Class Adaptive Conformal Training

Badr-Eddine Marani [*1,2] **Julio Silva-Rodríguez** [3] **Ismail Ben Ayed** [3]
Maria Vakalopoulou [4,5] **Stergios Christodoulidis** [4,5] **Jose Dolz** [3]

[1] Renault Group [2] U2IS Laboratory, ENSTA, Institut Polytechnique de Paris
[3] ÉTS Montréal
[4] Université Paris-Saclay, CentraleSupélec, Gustave Roussy, INSERM, IHU PRISM, Cancer Data Science Unit, France
[5] Université Paris-Saclay, CentraleSupélec, MICS Laboratory, France

Abstract

Deep neural networks have achieved remarkable success across a variety of tasks, yet they often suffer from unreliable probability estimates, leading to overconfident predictions. Conformal Prediction (CP) offers a principled framework for uncertainty quantification, yielding prediction sets with rigorous coverage guarantees. Recent work has proposed integrating conformal objectives into training, optimizing for overall set size. Nevertheless, shaping the prediction sets in a class-conditional manner is not straightforward and typically requires prior knowledge of the data distribution. While class-conditional efficiency has been studied at the post-hoc calibration level (*e.g.*, via class-conditional scoring or calibration partitioning), no prior conformal *training* method explicitly addresses it. To fill this gap, we introduce Class Adaptive Conformal Training (CaCT), which formulates conformal training as an augmented Lagrangian optimization problem that adaptively shapes prediction sets class-conditionally without relying on parametric distributional assumptions, beyond a user-specified global coverage constraint. Experiments on multiple benchmark datasets, including standard and long-tailed image recognition as well as text classification, demonstrate that CaCT consistently outperforms prior conformal training methods, reducing prediction set size while maintaining the desired coverage guarantees.

1 Introduction

Deploying machine learning based systems in high-stakes, real-world scenarios requires not only strong predictive performance but also reliable measures of uncertainty. Conformal prediction (CP) has emerged as a principled approach to meet this requirement [1, 2]. Given a pre-trained model, a calibration dataset, and a new test example $X_{\text{test}} \in \mathcal{X}$ with unknown label $Y_{\text{test}} \in \mathcal{Y}$, CP outputs prediction set $\mathcal{C}(X_{\text{test}}) \subseteq \mathcal{Y}$, which is agnostic to the model and data distribution, and statistically guaranteed to include the correct output with a user-defined miscoverage level $\alpha \in [0, 1]$:

$$\Pr(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \geq 1 - \alpha. \tag{1}$$

This distribution-free guarantee is a significant property, as it ensures the validity of the prediction sets. Yet, while validity is crucial, the practical value of a prediction set depends on its efficiency, which is typically evaluated as the expected size of the predicted sets $\mathbb{E}_X|\mathcal{C}(X)|$. For instance, a trivial set that contains all labels is *valid* but *uninformative*. Moreover, CP sets should exhibit *adaptivity*, producing compact sets for instances deemed easy by the model and larger ones for more uncertain or ambiguous cases.

Methods for enhancing the efficiency of conformal predictors span from designing effective non-conformity scores [3–8] to explicit optimization-based strategies [9–13]. While the former approach

*Work done during an internship at CentraleSupélec. Corresponding author: badr-eddine.marani@outlook.com.

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

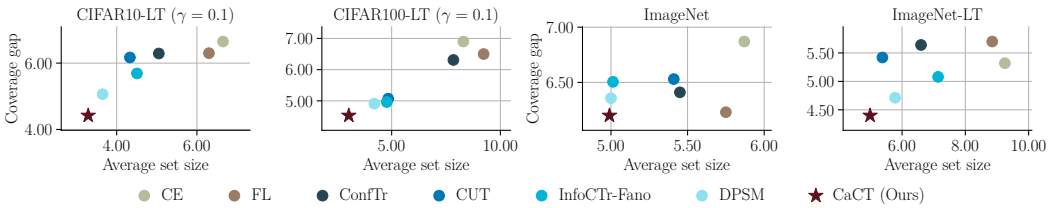

Figure 1: **Existing conformal training methods fail to account for datasets with a large number of classes or long-tailed distributions.** We compare **CaCT** against conformal training methods, and report coverage gap and set size across relevant benchmarks. Lower values for both metrics indicate better conformalization and more informative prediction sets (*i.e.*, better *efficiency*) across all classes. Among all methods, **CaCT** exhibits superior performance.

enjoys widespread adoption, applying CP solely as a post-training procedure prevents the underlying model from adapting to the construction of confidence sets [9]. Consequently, training conformal models end-to-end has emerged as a particularly promising strategy for achieving both validity and improved efficiency. In particular, ConfTr [9] represents one of the pioneer efforts to integrate model training with conformal prediction end-to-end. By resorting to smooth implementations of recent non-conformity scores, ConfTr introduces a differentiable formulation of CP, allowing the adoption of existing CP methods during training. Compared to standard training, this strategy reduces the average confidence set size of state-of-the-art CP approaches.

Nevertheless, although ConfTr advocates for improved efficiency, their adaptivity remains fundamentally constrained by the underlying formulation. To illustrate this limitation, let us examine its main learning objective, which introduces a regularization term to control efficiency by penalizing the size of the prediction set. This penalty is modulated by a single hyperparameter λ, applied uniformly across all instances and classes regardless of their difficulty. While simple, this design makes the model's performance highly sensitive to the choice of λ, especially in large-scale or long-tailed settings. In addition, this weight is typically set before training and remains fixed, preventing the model from learning an optimal trade-off between accuracy and conformalization. Such a class-agnostic, rigid formulation is suboptimal, as different classes often exhibit distinct underlying characteristics. This limitation is further exacerbated in long-tailed scenarios, a common setting in real-world applications, where the majority classes often dominate the optimization dynamics, leading to biased conformalization and reduced adaptivity for underrepresented classes (Fig. 2).

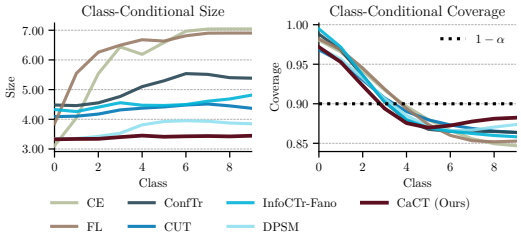

Figure 2: **Impact of per-class conformal set size and coverage with class frequency** ($\alpha = 0.1$) **in CIFAR10-LT**. As classes become underrepresented (from head to tail classes, moving toward the right of the x-axis), existing methods deteriorate their conformal performance, *i.e.*, increase set size and decrease class-conditional coverage. In contrast, CaCT is less sensitive to class imbalance, yielding smaller class set sizes (*left*) yet maintaining higher class coverage (*right*) for these classes.

In light of the above limitations, we propose a novel conformal training strategy for *efficient* and *adaptive* conformal prediction. The key contributions can be summarized as:

- We identify a structural limitation of existing conformal training methods: using a single global penalty weight systematically biases optimization toward majority classes and inflates prediction sets for minority or difficult classes, particularly under class imbalance. This problem cannot be resolved by simply tuning the global weight (illustrated in Figs. 2 and 3a).
- We introduce Class Adaptive Conformal Training (CaCT), a framework that addresses this limitation by reformulating conformal training as a class-wise constrained optimization problem, and employing class-wise multipliers instead of a single global weight to construct *efficient* and *adaptive* prediction sets. This solves the issues highlighted above: (i) scaling to datasets with many classes, which exhibit diverse intrinsic difficulties, and (ii) learning effectively under class imbalance.

- The resulting constrained optimization problem is efficiently solved by the proposed modified Augmented Lagrangian algorithm, which learns adaptive weights for the class-wise penalties. Unlike prior conformal training methods, CaCT shapes class-conditional prediction sets without designing new loss functions or assumptions about the data distribution, and naturally handles datasets with many classes (*e.g.*, ImageNet), while mitigating class imbalance issues (*e.g.*, ImageNet-LT).

- Comprehensive experiments across multiple benchmarks and distinct tasks, including standard image classification, long-tailed image recognition, and text classification, show that CaCT produces smaller and more informative prediction sets, while maintaining coverage, outperforming state-of-the-art conformal training methods, as shown in Fig. 1.

2 Related Work

Conformal prediction (CP) provides a principled framework to quantify uncertainty [2, 14]. Rather than returning a single label, CP outputs a prediction set that contains the true outcome with a user-specified probability. For instance, in medical diagnosis, a conformal predictor might return a set of possible diseases and guarantee that the correct one is included 90% of the time. This framework has recently attracted significant attention [14] and has been successfully applied across tasks including regression [15], classification [3, 4, 16, 17], image segmentation [18, 19], and language modeling [20]. Split CP [2, 21] is a widely used method for constructing prediction sets. Its core appeal lies in its simplicity and generality: given a pre-trained model and a calibration set, conformal prediction wraps around the model, outputting prediction sets with guaranteed coverage on a held-out sample under exchangeability, at a fraction of the computational cost, and without making any assumptions about the underlying data distribution. Despite its broad applicability, CP is a post-hoc and model-agnostic procedure, and its effectiveness depends strongly on the quality of the underlying predictive model. When the model is not well aligned with the goals of CP, the resulting sets may be excessively large [9]. Although such sets are technically valid, they are often uninformative [12], which raises the question of how models could be trained to produce more informative and efficient prediction sets.

Conformal training incorporates regularization terms into the training objective in order to enforce conformal properties [9–13]. For example, SCPO-IC [12] and ConfTr [9] simulate Split CP during training by introducing a regularization term that computes a differentiable approximation of the prediction set size, encouraging the model to construct smaller prediction sets. Direct Prediction Set Minimization (DPSM) [13] further improves ConfTr by reformulating conformal training as a bi-level optimization problem. Instead of using a stochastic approximation for the quantile, it directly learns the quantile of the conformity scores. Other recent works improve specific components of the ConfTr pipeline: Jia et al. [22] replace the sigmoid surrogate of the set-size indicator with a rank-weighted objective that provably upper-bounds the expected set size, while VR-ConfTr [23] reduces the variance of ConfTr's quantile-based gradient estimator. These contributions are orthogonal to ours and could be plugged into CaCT. Conformalized Uncertainty-aware Training (CUT) [11] penalizes deviations from the uniform distribution of non-conformity scores. A major limitation of all of the aforementioned methods, however, is that the regularization weight is applied uniformly across all samples, regardless of class or difficulty, limiting their ability to adapt to intrinsic differences or imbalances among classes. Though these methods manage to reduce inefficiency marginally, they fail to reduce it in under-represented classes while maintaining coverage, as shown in Fig. 2. Crucially, no existing conformal training method has explicitly formulated or solved the class-wise constrained optimization problem we address here.

Learning from long-tailed data is a central challenge in machine learning, as standard models are biased toward frequent classes, leading to poor generalization on under-represented categories. In the context of CP, marginal coverage guarantees ensure that the true label lies in the prediction set on average across all classes. However, these guarantees do not protect minority classes, which can lead to systematically over- or under-covered predictions. Label-conditional CP (Label CP) [1] mitigates this problem by constructing prediction sets separately for each class, effectively applying Split CP per-class. While this approach ensures the intended coverage, it often produces overly conservative and uninformative sets for rare classes, especially when the available data is limited [24]. Recent works, such as PAS [6] and sTACP [7] propose non-conformity score functions for long-tailed classification, which are complementary to conformal training methods like CaCT.

It is important to distinguish between these two frameworks: *post-hoc* methods (*e.g.*, score functions such as APS, RAPS, or PAS, and calibration strategies such as Label CP and Cluster CP [24]), operate on a fixed pre-trained model at calibration time, whereas *conformal training* (*e.g.*, ConfTr, CUT, DPSM, CaCT) modifies the model during training to produce outputs that are inherently better suited for conformal prediction. These two families are orthogonal and can be freely combined, as we show in our empirical validation. Furthermore, while the high-level goal of class-conditional efficiency has been explored at the post-hoc calibration level (*e.g.*, Label CP, Cluster CP, PAS), the conformal *training* literature has so far provided no principled mechanism to enforce class-conditional efficiency constraints during optimization, and CaCT directly addresses this gap.

3 Background and Motivation

Notations. Let us consider a standard supervised multi-class classification setting, where data is drawn from a distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$. Here, $\mathcal{X}$ denotes the input space, and $\mathcal{Y} = \{1, \dots, K\}$ is the set of class labels, with K the total number of classes. To keep the notation simple, we use $\mathcal{P}_X$ to denote the marginal distribution of X, and $\mathcal{P}_{X|Y=y}$ to denote the conditional distribution of X given a label $y \in \mathcal{Y}$. Note that we use uppercase X, Y to denote input and output random variables, including samples in the training, validation, calibration, and testing sets, and fixed values with lowercase letters. This convention is maintained throughout the paper for consistency. Let $\pi_\theta : \mathcal{X} \to \Delta_K$ be a soft classifier, *e.g.*, softmax scores, with parameters θ, where Δ_K denotes the $(K-1)$-dimensional probability simplex. The training dataset is given by $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^n$, while $\mathcal{D}_{\text{val}}$, $\mathcal{D}_{\text{cal}}$ and $\mathcal{D}_{\text{test}}$ represent the validation, calibration and test datasets, respectively. We denote the indices of samples corresponding to a class y as $\mathcal{I}_y = \{i \mid Y_i = y\}$, and $n_y = |\mathcal{I}_y|$.

Conformal prediction offers a principled framework for quantifying predictive uncertainty by constructing prediction sets that are guaranteed to contain the true label with high probability. Given a user-defined miscoverage level $\alpha \in [0, 1]$, a calibration dataset, a predictive model parameterized by θ, and a new test pair $(X_{\text{test}}, Y_{\text{test}}) \sim \mathcal{P}$, CP ensures that the prediction set $\mathcal{C} : \mathcal{X} \to 2^{|\mathcal{Y}|}$ guaranteed to contain the true label with high probability at least $1 - \alpha$, following Eq. (1). The construction of the prediction set relies on a non-conformity score function $S : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which measures how atypical a candidate label y is for a given input x relative to previously observed data [2]. For the i-th sample, we denote the corresponding score as $S_i = S_\theta(X_i, Y_i)$. The prediction set is then constructed by thresholding the non-conformity scores with the empirical quantile $\widehat{q}_\theta$ computed from $\mathcal{D}_{\text{cal}}$:

$$\mathcal{C}(X_{\text{test}}) = \{y \in \mathcal{Y} \mid S_\theta(X_{\text{test}}, y) \leq \widehat{q}_\theta\},$$

where $\widehat{q}_\theta$ is the $\lceil (m+1)(1-\alpha) \rceil / m$ empirical quantile (*i.e.*, the $\lceil (m+1)(1-\alpha) \rceil$-th order statistic) of $\{S_{\theta,i}\}_{i=1}^m$, and $m = |\mathcal{D}_{\text{cal}}|$. Under the standard exchangeability assumption [2], this construction ensures that the prediction set covers the true label with probability at least $1 - \alpha$.

