A Theoretical Bridge Between Long-Tailed Recognition and Continual Learning

Mahdiyar Molahasani, Michael Greenspan, Ali Etemad
Department of Electrical and Computer Engineering
Queen's University, Kingston, Canada
{m.molahasani,michael.greenspan, ali.etemad}@queensu.ca

Abstract

We theoretically and empirically establish a previously unexplored connection between Long-Tailed Recognition (LTR) and Continual Learning (CL). Specifically, we show that training on a long-tailed dataset with imbalance factor IF drives model parameters into an $\mathcal{O}(1/\sqrt{\text{IF}})$ neighborhood of the dominant-class solution, under both uniform and exponentially decaying cardinality distributions. Building on this result, we establish an objective-level reformulation of LTR as a sequential learning problem and prove that the CL objective upper-bounds the balanced LTR loss up to a constant determined by the residual Head loss. Motivated by this connection, we introduce Continual Learning for Long-Tailed Recognition (CLTR), a principled framework that leverages standard off-the-shelf CL methods to sequentially learn Head and Tail classes while mitigating catastrophic forgetting. Extensive experiments on CIFAR100-LT, CIFAR10-LT, ImageNet-LT, and Caltech256 validate our theoretical predictions and demonstrate strong performance across diverse LTR benchmarks. In the foundation-model regime, CLTR with DualPrompt on a CLIP backbone outperforms standard adaptation baselines and is competitive with specialized methods using external semantic supervision. Our work bridges LTR and CL both theoretically and empirically, providing a principled approach for addressing long-tailed learning using standard CL strategies.

1 Introduction

Real-world data often exhibits long-tailed distributions [1, 2, 3, 4], where a small set of "Head" classes dominates with abundant samples, while the vast majority of "Tail" classes are severely underrepresented. This imbalance poses a fundamental challenge for deep learning: the Head classes disproportionately influence gradients during training, causing models to converge toward solutions that favor the Head while failing to generalize to the Tail [5, 6, 7].

To address this issue, various methods have been proposed, including sample-wise balancing through over-sampling the Tail classes [8, 9, 10], transfer learning approaches to adapt knowledge from Head to Tail [11, 12, 13, 14], loss- or gradient-wise balancing to mitigate class size effects [15, 16, 17], and weight regularization for ensuring balanced weight magnitudes during training [5]. More recently, *multi-stage training* has emerged as a promising solution for reducing sensitivity to Tail variations and model variance [18, 19]. Concurrently, Continual Learning (CL) has evolved as a paradigm to learn sequences of tasks without forgetting previously acquired knowledge [20]. The structural similarity between multi-stage Long-Tailed Recognition (LTR) and sequential learning prompts a fundamental question: *Can solutions for continual learning be harnessed to solve long-tailed recognition?*

To answer this, we first analyze the optimization behavior induced by long-tailed training. We prove that training on a long-tailed dataset drives the learned parameters into an $\mathcal{O}(1/\sqrt{\text{IF}})$ neighborhood of the solution obtained by training exclusively on the Head classes (Fig. 1 left). This reveals a

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

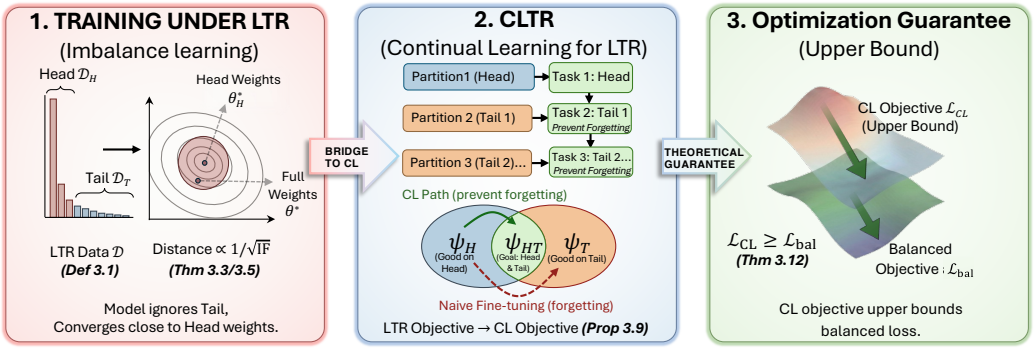

Figure 1: Overview of our framework: (left) theoretical analysis of convergence under the LTR setup, showing how imbalance drives convergence toward head-class solutions; (middle) our proposed CLTR reformulating LTR as a CL problem; and (right) the conditional optimization guarantee establishing that the CL objective upper-bounds the balanced LTR loss up to an additive constant.

structural bias in long-tailed optimization toward Head-dominated solutions and naturally motivates a sequential learning strategy: first learn the Head, then incorporate the Tail while preserving previously acquired Head knowledge. Building on this insight, we establish a principled connection between LTR and CL, showing that LTR can be reformulated as a sequential learning problem over cardinality-sorted partitions. Motivated by this perspective, we introduce Continual Learning for Long-Tailed Recognition (CLTR), a simple framework that sequentially learns Head and Tail partitions using standard off-the-shelf CL methods to mitigate catastrophic forgetting and maintain balanced performance (Fig. 1 middle). Finally, we provide a theoretical guarantee showing that the CL objective upper-bounds the balanced LTR objective, establishing CL as a principled optimization framework for LTR rather than merely a heuristic sequential training strategy (Fig. 1 right).

We validate CLTR through extensive experiments on five datasets: MNIST-LT, CIFAR100-LT, CIFAR10-LT, ImageNet-LT, and Caltech256. We first use MNIST-LT and CIFAR100-LT to empirically verify our theoretical predictions regarding the parameter dynamics of Head-only and full-dataset training. We then evaluate CLTR across standard LTR benchmarks under varying imbalance factors, consistently obtaining strong performances. Beyond standard ResNet settings, CLTR naturally integrates off-the-shelf CL methods, achieves strong results in the foundation-model adaptation regime, and extends effectively to the more challenging Long-Tailed Class-Incremental Learning (LT-CIL) setting.

Our contributions are summarized as follows: ❶ **Theoretical Analysis of Convergence:** We prove upper bounds on the distance between parameters learned on imbalanced partitions, showing this distance scales inversely with the square root of the imbalance factor, $\mathcal{O}(1/\sqrt{\text{IF}})$ (Theorems 3.3 and 3.5). ❷ **Reformulating LTR as CL:** We uncover a previously unexplored connection between long-tailed recognition and continual learning, establishing a formal objective-level reformulation of LTR as a CL problem (Proposition 3.9), directly motivated by the convergence analysis in Theorems 3.3 and 3.5. ❸ **The CLTR Framework:** Building on this reformulation, we introduce CLTR, a streamlined framework that learns Head and Tail classes sequentially using standard CL techniques to mitigate catastrophic forgetting (Algorithm 1). ❹ **Optimization Guarantee:** We establish sufficient conditions under which the CL objective upper-bounds the balanced LTR loss up to an additive constant (Theorem 3.12), providing a principled surrogate objective for long-tailed recognition. ❺ **Empirical Validation:** We demonstrate that CLTR achieves strong performance across standard LTR benchmarks, enables effective foundation model adaptation under severe class imbalance, and extends to complex LT-CIL scenarios.

2 Related Work

LTR methods have largely focused on re-balancing strategies, including re-sampling, loss re-weighting, decoupled training, and multi-stage optimization [16, 15, 21, 18, 22]. In parallel, CL methods address catastrophic forgetting in sequential learning through regularization, replay, expansion-based, or parameter-isolation mechanisms [23, 20, 24, 25, 26]. Despite structural similarities between multi-stage LTR and sequential CL, prior work has treated the two paradigms independently; in

contrast, we establish a formal theoretical connection showing that LTR admits an objective-level reformulation as a CL problem. Please see an overview of the related works in Appendix A.

