworthy confidence estimates in continually-updated classifiers, and we are not aware of any specific negative societal impact that warrants dedicated discussion.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We do not release any pretrained generative models, scraped datasets, or other artifacts with high misuse risk; the contribution is a post-hoc calibration method evaluated on existing public benchmarks.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets used (CIFAR-10/100, Tiny-ImageNet, ImageNet) are standard public benchmarks cited via their original papers and used under their standard academic-use terms. Baseline calibration and CIL methods, including T-CIL [16], ETS, IRM, PTS, AdaTS, and grouping calibration, are properly cited.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The MT-CC implementation is publicly available on GitHub. The method and experimental settings are documented in Section 4 and Appendix A.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve research with human subjects, so IRB review is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were used only for editing and language polishing of the manuscript and are not part of the core methodology.

Guidelines:

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