The prediction set size of a given input X, also known as a set efficiency, is defined as:

$$|\mathcal{C}(X)| = \sum_{y \in \mathcal{Y}} \mathbb{1}[y \in \mathcal{C}(X)] = \sum_{y \in \mathcal{Y}} \mathbb{1}[S_\theta(X, y) \leq \widehat{q}_\theta].$$

Conformal Training. From an optimization standpoint, the goal is to learn a model that: (i) makes accurate predictions, and (ii) yields minimal prediction sets at test-time. This leads to the following constrained optimization problem:

$$\min_\theta \quad \mathcal{L}_{\text{cls}}(\theta) \quad \text{subject to} \quad |\mathcal{C}(X)| \leq \eta, \text{ a.s. } X \sim \mathcal{P}_X, \tag{2}$$

where $\mathcal{L}_{\text{cls}}(\theta)$ is a classification loss (*e.g.*, cross-entropy) and η is the target set size. We can solve this problem using a penalty-based method, as in ConfTr [9], which converts (2) into the following unconstrained optimization problem:

$$\min_\theta \quad \mathcal{L}_{\text{ConfTr}}(\theta) \coloneqq \mathcal{L}_{\text{cls}}(\theta) + \lambda \, \mathcal{L}_{\text{s}}(\theta), \tag{3}$$

in which $\lambda \in \mathbb{R}_+$ is a hyperparameter that controls the importance of the second term, *i.e.*, the size loss, defined as:

$$\mathcal{L}_{\text{s}}(\theta) = \mathbb{E}_X \left[\Omega \left(\mathcal{C}(X) \right) \right], \quad \text{with} \quad \Omega \left(\mathcal{C}(X) \right) = \max \left(0, |\mathcal{C}(X)| - \eta \right). \tag{4}$$

The penalty term in (4) approximates prediction set size at a specific miscoverage level α and penalizes sets larger than η. Note that the objective in (3) admits a straightforward unbiased estimator using a training set of n i.i.d. samples $\{(X_i, Y_i)\}_{i=1}^n$:

$$\widehat{\mathcal{L}}_{\text{ConfTr}}(\theta) = \widehat{\mathcal{L}}_{\text{cls}}(\theta) + \lambda \frac{1}{n} \sum_{i=1}^{n} \Omega \left(\mathcal{C}(X_i) \right) \tag{5}$$

4 Proposed Approach

4.1 Class-wise Constraints for Conformal Training

The constraint in (2) enforces an upper bound η on the prediction set size *marginally* across all data points. However, when the number of classes is large, or under class imbalance, under-represented classes receive overly conservative prediction sets [17] (Fig. 2). To address this issue, we first replace the marginal constraint with class-conditional constraints, resulting in the following:

$$\min_{\theta} \quad \mathcal{L}_{\text{cls}}(\theta) \quad \text{subject to} \quad \mathbb{E}_{X|Y=k}\,|\mathcal{C}(X)| \leq \eta, \;\; \text{for all } k \in \mathcal{Y}, \tag{6}$$

which introduces $K = |\mathcal{Y}|$ constraints, one for each class, encouraging better control over class-wise efficiency. This modifies the learning objective of ConfTr [9] with the proposed unconstrained problem defined as:

$$\min_{\theta} \quad \widehat{\mathcal{L}}_{\text{cls}}(\theta) + \lambda \frac{1}{n} \sum_{y \in \mathcal{Y}} \sum_{i \in \mathcal{I}_y} \Omega(\mathcal{C}(X_i)). \tag{7}$$

Note that because the index sets $\mathcal{I}_y$ partition the training data, the double sum above is algebraically equivalent to the single marginal sum in (5). Thus, (7) illustrates that a naive relaxation of the class-wise constraints using a single penalty weight λ collapses back to the marginal ConfTr objective. Nevertheless, even though the above constraints account for set sizes in a class-wise fashion, treating all constraint violations equally with a single uniform penalty weight λ fails to accurately solve the associated constrained problem in (6). Indeed, it is approximated by the unconstrained problem in (7), which uses a single uniform penalty weight, regardless of the category[2]. This implicitly assumes that all classes have similar characteristics. However, in practice, classes exhibit vastly different behaviors: some classes may naturally produce smaller prediction sets due to high model confidence, while others, particularly minority classes in long-tailed distributions, may consistently violate the size constraint. Therefore, an enhanced training strategy would involve considering distinct penalty weights, one per category. This requires integrating K penalty weights, resulting in the following learning objective:

$$\min_{\theta} \quad \widehat{\mathcal{L}}_{\text{cls}}(\theta) + \frac{1}{n} \sum_{y \in \mathcal{Y}} \lambda_y \sum_{i \in \mathcal{I}_y} \Omega(\mathcal{C}(X_i)), \tag{8}$$

where $(\lambda_y)_{1 \leq y \leq K} \in \mathbb{R}_+^K$, and each weight λ_y balances the contribution of its associated class-wise penalty for category y in the main learning objective. While this formulation allows controlling set sizes per-class, determining appropriate values for the K penalty weights is not straightforward. Traditional grid search over $\boldsymbol{\lambda}$ quickly becomes infeasible, *e.g.*, with $K = 1000$ classes (as in ImageNet [25]), even 5 values per dimension require evaluating 5^{1000} configurations. This is further exacerbated under long-tailed distributions, as different classes need penalty weights on different scales. In the following, we present an augmented Lagrangian strategy that automatically learns the class-wise penalty weights.

4.2 Class Adaptive Conformal Training

General Augmented Lagrangian Methods (ALM). To automatically learn penalty weights, ALM, which are widely used in the broader field of optimization [26–28], offer an appealing approach, integrating penalty functions and primal-dual updates to solve constrained optimization problems.

We begin by considering a generic constrained optimization problem ($\boldsymbol{x} \in \mathbb{R}^d$ denotes a generic optimization variable in the Augmented Lagrangian formulation):

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \quad \text{subject to} \quad h_i(\boldsymbol{x}) \leq 0, \quad i \in [\![1, m]\!], \tag{9}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is the *objective function*, and $h_i : \mathbb{R}^d \to \mathbb{R}$ for $i \in [m]$ represent the *set of constraint functions*. One classical strategy is to convert Problem (9) into a sequence of $j \in \mathbb{N}$ unconstrained problems, where each problem is solved approximately w.r.t. $\boldsymbol{x}$:

$$\min_{\boldsymbol{x}} \quad \mathcal{L}^{(j)}(\boldsymbol{x}) = f(x) + \sum_{i=1}^{m} P(h_i(\boldsymbol{x}), \lambda_i^{(j)}, \rho_i^{(j)}), \tag{10}$$

[2] An analogous argument holds for ConfTr [9]: its constrained formulation in (2) is only approximated by the unconstrained objective in (5).

where $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}$ is a *penalty-Lagrangian function* $P(z, \lambda, \rho)$[3], whose partial derivative with respect to its first argument exists, is positive, and continuous for all $z \in \mathbb{R}$ and $\rho, \lambda \in \mathbb{R}^2_{++}$. Throughout this paper, we instantiate P with the PHR (Powell–Hestenes–Rockafellar) penalty [30, 31], defined in (15); two alternative penalty functions (P_2 and P_3) are described and compared in Appx. C. The parameters $\boldsymbol{\lambda}^{(j)} = (\lambda_i^{(j)})_{1 \le i \le m} \in \mathbb{R}_{++}$ and $\boldsymbol{\rho}^{(j)} = (\rho_i^{(j)})_{1 \le i \le m} \in \mathbb{R}_{++}$ are the Lagrange multipliers and penalty coefficients associated to P at iteration j, respectively. Minimizing $\mathcal{L}^{(j)}$ is referred to as the *inner* iterations, whereas the *outer* iterations denote the sequence of unconstrained problems.

After solving Problem (10), the penalty multipliers $\boldsymbol{\lambda}^{(j)}$ are updated by differentiating the penalty function with respect to its first argument (*i.e.*, last inner minimization):

$$\lambda_i^{(j+1)} = P'(h_i(\boldsymbol{x}^{(j)}), \lambda_i^{(j)}, \rho_i^{(j)}). \tag{11}$$

This update rule corresponds to a first-order multiplier estimate for the constrained problem, where $\boldsymbol{x}^{(j)}$ is the solution obtained at iteration j. To adapt the penalty parameters $\boldsymbol{\rho}^{(i)}$ during training, the behavior of the constraint function is monitored. If the constraint does not show improvement between iterations, the corresponding penalty parameter is scaled up by a factor ($\beta > 1$). This dynamic adjustment encourages the model to better satisfy constraints over time.

In convex settings, alternating between approximate minimization of (10) and updates to the multipliers (11) leads to convergence toward a feasible solution. The inner loop targets the primal formulation, while the outer loop progressively refines the dual variables. Although classical convergence guarantees for augmented Lagrangian methods are derived in convex settings, these techniques have also demonstrated strong empirical performance in nonconvex scenarios [29, 32]. Prior work has shown that ALM variants remain stable and effective in non-convex deep models [33–35], which is further confirmed by our experiments (further details are in Appx. D.1). Despite their potential and strong empirical performance, ALM-based approaches remain relatively overlooked in deep learning contexts, and more so in the CP literature, making them a natural and principled fit for class-adaptive conformal training.

Integrating ALM Into Conformal Training. We propose in this work to leverage ALM to solve the unconstrained problem in (8). Specifically, we reformulate (8) by integrating a penalty function P parameterized by $\boldsymbol{\lambda}, \boldsymbol{\rho} \in \mathbb{R}^K_{++} \times \mathbb{R}^K_{++}$, which can be formally defined as:

$$\min_{\theta} \quad \mathcal{L}_{\text{cls}}(\theta) + \sum_{k=1}^{K} P(\widehat{d}_k - \eta, \lambda_k, \rho_k), \tag{12}$$

where $\widehat{d}_k = \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} |\mathcal{C}(X_i)|$ is the average class-wise prediction set size. Note that the multipliers $\boldsymbol{\lambda}$ are *not* minimized jointly with θ. Instead they are updated via dual ascent (Eq. (13)), which is the standard ALM procedure. Directly estimating Lagrange multipliers from the training set could easily lead to overfitting the training data. We propose to leverage a validation set $\mathcal{D}_{\text{val}}$ to reliably estimate the penalty multipliers at each epoch. In particular, each training epoch approximates the minimization of the primary loss function, while the average penalty multiplier value is subsequently estimated as an expectation over the validation set $\mathcal{D}_{\text{val}}$. We formally define this procedure, where after training epoch j the penalty multipliers for epoch $j + 1$ are:

$$\lambda_k^{(j+1)} = P'(\widehat{d}_k - \eta, \lambda_k^{(j)}, \rho_k^{(j)}) \tag{13}$$

In this training scheme, the penalty multiplier is adjusted dynamically, increasing when the constraint is violated and decreasing when the constraint is satisfied. This mechanism effectively provides an adaptive strategy for selecting the penalty weight in a penalty-based method. As noted above, this differs from a joint min-max optimization over $(\theta, \boldsymbol{\lambda})$: $\boldsymbol{\lambda}$ is never updated by minimizing or maximizing the objective, but is held fixed during the inner minimization over θ in (12) and only updated afterward through (13). Consequently, the derivative P' in (13), taken with respect to the constraint residual, is not a stationarity condition $\partial \mathcal{L} / \partial \boldsymbol{\lambda} = 0$, but the standard ALM update rule that determines how each multiplier should increase or decrease. Moreover, since this rule increases λ_k whenever the k-th constraint is violated, by construction of P (Appx. C), the degenerate solution

[3]We provide in Appx. C the axioms that must be satisfied to consider a penalty function [29].

Table 1: **Main results across image and text classification datasets.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result. See Appx. G for results with RAPS.

	Method	MNIST			CIFAR10			CIFAR100			ImageNet			20 Newsgroups		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG	S	C	CG
THR	CE	2.96	0.90	4.45	3.31	0.90	5.67	3.41	0.90	5.56	5.87	0.90	6.87	5.27	0.90	7.75
	FL [38]	3.89	0.90	3.33	3.15	0.90	6.78	4.05	0.90	4.98	5.75	0.90	**6.23**	4.91	0.90	7.35
	ConfTr [9] ICLR'22	3.45	0.90	4.99	2.89	0.90	5.08	4.76	0.90	5.65	5.45	0.90	6.41	4.34	0.90	7.27
	CUT [11] NeurIPS'22	4.15	0.90	4.00	2.65	0.90	4.98	3.00	0.90	4.29	5.41	0.90	6.53	4.29	0.90	6.52
	InfoCTr Fano [10] NeurIPS'24	2.93	0.90	3.32	**2.31**	0.90	**4.37**	4.12	0.90	4.24	5.01	0.90	6.50	4.36	0.90	6.19
	DPSM [13] ICML'25	3.50	0.90	**3.14**	2.69	0.90	5.06	2.81	0.90	**3.69**	**5.00**	0.90	6.36	**3.61**	0.90	5.92
	CaCT-H (Ours)	**2.72**	0.90	3.50	2.51	0.90	4.37	**2.69**	0.90	4.34	5.86	0.90	6.74	3.72	0.90	**5.87**
	CaCT (Ours)	**2.14**	0.90	**3.21**	**2.45**	0.90	**4.12**	**2.56**	0.90	**4.21**	**4.99**	0.90	**6.20**	**3.59**	0.90	**5.74**
APS	CE	3.46	0.90	3.86	3.41	0.90	3.56	5.50	0.90	4.09	7.21	0.90	6.54	5.35	0.90	6.90
	FL [38]	3.00	0.90	3.51	3.74	0.90	**3.05**	5.49	0.90	4.62	5.52	0.90	**6.15**	5.09	0.90	6.55
	ConfTr [9] ICLR'22	2.95	0.90	3.32	3.45	0.90	3.59	**4.57**	0.90	3.82	5.95	0.90	6.42	**4.39**	0.90	5.94
	CUT [11] NeurIPS'22	2.71	0.90	3.78	**2.76**	0.90	3.47	4.87	0.90	3.80	5.89	0.90	7.70	4.54	0.90	6.12
	InfoCTr Fano [10] NeurIPS'24	2.43	0.90	**3.09**	3.08	0.90	3.86	4.95	0.90	3.84	5.70	0.90	6.65	4.56	0.90	5.69
	DPSM [13] ICML'25	2.81	0.90	3.20	**2.84**	0.90	**3.42**	4.77	0.90	3.44	**4.56**	0.90	6.24	4.74	0.90	5.40
	CaCT-H (Ours)	**2.28**	0.90	3.35	3.09	0.90	3.87	4.76	0.90	**3.40**	5.09	0.90	6.95	4.87	0.90	**5.36**
	CaCT (Ours)	**2.32**	0.90	**3.15**	2.97	0.90	3.56	**3.99**	0.90	**3.22**	**5.01**	0.90	**5.80**	**4.00**	0.90	**4.67**

$\boldsymbol{\lambda} \to \mathbf{0}$, in which all constraints would be discarded, is not reached. Finally, we update penalty parameters $\boldsymbol{\rho}$ as:

$$\rho_k^{(j+1)} = \begin{cases} \beta \rho_k^{(j)} & \text{if} \quad \widehat{d}_k^{(j)} - \eta > \max\left(0, \widehat{d}_k^{(j-1)} - \eta\right) \\ \rho_k^{(j)} & \text{otherwise,} \end{cases} \tag{14}$$

where we compute the average constraint value per-class on the validation set $\mathcal{D}_{\text{val}}$. Thus, for each category, if the average constraint remains positive and has not decreased compared to the previous epoch, the corresponding penalty parameter is multiplied by a scaling factor β. This update strategy ensures that $\boldsymbol{\rho}$ grows when constraint violations persist and stabilizes otherwise, effectively implementing an adaptive penalty adjustment process.

Last, the choice of an appropriate penalty function is paramount to the performance of ALM methods, particularly in non-convex scenarios [29]. Common penalties include P_2 [36], P_3 [37] and PHR [30, 31], whose definitions are provided in Appx. C. Following prior literature on ALM in the context of CNNs [33, 34], we resort to the PHR function as the penalty, which is defined as:

$$\text{PHR}(z, \lambda, \rho) = \begin{cases} \lambda z + \frac{1}{2}\rho z^2 & \text{if} \quad \lambda + \rho z \geq 0; \\ -\frac{\lambda^2}{2\rho} & \text{otherwise.} \end{cases} \tag{15}$$

Our whole approach is detailed in Alg. 1 (Appx. D), as well as details on its differentiability.

5 Experiments and Results

5.1 Experimental Setup

Datasets. CaCT is empirically validated in several computer vision classification tasks, including MNIST [39], CIFAR-10 and CIFAR-100 [40], ImageNet [25], as well as their long-tailed counterparts [41]. For experiments under imbalance distributions, we follow [41] to collect the long-tailed versions of MNIST, CIFAR10 and CIFAR100 with different degrees of data imbalance ratio, γ, as well as ImageNet-LT [42]. To show the general applicability of CaCT, we further evaluate it on the 20 Newsgroups dataset [43], a popular NLP benchmark for text classification.

Conformity Scoring Functions. We consider five popular non-conformity scores: Least Ambiguous Classifier (THR) [5], Adaptive Prediction Sets (APS) [4], Regularized Adaptive Prediction Sets (RAPS) [3], Sorted Adaptive Prediction Sets (SAPS) [8], and Prevalence-Adjusted Softmax (PAS) [6]. Results for RAPS, SAPS, and PAS are in Appx. G.

Baselines. We focus on learning-based methods: (i) CE, where we train the model with the standard cross-entropy loss; (ii) Focal loss (FL) [38], which mitigates the overconfident predictions by down-weighting easy samples, *i.e.* with high probability; (iii) ConfTr [9] minimizing the average differentiable prediction set size; (iv) CUT [11] aiming at decreasing overconfidence by minimizing the gap between the uniform distribution and CDF of non-conformity scores; (v) InfoCTr [10], proposing three different upper bounds to the conditional entropy (Fano, Model-Based (MB) Fano,

Table 2: **Main results across long-tailed datasets.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$ and an imbalance factor $\gamma = 0.1$. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, and second best in **Blue**. Results with RAPS in Appx. G.

	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG
THR	CE	4.32	0.90	6.93	6.65	0.90	6.65	8.30	0.90	6.90	9.25	0.90	5.32
	FL [38]	4.90	0.90	5.95	6.30	0.90	6.30	9.23	0.90	6.50	8.85	0.90	5.70
	ConfTr [9] ICLR'22	4.35	0.90	4.92	5.05	0.90	6.29	7.84	0.90	6.31	6.60	0.90	5.64
	CUT [11] NeurIPS'22	3.71	0.90	4.90	4.33	0.90	6.17	4.84	0.90	5.07	**5.38**	0.90	5.42
	InfoCTr Fano [10] NeurIPS'24	**2.88**	0.90	4.47	4.51	0.90	5.69	4.78	0.90	4.96	7.14	0.90	5.08
	DPSM [13] ICML'25	3.06	0.90	**4.32**	**3.65**	0.90	**5.06**	**4.21**	0.90	4.91	5.78	0.90	4.71
	CaCT-H (Ours)	3.00	0.90	4.42	3.76	0.90	5.14	5.38	0.90	**4.58**	5.72	0.90	**4.67**
	CaCT (Ours)	**2.62**	0.90	**4.22**	**3.29**	0.90	**4.42**	**3.03**	0.90	**4.53**	**4.99**	0.90	**4.40**
APS	CE	5.25	0.90	5.18	6.80	0.90	5.80	9.20	0.90	6.80	9.40	0.90	5.15
	FL [38]	5.48	0.90	4.72	6.45	0.90	5.45	9.85	0.90	6.07	9.54	0.90	5.64
	ConfTr [9] ICLR'22	5.15	0.90	4.80	5.19	0.90	4.67	8.17	0.90	5.31	7.99	0.90	5.01
	CUT [11] NeurIPS'22	5.03	0.90	4.26	3.94	0.90	4.56	6.97	0.90	4.55	**6.08**	0.90	4.74
	InfoConfTr Fano [10] NeurIPS'24	**3.69**	0.90	**3.78**	5.13	0.90	**3.52**	7.18	0.90	5.65	7.27	0.90	**4.18**
	DPSM [13] ICML'25	3.83	0.90	4.33	**3.55**	0.90	**4.28**	6.60	0.90	4.76	7.02	0.90	4.51
	CaCT-H (Ours)	4.02	0.90	4.34	3.62	0.90	4.29	**5.65**	0.90	**4.12**	6.97	0.90	4.62
	CaCT (Ours)	**3.53**	0.90	**4.12**	**2.99**	0.90	4.52	**4.49**	0.90	**4.10**	**5.89**	0.90	**3.90**

and Data Processing Inequality (DPI)) to achieve smaller average set size[4]; and (vi) DPSM [13], which learns the quantile of non-conformity scores instead of using a stochastic approximation.