3 Proposed Approach

3.1 Training on Long-Tailed Distributions

Here, we first define the LTR problem and then analyze the behavior of a model when trained under LTR setup to provide a theoretical basis for our proposed CLTR framework. For the summary of the notations see Appendix B.

Let's consider the input space to be $\mathbb{R}^d$, where each input is represented by $\mathbf{x}_i$, and the label space is $\{1, \ldots, k\}$, where each label is denoted by y_i. Let $\mathcal{D}$ denote the training set containing samples $(\mathbf{x}_i, y_i)$. $\mathcal{D}_c$ is a set of all samples belonging to class c denoted as $\mathcal{D}_c = \{(\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i = c\}$ and $|\mathcal{D}_c|$ represents its cardinality. Without loss of generality, let the classes be ordered by their cardinalities such that $|\mathcal{D}_i| \geq |\mathcal{D}_j|$ for all $i < j$. Following [27], let $\mathcal{D}_H$ and $\mathcal{D}_T$ represent the subsets of $\mathcal{D}$ corresponding to the Head set and Tail set, respectively as $\mathcal{D}_H = \{(\mathbf{x}_i, y_i) \in \mathcal{D} : y_i \leq c_k\}$ and $\mathcal{D}_T = \{(\mathbf{x}_i, y_i) \in \mathcal{D} : y_i > c_k\}$, where c_k denotes the number of classes in the Head set. As a result, every class in the Head has more samples than any class in the Tail. The loss function over $\mathcal{D}_c$ is defined as $\mathcal{L}(\mathcal{D}_c, \theta) = \frac{1}{|\mathcal{D}_c|} \sum_{i=1}^{|\mathcal{D}_c|} \ell((\mathbf{x}_i, y_i), \theta)$, where $(\mathbf{x}_i, y_i) \in \mathcal{D}_c$. Note that $\ell((\mathbf{x}_i, y_i), \theta)$ is the loss of each individual sample. For brevity, we will henceforth use notations $\mathcal{L}_{\mathcal{D}_c} = \mathcal{L}_{\mathcal{D}_c}(\theta) = \mathcal{L}(\mathcal{D}_c, \theta)$ and $\ell((\mathbf{x}_i, y_i)) = \ell_{(\mathbf{x}_i, y_i)}(\theta) = \ell((\mathbf{x}_i, y_i), \theta)$. We denote the average Head and Tail losses by $\mathcal{L}_H(\theta) = \mathcal{L}(\mathcal{D}_H, \theta)$ and $\mathcal{L}_T(\theta) = \mathcal{L}(\mathcal{D}_T, \theta)$, respectively.

LTR aims to address the challenge of learning from highly imbalanced data. This occurs when the training data $\mathcal{D}$ contains more samples in Head set ($\mathcal{D}_H$) and fewer in the Tail set ($\mathcal{D}_T$). The imbalance factor IF quantifies the severity of this issue in a dataset, defined as:

$$\text{IF} = \frac{|\mathcal{D}_{c^{\max}}|}{|\mathcal{D}_{c^{\min}}|}, \tag{1}$$

where $c^{\max} = \arg\max_c |\mathcal{D}_c|$, and $c^{\min} = \arg\min_c |\mathcal{D}_c|$, such that $\mathcal{D}_{c^{\max}} \in \mathcal{D}_H$ and $\mathcal{D}_{c^{\min}} \in \mathcal{D}_T$. Now we formally define a long-tailed dataset and the LTR problem:

Definition 3.1. A dataset is deemed *long-tailed* when $|\mathcal{D}_{c^{\max}}| \gg |\mathcal{D}_{c^{\min}}|$ or, in other words, IF $\gg 1$. When a model is trained on such a dataset and its performance is assessed on a test set where each class c has the same number of samples (i.e. $|\mathcal{D}_c| = \kappa$ for each class c within the test set where κ is a constant number), the problem is referred to as *Long-Tailed Recognition*.

3.2 Convex Case: Baseline Analysis

Here we address the case that the loss function is convex (as illustrated in Fig. 2 left), as detailed in the following:

Assumption 3.2. Let the size of all Head classes be equal, and the size of all Tail classes be equal, i.e. $|\mathcal{D}_i| = |\mathcal{D}_j|$ for all $\mathcal{D}_i, \mathcal{D}_j \in \mathcal{D}_H$ and $|\mathcal{D}_k| = |\mathcal{D}_l|$ for all $\mathcal{D}_k, \mathcal{D}_l \in \mathcal{D}_T$ (This assumption is a didactic simplification introduced to derive the $\mathcal{O}(1/\sqrt{\text{IF}})$ scaling in closed form. It is systematically relaxed in Assumptions 3.4 and 3.6.). Also, let the model be a classifier with parameters θ trained with a convex loss function $\mathcal{L}$, with an additional L^2 regularization term utilized in the training to prevent the weights from growing excessively.

We now introduce Theorem 3.3 demonstrating the relationship between the weights of the model when it is trained solely on the Head as well as on the entire dataset.

Theorem 3.3. *Given Assumption 3.2, if a model is trained in an LTR setting (Definition 3.1), then the weights of the model after training (θ^*) will lie within the bounded neighborhood of the model's weight if solely trained on Head (θ^*_H), with the scale of:*

$$\|\theta^* - \theta^*_H\| = \mathcal{O}(\frac{1}{\sqrt{\text{IF}}}). \tag{2}$$

For the full proof see Appendix C.1.

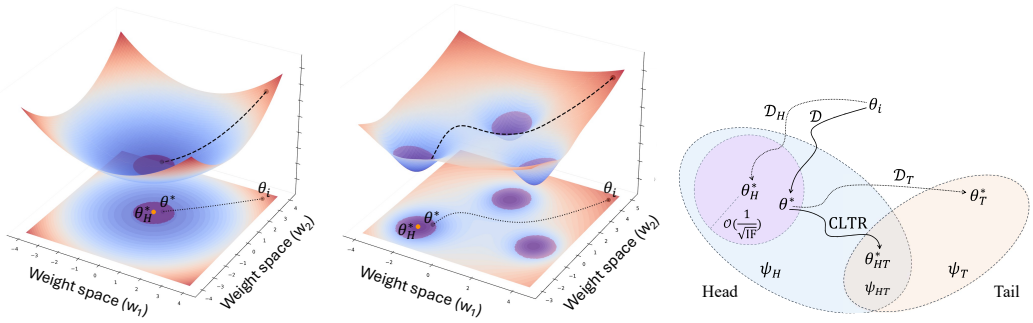

Figure 2: Left: Illustration of the Head loss landscape ($\mathcal{L}_H$) for convex setting (Assumption 3.2); the unique minimizer ($\theta_H^\star$) is marked in orange. Middle: Illustration of the Head loss landscape for feedforward deep network, KL((1/2)) setting (Assumption 3.4); multiple local minimizers ($\theta_{H,k}^\star$) are shown in orange. In both panels, the red region denotes the provable neighborhood that contains the parameter ($\theta^\star$) obtained when training on the full long-tailed dataset; its radius scales as ($\mathcal{O}(1/\sqrt{\text{IF}})$) by Theorems 3.3 and 3.5. (Top: surface; bottom: contour projection.). Right: Overview of learning under the LTR scenario and our proposed CLTR approach (symbols described in the text).

3.3 Extension to Feedforward Deep Networks

Assumption 3.2 posits a convex loss with L^2 regularization, which is convenient for analysis but not representative of modern deep networks, whose training losses are typically nonconvex. We now relax convexity and work under the widely used *Kurdyka–Łojasiewicz* (KL) framework, together with a local geometric condition that is standard for deep nets near minimizers [28] (Fig. 2 middle).