Implementation Details. We set $\gamma = 3$ in FL, and target size $\eta = 1$, as in ConfTr [9]. Unless otherwise stated, the miscoverage level $\alpha = 0.1$ for the non-conformity scores. For ALM hyper-parameters, we initialize $\boldsymbol{\lambda}^{(0)} = 10^{-6}$, $\boldsymbol{\rho}^{(0)} = 1$, $\beta = 1.2$, with $\boldsymbol{\rho}$ being updated every 10 epochs, as in [34], whereas the multipliers $\boldsymbol{\lambda}$ are updated after every epoch (Alg. 1). **Computational cost.** The per-epoch complexity of CaCT is $\mathcal{O}(nC + Km)$, compared to $\mathcal{O}(nC)$ for ConfTr, where n is the number of training samples, C the per-sample model cost, K the number of classes, and m the validation set size. The additional outer-loop term is small in practice, *e.g.*, CaCT increases the training time by only $\sim$ 4–5.5% on CIFAR100 and ImageNet (see Appendices G and H). We also introduce CaCT-H as a simpler and more computationally efficient baseline that captures the core intuition of class-adaptive penalization instead of the augmented Lagrangian updates (detailed in Appx. E). **Evaluation Metrics:** We report average set size (S), marginal coverage (C), and deviation of class-conditional coverage from the target coverage $1 - \alpha$ (CG) [24]. Additional details for implementation, datasets, baselines, and evaluation metrics are provided in Appx. F.

5.2 Main Results

Results on standard benchmarks (Tab. 1). We begin our experimental evaluation by assessing the performance of CaCT on established benchmarks spanning both image and text classification, using both THR and APS non-conformity scores. Across all settings, CaCT ranks first or second in 16 out of 20 metric-dataset combinations (80%), with an **average set efficiency improvement of nearly 9% and 14%** compared to the next two best methods, *i.e.*, DPSM and InfoCTr, respectively. From a coverage gap standpoint, our approach also leads to **average gains of 4% and 8%** compared to DPSM and InfoCTr. Thus, in standard benchmarks, CaCT reduces not only the conformal set size, but also the coverage gap compared to competing approaches.

Performance under long-tailed distributions (Tab. 2). After establishing the effectiveness of CaCT on standard benchmarks, we next examine its behavior under distributional imbalance. Long-tailed regimes provide a natural testbed for assessing whether class-wise penalties continue to offer benefits when the label distribution is highly skewed, with a few head classes containing many samples and numerous tail classes with limited observations. Long-tailed versions of each dataset are constructed using an imbalance factor $\gamma = 0.1$, while calibration and test sets remain unchanged to preserve exchangeability and maintain conformal coverage guarantees. *Set efficiency* (i.e., *set size):* CaCT yields the most compact prediction sets, outperforming the next-best method, DPSM, with an **average reduction of 18.2% in set size.** On CIFAR100-LT, the gains are even more pronounced, reaching 28% (THR) and 32% (APS) improvements over DPSM. *Coverage gap:* CaCT typically achieves the best coverage gap, ranking first in 6 out of the 8 settings, **outperforming DPSM by 7.3% as**

[4]We report results for the Fano bound and provide results for the MB Fano and DPI bounds in Appx. G.

Table 3: **Ablation studies on long-tailed distributions γ (a) and test-time mis-coverage level α (b) on CIFAR100-LT**. See Appx. G for results with APS and RAPS.

(a) Imbalance factor γ

	Method	$\gamma = 0.5$			$\gamma = 0.8$			$\gamma = 1.0$		
		S	C	CG	S	C	CG	S	C	CG
THR	CE	7.54	0.90	6.80	4.25	0.90	6.02	3.41	0.90	5.56
	FL	6.86	0.90	5.98	4.46	0.90	5.57	4.05	0.90	4.98
	ConfTr	6.43	0.90	6.01	4.94	0.90	5.69	4.76	0.90	5.65
	CUT	4.02	0.90	4.89	3.80	0.90	4.62	3.00	0.90	4.29
	InfoCTr Fano	4.56	0.90	4.90	4.22	0.90	**4.30**	4.12	0.90	4.24
	DPSM	**3.72**	0.90	4.77	**3.15**	0.90	4.35	2.81	0.90	**3.69**
	CaCT-H	4.30	0.90	**4.49**	3.61	0.90	4.40	**2.69**	0.90	4.34
	CaCT	**2.88**	0.90	**4.37**	**2.73**	0.90	**4.29**	**2.56**	0.90	**4.21**

(b) Test-time mis-coverage level α

	Method	$\alpha = 0.01$			$\alpha = 0.05$			$\alpha = 0.1$		
		S	C	CG	S	C	CG	S	C	CG
THR	CE	16.93	0.99	4.66	12.54	0.95	5.38	8.30	0.90	6.90
	FL	18.71	0.99	**3.59**	13.22	0.95	4.54	9.23	0.90	6.50
	ConfTr	20.34	0.99	4.06	11.78	0.95	**4.09**	7.84	0.90	6.31
	CUT	14.26	0.99	3.67	9.01	0.95	4.12	4.84	0.90	5.07
	InfoCTr Fano	**13.19**	0.99	4.44	9.55	0.95	4.83	4.78	0.90	4.96
	DPSM	14.87	0.99	4.78	8.58	0.95	4.84	**4.21**	0.90	4.91
	CaCT-H	14.69	0.99	4.32	**8.54**	0.95	4.57	5.38	0.90	**4.58**
	CaCT	**12.08**	0.99	**3.40**	**7.21**	0.95	**4.07**	**3.03**	0.90	**4.53**

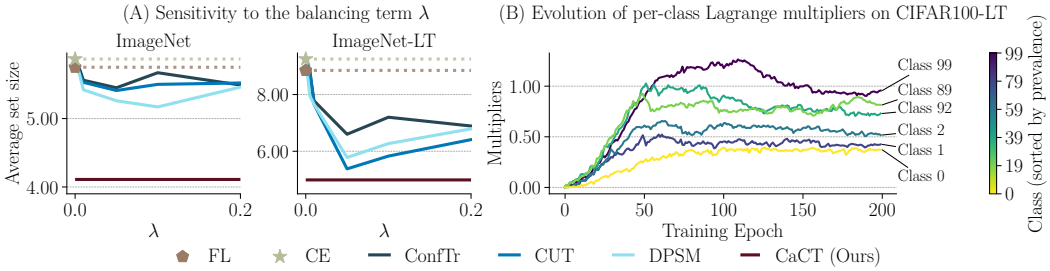

Figure 3: **Global penalty tuning is hard, CaCT learns class-aware multipliers. (A)** Set size of ConfTr, CUT, and DPSM as a function of global penalty λ on ImageNet and ImageNet-LT. Small changes in λ cause swings of up to 2 classes, and no single value matches CaCT across both settings. **(B)** Per-class evolution of learned Lagrange multipliers $\{\lambda_k\}$ on CIFAR100-LT.

average. Thus, results show that CaCT yields the best *efficiency* (*i.e.*, set size) across all datasets, while exhibiting a strong *adaptivity*. CaCT is therefore the most effective of the evaluated conformal training methods in this scenario: **it yields the best *efficiency* across all datasets, while exhibiting a strong *adaptivity***, as it achieves the lowest coverage gap in all but two cases.

Learned multipliers encode class difficulty and eliminate penalty tuning (Fig. 3). Fig. 3a illustrates two limitations of global-penalty methods. First, existing methods (ConfTr, CUT, DPSM) exhibit *high sensitivity* to λ: small changes lead to set-size swings of up to 2 classes on ImageNet-LT, making tuning impractical at scale. Second, even at their *optimal* λ, these methods are still outperformed by CaCT, demonstrating that a single global weight cannot correct the structural bias from class imbalance regardless of its value. Fig. 3b: Through ALM, CaCT eliminates this hyperparameter entirely, and per-class multipliers $\{\lambda_k\}$ are learned automatically during training. Beyond mere convenience, these multipliers carry meaningful structure: they naturally reflect per-class constraint pressure, with tail classes accumulating higher values than head classes, a pattern that emerges from constraint violations alone without any explicit use of class-frequency information.

Varying imbalance factor γ (Tab. 3a). To further investigate the robustness of CaCT, we conduct a controlled study across different values of the imbalance parameter $\gamma \in \{0.5, 0.8, 1.0\}$, which modulates the long-tailedness of the data distribution. Specifically, γ determines the relative frequency decay between head and tail classes, with lower values corresponding to more severe imbalance. Similarly to previous scenarios, CaCT yields the smallest prediction sets while also typically providing the best coverage gaps. In particular, for the most extreme case ($\gamma = 0.5$), **set size and coverage gap are reduced by 22.5% and 8.4%,** respectively, compared to the second best baseline, *i.e.*, DPSM. Furthermore, even as the distribution becomes more balanced ($\gamma \to 1.0$), CaCT consistently learns to optimally shape class-conditional set sizes, confirming that the benefits of class-adaptive penalties persist across several imbalance regimes, not only under extreme skew.

Varying miscoverage level α at test-time (Tab. 3b). In conformal training, we need to fix a target miscoverage level to compute batch-level quantiles. We now examine whether the performance of the resulting models deteriorates at different coverage rates, potentially "overfitting" to the α value used for training. In Tab. 3b, we report how average set size and coverage gap vary for different miscoverage levels α at test time for models trained with $\alpha = 0.01$ and models trained with CE and FL, which are agnostic to the coverage rate. CaCT achieves the **best performance across miscoverage levels,** with an **average improvement of 19.4%** (over DPSM) and **18.8%** (over InfoCTr) in terms of set size, and **17.4%** (DPSM) and **15.5%** (InfoCTr) from a coverage gap standpoint. Notably, at the strictest level ($\alpha = 0.01$), CaCT **reduces both set size and coverage gap by 17–20%** compared to CaCT-H, confirming that ALM's principled dual updates provide

measurable benefits at demanding operating points. These results show that CaCT does not overfit to the training α, maintaining robustness across the full range of coverage requirements.

Does conformal training provide meaningful gains over applying a long-tail-aware non-conformity score (*e.g.*, PAS [6], ICLR'26) to a fixed CE-trained model? PAS is a recent method designed to improve post-hoc CP on CE-trained models in long-tailed scenarios (detailed in Appx. B.1). We compare it against CaCT trained with a simple THR score (Tab. 4). On both long-tailed datasets, CaCT with THR score not only consistently produces substantially smaller prediction sets than PAS applied post-hoc on a fixed CE model, but also reduces significantly the coverage gap. In particular, CaCT brings **improvements of 48.93% and 67.24%** in terms of set efficiency, and of **12.87% and 27.75%** from a coverage gap standpoint. These results demonstrate that training-time optimization plays an important role, as it induces representation changes that post-hoc methods cannot replicate.

Table 4: **Conformal training (*Ours*) vs. post-hoc long-tailed scoring (PAS [6])** ($\alpha = 0.1$).

		PAS [6] + CE	CaCT with THR (*Ours*)
CIFAR100-LT	S	9.25	**3.03** (+67.24%)
	CG	6.27	**4.53** (+27.75%)
ImageNet-LT	S	9.77	**4.99** (+48.93%)
	CG	5.05	**4.40** (+12.87%)

6 Conclusion

We showed that global penalty weights impose a structural limitation in conformal training, systematically biasing optimization toward majority classes and inflating prediction sets for underrepresented ones. To overcome this, we introduced CaCT, a class-adaptive conformal training that formulates the problem as a class-wise constrained optimization, adaptively learning class-specific multipliers via a modified augmented-Lagrangian procedure. CaCT enables principled class-conditional shaping of prediction sets and naturally scales to many-class and imbalanced regimes, consistently reducing prediction-set size and coverage gap across multiple image and text classification benchmarks.

Acknowledgments

This research work was partially supported by France 2030 funding managed by the National Research Agency (ANR) as part of IA CLUSTER program, reference ANR-23-IACL-0003 – DATAIA CLUSTER. Experiments have been conducted using HPC resources from the Mésocentre computing center of CentraleSupélec and École Normale Supérieure Paris-Saclay, supported by CNRS and Région Île-de-France

References

[1] Vladimir Vovk, David Lindsay, Ilia Nouretdinov, and Alex Gammerman. Mondrian Confidence Machine, 2003. URL https://alrw.net/old/04.pdf. Pre-print.

[2] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer Cham, 2005. ISBN 978-3-031-06649-8.

[3] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty Sets for Image Classifiers Using Conformal Prediction. In *International Conference on Learning Representations (ICLR)*, 2022.

[4] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification With Valid and Adaptive Coverage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least Ambiguous Set-Valued Classifiers With Bounded Error Levels. In *J. Amer. Statist. Assoc.*, 2018.

[6] Tiffany Ding, Jean-Baptiste Fermanian, and Joseph Salmon. Conformal prediction for long-tailed classification. In *International Conference on Learning Representations (ICLR)*, 2026.

[7] Shuqi Liu, Jianguo Huang, and Luke Ong. Conformal Prediction Meets Long-tail Classification. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2026.

[8] Jianguo Huang, Huajun Xi, Linjun Zhang, Huaxiu Yao, Yue Qiu, and Hongxin Wei. Conformal prediction for deep classifier via label ranking. In *International Conference on Machine Learning (ICML)*, 2023.

[9] David Stutz, Krishnamurthy, Dvijotham, Ali Taylan Cemgil, and Arnaud Doucet. Learning Optimal Conformal Classifiers. In *International Conference on Learning Representations (ICLR)*, 2022.

[10] Alvaro H. C. Correia, Fabio Valerio Massoli, Christos Louizos, and Arash Behboodi. An Information Theoretic Perspective on Conformal Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[11] Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training Uncertainty-Aware Classifiers with Conformalized Deep Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[12] Anthony Bellotti. Optimized Conformal Classification Using Gradient Descent Approximation, 2021. URL https://arxiv.org/abs/2105.11255. Pre-print.

[13] Yuanjie Shi, Hooman Shahrokhi, Xuesong Jia, Xiongzhi Chen, Jana Doppa, and Yan Yan. Direct Prediction Set Minimization via Bilevel Conformal Classifier Training. In *International Conference on Machine Learning (ICML)*, 2025.

[14] Anastasios N. Angelopoulos and Stephen Bates. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification, 2022. URL https://arxiv.org/abs/2107.07511. Pre-print.

[15] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized Quantile Regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[16] Julio Silva-Rodríguez, Ismail Ben Ayed, and Jose Dolz. Conformal Prediction for Zero-Shot Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[17] Leo Fillioux, Julio Silva-Rodríguez, Ismail Ben Ayed, Paul-Henry Cournède, Maria Vakalopoulou, Stergios Christodoulidis, and Jose Dolz. Are Foundation Models for Computer Vision Good Conformal Predictors? In *Transactions on Machine Learning Research (TMLR)*, 2026.

[18] Joren Brunekreef, Eric Marcus, Ray Sheombarsing, Jan-Jakob Sonke, and Jonas Teuwen. Kandinsky Conformal Prediction: Efficient Calibration of Image Segmentation Algorithms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] Luca Mossina, Joseba Dalmau, and Léo Andéol. Conformal Semantic Image Segmentation: Post-Hoc Quantification of Predictive Uncertainty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

[20] Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F. Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking LLMs via Uncertainty Quantification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[21] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning (ECML)*, 2002.

[22] Xuesong Jia, Yuanjie Shi, Ziquan Liu, Yi Xu, and Yan Yan. Cost-Sensitive Conformal Training with Provably Controllable Learning Bounds. In *AAAI Conference on Artificial Intelligence*, 2026.

[23] Sima Noorani, Orlando Romero, Nicolo Dal Fabbro, Hamed Hassani, and George J. Pappas. Conformal Risk Minimization with Variance Reduction. *arXiv preprint arXiv:2411.01696*, 2024.

[24] Tiffany Ding, Anastasios N. Angelopoulos, Stephen Bates, Michael I. Jordan, and Ryan J. Tibshirani. Class-Conditional Conformal Prediction With Many Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In *International Journal of Computer Vision (IJCV)*, 2015.

[26] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Optimization and Neural Computation Series. Athena Scientific, 1st edition, 1996.

[27] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer New York, NY, 2006. ISBN 978-0-387-40065-5.

[28] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016. ISBN 978-1-886529-05-2.

[29] Ernesto G. Birgin, Rómulo A. Castillo, and José Mario Martínez. Numerical Comparison of Augmented Lagrangian Algorithms for Nonconvex Problems. *Computational Optimization and Applications*, 2005.

[30] Magnus R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 1969.

[31] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In *Optimization*. Academic Press, 1969.

[32] Boris Houska, Janick Frasch, and Moritz Diehl. *An augmented Lagrangian based algorithm for distributed nonconvex optimization*. SIAM, 2016.

[33] Jérôme Rony, Eric Granger, Marco Pedersoli, and Ismail Ben Ayed. Augmented lagrangian adversarial attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[34] Bingyuan Liu, Jérôme Rony, Adrian Galdran, Jose Dolz, and Ismail Ben Ayed. Class Adaptive Network Calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[35] Sara Sangalli, Ertunc Erdil, Andreas Hoetker, Olivio Donati, and Ender Konukoglu. Constrained Optimization to Train Neural Networks on Critical and Under-Represented Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[36] Barry W. Kort and Dimitri P. Bertsekas. Combined Primal–Dual and Penalty Methods for Convex Programming. *SIAM Journal on Control and Optimization*, 1976.

[37] H. Nakayama, H. Samaya, and H. Samaya. A Generalized Lagrangian Function and Multiplier Method. *Journal of Optimization Theory and Applications*, 1975.

[38] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision (ICCV)*, 2017.

[39] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proc. IEEE*, 1998.

[40] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Master's thesis, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[41] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[42] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-Scale Long-Tailed Recognition in an Open World. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Tom Mitchell. Twenty Newsgroups. UCI Machine Learning Repository, 1997. URL https://doi.org/10.24432/C5C323.

[44] R.M. Fano. *Transmission of Information: A Statistical Theory of Communications*. M.I.T. Press, 1961.

[45] Vladimir Vovk. Conditional Validity of Inductive Conformal Predictors. In *Asian Conference on Machine Learning (ACML)*, 2012.

[46] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[47] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 2011.

[48] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic Optimization of Sorting Networks via Continuous Relaxations. In *International Conference on Learning Representations (ICLR)*, 2019.

[49] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Monotonic Differentiable Sorting Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019.

[52] TorchVision Contributors. TorchVision: PyTorch's Computer Vision library. GitHub Repository, 2016. URL https://github.com/pytorch/vision.

A Broader Impact, Limitations, and Future Work

Broader Impact. This paper aims to advance the methodology of conformal prediction and reliable machine learning. By improving the efficiency of prediction sets while preserving formal coverage guarantees, our work contributes to the broader goal of producing models with more transparent and calibrated uncertainty estimates. Such improvements may be beneficial in high-stakes applications where understanding model uncertainty is important, including scientific analysis and decision-support systems. We do not anticipate negative societal impacts beyond those already associated with supervised learning and conformal prediction, and we identify no domain-specific risks that require special attention.

Limitations and Future Work. While CaCT provides a general framework for class-adaptive conformal training, it introduces class-wise constraints whose estimates depend on the amount of data available per class. In regimes with extremely scarce labels, such as long-tailed regimes, rare classes may yield noisy estimates and slower convergence of class-wise constraints. This limitation is shared by most conformal training methods, yet our experiments show that CaCT remains robust and effective under large number of classes and severe imbalance. Future work could mitigate this issue by sharing or smoothing constraints across classes, for example by learning penalties over clusters of semantically or statistically similar labels [24], which might further improve stability and scalability. Finally, the class-wise ALM framework of CaCT is not tied to a specific set-size surrogate or gradient estimator: its outer loop accepts any differentiable per-class proxy of the set size. It is therefore orthogonal to recent efforts that (i) design tighter, provably-bounded surrogates of the set-size indicator (*e.g.*, rank-weighted objectives [22]), or (ii) reduce the variance of the quantile and gradient estimators used in conformal training [23]. Integrating such surrogates or variance-reduced estimators within the inner loop of CaCT is a promising avenue to combine their benefits with the class-conditional adaptivity of our approach.

B Additional Background Details

B.1 Non-conformity Scoring Functions

Non-conformity scores are measures of the predictive model's uncertainty. They quantify the level of agreement between the input and its corresponding output, with higher scores indicating greater uncertainty. These scores are a fundamental component of conformal prediction, and many advancements in the field stem from the development of new scoring functions. In the following, we present three widely used scoring functions in the literature.

Least Ambiguous Classifier. The least ambiguous classifier (THR) [5] is the simplest and most commonly used non-conformity function. It yields the smallest possible prediction sets by only considering the confidence assigned to the true label. The score is defined as

$$S_\theta^{\mathrm{THR}}(X, Y) = 1 - \pi_\theta(X)_Y, \tag{16}$$

where $\pi_\theta(x)_y$ is the predicted probability for the true label y. Despite its efficiency, it tends to produce empty prediction sets when the model is highly uncertain, and undercover hard subgroups and overcover easy ones [14].

Adaptive Prediction Sets. To address this issue, adaptive prediction sets (APS) [4] were proposed. APS aims to handle ambiguity by considering the entire distribution over classes rather than focusing solely on the true label. The intuition behind APS is to build prediction sets by sequentially including labels in descending order of their predictive probabilities until the true label is covered. This approach ensures that the prediction set adapts to the model's confidence level and avoids having empty sets. If the model assigns high probability to the true label, the resulting set is small; otherwise, it grows accordingly. Formally, the APS score is defined as

$$S_\theta^{\mathrm{APS}}(X, Y) = \sum_{i=1}^{r_\theta(X,Y)-1} \pi_\theta(X)_{(i)} + U \cdot \pi_\theta(X)_Y, \tag{17}$$

where r_θ gives the rank of label y among the predictive probabilities $\pi_\theta(x)$ sorted in descending order, $\pi_\theta(X)_{(i)}$ denotes the i-th largest predictive probability (*i.e.*, the i-th order statistic in descending order), $\pi_\theta(X)_Y$ is the probability assigned to the true label Y, and $U \in [0, 1]$ is a uniform random variable added to achieve exact marginal coverage.

Regularized Adaptive Prediction Sets. Building on APS, the regularized adaptive prediction sets (RAPS) [3] introduce a regularization term that discourages the inclusion of too many low-confidence labels. This function encourages smaller prediction sets by penalizing cases where the true label is not among the top-k candidates. The RAPS score is given by

$$S_\theta^{\mathrm{RAPS}}(X,Y) = S_\theta^{\mathrm{APS}}(X,Y) + \lambda_{\mathrm{reg}} \cdot \max\left(0, r_\theta(X,Y) - k\right), \tag{18}$$

where λ_{reg} is a regularization coefficient that controls the strength of the penalty, and k is the desired target rank.

Sorted Adaptive Prediction Sets (SAPS). Huang et al. [8] modify the APS score and introduce Sorted Adaptive Prediction Sets (SAPS). They argue that relying on all softmax probabilities can lead to large prediction sets. To address this issue, SAPS discards all probability values except the maximum softmax probability when defining the non conformity measure. Formally, it is defined as

$$S_\theta^{\mathrm{SAPS}}(X,Y) = \begin{cases} U \cdot \pi_\theta^\star(X), & \text{if} \quad r_\theta(X,Y) = 1, \\ \pi_\theta^\star(X) + (r_\theta(X,Y) - 2 + U) \cdot \lambda & \text{otherwise.} \end{cases} \tag{19}$$

where $\pi_\theta^\star(X) = \max_{y \in \mathcal{Y}} \pi_\theta(X)_y$ is the maximum softmax probability, and λ is a hyperparameter.

Prevalence-Adjusted Softmax (PAS). In contrast to previous scores, Ding et al. [6] address conformal prediction in long tail classification settings. Their idea is to recalibrate the model label score $\pi_\theta(X)_Y$ to account for class rarity. Formally, the Prevalence-Adjusted Softmax score is defined as

$$S_\theta^{\mathrm{PAS}}(X,Y) = 1 - \frac{\pi_\theta(X)_Y}{n_Y / |\mathcal{D}_{\mathrm{cal}}|}, \tag{20}$$

where n_Y denotes the number of calibration samples belonging to class Y, and $|\mathcal{D}_{\mathrm{cal}}|$ is the size of the calibration set. They further propose a weighted variant, referred to as Weighted Prevalence-Adjusted Softmax (WPAS), which provides additional control through user-defined class weights $w(y)$ for $y \in \mathcal{Y}$ satisfying $\sum_{y \in \mathcal{Y}} w(y) = 1$. The corresponding score is given by

$$S_\theta^{\mathrm{WPAS}}(X,Y) = 1 - w(y)\, \frac{\pi_\theta(X)_Y}{n_Y / |\mathcal{D}_{\mathrm{cal}}|}. \tag{21}$$

B.2 Conformal Training Objectives

ConfTr [9] estimates a soft measure of the average prediction set size, defined as:

$$\widehat{\mathcal{L}}_{s,\mathrm{ConfTr}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} \widehat{\mathbb{1}}\left[S_\theta(X_i, y) \leq \widehat{q}_\theta\right] \tag{22}$$

CUT [11] measures the maximum deviation from uniformity of the non-conformity scores, defined as,

$$\widehat{\mathcal{L}}_{s,\mathrm{CUT}}(\theta) = \sup_{w \in [0,1]} \left| \widehat{F}_S(w) - w \right|, \tag{23}$$

where $\widehat{F}_S(w)$ is the empirical cumulative distribution function (CDF) of $S_i = S_\theta(X_i, Y_i)$ for $i \in [n]$. CUT could also be written with respect to the quantiles as follows,

$$\sup_{\alpha \in [0,1]} \left| (1-\alpha) - \widehat{q}_\theta \right| \tag{24}$$

InfoCTr [10] proposes three different ways to upper bound the conditional entropy $H[Y|X]$, which can serve as better alternatives to push the model closer to the true distribution and achieve a smaller average set size. Correia et al. [10] propose: (i) a bound derived from the data processing inequality (DPI bound), and (ii) two bounds based on a variation of Fano's inequality [44]: a model-agnostic version (simple Fano bound) and a version informed by the predictive model itself (model-based Fano bound).

ConfTr, CUT, and InfoCTr use a stochastic approximation to estimate quantiles. DPSM [13] proposes a bi-level optimization approach for conformal training, where the quantile value is learned by minimizing the pinball loss on non-conformity scores. This serves as a constraint to achieve accurate quantiles while simultaneously optimizing for efficiency, similar to the objective used in ConfTr.

B.3 Conditional Conformal Prediction

Standard Conformal Prediction (CP), *i.e.*, Split CP, provides marginal coverage guarantees, meaning that the prediction set contains the true label with probability at least $1-\alpha$ on average over the data distribution. However, this guarantee may fail to protect minority groups and rare but important classes.

Label-Conditional CP guarantees that each ground-truth class y has at least $1-\alpha$ probability of being included in the prediction set [1, 45]. It computes a separate conformal threshold for each class in the calibration set, and the prediction set is defined as

$$\mathcal{C}(X) = \{y \in \mathcal{Y} \mid S_\theta(X, y) \leq \widehat{q}_\theta^{\,y}\}\,, \tag{25}$$

where the threshold $\widehat{q}_\theta^{\,y}$ for label y is given by

$$\widehat{q}_\theta^{\,y} = \mathcal{Q}\left(\{S_\theta(X_i, Y_i)\}_{i\in\mathcal{I}_y}; \frac{\lceil (n_y+1)(1-\alpha)\rceil}{n_y}\right), \tag{26}$$

and n_y is the number of calibration samples belonging to class y, with $\mathcal{I}_y$ denoting their indices. If a class contains too few samples to compute a threshold (*i.e.*, $n_y < (1/\alpha) - 1$, we set $\widehat{q}_\theta^{\,y} = \infty$, which would include the entire label space $\mathcal{Y}$ in the prediction set. Label CP may still suffer from uninformative and overly conservative prediction sets (*i.e.*, large sets) for classes with limited calibration samples [24].

Cluster-Conditional CP outperforms Split CP and Label CP, in terms of class-conditional coverage, when there is limited access to calibration data per-class [24]. Cluster CP groups the classes into M clusters based on their non-conformity scores and then computes a shared conformal threshold for each cluster:

$$\mathcal{C}(X) = \left\{y \in \mathcal{Y} \mid S_\theta(X, y) \leq \widehat{q}_\theta^{\,h(y)}\right\}, \tag{27}$$

where $h : \mathcal{Y} \to \{1, \ldots, M\} \cup \{\text{null}\}$ is a clustering function (*e.g.*, K-Means). The threshold for each cluster is given by

$$\widehat{q}_\theta^{\,k} = \mathcal{Q}\left(\{S_\theta(X_i, Y_i)\}_{i\in\mathcal{I}_k}; \frac{\lceil (n_k+1)(1-\alpha)\rceil}{n_k}\right), \tag{28}$$

$$\widehat{q}_\theta^{\,\text{null}} = \mathcal{Q}\left(\{S_\theta(X_i, Y_i)\}_{i\in\mathcal{I}}; \frac{\lceil (n+1)(1-\alpha)\rceil}{n}\right), \tag{29}$$

where $\mathcal{I}_k = \{i \mid h(Y_i) = k\}$ and $n_k = |\mathcal{I}_k|$ denote the indices and the number of calibration samples in cluster k, respectively. If a class has too few samples to be assigned to any cluster, it is placed in the null group, for which the conformal threshold is computed using all calibration samples.

In addition to the experiments we have carried out in the main manuscript, we assess the performance of conformal training objectives with Label CP and Cluster CP in Tabs. 12 and 13.

C Penalty Functions for ALM

We outline here the requirements for a penalty function P in the Augmented Lagrangian Method (ALM), detailed in Sec. 4.2.

A function $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}$ is considered a valid Penalty-Lagrangian function if it satisfies the following properties: (i) *Boundedness below:* $P(z, \lambda, \rho)$ is bounded below (not necessarily non-negative) for all $z \in \mathbb{R}, \lambda, \rho \in \mathbb{R}_{++}$, and (ii) *Regularity:* P is continuous in all arguments, and its partial derivative with respect to z, denoted $P'(z, \lambda, \rho) \equiv \partial P(z, \lambda, \rho)/\partial z$, exists and is continuous over the same domain. In addition, a penalty function P should satisfy the following four axioms [29]:

Axiom 1. $P'(z, \lambda, \rho) \geq 0 \quad \forall\, z \in \mathbb{R}, \lambda, \rho \in \mathbb{R}^2_{++}$.

Axiom 2. $P'(0, \lambda, \rho) = \lambda \quad \forall\, \lambda, \rho \in \mathbb{R}^2_{++}$.

Axiom 3. If, for all $j \in \mathbb{N}$, $\lambda^{(j)} \in [\lambda_{\min}, \lambda_{\max}]$, where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$, then: $\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} z^{(j)} > 0$ imply that $\lim_{j\to\infty} P'(z^{(j)}, \lambda^{(j)}, \rho^{(j)}) = \infty$.

Axiom 4. If, for all $j \in \mathbb{N}$, $\lambda^{(j)} \in [\lambda_{\min}, \lambda_{\max}]$, where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$, then: $\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} z^{(j)} < 0$ imply that $\lim_{j\to\infty} P'(z^{(j)}, \lambda^{(j)}, \rho^{(j)}) = 0$.

The first two axioms guarantee that the derivative of the Penalty-Lagrangian function P w.r.t. z is positive and equals to λ when $z = 0$. The last two axioms guarantee that the derivative tends to infinity when the constraint is not satisfied, and zero otherwise. There exist many penalty functions in the literature. In this paper, we use the PHR function (Eq. (15)) and empirically compare it with two other penalty functions, P_2 and P_3 (Fig. 5, right),

$$\mathrm{P}_2(z, \lambda, \rho) = \begin{cases} \lambda z + \lambda\rho z^2 + \frac{1}{6}\rho^2 z^3 & \text{if } z \geq 0, \\ \frac{\lambda z}{1-\rho z} & \text{otherwise,} \end{cases} \tag{30}$$

$$\mathrm{P}_3(z, \lambda, \rho) = \begin{cases} \lambda z + \lambda\rho z^2 & \text{if } z \geq 0, \\ \frac{\lambda z}{1-\rho z} & \text{otherwise.} \end{cases} \tag{31}$$

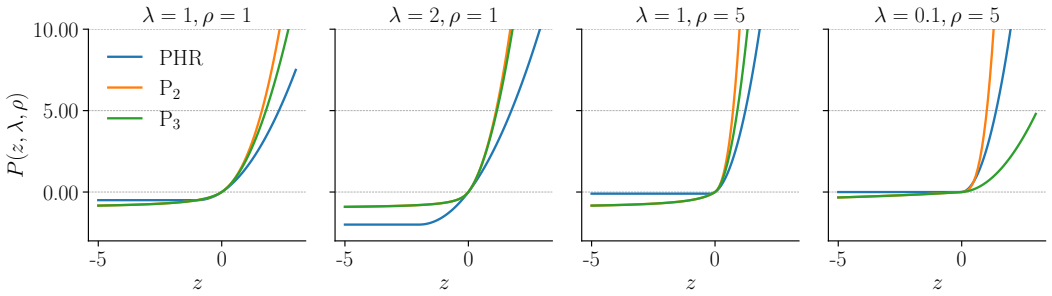

Figure 4: **Examples of penalty functions, P_2, P_3, and PHR for different values of λ and ρ.** The validity of such functions is discusses in Appx. C.