Assumption 3.4. The model is a feedforward deep neural network with piecewise-analytic/definable activations (e.g., ReLU/LeakyReLU/softplus) and a standard empirical loss (e.g., cross-entropy or squared loss). Hence, the loss $\mathcal{L}$ enjoys the Kurdyka–Łojasiewicz (KL) property. We assume that the Head loss $\mathcal{L}_H$ has KL exponent $1/2$ and satisfies a local Hölder-$1/2$ error bound in a neighborhood of $S_H = \arg\min_\theta \mathcal{L}_H(\theta)$, the set of Head-loss minimizers.

For a set S, we define $\text{dist}(\theta, S) = \inf_{\bar{\theta} \in S} \|\theta - \bar{\theta}\|_2$. Under this relaxed assumption we obtain the same scaling as in the convex case:

Theorem 3.5. *Given Assumption 3.4, there exists a bound for the distance between the weights of the model after training (θ^*) and one of the minima of the loss when training on the head solely, scaling inversely with the square root of the imbalance factor:*

$$\text{dist}(\theta^*, S_H) = \mathcal{O}(\frac{1}{\sqrt{\text{IF}}}). \tag{3}$$

For the full proof see Appendix C.3.

3.4 Extension to Multiple Head–Tail Partitions

Theorems 3.3 and 3.5 assume that there is only one Head and one Tail in the dataset, which is not the case in many real-world datasets. We relax our assumption regarding the data as follows:

Assumption 3.6. We relax Assumptions 3.2 and 3.4 by allowing individual classes within the respective Head or Tail sets to vary in size.

Under the relaxed assumption where class cardinalities within the Head and Tail sets may differ, we partition the classes into n consecutive, class-disjoint subsets ordered by decreasing cardinality. These partitions are constructed to reduce within-partition imbalance while preserving the Head-to-Tail ordering. Theorem 3.7 extends the analysis of Theorems 3.3 and 3.5 to multiple partitions, including the nonconvex setting where the Head objective may have multiple minimizers.

Theorem 3.7. *Following Assumption 3.6, let $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}^i$ be divided into $n \geq 2$ disjoint partitions, ordered such that $|\mathcal{D}^1| \geq \cdots \geq |\mathcal{D}^n|$. Define the aggregated Tail as $\mathcal{D}_T = \bigcup_{i=2}^{n} \mathcal{D}^i$ and the Head minimizer set as $S_{\mathcal{D}^1} = \arg\min_\theta F_{\mathcal{D}^1}(\theta)$, where $F_A(\theta) = \mathcal{L}(A, \theta) + \mathcal{R}(\theta)$ denotes the training objective with a shared regularizer. Let θ^* minimize $F_{\mathcal{D}}$. Under the local error-bound and bounded loss-contrast conditions of Theorem 3.5, applied to the Head and aggregated Tail, and assuming θ^* lies within the corresponding neighborhood, the following bound holds for fixed n:*

$$\operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) \leq C\sqrt{2M\frac{|\mathcal{D}_T|}{|\mathcal{D}|}} \leq C\sqrt{2M(n-1)\frac{|\mathcal{D}^2|}{|\mathcal{D}^1|}} = \mathcal{O}_n\left(\frac{1}{\sqrt{\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}}}\right), \tag{4}$$

where $\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2} = |\mathcal{D}^1|/|\mathcal{D}^2|$, and $C, M > 0$ are constants defined by the local error bound and loss-contrast condition, respectively.
When the Head objective has a unique minimizer $\theta^*_{\mathcal{D}^1}$, the set-valued distance reduces to $\|\theta^* - \theta^*_{\mathcal{D}^1}\|$. The formal proof is presented in Appendix C.4.

Remark 3.8 addresses the exponential class-frequency decay commonly used in LTR benchmarks.

Remark 3.8 (Exponential long-tailed datasets)**.** *The same inverse-square-root geometry in Theorem 3.7 holds for exponentially decaying class frequencies, which are commonly used to construct LTR benchmarks. The full statement and proof are provided in Appendix C.5.*

3.5 Continual Learning for Long-Tailed Recognition

Let us assume an LTR problem and a learner with a set of parameters denoted as θ (recall definition 3.1). Initially, the learner is trained on a highly imbalanced dataset $\mathcal{D}$, as shown in Fig. 2 right, where θ_i is the initialized model in the weight space. Owing to the larger number of Head samples in each iteration, they dominate the evolution of the gradients, resulting in a learner that performs significantly better on the Head set than on the Tail set at the end of training. This process leads the parameters to converge to θ^*.

Under the respective assumptions, Theorems 3.3 and 3.7 establish that the full-dataset minimizer θ^* lies within a bounded neighborhood of the Head-only solution in the convex case, or the Head minimizer set in the nonconvex case, with an inverse-square-root dependence on the corresponding imbalance factor.

Algorithm 1 CLTR: Head-to-Tail Learning

Input: Long-tailed dataset $\mathcal{D}$, initial parameters θ_0, number of partitions N, CL algorithm $\mathcal{A}$
Output: Final parameters θ^*_{HT}
Sort classes by cardinality: $|\mathcal{D}_{c_1}| \geq \cdots \geq |\mathcal{D}_{c_k}|$
Partition the sorted classes into N consecutive, class-disjoint subsets $\{\mathcal{D}^t\}_{t=1}^{N}$, with $R_t = \frac{\max_{\mathcal{D}_c \subseteq \mathcal{D}^t} |\mathcal{D}_c|}{\min_{\mathcal{D}_c \subseteq \mathcal{D}^t} |\mathcal{D}_c|}$ (Proposition 3.10)
Initialize CL state $S_0 \leftarrow \emptyset$
// Stage 1: Head-dominated initialization
$\mathcal{D}_{\text{init}} \leftarrow \mathcal{D}^1$ *(or $\mathcal{D}$; see Appendix D)*
$\theta_1 \leftarrow \mathrm{Train}(\theta_0, \mathcal{D}_{\text{init}})$
$S_1 \leftarrow \mathrm{InitState}_{\mathcal{A}}(\theta_1, \mathcal{D}^1)$
// Stages 2–N: Learn Tail partitions
for $t = 2$ **to** N **do**
  $(\theta_t, S_t) \leftarrow \mathrm{Update}_{\mathcal{A}}(\theta_{t-1}, \mathcal{D}^t, S_{t-1})$
end for
$\theta^*_{HT} \leftarrow \theta_N$
return θ^*_{HT}

This neighborhood falls in ψ_H which represents an area within the weight space where the network performs well on the Head set. At this stage, the model should learn the Tail; however, if it is simply fine-tuned on the Tail ($\mathcal{D}_T$), then the parameters may move toward the Tail-only solution θ^*_T within the Tail-performance region ψ_T, potentially leaving ψ_H. This phenomenon occurs in sequential learning and it is known as catastrophic forgetting. To mitigate this problem and guide the model to the intersection of ψ_H and ψ_T denoted as ψ_{HT}, where the model performs well on both Head and Tail, the Tail should be learned without forgetting the Head. To this end, we propose using the standard CL methods for sequentially learning the Tail after the Head while mitigating catastrophic forgetting and seeking a solution θ^*_{HT} that performs well on both Head and Tail classes.