D CaCT Algorithm

D.1 Optimality Gap and Non-Convex Regime

In the classical *convex* setting, strong duality under Slater's condition yields an explicit bound on the duality gap [46]. The conformal training objective paired with a deep network is inherently non-convex, so neither strong duality nor a gap bound can be established in general; we do not claim one. Instead, ALM serves as a *principled and widely adopted mechanism* for enforcing inequality constraints in non-convex training [33–35]: the quadratic penalty stabilizes the landscape while the adaptive multiplier updates drive each class toward feasibility without requiring $\rho_k \to \infty$. The empirical behavior of CaCT (*i.e.*, stable multiplier trajectories, consistent constraint satisfaction, and significant efficiency gains) confirms that this mechanism is effective in practice. We elaborate on each point below.

Classical (convex) setting. In the convex case, when both the objective and constraints are convex and Slater's qualification holds, strong duality holds: primal and dual optima coincide, and the duality gap is zero. Moreover, for the augmented Lagrangian, an explicit upper bound on residual primal infeasibility as a function of ρ can be derived [46]. In this regime, convergence to the global optimum can be guaranteed.

Non-convex setting (our case). The conformal training objective in CaCT involves a deep neural network, making it inherently non-convex. In this regime: (i) strong duality does not hold in general; (ii) KKT conditions characterize only stationary points, not global optima; and (iii) feasibility is not guaranteed. Consequently, a meaningful optimality-gap bound (in the sense of convex theory) cannot be established for CaCT, and we do not claim one.

Role of ALM in non-convex settings. In non-convex optimization, ALM is used as a *principled and practical mechanism for constraint enforcement during training* [33–35] — analogous to how SGD is used for stochastic optimization without a convergence certificate for non-convex objectives. Its key advantages are: (i) the quadratic penalty stabilizes the landscape by penalizing large violations more severely as ρ_k grows; (ii) the adaptive multiplier update $\lambda_k^{(j+1)} = P'(\widehat{d}_k - \eta, \lambda_k^{(j)}, \rho_k^{(j)})$ increases

λ_k when class k violates its constraint and decreases it when satisfied, achieving empirically tight enforcement without requiring $\rho_k \to \infty$ (which would ill-condition the Hessian).

Why is $\boldsymbol{\lambda}$ alone not sufficient? Using the multipliers alone, *i.e.*, a standard Lagrangian dual-ascent scheme without the quadratic term controlled by $\boldsymbol{\rho}$, is insufficient in our setting. Classical dual ascent is guaranteed to converge only under fairly restrictive conditions, such as strict convexity of the objective [47, §2.3]. Without these assumptions, the primal minimizer at each step need not be unique, and the alternating primal-dual iterations may behave poorly or oscillate. The augmented Lagrangian addresses this by adding a quadratic term controlled by ρ, which improves the conditioning of the penalized objective and stabilizes the primal updates, even when the original objective is non-convex, and it remains a principled and widely used mechanism for constrained optimization of deep networks [33–35]. Since θ parameterizes a deep, non-convex model, strict convexity cannot be assumed in our setting. This is precisely the role of $\boldsymbol{\rho}$ in CaCT: *it locally stabilizes the primal updates for classes whose constraint is difficult to satisfy, while $\boldsymbol{\lambda}$ retains its role of driving each constraint towards satisfaction*. Concretely, the per-class ρ-update in (14) increases ρ_k only for classes whose constraint violation persists across updates, adaptively stabilizing exactly those classes that need it.

Empirical evidence. The effectiveness of this mechanism is confirmed throughout the paper: (i) Figs. 3a and 5 shows that multipliers and set sizes converge and stabilize during training; (ii) Tab. 5 shows ALM achieves consistently lower set sizes *and* coverage gaps compared to fixed-weight or heuristic alternatives; (iii) per-class multiplier trajectories (Fig. 3a) show that constraint violations are reliably corrected across classes with different difficulty levels. This empirical behavior is consistent with how ALM has been successfully applied in other constrained deep learning settings [33–35], where the lack of a convex-theory gap guarantee does not preclude practical effectiveness.

D.2 CaCT Differentiability

The original objective in CaCT is non-smooth as it involves indicator functions to compute the size of the prediction set and requires computing the empirical batch-level quantile in (2). To make it differentiable, we will need to smooth the objective, following ConfTr [9]. In particular, we use the sigmoid function to smooth the indicator $\mathbb{1}[a \leq b]$, defined as: $\widehat{\mathbb{1}}[a \leq b] = 1/(1+\exp(-(b-a)/T))$ where $T > 0$ is a tunable hyperparameter to control its smoothness. Moreover, we compute the conformal threshold using a differentiable quantile operator, derived using smooth sorting techniques [48, 49]. Smooth approximations relax the calibration step in Split CP, resulting in the loss of the marginal coverage guarantee. However, as the temperature hyperparameter $T \to 0$, the smooth formulation converges to the original non-smooth computations, and the corresponding coverage guarantee. This is sufficient because the smoothing operations are replaced *only* during training to enable gradient-based optimization. At test time, we use the original non-smooth implementation, and the formal coverage guarantee follows directly [3, 4, 14].

D.3 CaCT Training Procedure

Model Optimization: In the *inner* step, the model parameters θ are updated via stochastic gradient descent (SGD). For each mini-batch drawn from the training set $\mathcal{D}_{\text{train}}$, the data is randomly partitioned into a calibration subset $\mathcal{B}_{\text{cal}}$ and a prediction subset $\mathcal{B}_{\text{pred}}$. The main learning objective, which incorporates both the conformal objective and penalty terms, is evaluated on these subsets. Then, the gradient of this loss with respect to θ is used to perform a parameter update. Note that this step corresponds to solving the primal problem under the current penalty configuration. *Multiplier and Penalty Updates:* Then, during the *outer* iteration, the algorithm adjusts the penalty multipliers $\boldsymbol{\lambda}$ and penalty parameters $\boldsymbol{\rho}$ for each class $k \in \mathcal{Y}$. This reflects the dual ascent mechanism commonly used in ALM frameworks.

E Heuristic Version of CaCT

As stated in the main paper, we consider a simpler and more computationally efficient baseline that captures the core intuition of class-adaptive penalization instead of the augmented Lagrangian updates, which is referred to as CaCT-H. Formally, in CaCT-H, the multiplier $\lambda_k^{(j+1)}$ is scaled based

[5] $\mathcal{I}_k = \{i \in \mathcal{D}_{\text{val}} \mid Y_i = y\}$ is the set of indices of validation samples that belong to class $k \in \mathcal{Y}$, and $n_k = |\mathcal{I}_k|$.

Algorithm 1 Class Adaptive Conformal Training with ALM

Require: Training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$, learning rate $\nu > 0$, miscoverage level $\alpha \in [0, 1]$, training epochs T, target size $\eta \geq 0$, penalty parameter increase rate $\beta > 1$, penalty function P, initial multiplier $\boldsymbol{\lambda}^{(0)}$, initial penalty parameter $\boldsymbol{\rho}^{(0)}$

for $j = 1, \ldots, T$ **do**
  ▷ *Inner Step (Model optimization)*
  for each mini-batch $\mathcal{B}$ from $\mathcal{D}_{\text{train}}$ **do**
    Randomly split $\mathcal{B}$ into $\mathcal{B}_{\text{cal}}$ and $\mathcal{B}_{\text{pred}}$
    Compute $\widehat{q}_\theta$ as the $(1-\alpha)$-quantile of non-conformity scores on $\mathcal{B}_{\text{cal}}$
    $\widehat{\nabla}_\theta^{(j)} \leftarrow \nabla_\theta \widehat{\mathcal{L}}_{\text{CaCT}}(\theta^{(j)}, \boldsymbol{\lambda}^{(j)}, \boldsymbol{\rho}^{(j)})$ on $\mathcal{B}_{\text{pred}}$
    $\theta^{(j+1)} \leftarrow \theta^{(j)} - \nu \cdot \widehat{\nabla}_\theta^{(j)}$
  end for
  ▷ *Outer Step (Multiplier and Penalty Updates)*
  for $k = 1, \ldots, |\mathcal{Y}|$ **do**
    $\widehat{d}_k \leftarrow \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} |\mathcal{C}(X_i)|$ [5]
    $\lambda_k^{(j+1)} \leftarrow P'\left(\frac{\widehat{d}_k}{\eta} - 1, \lambda_k^{(j)}, \rho_k^{(j)}\right)$
    if the k-th constraint does not improve (Eq. (14)) **then**
      $\rho_k^{(j+1)} \leftarrow \beta \cdot \rho_k^{(j)}$
    end if
  end for
end for

Ensure: Model parameters $\theta^{(T)}$

on the value of the k-th penalty $P_k^{(j)}$ at iteration j:

$$\lambda_k^{(j+1)} = \begin{cases} \mu \cdot \lambda_k^{(j)} & \text{if } P_k^{(j)} > \tau P_k^{(j-1)}, \\ \lambda_k^{(j)} / \mu & \text{if } P_k^{(j-1)} > \tau P_k^{(j)}, \\ \lambda_k^{(j)} & \text{otherwise.} \end{cases} \tag{32}$$

where $\mu, \tau > 1$ are both fixed to 1.1. Here $P_k^{(j)}$ denotes the constraint violation $\max(0, \widehat{d}_k^{(j)} - \eta)$ evaluated at the end of epoch j.

F Additional Experimental Setup Details

Dataset and Split. All datasets come with pre-defined training/test splits. We also experiment with long-tailed datasets: manually resampled long-tailed versions of MNIST, CIFAR10 and CIFAR100 with an exponential decay of γ (*i.e.*, $\gamma = \min_i n_i / \max_i n_i$, where n_i is the number of samples in each class), an imbalance factor following [41], and ImageNet-LT [42] which is truncated from ImageNet [25] by sampling a subset so that the labels of the training set follow a long-tailed distribution. In our experiments, we further split the test set 50:50 into a validation subset and a final test set $\mathcal{D}_{\text{test}}$. The validation subset is then further divided 20:80 into a validation set $\mathcal{D}_{\text{val}}$ (used for ALM optimization) and a calibration set $\mathcal{D}_{\text{cal}}$. This final partition ensures that $\mathcal{D}_{\text{cal}}$ remains exchangeable with the test samples; reusing the same data for both ALM updates and calibration would violate exchangeability. This mimics the proceedure followed in other conformal training approaches, such as [9].

Models and Training. We use a linear model on MNIST, ResNet-34/50 [50] on CIFAR10 and CIFAR100, ResNet-50 for ImageNet, and a pre-trained BERT [51] model with linear layer on top of it for 20 Newsgroups. For CIFAR datasets, since the default ResNet implementation produces a 1×1 feature map before global pooling, we modify the first convolution to use a kernel size of 3, stride 1, and padding 1. We train using stochastic gradient descent (SGD) with momentum set to 0.9 and Nesterov gradients. The learning rate followed a step-wise schedule, where the initial learning rate was multiplied by 0.1 at $2/5$, $3/5$, and $4/5$ of the total number of training epochs. all models are trained for 200 epochs, with the exception of MNIST and 20 Newsgroups dataset which were trained only for 50 epochs. The learning rate and batch size are optimized alongside the hyperparameters using grid search, see next paragraph. We only use data augmentations on the CIFAR and ImageNet datasets, and following [10], we only apply random flipping and cropping. Note that the same thing applies to the corresponding long-tailed version of each dataset.

Numerical Stability. To mitigate numerical instabilities in ALM, often encountered with non-linear penalties, in practice we normalize the constraints using the target size parameter $\eta > 0$. Specifically, the original constraint $\widehat{d}_k - \eta \leq 0$ is equivalently rewritten as $\widehat{d}_k/\eta - 1 \leq 0$. This normalization improves the numerical behavior of the ALM updates and ensures more stable convergence during training [34].

Choice of Non-Conformity Score During Training. We perform conformal training by using the log probabilities as model outputs rather than the THR or APS scores. This choice follows the findings of Stutz et al. [9] and is consistent with the observations of Correia et al. [10] and Shi et al. [13]. We found this approach to be efficient and to improve stability during training. The models generalize well across all non-conformity score functions evaluated at test time, which confirms the robustness of this design choice.

Fine-tuning CIFAR and ImageNet. Base models are first trained on the full training set using cross-entropy loss, and then fine-tuned using the different conformal training approaches: ConfTr [9], CUT [11], InfoCTr [10], DPSM [13], and our approach CaCT. During fine-tuning, the final (logit) layer is reinitialized and trained using the same data augmentation as in the base training phase. For ImageNet, the base model is a pre-trained ResNet-50 available in Torchvision [52].

Hyperparameters. The final hyperparameters were obtained using grid search over the following hyperparameters: batch size in $\{100, 500, 1000\}$; learning rate in $\{0.05, 0.01, 0.005\}$; temperature parameter $T \in \{0.01, 0.1, 0.5, 1\}$; regularization weight $\lambda \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$, used in ConfTr [9], CUT [11], InfoCTr [10], and DPSM [13] to control the trade-off between the classification loss and regularization term; steepness, which controls the smoothness of the differentiable sorting algorithm [49] in $\{1, 10, 100\}$. Higher values make the output closer to standard (hard) sorting. Following [9], hyperparameter optimization is done using the training dataset. For each hyperparameter configuration, we evaluate the average prediction set size across 10 random calibration/test splits, using THR at test time. We select the hyperparameters that yield the smallest prediction set size.

Evaluation Metrics. We formally define the metrics introduced in the empirical validation. Using the test set $\mathcal{D}_{\text{test}}$ across all datasets, we report:

- The empirical coverage to measure the degree of satisfaction of the marginal coverage guarantees. It is defined as:

$$\text{Cov} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} \mathbb{1}[Y_i \in \mathcal{C}(X_i)] \,.$$

- The average set size, also known as inefficiency [9], is a widely employed metric for assessing the utility of CP methods for multi-class classification problems. An optimum conformal method in terms of efficiency should provide a lower set size. It is formally defined as:

$$\text{Size} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} |\mathcal{C}(X_i)| \,. \tag{33}$$

- The deviation of class-conditional coverage from the desired coverage level $1 - \alpha$, also known as coverage gap [24]. It is defined as

$$\text{CovGap} = 100 \times \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} |c_y - (1 - \alpha)|, \tag{34}$$

  where $c_y = \frac{1}{n_y} \sum_{i \in \mathcal{I}_y} \mathbb{1}[Y_i \in \mathcal{C}(X_i)]$, and n_y is the number of samples in the dataset for class y.

- Top-k accuracy, to assess the discriminative performance of the underlying predictive model. For each input, the model produces class scores that determine a ranked list of labels. Top-k accuracy measures the proportion of test examples for which the true label appears among the k highest scoring predictions. We report the Top-1 accuracy, together with the macro-accuracy (*i.e.*, the average of per-class Top-1 accuracies), in Tab. 16.

G Additional Experimental Results

Convergence Analysis of ALM. In Fig. 5, we ablate the role of the target size η and evaluate the impact of three commonly used penalty functions, P_2, P_3, and PHR, for ALM (see App. C for details). *Left figures* illustrate the evolution of the Lagrange multipliers $\boldsymbol{\lambda}$ and average set size over training epochs, where we observe two distinct training regimes. The multipliers are initialized close to zero. As training begins, they increase in response to class-wise constraint violations, enforcing stronger penalties on classes whose prediction set sizes exceed the target η. As the model improves and constraints become better satisfied, both the multipliers and the average set size gradually decrease, indicating that the model has learned to satisfy the constraints with reduced penalization. Crucially, the per-class evolution visible in the lower-left plot reveals that the multipliers $(\lambda_y)_{1\leq y\leq K}$ are strongly correlated with class rank in the long-tailed distribution: tail classes (low-frequency) consistently acquire higher multipliers, while head classes (high-frequency) receive lower values (Fig. 3b). This behavior emerges entirely from constraint violations on the validation set rather than from any direct use of class-frequency information, confirming that CaCT learns adaptive class-wise regularization driven by empirical constraint violations. More specifically, classes whose set-size constraints remain violated over several epochs receive repeated increases in their multipliers through (13). Consequently, classes with persistently large set sizes accumulate higher values of λ, which strengthens the optimization pressure applied to them until their average set sizes move closer to the target η. This qualitative relationship is consistent with the update rule and with Fig. 3b, where the more difficult tail classes generally receive larger multipliers. Moreover, ConfTr's soft set size plateaus slightly earlier, but at a higher value, whereas CaCT keeps refining its sets until convergence, with the multipliers stabilizing after roughly 100 epochs. *Right figure* compares performance across the three penalty functions with different values of target set size η. We find that PHR yields the best results, and setting $\eta = 1.0$ (as in [9]) achieves the best trade-off between prediction set size and coverage gap.