Following [29], a general CL problem can be formulated as a model exposed to a stream of N incoming training datasets $\mathcal{D}_{\mathcal{Y}_t} = \{(x_i, y_i) | y_i \in \mathcal{Y}_t\}$ for $1 \leq t \leq N$, where $\mathcal{Y}_t$ is the corresponding set of labels. Up to the current timestep t, the set of labels $\bigcup_{i=1}^{t} \mathcal{Y}_i$ in dataset $\bigcup_{i=1}^{t} \mathcal{D}_{\mathcal{Y}_i}$ has been

previously used in training of the network. The objective at the next timestep $t+1$ is to find a mapping $f_\theta : x \to y$ that accurately maps sample x to $\bigcup_{i=1}^{t} \mathcal{Y}_i \cup \mathcal{Y}_{t+1}$, where $\mathcal{Y}_{t+1}$ is the set of new unseen labels in the incoming new dataset $\mathcal{D}_{\mathcal{Y}_{t+1}} = \{(x_i, y_i) | y_i \in \mathcal{Y}_{t+1}\}$. Therefore the ultimate objective of CL is to find an accurate mapping $f_\theta : x \to y \quad \text{for all} \quad (x, y) \in \bigcup_{i=1}^{N} \mathcal{D}_{\mathcal{Y}_i}$. Consider dataset $\mathcal{D}$ under the LTR setup (Definition 3.1) divided into N partitions with substantially different sizes sorted based on cardinality such that $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}^i$ and $|\mathcal{D}^i| \gg |\mathcal{D}^{i+1}|$.

Under the conditions of Theorem 3.7, the full-dataset minimizer θ^* lies within a bounded neighborhood of the minimizer set $S_{\mathcal{D}^1}$ associated with the largest partition $\mathcal{D}^1$. This proximity characterizes the Head-dominated geometry of long-tailed optimization and motivates sequentially learning the remaining partitions while preserving knowledge of the Head. On the other hand, following Definition 3.1, the objective of LTR is to learn $f_\theta : x \to y$ for all $(x, y) \in \mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}^i$. Hence, additional training steps are required for the model to further learn the rest of the partitions of the dataset ($\bigcup_{i=2}^{N} \mathcal{D}^i$). Thus, if we consider each of the partitions of the LTR dataset ($\mathcal{D}^i$ for $1 \leq i \leq N$) as an incoming CL dataset ($\mathcal{D}_{\mathcal{Y}_t}$ for $1 \leq t \leq N$), the learning objective of LTR can be cast as the objective of a CL problem (Proposition 3.9). This reformulation operates at the objective level, showing that the same hypothesis class and dataset can be learned through a CL stream sorted by partition cardinality. It does *not* claim equivalence of optimization trajectories, gradient dynamics, or solution geometry in general non-convex settings.

Proposition 3.9. *Let dataset $\mathcal{D}$ under the LTR setup (Definition 3.1) be divided into N partitions sorted by cardinality such that $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}^i$, where $|\mathcal{D}^i| \gg |\mathcal{D}^{i+1}|$. If each partition $\mathcal{D}^i$ is treated as an incoming dataset in a continual learning stream, then the objective of the LTR problem admits an objective-level reformulation as the objective of a CL problem , with the goal of learning: $f_\theta : x \to y \quad s.t. \quad (x, y) \in \bigcup_{t=1}^{N} \mathcal{D}_{\mathcal{Y}_t}$ and $\mathcal{D}_{\mathcal{Y}_1} = \mathcal{D}^1$, $\mathcal{D}_{\mathcal{Y}_2} = \mathcal{D}^2$, $\ldots$, $\mathcal{D}_{\mathcal{Y}_N} = \mathcal{D}^N$.*

A corollary of Proposition 3.9 is that CL solutions can be employed to address LTR. To operationalize this insight, we propose CLTR, a simple yet effective framework that leverages off-the-shelf CL methods to solve LTR. The complete procedure is summarized in Algorithm 1.

First, the long-tailed dataset is partitioned into N consecutive, class-disjoint subsets based on class cardinality, with boundaries selected to approximately equalize the local imbalance factors following Proposition 3.10. Starting with the largest partition (Head), each subsequent partition is treated as a new task and learned using a selected CL method $\mathcal{A}$ while preserving knowledge from previous partitions. The method-specific state S_t maintains the information required by $\mathcal{A}$, such as replay samples, regularization statistics, or teacher models. This Head-to-Tail ordering ensures that the solution obtained on the first partition serves as the Head solution θ_H^* underlying our analysis in Proposition 3.9 and Theorem 3.7, and is consistent with established LTR practice [18, 19]. For $N = 2$, the procedure reduces to the two-stage Head–Tail setting analyzed in Theorem 3.12. We empirically demonstrate the importance of this ordering through a task-order ablation in Appendix F.4.

Proposition 3.10. *For a CL algorithm $\mathcal{A}$, let $\eta_{\mathcal{A}} > 0$ denote its effective forgetting coefficient. Smaller values indicate stronger preservation of previously learned partitions. Let $B > 0$ denote a uniform upper bound on per-class losses over the considered parameter region. Under the exponential long-tailed class-count profile used in standard LTR benchmarks, the optimal number of partitions is given by*

$$N_{\mathcal{A}}^{\star} \in \arg\min_{N} \left[2B \frac{\mathrm{IF}^{1/(2N)} - 1}{\mathrm{IF}^{1/(2N)} + 1} + (N-1)\eta_{\mathcal{A}} \right]. \tag{5}$$

Moreover, for any two CL algorithms $\mathcal{A}_i$ and $\mathcal{A}_j$: $\eta_{\mathcal{A}_i} < \eta_{\mathcal{A}_j} \implies N_{\mathcal{A}_i}^{\star} \geq N_{\mathcal{A}_j}^{\star}$. Hence, stronger CL methods support a larger optimal number of partitions, while weaker CL methods should use fewer partitions. For any fixed N, the optimal partition boundaries equalize the local imbalance factors: $R_1 = R_2 = \cdots = R_N = \mathrm{IF}^{1/N}$, $R_i = \frac{\max_{\mathcal{D}_c \subseteq \mathcal{D}^i} |\mathcal{D}_c|}{\min_{\mathcal{D}_c \subseteq \mathcal{D}^i} |\mathcal{D}_c|}$. The full bias-forgetting bound and proof are given in Appendix C.6.

Proposition 3.10 provides a principled criterion for selecting the number of partitions by balancing within-partition imbalance against accumulated forgetting. In practice, the effective forgetting coefficient $\eta_{\mathcal{A}}$ can be estimated from the increase in retained-partition loss after a short pilot transition, while the observed per-class loss range on held-out data provides a proxy for B. The criterion can then be evaluated over a small set of candidate values of N. Following this principle, we use $N = 2$ for more forgetting-sensitive CL methods and $N = 4$ for stronger methods such as FOSTER and TPL. For fixed N, partition boundaries approximately equalize the local imbalance factors $R_i \approx IF^{1/N}$. We empirically evaluate sensitivity to both the number of partitions and boundary placement in Appendix F.6.