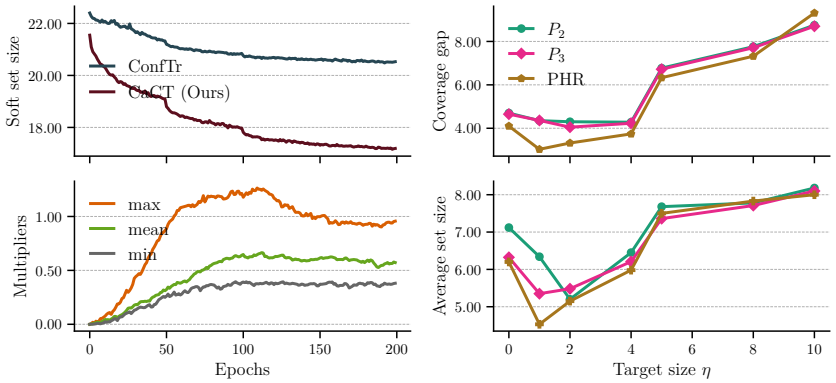

Figure 5: **Left: Evolution of average soft set size** (approximated using Sigmoid function) of CaCT and ConfTr (*top*), and per-class evolution of the Lagrange multipliers $(\lambda_y)_{1\leq y\leq K}$ for CaCT over training epochs (*bottom*), showing that tail classes accumulate higher multipliers than head classes. **Right: Effect of penalty functions and target size**: Coverage gap and average set size on test set are shown across three popular penalty functions and different target size values. Ablation study on CIFAR100-LT using ResNet and THR score.

Results with inverse-frequency weighting baseline. To isolate whether the gains of CaCT stem from (i) simply having class-wise weights, or (ii) from the ALM-driven adaptive optimization of those weights, we introduce two progressively stronger baselines: (a) $\lambda_k := 1/n_k$ (inverse-frequency weighting), which adds class-wise weights to the ConfTr objective but uses fixed, non-adaptive values, and (b) CaCT-H, which uses class-wise weights but updates them heuristically rather than via constrained optimization. Results on long-tailed datasets using the THR score confirm that the source of improvement matters: fixed class-wise weights help set size but worsen coverage gap, the heuristic rule provides a better trade-off, and only ALM's constraint-driven updates jointly minimize both metrics. Specifically, inverse-frequency weighting ($\lambda_k := 1/n_k$) improves prediction set size over ConfTr (*e.g.*, CIFAR100-LT: 5.48 vs. 7.84) but worsens coverage gap (5.36 vs. 6.31), as the fixed weights are not calibrated to actual constraint violations. CaCT-H achieves a markedly better trade-off (5.38/4.58 on CIFAR100-LT), and CaCT consistently outperforms all variants on both metrics simultaneously (3.03/4.53 on CIFAR100-LT, 4.99/4.40 on ImageNet-LT). This confirms that

Table 5: **Ablation: inverse-frequency weighting baseline** (THR score, long-tailed datasets). Lower is better. Best result in **Red**, second best in **Blue**.

	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG
THR	ConfTr [9] ICLR'22	4.35	0.90	4.92	5.05	0.90	6.29	7.84	0.90	6.31	6.60	0.90	5.64
	$\lambda_k := 1/n_k$	3.77	0.90	5.87	4.14	0.90	6.43	5.48	0.90	5.36	5.81	0.90	5.32
	CaCT-H (Ours)	**3.00**	0.90	**4.42**	**3.76**	0.90	**5.14**	**5.38**	0.90	**4.58**	**5.72**	0.90	**4.67**
	CaCT (Ours)	**2.62**	0.90	**4.22**	**3.29**	0.90	**4.42**	**3.03**	0.90	**4.53**	**4.99**	0.90	**4.40**

the gains of CaCT stem from the adaptive constraint-driven updates of ALM, not merely from adding class-wise weights.

Performance across the three versions of InfoCTr [10]. Correia et al. [10] introduced three bounds, namely Fano, MB Fano and DPI, which we review in Appx. B.2. For clarity in our empirical study, we reported in the main paper only the results obtained with the Fano bound because it consistently delivered the strongest performance among the three. To support this choice, Tabs. 6 and 7 provide the complete comparison of all three bounds against CaCT. The results show that the Fano bound achieves the strongest performance among the three bounds, which aligns with the findings of Correia et al. [10]. This behavior is expected because the Fano bound provides a tighter lower bound on the set size. In Tab. 6, for the **long tailed setting**, where the number of training samples decays exponentially across classes, our method achieves the best results for every dataset. Furthermore, when using RAPS as non-conformity score, we also observe that the heuristic rule for optimizing the multipliers performs second best after the ALM version, which further supports the robustness of our approach. In Tab. 7, which reports the results for the **balanced setting**, where all classes have the same number of training samples, CaCT remains the strongest or second strongest method across all image classification benchmarks. This holds regardless of the choice of non conformity score for Split CP. Furthermore, CaCT performs competitively on the text classification dataset, yielding the best results across the three non-conformity scores, highlighting the versatility of our method and showing that it is not limited to vision based tasks.

Table 6: **Extension of Tab. 2 – Comparison between CaCT and InfoCTr [10] bounds across long-tailed image classification datasets.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$ and an imbalance factor of $\gamma = 0.1$ for all datasets except ImageNet-LT [42]. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

Score	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG
THR	InfoCTr Fano [10] NeurIPS'24	**2.88**	0.90	4.47	4.51	0.90	5.69	**4.78**	0.90	4.96	7.14	0.90	5.08
	InfoCTr MB Fano [10] NeurIPS'24	3.05	0.90	5.28	4.93	0.90	**5.08**	7.02	0.90	5.11	7.11	0.90	**4.66**
	InfoCTr DPI [10] NeurIPS'24	2.95	0.90	4.45	5.38	0.90	5.72	7.32	0.90	6.05	6.68	0.90	4.99
	CaCT-H (Ours)	3.00	0.90	**4.42**	**3.76**	0.90	5.14	5.38	0.90	**4.58**	**5.72**	0.90	4.67
	CaCT (Ours)	**2.62**	0.90	**4.22**	**3.29**	0.90	**4.42**	**3.03**	0.90	**4.53**	**4.99**	0.90	**4.40**
APS	InfoCTr Fano [10] NeurIPS'24	**3.69**	0.90	**3.78**	5.13	0.90	**3.52**	7.18	0.90	5.65	7.27	0.90	**4.18**
	InfoCTr MB Fano [10] NeurIPS'24	4.80	0.90	4.65	3.77	0.90	4.91	7.36	0.90	4.46	**6.63**	0.90	4.32
	InfoCTr DPI [10] NeurIPS'24	4.56	0.90	**4.06**	5.30	0.90	4.48	7.24	0.90	4.97	8.10	0.90	4.61
	CaCT-H (Ours)	4.02	0.90	4.34	**3.62**	0.90	**4.29**	**5.65**	0.90	**4.12**	6.97	0.90	4.62
	CaCT (Ours)	**3.53**	0.90	4.12	**2.99**	0.90	4.52	**4.49**	0.90	**4.10**	**5.89**	0.90	**3.90**
RAPS	InfoCTr Fano [10] NeurIPS'24	4.21	0.90	4.29	6.68	0.90	4.89	**7.21**	0.90	**4.29**	7.39	0.90	5.04
	InfoCTr MB Fano [10] NeurIPS'24	3.39	0.90	4.48	5.60	0.90	5.34	9.09	0.90	4.99	6.78	0.90	5.07
	InfoCTr DPI [10] NeurIPS'24	4.86	0.90	4.77	6.25	0.90	5.01	8.84	0.90	4.83	5.92	0.90	4.98
	CaCT-H (Ours)	**3.34**	0.90	**3.84**	**4.05**	0.90	**4.84**	7.53	0.90	5.06	**5.65**	0.90	**4.24**
	CaCT (Ours)	**3.07**	0.90	**3.78**	**3.43**	0.90	**4.72**	**6.22**	0.90	**4.01**	**5.61**	0.90	**3.84**
SAPS	InfoCTr Fano [10] NeurIPS'24	4.40	0.90	4.33	6.75	0.90	4.98	**7.43**	0.90	**4.33**	7.79	0.90	4.63
	InfoCTr MB Fano [10] NeurIPS'24	3.31	0.90	4.32	5.80	0.90	4.97	9.12	0.90	5.16	6.68	0.90	5.06
	InfoCTr DPI [10] NeurIPS'24	5.07	0.90	4.83	6.49	0.90	**4.90**	8.83	0.90	4.74	5.88	0.90	4.94
	CaCT-H (Ours)	**3.07**	0.90	**3.52**	4.19	0.90	5.15	7.73	0.90	5.17	**5.78**	0.90	**4.31**
	CaCT (Ours)	**3.12**	0.90	**3.81**	**3.32**	0.90	**4.54**	**6.36**	0.90	**4.12**	**5.75**	0.90	**4.26**
PAS	InfoCTr Fano [10] NeurIPS'24	3.93	0.90	3.90	6.39	0.90	**4.79**	7.16	0.90	**4.58**	7.52	0.90	5.25
	InfoCTr MB Fano [10] NeurIPS'24	3.50	0.90	4.62	5.81	0.90	4.85	9.14	0.90	4.66	7.02	0.90	5.15
	InfoCTr DPI [10] NeurIPS'24	5.21	0.90	4.90	6.28	0.90	4.89	8.79	0.90	4.79	**5.39**	0.90	4.97
	CaCT-H (Ours)	**3.25**	0.90	**3.72**	**4.14**	0.90	4.84	7.49	0.90	4.97	5.55	0.90	**4.20**
	CaCT (Ours)	**3.34**	0.90	**3.87**	**3.31**	0.90	**4.32**	**6.05**	0.90	**3.65**	**5.30**	0.90	**3.65**

Table 7: **Extension of Tab. 1 – Comparison between CaCT and InfoCTr [10] bounds across image and text classification datasets.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

Score	Method	MNIST			CIFAR10			CIFAR100			ImageNet			20 Newsgroups		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG	S	C	CG
THR	InfoCTr Fano [10] NeurIPS'24	2.93	0.90	**3.32**	**2.31**	0.90	**4.37**	4.12	0.90	**4.24**	**5.01**	0.90	6.50	4.36	0.90	6.19
	InfoCTr MB Fano [10] NeurIPS'24	3.00	0.90	3.40	2.98	0.90	5.39	3.49	0.90	4.56	5.49	0.90	**6.33**	4.51	0.90	6.46
	InfoCTr DPI [10] NeurIPS'24	3.17	0.90	3.35	2.71	0.90	4.79	3.68	0.90	4.74	5.10	0.90	6.45	4.29	0.90	7.14
	CaCT-H (Ours)	**2.72**	0.90	3.50	2.51	0.90	4.37	**2.69**	0.90	4.34	5.86	0.90	6.74	**3.72**	0.90	**5.87**
	CaCT (Ours)	**2.14**	0.90	**3.21**	**2.45**	0.90	**4.12**	**2.56**	0.90	**4.21**	**4.99**	0.90	**6.20**	**3.59**	0.90	**5.74**
APS	InfoCTr Fano [10] NeurIPS'24	2.43	0.90	**3.09**	3.08	0.90	3.86	4.95	0.90	3.84	5.70	0.90	6.65	4.56	0.90	5.69
	InfoCTr MB Fano [10] NeurIPS'24	2.56	0.90	3.29	**2.98**	0.90	**3.75**	4.93	0.90	3.72	5.29	0.90	**6.55**	**4.44**	0.90	5.50
	InfoCTr DPI [10] NeurIPS'24	2.62	0.90	3.16	3.19	0.90	3.84	5.01	0.90	3.83	5.18	0.90	6.60	4.68	0.90	6.75
	CaCT-H (Ours)	**2.28**	0.90	3.35	3.09	0.90	3.87	**4.76**	0.90	**3.40**	**5.09**	0.90	6.95	4.87	0.90	**5.36**
	CaCT (Ours)	**2.32**	0.90	**3.15**	**2.97**	0.90	**3.56**	**3.99**	0.90	**3.22**	**5.01**	0.90	**5.80**	**4.00**	0.90	**4.67**
RAPS	InfoCTr Fano [10] NeurIPS'24	3.13	0.90	**3.08**	**3.40**	0.90	3.54	**3.45**	0.90	3.41	5.78	0.90	**5.79**	4.67	0.90	**4.68**
	InfoCTr MB Fano [10] NeurIPS'24	3.19	0.90	3.26	3.51	0.90	3.46	4.76	0.90	**3.28**	5.82	0.90	5.87	4.86	0.90	4.99
	InfoCTr DPI [10] NeurIPS'24	**3.03**	0.90	3.15	4.09	0.90	**3.44**	4.85	0.90	3.31	**5.68**	0.90	5.84	4.65	0.90	6.22
	CaCT-H (Ours)	3.75	0.90	3.31	4.19	0.90	3.63	4.19	0.90	3.78	5.83	0.90	6.86	**4.57**	0.90	5.07
	CaCT (Ours)	**3.09**	0.90	**3.10**	**3.06**	0.90	**3.21**	**3.22**	0.90	**3.18**	**5.50**	0.90	**5.10**	**3.83**	0.90	**4.60**
SAPS	InfoCTr Fano [10] NeurIPS'24	4.40	0.90	4.33	6.75	0.90	4.98	**7.43**	0.90	**4.33**	7.79	0.90	4.63	4.42	0.90	**4.55**
	InfoCTr MB Fano [10] NeurIPS'24	3.31	0.90	4.32	5.80	0.90	4.97	9.12	0.90	5.16	6.68	0.90	5.06	4.67	0.90	5.10
	InfoCTr DPI [10] NeurIPS'24	5.07	0.90	4.83	6.49	0.90	**4.90**	8.83	0.90	4.74	5.88	0.90	4.94	4.28	0.90	6.38
	CaCT-H (Ours)	**3.07**	0.90	**3.52**	**4.19**	0.90	5.15	7.73	0.90	5.17	**5.78**	0.90	**4.31**	**4.16**	0.90	5.22
	CaCT (Ours)	**3.12**	0.90	**3.81**	**3.32**	0.90	**4.54**	**6.36**	0.90	**4.12**	**5.75**	0.90	**4.26**	**3.50**	0.90	**4.54**
PAS	InfoCTr Fano [10] NeurIPS'24	**2.93**	0.90	**3.00**	**3.17**	0.90	3.68	**3.43**	0.90	3.46	5.73	0.90	**5.72**	4.51	0.90	**4.85**
	InfoCTr MB Fano [10] NeurIPS'24	2.95	0.90	3.35	3.70	0.90	3.59	4.86	0.90	**3.02**	6.03	0.90	5.74	5.02	0.90	5.05
	InfoCTr DPI [10] NeurIPS'24	**2.80**	0.90	**3.29**	4.22	0.90	**3.32**	4.64	0.90	3.27	5.65	0.90	5.75	4.51	0.90	6.27
	CaCT-H (Ours)	3.64	0.90	3.55	4.15	0.90	3.64	4.45	0.90	3.77	**5.52**	0.90	6.97	**4.31**	0.90	5.09
	CaCT (Ours)	2.94	0.90	3.31	**3.15**	0.90	**3.21**	**3.13**	0.90	**3.07**	**5.48**	0.90	**5.34**	**3.78**	0.90	**4.43**

Results with RAPS [3], SAPS [8], and PAS [6]. Due to length constraints, Tabs. 1 and 2 in main paper present results only for THR and APS as non-conformity scores. To complement these results, Tabs. 8 and 9 report the performance of all methods on long-tailed and balanced datasets, respectively, when RAPS, SAPS, and PAS are used at testing. The results exhibit a performance trend similar to that observed for THR and APS. Across all conformal training objectives, CaCT typically attains superior performance and shows greater robustness across datasets compared with alternative approaches.

Varying imbalance factor γ (extension of Tab. 3a). Tab. 10 presents the performance when varying the imbalance severity of the training set using γ (*i.e.*, as $\gamma \to 1.0$, the per-class counts become equal). In line with the THR results in Tab. 3a, CaCT consistently outperforms all other existing conformal training objectives, as well as CE and FL, in terms of prediction set size and coverage gap. These findings remain consistent under different levels of training-set imbalance, further demonstrating the robustness of CaCT to changes in class distribution.

Varying miscoverage level α at test-time (extension of Tab. 3b). Tab. 11 extends the results reported in Tab. 3b to APS and RAPS. The experiments were conducted on CIFAR100-LT with imbalance factor $\gamma = 0.1$. During training, we fixed the miscoverage level to $\alpha = 0.01$. At test time, we varied the value of α in Split CP in order to examine whether the model overfits the training miscoverage level. The observations made for THR also hold for APS and RAPS. In particular, the results show that CaCT with ALM achieves the smallest set sizes and the lowest coverage gaps, and it performs better than the CE and FL objectives that do not depend on this parameter.

Beyond Split CP. All experiments have been conducted under the Split CP setting. We further evaluate whether the same findings hold across different settings, specifically Label CP, which is a Mondrian conformal variant [2] and Cluster CP [24]. Details on how prediction sets are produced with Label CP and Cluster CP are provided in Appx. B.3. To complement the results in main paper, Tabs. 12 and 13 present the performance of all methods on image and text classification datasets under these two CP algorithms. The results show that CaCT consistently outperforms the other objectives, regardless of the CP procedure or the choice of non conformity score. This superiority comes from the fact that CaCT, combined with ALM, allows the model to adapt its behavior in a class-conditionally fashion, which leads to more reliable and better shaped prediction set sizes.

How does the temperature parameter T affect the performance of conformal training? It is important to assess whether making Split CP differentiable for backpropagation through the objective and model parameters impacts model performance with respect to prediction set size and coverage

Table 8: **Extension of Tab. 2 – Results across long-tailed image classification datasets (with RAPS [3], SAPS [8], and PAS [6] as non-conformity scores).** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$ and an imbalance factor of $\gamma = 0.1$ for all datasets except ImageNet-LT [42]. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

Score	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG
RAPS	CE	5.21	0.90	4.04	6.87	0.90	6.12	9.15	0.90	6.12	9.56	0.90	5.23
	FL [38]	5.33	0.90	4.80	6.50	0.90	6.50	9.85	0.90	6.60	9.10	0.90	4.90
	ConfTr [9] ICLR'22	4.69	0.90	4.27	5.67	0.90	5.87	9.23	0.90	5.16	8.26	0.90	4.85
	CUT [11] NeurIPS'22	3.88	0.90	**3.74**	5.02	0.90	5.48	8.03	0.90	4.90	**4.76**	0.90	4.64
	InfoCTr Fano [10] NeurIPS'24	4.21	0.90	4.29	6.68	0.90	4.89	7.21	0.90	**4.29**	7.39	0.90	5.04
	DPSM [13] ICML'25	3.42	0.90	**3.77**	4.21	0.90	**4.71**	**7.18**	0.90	4.87	5.76	0.90	4.30
	CaCT-H (Ours)	**3.34**	0.90	3.84	**4.05**	0.90	4.84	7.53	0.90	5.06	5.65	0.90	**4.24**
	CaCT (Ours)	**3.07**	0.90	3.78	**3.43**	0.90	**4.72**	**6.22**	0.90	**4.01**	**5.61**	0.90	**3.84**
SAPS	CE	5.11	0.90	4.29	7.23	0.90	6.30	8.90	0.90	6.04	9.52	0.90	5.31
	FL [38]	5.38	0.90	5.12	6.47	0.90	6.40	10.09	0.90	6.44	9.09	0.90	4.98
	ConfTr [9] ICLR'22	4.90	0.90	4.29	5.53	0.90	6.09	9.36	0.90	5.40	8.19	0.90	4.78
	CUT [11] NeurIPS'22	3.94	0.90	3.84	5.43	0.90	5.38	7.73	0.90	4.79	**4.91**	0.90	4.71
	InfoCTr Fano [10] NeurIPS'24	4.40	0.90	4.33	6.75	0.90	4.98	7.43	0.90	**4.33**	7.79	0.90	4.63
	DPSM [13] ICML'25	3.51	0.90	3.95	**4.09**	0.90	**4.88**	**7.04**	0.90	5.02	6.02	0.90	**4.25**
	CaCT-H (Ours)	**3.07**	0.90	**3.52**	4.19	0.90	5.15	7.73	0.90	5.17	5.78	0.90	4.31
	CaCT (Ours)	**3.12**	0.90	**3.81**	**3.32**	0.90	**4.54**	**6.36**	0.90	**4.12**	**5.75**	0.90	**4.26**
PAS	CE	5.31	0.90	3.97	6.68	0.90	5.80	9.25	0.90	6.27	9.77	0.90	5.05
	FL [38]	5.10	0.90	4.60	6.48	0.90	6.44	9.58	0.90	7.10	8.78	0.90	5.40
	ConfTr [9] ICLR'22	4.21	0.90	4.18	5.74	0.90	5.92	8.99	0.90	**4.46**	8.68	0.90	4.56
	CUT [11] NeurIPS'22	4.21	0.90	4.13	4.77	0.90	5.54	8.02	0.90	5.33	**4.97**	0.90	4.42
	InfoCTr Fano [10] NeurIPS'24	3.93	0.90	3.90	6.39	0.90	4.79	7.16	0.90	4.58	7.52	0.90	5.25
	DPSM [13] ICML'25	3.39	0.90	**3.70**	**4.08**	0.90	**4.56**	**7.06**	0.90	4.95	6.10	0.90	4.33
	CaCT-H (Ours)	**3.25**	0.90	**3.72**	4.14	0.90	4.84	7.49	0.90	4.97	5.55	0.90	**4.20**
	CaCT (Ours)	**3.34**	0.90	3.87	**3.31**	0.90	**4.32**	**6.05**	0.90	**3.65**	**5.30**	0.90	**3.65**

Table 9: **Extension of Tab. 1 – Results across image and text classification datasets (with RAPS [3], SAPS [8], and PAS [6] as non-conformity scores).** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

Score	Method	MNIST			CIFAR10			CIFAR100			ImageNet			20 Newsgroups		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG	S	C	CG
RAPS	CE	3.64	0.90	3.22	3.48	0.90	3.94	4.62	0.90	3.37	6.13	0.90	6.77	5.11	0.90	6.12
	FL [38]	3.39	0.90	3.18	3.98	0.90	**3.32**	4.68	0.90	3.63	6.27	0.90	5.67	5.01	0.90	6.80
	ConfTr [9] ICLR'22	3.26	0.90	3.28	3.18	0.90	3.49	4.30	0.90	3.83	6.35	0.90	**5.33**	4.26	0.90	5.88
	CUT [11] NeurIPS'22	3.63	0.90	3.84	3.12	0.90	3.92	4.55	0.90	**3.16**	**5.47**	0.90	5.84	**4.04**	0.90	5.58
	InfoCTr Fano [10] NeurIPS'24	**3.13**	0.90	**3.08**	3.40	0.90	3.54	**3.45**	0.90	3.41	5.78	0.90	5.79	4.67	0.90	**4.68**
	DPSM [13] ICML'25	3.50	0.90	3.28	**3.09**	0.90	3.45	4.08	0.90	3.49	5.54	0.90	5.83	4.39	0.90	5.19
	CaCT-H (Ours)	3.75	0.90	3.31	4.19	0.90	3.63	4.19	0.90	3.78	5.83	0.90	6.86	4.57	0.90	5.07
	CaCT (Ours)	**3.09**	0.90	**3.10**	**3.06**	0.90	**3.21**	**3.22**	0.90	**3.18**	**5.50**	0.90	**5.10**	**3.83**	0.90	**4.60**
SAPS	CE	5.11	0.90	4.29	7.23	0.90	6.30	8.90	0.90	6.04	9.52	0.90	5.31	4.92	0.90	6.33
	FL [38]	5.38	0.90	5.12	6.47	0.90	6.40	10.09	0.90	6.44	9.09	0.90	4.98	5.03	0.90	6.82
	ConfTr [9] ICLR'22	4.90	0.90	4.29	5.53	0.90	6.09	9.36	0.90	5.40	8.19	0.90	4.78	4.42	0.90	6.12
	CUT [11] NeurIPS'22	3.94	0.90	3.84	5.43	0.90	5.38	7.73	0.90	4.79	**4.91**	0.90	4.71	**3.94**	0.90	5.38
	InfoCTr Fano [10] NeurIPS'24	4.40	0.90	4.33	6.75	0.90	4.98	7.43	0.90	**4.33**	7.79	0.90	4.63	4.42	0.90	**4.55**
	DPSM [13] ICML'25	3.51	0.90	3.95	**4.09**	0.90	**4.88**	**7.04**	0.90	5.02	6.02	0.90	**4.25**	4.45	0.90	5.52
	CaCT-H (Ours)	**3.07**	0.90	**3.52**	4.19	0.90	5.15	7.73	0.90	5.17	5.78	0.90	4.31	4.16	0.90	5.22
	CaCT (Ours)	**3.12**	0.90	**3.81**	**3.32**	0.90	**4.54**	**6.36**	0.90	**4.12**	**5.75**	0.90	**4.26**	**3.50**	0.90	**4.54**
PAS	CE	3.74	0.90	**3.03**	3.54	0.90	3.74	4.57	0.90	3.49	6.27	0.90	6.51	5.20	0.90	6.58
	FL [38]	3.45	0.90	3.15	3.73	0.90	3.43	4.88	0.90	3.87	6.26	0.90	5.74	5.02	0.90	6.41
	ConfTr [9] ICLR'22	**2.87**	0.90	3.15	**2.82**	0.90	3.51	4.58	0.90	3.59	6.33	0.90	**5.35**	**4.18**	0.90	5.70
	CUT [11] NeurIPS'22	3.39	0.90	4.09	3.12	0.90	3.91	4.98	0.90	**3.44**	**5.38**	0.90	6.17	4.25	0.90	5.63
	InfoCTr Fano [10] NeurIPS'24	2.93	0.90	**3.00**	3.17	0.90	3.68	**3.43**	0.90	3.46	5.73	0.90	5.72	4.51	0.90	**4.85**
	DPSM [13] ICML'25	3.07	0.90	3.23	**2.84**	0.90	**3.19**	3.88	0.90	3.44	**5.35**	0.90	6.18	4.33	0.90	5.07
	CaCT-H (Ours)	3.64	0.90	3.55	4.15	0.90	3.64	4.45	0.90	3.77	5.52	0.90	6.97	4.31	0.90	5.09
	CaCT (Ours)	**2.94**	0.90	3.31	3.15	0.90	**3.21**	**3.13**	0.90	**3.07**	5.48	0.90	**5.34**	**3.78**	0.90	**4.43**

Table 10: **Extension of Tab. 3a – Ablation study of imbalance factor** γ on CIFAR100-LT with APS [4] and RAPS [3] as non-conformity scores. Performance comparison across different levels of class imbalance severity (lower values indicate more severe imbalance). Lower is better. **Red**: best result; **Blue**: second best.

	Method	$\gamma = 0.5$			$\gamma = 0.8$			$\gamma = 1.0$		
		S	C	CG	S	C	CG	S	C	CG
APS	CE	6.47	0.90	6.41	6.01	0.90	5.63	5.50	0.90	4.09
	FL [38]	8.79	0.90	5.61	7.07	0.90	4.78	5.49	0.90	4.62
	ConfTr [9] ICLR'22	6.85	0.90	5.24	5.95	0.90	4.51	**4.57**	0.90	3.82
	CUT [11] NeurIPS'22	**5.09**	0.90	4.00	4.90	0.90	3.99	4.87	0.90	3.80
	InfoCTr Fano [10] NeurIPS'24	6.16	0.90	4.22	5.66	0.90	3.93	4.94	0.90	3.84
	DPSM [13] ICML'25	5.54	0.90	**3.82**	5.00	0.90	3.78	4.77	0.90	3.44
	CaCT-H (Ours)	5.49	0.90	4.09	**4.88**	0.90	**3.76**	4.76	0.90	**3.40**
	CaCT (Ours)	**4.21**	0.90	**3.89**	**4.02**	0.90	**3.70**	**3.99**	0.90	**3.22**
RAPS	CE	7.44	0.90	5.56	5.90	0.90	4.78	4.62	0.90	3.37
	FL [38]	8.97	0.90	6.08	5.75	0.90	4.81	4.68	0.90	3.63
	ConfTr [9] ICLR'22	7.74	0.90	4.98	5.05	0.90	4.74	4.30	0.90	3.83
	CUT [11] NeurIPS'22	**6.11**	0.90	4.80	4.80	0.90	**4.01**	4.55	0.90	**3.16**
	InfoCTr Fano [10] NeurIPS'24	6.13	0.90	**4.11**	**4.66**	0.90	4.02	**3.45**	0.90	3.41
	DPSM [13] ICML'25	7.02	0.90	4.61	4.74	0.90	4.51	4.08	0.90	3.49
	CaCT-H (Ours)	6.50	0.90	4.95	5.98	0.90	4.88	4.19	0.90	3.78
	CaCT (Ours)	**5.25**	0.90	**3.67**	**4.35**	0.90	**3.44**	**3.22**	0.90	**3.18**

Table 11: **Extension of Tab. 3b – Ablation study of test-time mis-coverage level** α on CIFAR100-LT (the imbalance factor γ is set 0.1) with APS [4] and RAPS [3] as non-conformity scores. Lower is better. **Red**: best result; **Blue**: second best.

	Method	$\alpha = 0.01$			$\alpha = 0.05$			$\alpha = 0.1$		
		S	C	CG	S	C	CG	S	C	CG
APS	CE	21.75	0.99	4.96	13.34	0.95	5.26	9.20	0.90	6.80
	FL [38]	20.65	0.99	4.77	14.09	0.95	5.07	9.85	0.90	6.07
	ConfTr [9] ICLR'22	**17.17**	0.99	3.91	12.26	0.95	4.46	8.17	0.90	5.31
	CUT [11] NeurIPS'22	18.69	0.99	**3.26**	11.09	0.95	**3.69**	6.97	0.90	4.55
	InfoCTr Fano [10] NeurIPS'24	19.53	0.99	3.65	10.95	0.95	4.27	7.18	0.90	5.65
	DPSM [13] ICML'25	18.41	0.99	3.73	10.82	0.95	4.10	6.60	0.90	4.76
	CaCT-H (Ours)	17.34	0.99	3.53	**9.78**	0.95	4.08	**5.65**	0.90	**4.12**
	CaCT (Ours)	**16.07**	0.99	**3.07**	**8.62**	0.95	**3.66**	**4.49**	0.90	**4.10**
RAPS	CE	18.19	0.99	4.55	13.23	0.95	4.74	9.15	0.90	6.12
	FL [38]	19.11	0.99	3.43	13.94	0.95	4.94	9.85	0.90	6.60
	ConfTr [9] ICLR'22	20.95	0.99	3.50	13.51	0.95	**3.79**	9.23	0.90	5.16
	CUT [11] NeurIPS'22	**16.21**	0.99	3.53	12.03	0.95	3.85	8.03	0.90	4.90
	InfoCTr Fano [10] NeurIPS'24	16.62	0.99	**3.23**	12.55	0.95	3.95	7.21	0.90	**4.29**
	DPSM [13] ICML'25	17.17	0.99	**3.18**	11.98	0.95	3.96	**7.18**	0.90	4.87
	CaCT-H (Ours)	17.14	0.99	4.02	**11.27**	0.95	4.19	7.53	0.90	5.06
	CaCT (Ours)	**15.89**	0.99	3.38	**10.34**	0.95	**3.58**	**6.22**	0.90	**4.01**

gap. In this paper, we use a sigmoid function to smooth the indicator function in the definition of the prediction set and a temperature parameter T to control the smoothness of this approximation (*e.g.*, as $T \to \infty$, we recover the original formulation). Fig. 6 ablate the effect of this parameter on CP performance with THR, APS, and RAPS. This experiment is conducted on CIFAR100-LT. During training, we set the miscoverage level to $\alpha = 0.01$, while inference uses $\alpha = 0.1$. The results show that CaCT consistently achieves stronger performance, especially when the temperature T is small. This is expected because as T approaches 0 the sigmoid function becomes a soft approximation of the indicator function. This ablation applies only to CaCT, ConfTr [9], InfoCTr [10], and DPSM [13], because these are the only objectives that use a temperature parameter to smooth the indicator during training.

Comparison with post-hoc class-conditional CP methods (Tab. 14). While CaCT operates at training time, methods such as Label CP [1] and Cluster CP [24] pursue class-conditional efficiency at calibration time, on top of a fixed pre-trained model. Although these approaches target the same high-level goal, they are not directly comparable because conformal training and post-hoc CP operate at different stages of the pipeline and are in fact complementary (see detailed results combining both in Tabs. 12 and 13). Nevertheless, to give an indicative comparison, we report below the prediction set size (S) and coverage gap (CG) for post-hoc methods applied to a standard CE-trained model vs. SplitCP applied to CaCT +ALM (THR score, $\alpha = 0.1$):

Table 12: **Results across long-tailed image classification datasets using LabelCP.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$ and an imbalance factor of $\gamma = 0.1$ for all datasets except ImageNet-LT [42]. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