3.6 General Guarantees for Effectiveness of CLTR

We define the balanced LTR objective as $\mathcal{L}_{\mathrm{bal}}(\theta) = \frac{1}{2}\mathcal{L}_H(\theta) + \frac{1}{2}\mathcal{L}_T(\theta)$. Here we provide a theoretical *guarantee* for the effectiveness of the proposed CLTR, demonstrating that employing any off-the-shelf CL method can address LTR. To this end, we first employ the unified and general formulation of CL solutions proposed by [30], which covers a wide variety of off-the-shelf CL methods:

$$\mathcal{L}_{\mathrm{CL}}(\theta) := \mathcal{L}_{\mathrm{task}}(\theta) + \alpha\, D_\Phi\big(h_\theta(\cdot), z\big) + \beta\, D_\Psi\big(\theta, \theta_H^\star\big), \qquad \alpha, \beta \geq 0 \tag{6}$$

where h_θ is the model, D_Φ and D_Ψ are Bregman divergences generated by convex C^1 functions Φ and Ψ, and z is a fixed Head reference (e.g., stored logits/predictions at $\theta_H^\star$). In CLTR stage-2, $\mathcal{L}_{\mathrm{task}}(\theta)$ is the Tail loss ($\mathcal{L}_{\mathrm{task}} \equiv \mathcal{L}_T$). The assumptions that we have regarding the CL solution on a neighborhood $\mathcal{N} \subset \mathbb{R}^d$ of $\theta_H^\star$ are as follows:

Assumption 3.11. (Head smoothness) $\mathcal{L}_H$ is L_H^{sm}-smooth at $\theta_H^\star$: for all $\theta \in \mathcal{N}$,

$$\mathcal{L}_H(\theta) \;\leq\; \mathcal{L}_H(\theta_H^\star) \;+\; \tfrac{L_H^{\mathrm{sm}}}{2}\,\|\theta - \theta_H^\star\|^2, \;\; \forall \theta \in \mathcal{N}. \tag{7}$$

(Parameter divergence curvature) $D_\Psi(\cdot, \theta_H^\star)$ is m_Ψ-strongly convex:

$$D_\Psi(\theta, \theta_H^\star) \;\geq\; \tfrac{m_\Psi}{2}\,\|\theta - \theta_H^\star\|^2, \;\; \forall \theta \in \mathcal{N}. \tag{8}$$

(Output calibration) There exists $c_\Phi > 0$ and a fixed Head target $z = z_H^\star$ such that:

$$\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star) \;\leq\; c_\Phi\, D_\Phi\big(h_\theta(\cdot), z_H^\star\big), \;\; \forall \theta \in \mathcal{N}. \tag{9}$$

Here we explain *why Assumption 3.11 is mild.* (i) Eq. 7 is a standard local smoothness bound; it always holds with some finite L_H^{sm} if $\mathcal{L}_H$ has locally Lipschitz gradient. (ii) Eq. 8 holds for quadratic/EWC-type $\Psi(\theta) = \frac{1}{2}(\theta - \theta_H^\star)^\top H_H(\theta - \theta_H^\star)$ with a damped positive-definite $H_H \succeq m_\Psi I$ (standard in practice). It also holds on piecewise strongly convex basins (e.g., deep networks with ReLU + weight decay [31]). (iii) Eq. 9 is the usual distillation calibration (e.g., cross-entropy controlled by KL between predictions).

We now propose the general guarantee for the effectiveness of CLTR through the following theorem.

Theorem 3.12. *Suppose Assumption 3.11 holds on $\mathcal{N}$. Define the thresholds*

$$\beta_{\min} \;:=\; \frac{L_H^{\mathrm{sm}}}{2\, m_\Psi}, \qquad \alpha_{\min} \;:=\; \frac{c_\Phi}{2}, \tag{10}$$

where L_H^{sm} is local smoothness constant of $\mathcal{L}_H$ at $\theta_H^\star$, m_Ψ is strong-convexity constant of $D_\Psi(\cdot, \theta_H^\star)$, and c_Φ is calibration constant relating $\mathcal{L}_H$ to D_Φ. If either $\beta \geq \beta_{\min}$ (Parameter route) *or $\alpha \geq \alpha_{\min}$* (Output route)*, then for any $\theta \in \mathcal{N}$,*

$$\mathcal{L}_{\mathrm{bal}}(\theta) \;\leq\; \mathcal{L}_{\mathrm{CL}}(\theta) \;+\; \tfrac{1}{2}\,\mathcal{L}_H(\theta_H^\star). \tag{11}$$

In words: if either $\beta \geq \beta_{\min}$ or $\alpha \geq \alpha_{\min}$, then the CL objective $\mathcal{L}_{\mathrm{CL}}(\theta)$ upper-bounds the balanced loss $\mathcal{L}_{\mathrm{bal}}(\theta)$ up to an additive θ-independent constant. For the full proof see Appendix C.7.

Under the local conditions of Theorem 3.12, the unified CL objective (Eq. 6) upper-bounds the balanced LTR loss up to the additive constant $\frac{1}{2}\mathcal{L}_H(\theta_H^*)$, provided that either $\beta \geq \beta_{\min}$ or $\alpha \geq \alpha_{\min}$. As discussed in Appendix F.5, empirical estimates suggest that these thresholds can be satisfied

by standard CL regularization strengths, often by orders of magnitude in the evaluated settings. Importantly, the additive term is determined by the residual Head loss and is small when Head training achieves low empirical loss. Therefore, if CL optimization yields $\hat{\theta} \in \mathcal{N}$ with a low $\mathcal{L}_{\mathrm{CL}}(\hat{\theta})$, the theorem certifies a correspondingly low balanced LTR loss. Moreover, decreasing $\mathcal{L}_{\mathrm{CL}}$ decreases the corresponding upper-bound certificate along iterates within $\mathcal{N}$, although this does not guarantee a monotonic decrease in the balanced loss itself. Thus, the significance of Theorem 3.12 lies in controlling the balanced loss through a decreasing surrogate, without requiring the upper bound to be numerically tight.

Furthermore, the CL objective minimizes the full Tail-task loss $\mathcal{L}_{task} = \mathcal{L}_T$, rather than the weighted Tail component $\frac{1}{2}\mathcal{L}_T$ of the balanced objective. This gives additional emphasis to Tail learning while the CL regularizers aim to preserve previously acquired Head knowledge, counteracting the Head bias induced during initial training. Hence, CLTR provides a principled surrogate for controlling the balanced LTR loss under the stated conditions. In the special convex setting analyzed in Appendix C.8, a particular choice of the CL objective components and coefficients yields an objective that coincides with the balanced loss up to a θ-independent constant, thereby sharing identical minimizers.

4 Experiments

Datasets. First, we use the **MNIST-LT** [32] toy dataset with different IF values to study the behavior of the upper bound (Eq. 2) and its compliance with our Theorem 3.3. Then we employed **CIFAR100-LT** [15] to numerically verify Theorem 3.5. Next, to evaluate the performance of CLTR in addressing the LTR problem, we employ three widely used LTR datasets: **CIFAR100-LT**, **CIFAR10-LT** [15], and **ImageNet-LT** [11]. These datasets represent long-tailed versions of the original CIFAR100, CIFAR10, and ImageNet datasets, maintaining the same number of classes while the number of samples in each class decreases exponentially according to the IF, where the first class has the maximum number of samples and the last class contains the least number of samples, as illustrated in Appendix E. We also demonstrate the efficacy of CLTR in addressing LT-CIL using the CIFAR100-LT benchmark (Appendix F.2). Finally, to further highlight the benefits of using CLTR, we carry out additional experiments using the naturally skewed **Caltech256** dataset [33] (Appendix F.3).

Implementation Details. Following the experimental setup of [5] and [4], we use ResNet-32 [34], ResNeXt-50 [35], and CLIP ViT-B/32 [36] for CIFAR benchmarks, ImageNet benchmark, and foundation model experiments, respectively. The LTR methods selected for comparison are state-of-the-art solutions in the area. We also employ various standard and state-of-the-art CL methods in CLTR, namely LwF [20], EWC [23], Modified EWC [37], GPM [26], FOSTER [25], SGP [38], TPL [39], and DualPrompt [40]. Following proposition 3.10, we divided the dataset into 2 partitions for LwF, EWC, Modified EWC, GPM, and SGP and 4 partitions for FOSTER and TPL and the boundaries are determined by our theory. All trainings were conducted using an NVIDIA RTX 3090 GPU with 24GB VRAM. More details on the implementation specifics are provided in Appendix D.

Evaluation. For the LTR datasets (MNIST-LT, CIFAR100-LT, CIFAR10-LT, ImageNet-LT), we first train the model on the long-tailed imbalanced training set and then evaluate it on the balanced test set, following the evaluation protocol of [5]. All reported values represent classification accuracy. The results of our proposed approach are highlighted in the Tables.

Empirical support for Theorems 3.3 and 3.5. Prior works have shown that complementing theoretical guarantees with focused empirical verification can yield valuable insights into the practical relevance of the theory [62, 63, 64]. Motivated by this, our experiments are structured to test the key behaviors and scaling predicted by our analysis. To evaluate the validity of Theorem 3.3 on the upper bound for the distance between the learner's

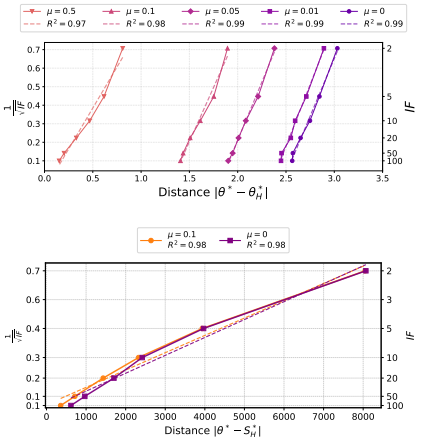

Figure 3: Empirical validation for Theorems 3.3 and 3.5. Top: log-reg on MNIST-LT with varying imbalance factors and L^2 strengths μ. Bottom: the same analysis for ResNet-18 on CIFAR-100, with and without weight decay.

Table 1: LTR benchmarks for (left) CIFAR100-LT, (middle) CIFAR10-LT, and (right) ImageNet-LT. **Bold** and underlined values denote the best and second-best results in each column.

Model	IF 100	IF 50	IF 10
Baseline [16]	38.3	43.9	55.7
Baseline + CB [16]	39.6	45.3	58.0
Focal loss [41]	38.4	44.3	55.8
Focal+CB [16]	39.6	45.2	58.0
τ-norm [21]	47.7	52.5	63.8
LDAM-DRW [15]	42.0	46.6	58.7
BBN [18]	42.6	47.0	59.1
LogitAdjust [42]	42.0	47.0	57.7
LDAM+SSP [43]	43.4	47.1	58.9
De-confound [17]	44.1	50.3	59.6
SSD [44]	46.0	50.5	62.3
DiVE [45]	45.4	51.1	62.0
DRO-LT [46]	47.3	57.6	63.4
WD [5]	46.0	52.7	66.1
WD+Max [5]	**53.4**	<u>57.7</u>	**68.7**
ABL [47]	46.8	52.1	61.8
BS + BSD [48]	50.6	55.4	64.0
PHASE [49]	49.1	52.8	61.2
CLTR (LwF)	45.1	49.3	58.7
CLTR (EWC)	44.4	50.3	58.8
CLTR (Modified EWC)	45.9	51.0	60.7
CLTR (GPM)	48.3	54.7	64.7
CLTR (FOSTER)	48.7	54.4	63.6
CLTR (SGP)	<u>50.7</u>	**58.0**	<u>67.2</u>
CLTR (TPL)	48.4	54.0	62.1

Model	IF 100	IF 50
Baseline [16]	69.8	75.2
Baseline + CB [16]	74.7	79.3
Focal loss[41]	70.4	75.3
PG Re-sampling [50]	67.1	75.0
Focal+CB[16]	74.6	79.3
LDAM-DRW[15]	77.0	79.3
BBN [18]	79.8	82.2
Manifold mixup [16]	73.0	78.1
CBA-LDAM [16]	80.3	82.2
ELF (LDAM)+DRW [16]	78.1	82.4
De-confound [17]	80.6	83.6
Hybrid-SC [51]	81.4	85.4
MiSLAS [52]	82.1	85.7
BCL [53]	<u>84.3</u>	<u>87.2</u>
WD+Max [5]	81.4	85.2
FNO [54]	82.6	85.1
BS + BSD [48]	79.7	82.7
PHASE [49]	78.9	82.7
CLTR (LwF)	76.3	78.6
CLTR (EWC)	75.1	80.1
CLTR (Modified EWC)	77.8	81.3
CLTR (GPM)	81.2	84.8
CLTR (FOSTER)	81.7	85.9
CLTR (SGP)	83.0	85.5
CLTR (TPL)	**84.7**	**87.6**

Model	Top-1 accuracy
Baseline [16]	44.4
Baseline + CB [16]	33.2
KD [55]	35.8
Focal [41]	30.5
SR Re-sampling [56]	46.8
OLTR [11]	35.6
cRT [21]	49.6
τ-norm [21]	49.4
LFME [57]	37.5
De-confound [17]	51.8
Seesaw Loss [58]	50.4
DiVE [45]	53.1
DisAlign [59]	52.9
WD [5]	48.6
WD+Max [5]	**53.9**
ABL [47]	53.2
EWB-FDR [60]	53.2
SynBalance (n=100) [61]	<u>53.6</u>
CLTR (LwF)	47.6
CLTR (EWC)	48.9
CLTR (Modified EWC)	49.1
CLTR (GPM)	51.7
CLTR (FOSTER)	52.7
CLTR (SGP)	53.2
CLTR (TPL)	**53.9**

weights when trained on $\mathcal{D}$ and $\mathcal{D}_H$ ($\|\theta^* - \theta^*_H\|$), we first
train a logistic regression model with cross-entropy loss on MNIST-LT with varying IF to ensure the convexity of the loss function [65]. We use different values for the regularizer coefficient as well (μ). Then, we compute the Euclidean distance between the two sets of weights and plot them against $\frac{1}{\sqrt{\text{IF}}}$, as shown in Fig. 3 top. A regression line is fitted to the data, and the R^2 score is calculated. Consistent with Eq. 2, the difference between the weights is bounded and scales linearly with $\frac{1}{\sqrt{\text{IF}}}$ with high confidence ($R^2 \geq 0.97$). We then extend this experiment to a deeper architecture, ResNet-18 (a feedforward deep network), trained on the CIFAR-100 dataset. We measure the distance $\|\theta^* - \theta^*_H\|$ with and without weight decay and report the results in Fig. 3 bottom. In both settings, the fitted line achieves $R^2 \geq 0.98$, indicating that the bound scales linearly with $\frac{1}{\sqrt{\text{IF}}}$ even in deep networks, where the loss is not convex, as predicted by Theorem 3.5.

Empirical support for Theorem 3.12. We compare the performance of CLTR with existing state-of-the-art LTR solutions on three benchmarks, CIFAR100-LT, CIFAR10-LT, and ImageNet-LT, as presented in Table 1. We also include two baselines trained on imbalanced data, with and without a class-balanced loss. Beyond classification accuracy, we directly examine the upper-bound inequality of Theorem 3.12 throughout Tail training on CIFAR100-LT. Across 33 recorded checkpoints per method, the inequality holds for both CLTR (LwF) and CLTR (EWC), whereas naive sequential training violates it at every checkpoint. These results demonstrate that the predicted upper-bound relationship holds throughout the evaluated CLTR training trajectories, while consistently failing under naive sequential training. Full results are provided in Appendix F.5.

Following the prior works such as [5], we avoid direct comparisons with solutions with "bells and whistles" such as RIDE [22], ACE [66], SSD [44], and PaCo [67], which employ aggressive data augmentations, ensemble learning, multi-expert architectures, and self-supervised pretraining. For completeness, we report these methods in Appendix F.1 while keeping them separate from the main comparison due to protocol differences. It is worth mentioning that some previous LTR solutions like BBN [18] learn the Head and Tail separately in a multi-stage manner. They rely on various techniques to prevent performance loss on the Head while learning the Tail. However, unlike these methods, our approach only uses one model through the entire training process, and the results demonstrate that employing standard CL methods yields strong performance in the LTR benchmarks. We further demonstrate the strong performance of CLTR on the more challenging long-tailed class-incremental learning (LT-CIL) benchmarks and on the naturally skewed Caltech256 dataset in Appendices F.2 and F.3, respectively. We also provide additional discussion on the impact of the number of incremental steps (Appendix F.6), replay memory size (Appendix F.7), runtime (Appendix F.8), weight imbalance (Appendix F.9), backward/forward transfer as well as catastrophic

forgetting (Appendix F.10), and CLTR performance on more advanced backbones (Appendix F.11) to better elucidate the mechanisms underlying CLTR. We further evaluate CLTR on the 20 Newsgroups-LT text classification benchmark, providing preliminary evidence that the framework extends beyond vision, with improved Tail and balanced accuracy over joint long-tailed training (Appendix F.12).