	Score	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT			20 Newsgroups		
			S	C	CG	S	C	CG	S	C	CG	S	C	CG	S	C	CG
Label CP	THR	CE	7.21	0.90	5.33	7.70	0.90	5.76	6.00	0.90	6.75	6.45	0.90	5.01	6.60	0.90	7.54
		FL [38]	5.61	0.90	6.03	4.86	0.90	7.46	5.48	0.90	7.46	6.09	0.90	6.88	4.55	0.90	7.06
		ConfTr [9] ICLR'22	5.42	0.90	6.11	4.52	0.90	7.66	5.42	0.90	7.55	6.05	0.90	7.10	4.30	0.90	7.00
		CUT [11] NeurIPS'22	4.97	0.90	5.81	4.42	0.90	7.19	5.06	0.90	7.29	5.59	0.90	**4.45**	4.23	0.90	6.83
		InfoCTr Fano [10] NeurIPS'24	4.15	0.90	7.70	5.01	0.90	**5.25**	5.60	0.90	7.15	5.65	0.90	6.80	4.00	0.90	6.50
		DPSM [13] ICML'25	3.57	0.90	5.27	**3.73**	0.90	5.26	3.91	0.90	6.70	**4.99**	0.90	5.57	**3.81**	0.90	6.88
		CaCT-H (Ours)	**3.47**	0.90	**4.82**	4.10	0.90	5.61	**3.87**	0.90	**6.42**	5.64	0.90	6.11	3.98	0.90	**6.27**
		CaCT (Ours)	**2.75**	0.90	**4.35**	**3.95**	0.90	**4.85**	**3.30**	0.90	**6.01**	**5.25**	0.90	**4.38**	**3.86**	0.90	**6.00**
	APS	CE	7.47	0.90	**3.44**	5.00	0.90	4.97	8.50	0.90	8.10	8.65	0.90	5.09	6.73	0.90	7.02
		FL [38]	7.15	0.90	3.88	5.26	0.90	5.23	7.38	0.90	7.99	7.87	0.90	4.80	5.97	0.90	6.91
		ConfTr [9] ICLR'22	6.50	0.90	4.76	5.78	0.90	5.75	5.15	0.90	7.76	6.30	0.90	4.23	4.46	0.90	6.70
		CUT [11] NeurIPS'22	5.53	0.90	4.37	5.24	0.90	5.31	4.90	0.90	7.15	**5.74**	0.90	4.06	4.26	0.90	6.44
		InfoCTr Fano [10] NeurIPS'24	4.25	0.90	5.09	5.15	0.90	5.40	4.80	0.90	7.32	5.95	0.90	3.82	4.11	0.90	6.30
		DPSM [13] ICML'25	4.55	0.90	3.83	4.97	0.90	4.67	5.03	0.90	6.71	5.86	0.90	4.21	4.14	0.90	**5.35**
		CaCT-H (Ours)	**3.86**	0.90	3.71	**4.32**	0.90	**4.56**	**4.46**	0.90	**6.11**	5.79	0.90	**3.78**	**3.91**	0.90	6.00
		CaCT (Ours)	**3.09**	0.90	**3.40**	**3.90**	0.90	**4.21**	**4.26**	0.90	**5.63**	**5.35**	0.90	**3.65**	**3.75**	0.90	**5.80**
	RAPS	CE	6.86	0.90	3.45	**4.98**	0.90	4.38	5.05	0.90	6.17	6.70	0.90	4.62	5.85	0.90	5.45
		FL [38]	6.58	0.90	4.31	5.51	0.90	5.42	4.51	0.90	7.22	5.67	0.90	4.41	4.89	0.90	6.26
		ConfTr [9] ICLR'22	6.48	0.90	4.62	5.70	0.90	5.79	4.32	0.90	7.60	**5.30**	0.90	4.33	4.55	0.90	6.55
		CUT [11] NeurIPS'22	5.96	0.90	**3.44**	5.52	0.90	**4.24**	4.22	0.90	**5.23**	**5.06**	0.90	4.47	4.49	0.90	**5.33**
		InfoCTr Fano [10] NeurIPS'24	**4.21**	0.90	3.75	5.30	0.90	5.30	4.10	0.90	7.14	5.70	0.90	4.05	**4.20**	0.90	6.15
		DPSM [13] ICML'25	4.47	0.90	3.46	5.73	0.90	4.36	**3.72**	0.90	6.63	6.07	0.90	4.09	4.23	0.90	6.25
		CaCT-H (Ours)	4.90	0.90	4.08	5.15	0.90	4.87	4.03	0.90	6.47	5.58	0.90	**3.86**	4.38	0.90	5.86
		CaCT (Ours)	**2.98**	0.90	**3.42**	**4.49**	0.90	**3.75**	**3.67**	0.90	**5.09**	5.70	0.90	**3.29**	**4.17**	0.90	**5.03**

Table 13: **Results across long-tailed image classification datasets using ClusterCP.** Prediction set size (S), empirical coverage (C) and coverage gap (CG) values with $\alpha = 0.1$ and an imbalance factor of $\gamma = 0.1$ for all datasets except ImageNet-LT [42]. The best model yields the smallest set size and coverage gap, while maintaining coverage close to the desired level. Best result in **Red**, whereas **Blue** indicates the second best result.

	Score	Method	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT			20 Newsgroups		
			S	C	CG	S	C	CG	S	C	CG	S	C	CG	S	C	CG
Cluster CP	THR	CE	6.65	0.90	4.83	6.95	0.90	**4.99**	5.65	0.90	5.98	8.45	0.90	4.40	6.50	0.90	6.95
		FL [38]	6.44	0.90	4.56	6.08	0.90	5.60	4.41	0.90	4.88	6.43	0.90	4.02	5.89	0.90	6.98
		ConfTr [9] ICLR'22	5.10	0.90	4.44	5.50	0.90	5.50	3.87	0.90	4.10	5.63	0.90	3.42	4.68	0.90	6.58
		CUT [11] NeurIPS'22	4.44	0.90	4.32	4.51	0.90	5.17	3.96	0.90	3.50	5.31	0.90	3.48	**3.43**	0.90	6.22
		InfoCTr Fano [10] NeurIPS'24	4.08	0.90	3.96	4.64	0.90	5.40	3.75	0.90	3.73	6.06	0.90	3.48	4.88	0.90	6.15
		DPSM [13] ICML'25	4.46	0.90	**3.51**	4.59	0.90	**4.78**	3.71	0.90	3.68	**5.10**	0.90	**2.67**	3.51	0.90	**5.86**
		CaCT-H (Ours)	**3.85**	0.90	3.84	**4.29**	0.90	5.03	**3.42**	0.90	**3.19**	**5.05**	0.90	**3.35**	3.86	0.90	5.98
		CaCT (Ours)	**2.63**	0.90	**3.56**	**3.47**	0.90	5.15	**3.16**	0.90	**3.42**	5.22	0.90	3.60	**3.47**	0.90	**5.88**
	APS	CE	6.70	0.90	4.40	6.11	0.90	**4.06**	9.13	0.90	6.67	7.98	0.90	4.51	7.08	0.90	6.10
		FL [38]	6.47	0.90	4.24	5.48	0.90	5.05	7.65	0.90	6.30	6.12	0.90	4.18	5.72	0.90	5.96
		ConfTr [9] ICLR'22	6.07	0.90	3.79	5.22	0.90	4.66	6.01	0.90	6.35	5.76	0.90	4.00	5.31	0.90	5.88
		CUT [11] NeurIPS'22	4.78	0.90	**3.47**	4.42	0.90	4.20	**4.01**	0.90	5.90	**5.49**	0.90	3.57	**4.68**	0.90	5.20
		InfoCTr Fano [10] NeurIPS'24	4.57	0.90	3.97	3.73	0.90	5.29	6.05	0.90	6.08	5.80	0.90	3.65	5.30	0.90	5.33
		DPSM [13] ICML'25	4.72	0.90	3.65	**3.12**	0.90	5.11	4.03	0.90	5.49	5.61	0.90	3.85	5.52	0.90	5.02
		CaCT-H (Ours)	**4.55**	0.90	3.97	3.55	0.90	5.87	4.54	0.90	**5.33**	**5.60**	0.90	**3.48**	4.86	0.90	**4.90**
		CaCT (Ours)	**4.31**	0.90	**3.51**	**3.26**	0.90	**3.58**	**3.88**	0.90	**4.83**	5.87	0.90	**3.29**	**4.09**	0.90	**4.81**
	RAPS	CE	6.65	0.90	4.10	6.10	0.90	6.83	5.99	0.90	6.10	6.07	0.90	5.19	6.67	0.90	5.24
		FL [38]	5.07	0.90	3.49	5.53	0.90	3.65	5.00	0.90	6.08	5.64	0.90	4.65	5.65	0.90	5.13
		ConfTr [9] ICLR'22	4.29	0.90	3.44	4.38	0.90	3.19	**3.92**	0.90	5.34	**5.32**	0.90	4.17	**4.24**	0.90	5.06
		CUT [11] NeurIPS'22	**3.26**	0.90	**3.12**	3.98	0.90	2.89	**4.52**	0.90	5.39	5.80	0.90	3.68	4.46	0.90	4.66
		InfoCTr Fano [10] NeurIPS'24	3.73	0.90	3.60	4.31	0.90	3.36	5.42	0.90	5.19	5.91	0.90	3.60	4.82	0.90	4.84
		DPSM [13] ICML'25	3.36	0.90	**3.33**	**3.05**	0.90	5.07	5.82	0.90	**4.69**	6.22	0.90	3.80	4.54	0.90	**4.09**
		CaCT-H (Ours)	**3.06**	0.90	3.74	3.47	0.90	**2.64**	5.90	0.90	**5.18**	5.93	0.90	**3.45**	4.26	0.90	4.52
		CaCT (Ours)	3.43	0.90	3.36	**2.95**	0.90	**2.48**	4.70	0.90	5.29	**5.48**	0.90	**3.01**	**3.93**	0.90	**3.93**

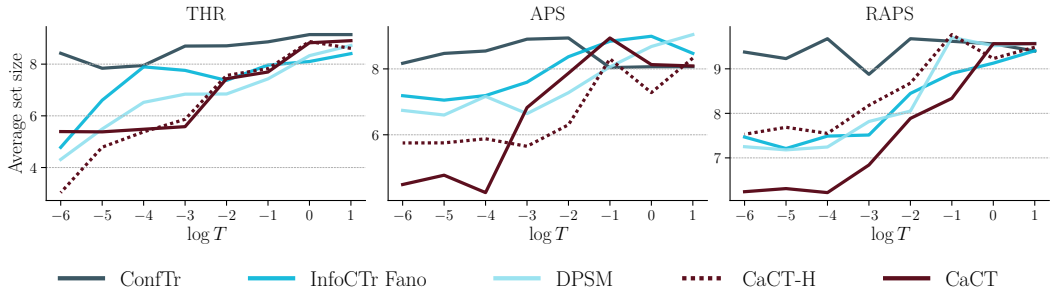

Figure 6: **Ablation study on the effect of temperature parameter** T on average set size on CIFAR100-LT (the imbalance factor γ is set 0.1) with THR, APS, and RAPS as non-conformity scores.

Table 14: **Direct comparison with post-hoc class-conditional CP methods** (THR, $\alpha = 0.1$). Post-hoc methods are applied on a CE-trained model; CaCT +ALM uses SplitCP at calibration time. Best result in **Red**, second best in **Blue**. Full results presented in Tabs. 12 and 13.

CP Method	Training	MNIST-LT			CIFAR10-LT			CIFAR100-LT			ImageNet-LT		
		S	C	CG	S	C	CG	S	C	CG	S	C	CG
Split CP	CE	**4.32**	0.90	6.93	**6.65**	0.90	6.65	8.30	0.90	6.90	9.25	0.90	5.32
Label CP		7.21	0.90	5.33	7.70	0.90	5.76	6.00	0.90	6.75	**6.45**	0.90	**5.01**
Cluster CP		6.65	0.90	**4.83**	6.95	0.90	**4.99**	**5.65**	0.90	**5.98**	8.45	0.90	**4.40**
Split CP	CaCT (Ours)	**2.62**	0.90	**4.22**	**3.29**	0.90	**4.42**	**3.03**	0.90	**4.53**	**4.99**	0.90	**4.40**

CaCTachieves consistently achieves substantially smaller prediction sets than all post-hoc approaches on the four datasets (40–61% relative improvement over Cluster CP+CE), while maintaining (*e.g.*, ImageNet-LT) or improving (*e.g.*, CIFAR100-LT) the coverage gap. This confirms that explicitly optimizing the model for class-conditional efficiency during training provides measurable benefits beyond what post-hoc calibration strategies can achieve on a fixed model.

Worst-Slice Coverage (Tab. 15). To further assess the robustness of the coverage beyond marginal and class-conditional coverage, we report the Worst-Slice Coverage (WSC; higher is better) for CE and the conformal training methods on CIFAR100-LT ($\gamma = 0.1$), using THR with $\alpha = 0.1$. CaCT achieves the highest WSC, indicating stronger coverage robustness across challenging slices of the input space.

Table 15: **Worst-Slice Coverage** (WSC, %; ↑) on CIFAR100-LT ($\gamma = 0.1$, THR, $\alpha = 0.1$).

	CE	ConfTr	DPSM	CaCT
WSC	79.6	80.7	80.9	**82.5**

Classification accuracy and macro-accuracy (Tab. 16). To verify that the gains in predictive efficiency do not come at the expense of the discriminative performance of the underlying model, Tab. 16 reports the overall (Top-1) accuracy and the macro-accuracy (*i.e.*, the average of per-class accuracies) on the long-tailed benchmarks. As expected, CE achieves higher classification accuracy because it does not optimize a conformal objective; however, it produces substantially larger prediction sets (Tab. 2). Among conformal training methods, CaCT achieves the highest accuracy on all long-tailed datasets except CIFAR10-LT, where it is within one percentage point of DPSM. Furthermore, CaCT achieves the highest macro-accuracy among conformal training methods on all long-tailed datasets, indicating that its efficiency gains do not come at the expense of under-represented classes relative to other conformal training methods.

Table 16: **Macro-accuracy / accuracy on long-tailed benchmarks.** Macro-accuracy is computed by averaging the per-class accuracies, whereas accuracy is computed over all test samples. Best result among conformal training methods in bold.

	MNIST-LT	CIFAR10-LT	CIFAR100-LT	ImageNet-LT
CE	0.84 / 0.88	0.82 / 0.86	0.53 / 0.56	0.35 / 0.39
ConfTr [9]	0.76 / 0.80	0.72 / 0.76	0.43 / 0.46	0.21 / 0.25
CUT [11]	0.75 / 0.79	0.68 / 0.72	0.40 / 0.43	0.20 / 0.24
DPSM [13]	0.76 / 0.80	0.79 / **0.84**	0.47 / 0.50	0.31 / 0.35
CaCT (Ours)	**0.80** / **0.84**	**0.80** / 0.83	**0.49** / **0.52**	**0.33** / **0.37**

H Details on Computational Efficiency

Hardware. The system we use is equipped with a 2x Intel Xeon Gold 6346 16C @ 3.1GHz while the GPUs are 4x NVIDIA A100 (40GB VRAM each), for a total of 160GB GPU VRAM.

Runtime Analysis in CaCT ALM. Tab. 17 reports the average training wall clock time per epoch, in minutes, across datasets with varying numbers of classes. The goal is to assess whether incorporating

Table 17: **Training wall-clock (minutes per epoch) across datasets with different number of classes.** Batch size is 256 and all experiments use a single A100 GPU. K is the number of classes.

Model	Dataset	K	Method	Wall-clock time (min)
Linear	MNIST	10	ConfTr	0.32
			CaCT	0.80
ResNet34	CIFAR10	10	ConfTr	1.11
			CaCT	1.15
BERT	20 Newsgroups	20	ConfTr	1.53
			CaCT	1.98
ResNet50	CIFAR100	100	ConfTr	2.61
			CaCT	2.70
ResNet50	ImageNet	1000	ConfTr	18.01
			CaCT	19.02

ALM into CaCT introduces any additional computational overhead. We recall that each training epoch of CaCT with ALM consists of two stages. First, *Model Optimization*, where the model parameters are updated using SGD. Second, *Multiplier and Penalty Updates*, where the Lagrange multipliers $\boldsymbol{\lambda}$ and the penalty parameter $\boldsymbol{\rho}$ are adjusted for each class. Further algorithmic details are provided in Appx. D. Formally, the per-epoch complexity of CaCT is $\mathcal{O}(nC + Km)$, where n is the number of training samples, C is the model-dependent cost per sample (forward pass, non-conformity score, differentiable quantile estimation, size loss, and backward pass), K is the number of classes, and m is the validation set size. The additional ALM term $\mathcal{O}(Km)$ is negligible in practice since $m \ll n$ and each per-class multiplier update is a simple scalar operation. On average across all datasets, the difference in per epoch runtime between the baseline ConfTr method and CaCT is below 0.7 minutes, which is small in practice. Combined with the performance gains demonstrated throughout the paper, these results indicate that *CaCT remains computationally efficient while producing smaller prediction sets in nearly all settings*.

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: All of our claims are supported by our experimental results and ablation studies in the main body and appendix.

   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: Limitations are discussed in Appx. A.

   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: [N/A]

   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: Appx. F provides full experimental details: dataset splits, network architectures, optimizer, learning rate schedule, batch sizes, ALM hyperparameters, grid search procedure, and calibration/test split protocol. Alg. 1 gives the complete pseudocode for CaCT.

   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: All datasets used (MNIST, CIFAR-10/100, ImageNet, ImageNet-LT, 20 Newsgroups) are standard publicly available benchmarks. We will release code upon acceptance.

   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: Appx. F details dataset splits, network architectures, optimizer and learning rate schedule, batch sizes, ALM hyperparameters, and the grid search procedure used to select them. The main body (Sec. 5.2) summarizes key experimental conditions.

   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: Results are evaluated over 10 random calibration/test splits following the standard protocol of [3, 4, 9].

   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: Appx. G reports the compute setup (2× Intel Xeon Gold 6346 CPUs, 4× NVIDIA A100 40GB GPUs) and Tab. 17 in Appx. H provides training wall-clock time per epoch for each dataset.

   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: This paper presents a methodological contribution to conformal prediction using standard, publicly available classification benchmarks. The work does not involve human subjects, private data, or any application with direct ethical risks.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Appx. A discusses positive impacts (more transparent uncertainty estimates for high-stakes applications) and explicitly states that no negative societal impacts beyond those already associated with supervised learning and conformal prediction are anticipated.

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: [N/A]

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 (MNIST, CIFAR-10/100, ImageNet, ImageNet-LT, 20 Newsgroups) and all baseline methods (ConfTr, CUT, InfoCTr, DPSM, etc.) are properly cited in the paper. All datasets used are standard academic benchmarks with established research-use licenses.

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: [N/A]

Justification: [N/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: [N/A]

    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: [N/A]

    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: [No]

Justification: LLMs were not used as part of the core methodology. Any LLM usage was limited to writing assistance and does not impact the scientific content, methodology, or originality of the research.

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.