Adapting Foundation Models to Long-Tailed Data. Modern LTR increasingly involves adapting pretrained foundation models to scarce and imbalanced data [70]. This setting is practically important because many real-world applications have access to strong pretrained foundation models but only heavily skewed task-specific data. CLTR is naturally suited to this regime. Table 2 evaluates this setting on CIFAR100-LT with a CLIP backbone. Standard adaptation baselines expose complementary failures: zero-shot CLIP cannot exploit task data, linear probing and prompt learning have limited capacity to correct Tail bias, full fine-tuning remains Head-biased under imbalanced updates, and naive sequential training catastrophically forgets earlier partitions. CLTR (DualPrompt) directly addresses this problem through employing CL methods (details in Appendix D) and achieves the best performance without additional semantic supervision, improving over full fine-tuning by 5.8, 7.1, and 3.0 points. Gray rows report specialized VLM-LTR methods using external semantic supervision or dedicated cross-modal modules only for context; despite this stricter setting, CLTR is competitive with TASA at all IF values.

Table 2: Foundation-model adaptation on CIFAR100-LT. Bold denotes the best result without additional semantic supervision. Gray rows are contextual, non-protocol-identical comparisons.

Method	IF		
	100	50	10
Zero-shot CLIP	64.5	64.5	64.5
Linear probe	60.0	67.2	75.6
Full fine-tuning	73.4	74.8	81.0
Prompt learning	60.4	63.1	66.9
Naive sequential	48.0	51.2	53.1
CLTR (DualPrompt)	**79.2**	**81.9**	**84.0**
External semantic supervision			
BALLAD [68]	77.8	NR	NR
MMS [69]	75.2	77.5	82.0
TASA [70]	79.4	80.6	84.4

Limitations. We identify the following limitations: **Theoretical assumptions.** Theorems 3.3–3.7 characterize parameter-space distance from the Head solution under partition-based simplifications of the cardinality distribution. While Assumption 3.6 accommodates exponential decay via recursive partitioning, a closed-form treatment of fully continuous cardinality distributions remains open. **Stability–plasticity completeness.** We characterize sufficient conditions on α, β for the upper-bound to hold; a necessary characterization of the optimal stability-plasticity trade-off in the LTR regime is left to future work.

Scope. CLTR is primarily evaluated on image classification benchmarks, including CIFAR100-LT, CIFAR10-LT, ImageNet-LT, Caltech256, and CIFAR100-LT-CIL. Our experiments with a CLIP backbone further demonstrate applicability to vision-language pretrained foundation models, while the 20 Newsgroups-LT experiment provides preliminary evidence of generalization to text classification. However, joint image-text adaptation under long-tailed distributions, as well as broader evaluation on NLP, detection, segmentation, and regression tasks, remains future work. **Adaptive LTR-aware CL.** CLTR deliberately leverages off-the-shelf CL methods rather than introducing a new optimizer; its novelty lies in the theoretical LTR-to-CL connection and partitioning framework. While Proposition 3.10 provides a criterion for selecting a fixed number of partitions, a promising direction is to develop an LTR-aware CL method that estimates forgetting during training and dynamically adjusts partition boundaries and Head-preservation strength at each transition.

5 Conclusion

We considered Long-Tailed Recognition (LTR) through the lens of Continual Learning (CL). We showed that, under the stated assumptions, optimizing a long-tailed objective yields parameters within an $\mathcal{O}(1/\sqrt{\mathrm{IF}})$ neighborhood of the Head-only solution or minimizer set, in both convex and nonconvex settings. Building on this geometric insight, we established an objective-level reformulation of LTR as a sequential learning problem and proved that, under explicit local conditions, a unified CL objective upper-bounds the balanced LTR loss up to a constant determined by the residual Head loss. These findings provide a principled theoretical connection between LTR and CL. Motivated by this connection, we introduced CLTR, a framework that leverages off-the-shelf CL methods to incorporate Tail classes while preserving previously acquired Head knowledge. Extensive experiments on established LTR benchmarks, including foundation-model adaptation, demonstrated the effectiveness of this approach. We hope that viewing LTR through the lens of CL will inspire new algorithms and theoretical analyses that jointly advance both fields. Broader impact is discussed in Appendix G.

Acknowledgments and Disclosure of Funding

We would like to thank Geotab Inc., the City of Kingston, and NSERC for supporting this work. The authors declare no competing interests.

References

[1] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[2] William J Reed, "The pareto, zipf and other power laws," *Economics letters*, vol. 74, no. 1, pp. 15–19, 2001.

[3] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng, "Deep long-tailed learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[4] Yu Fu, Liuyu Xiang, Yumna Zahid, Guiguang Ding, Tao Mei, Qiang Shen, and Jungong Han, "Long-tailed visual recognition with deep models: A methodological survey and evaluation," *Neurocomputing*, 2022.

[5] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong, "Long-tailed recognition via weight balancing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6897–6907.

[6] Arman Haghanifar, Mahdiyar Molahasani Majdabadi, Younhee Choi, S Deivalakshmi, and Seokbum Ko, "Covid-cxnet: Detecting covid-19 in frontal chest x-ray images using deep learning," *Multimedia tools and applications*, vol. 81, no. 21, pp. 30615–30645, 2022.

[7] Wei Cui, Zhanyun Feng, Jiale Chen, Xing Xu, Yueling Tian, Huilin Zhao, and Chenglei Wang, "Long-tailed effect study in remote sensing semantic segmentation based on graph kernel principles," *Remote Sensing*, vol. 16, no. 8, 2024.

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[9] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[10] Chengjian Feng, Yujie Zhong, and Weilin Huang, "Exploring classification equilibrium in long-tailed object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3417–3426.

[11] 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, pp. 2537–2546.

[12] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert, "Learning to model the tail," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[13] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang, "Unequal-training for deep face recognition with long-tailed noisy data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7812–7821.

[14] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7610–7619.

[15] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[16] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie, "Class-balanced loss based on effective number of samples," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9268–9277.

[17] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1513–1524, 2020.

[18] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9719–9728.

[19] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng, "Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 34077–34090, 2022.

[20] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *International Conference on Learning Representations (ICLR)*, 2020.

[22] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu, "Long-tailed recognition by routing diverse distribution-aware experts," in *International Conference on Learning Representations (ICLR)*, 2021.

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[24] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny, "Efficient lifelong learning with a-GEM," in *International Conference on Learning Representations (ICLR)*, 2019.

[25] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 398–414.

[26] Gobinda Saha, Isha Garg, and Kaushik Roy, "Gradient projection memory for continual learning," in *International Conference on Learning Representations (ICLR)*, 2021.

[27] Yan Hong, Jianfu Zhang, Zhongyi Sun, and Ke Yan, "Proaug: Prototype-based augmentation for long-tailed image classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 3035–3039.

[28] Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee, "Stochastic subgradient method converges on tame functions," *Foundations of computational mathematics*, vol. 20, no. 1, pp. 119–154, 2020.

[29] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 524–540.

[30] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang, "A unified and general framework for continual learning," *arXiv preprint arXiv:2403.13249*, 2024.

[31] Tristan Milne, "Piecewise strong convexity of neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] Gregory Griffin, Alex Holub, and Pietro Perona, "Caltech-256 object category dataset," *Technical report*, 2007.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[35] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1492–1500.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[37] Mahdiyar Molahasani, Ali Etemad, and Michael Greenspan, "Continual learning for out-of-distribution pedestrian detection," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 2685–2689.

[38] Gobinda Saha and Kaushik Roy, "Continual learning with scaled gradient projection," *arXiv preprint arXiv:2302.01386*, 2023.

[39] Haowei Lin, Yijia Shao, Weinan Qian, Ningxin Pan, Yiduo Guo, and Bing Liu, "Class incremental learning via likelihood ratio based task prediction," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[40] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al., "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European conference on computer vision*. Springer, 2022, pp. 631–648.

[41] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[42] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar, "Long-tail learning via logit adjustment," in *International Conference on Learning Representations (ICLR)*, 2020.

[43] Yuzhe Yang and Zhi Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 19290–19301, 2020.

[44] Tianhao Li, Limin Wang, and Gangshan Wu, "Self supervision to distillation for long-tailed visual recognition," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 630–639.

[45] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei, "Distilling virtual examples for long-tailed recognition," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 235–244.

[46] Dvir Samuel and Gal Chechik, "Distributional robustness loss for long-tail learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9495–9504.

[47] Lianbao Jin, Dayu Lang, and Na Lei, "An optimal transport view of class-imbalanced visual recognition," *International Journal of Computer Vision*, vol. 131, no. 11, pp. 2845–2863, 2023.

[48] Ning Ren, Xiaosong Li, Yanxia Wu, and Yan Fu, "Balanced self-distillation for long-tailed recognition," *Knowledge-Based Systems*, vol. 290, pp. 111504, 2024.

[49] Jiale Li, Jicong Duan, Changbin Shao, and Hualong Yu, "Phase: Progressive hierarchical adaptation for sample-efficient rebalancing in long-tail classification," *Symmetry*, vol. 17, no. 12, pp. 2040, 2025.

[50] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4109–4118.

[51] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 943–952.

[52] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia, "Improving calibration for long-tailed recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16489–16498.

[53] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang, "Balanced contrastive learning for long-tailed visual recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6908–6917.

[54] Hoyong Kim and Kangil Kim, "Fixed non-negative orthogonal classifier: Inducing zero-mean neural collapse with feature dimension separation," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[55] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[56] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 181–196.

[57] Liuyu Xiang, Guiguang Ding, and Jungong Han, "Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 247–263.

[58] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin, "Seesaw loss for long-tailed instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9695–9704.

[59] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun, "Distribution alignment: A unified framework for long-tail visual recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2361–2370.

[60] Naoya Hasegawa and Issei Sato, "Exploring weight balancing on long-tailed recognition problem," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[61] Zhongyu Jiang, Jiarui Cai, Chang Liu, Dongsheng An, and Jonathan Wu, "Synbalance: Harnessing synthetic data in long-tailed recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 6276–6285.

[62] Farhad Pourpanah, Mahdiyar Molahasani, Milad Soltany, Michael Greenspan, and Ali Etemad, "Federated unsupervised domain generalization using global and local alignment of gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, vol. 39, pp. 19948–19958.

[63] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent," *Advances in neural information processing systems*, vol. 32, 2019.

[64] Xinrui Wang, Chuanxing Geng, Wenhai Wan, Shao-Yuan Li, and Songcan Chen, "Forgetting, ignorance or myopia: Revisiting key challenges in online continual learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 58341–58375, 2024.

[65] Marek Rychlik, "A proof of convergence of multi-class logistic regression network," *arXiv preprint arXiv:1903.12600*, 2019.

[66] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang, "Ace: Ally complementary experts for solving long-tailed recognition in one-shot," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 112–121.

[67] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia, "Parametric contrastive learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 715–724.

[68] Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Yu Qiao, "A simple long-tailed recognition baseline via vision-language model," *arXiv preprint arXiv:2111.14745*, 2021.

[69] Yidong Wang, Bowen Zhang, Wenxin Hou, Zhen Wu, Jindong Wang, and Takahiro Shinozaki, "Margin calibration for long-tailed visual recognition," in *Asian conference on machine learning*. PMLR, 2023, pp. 1101–1116.

[70] Long Li, Tinglei Jia, Huaizhi Yue, Huize Cheng, Yongfeng Bu, and Zhaoyang Zhang, "Tasa: Text-anchored state–space alignment for long-tailed image classification," *Journal of Imaging*, vol. 11, no. 11, pp. 410, 2025.

[71] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing: International Conference on Intelligent Computing (ICIC)*, 2005, pp. 878–887.

[72] Chris Drummond, Robert C Holte, et al., "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from Imbalanced Datasets II*, 2003, vol. 11, pp. 1–8.

[73] Li Shen, Zhouchen Lin, and Qingming Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 467–482.

[74] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang, "Deep imbalanced learning for face recognition and attribute prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2781–2794, 2019.

[75] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao, "Striking the right balance with uncertainty," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 103–112.

[76] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker, "Feature transfer learning for face recognition with under-represented data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5704–5713.

[77] Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao, "Procrustean training for imbalanced deep learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 92–102.

[78] Emanuele Francazi, Marco Baity-Jesi, and Aurelien Lucchi, "A theoretical analysis of the learning dynamics under class imbalance," *International Conference on Machine Learning (ICML)*, 2023.

[79] Syed Shakib Sarwar, Aayush Ankit, and Kaushik Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," *IEEE Access*, vol. 8, pp. 4615–4628, 2019.

[80] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *International Conference on Machine Learning (ICML)*, 2019, pp. 3925–3934.

[81] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang, "Scalable and order-robust continual learning with additive parameter decomposition," in *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[82] Gobinda Saha, Isha Garg, Aayush Ankit, and Kaushik Roy, "Space: Structured compression and sharing of representational space for continual learning," *IEEE Access*, vol. 9, pp. 150480–150494, 2021.

[83] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020, pp. 3762–3773.

[84] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *International Conference on Learning Representations (ICLR)*, 2018.

[85] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang, "Online class-incremental continual learning with adversarial shapley value," in *AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 9630–9638.

[86] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola, "Exploring visual prompts for adapting large-scale models," *arXiv preprint arXiv:2203.17274*, 2022.

[87] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim, "Imbalanced continual learning with partitioning reservoir sampling," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 411–428.

[88] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng, "Long-tailed class incremental learning," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 495–512.

[89] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and X Yu Stella, "Open long-tailed recognition in a dynamic world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[90] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, "Learning a unified classifier incrementally via rebalancing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 831–839.

[91] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 86–102.

[92] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari, "End-to-end incremental learning," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 233–248.

[93] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan, "Delta: Deep learning transfer using feature map with attention for convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[94] Bingyan Liu, Yifeng Cai, Yao Guo, and Xiangqun Chen, "Transtailor: Pruning the pre-trained model for improved transfer learning," in *AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 8627–8634.

[95] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, Davide Maltoni, et al., "Don't forget, there is more than forgetting: new metrics for continual learning," in *Continual Learning Workshop at NeurIPS*, 2018, pp. 1–7.

[96] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang, "Beyond not-forgetting: Continual learning with backward knowledge transfer," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 16165–16177, 2022.

[97] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 and cifar-100 datasets," *https://www.cs.toronto.edu/kriz/cifar.html*, 2009.

[98] Ken Lang, "Newsweeder: Learning to filter netnews," in *Machine learning proceedings 1995*, pp. 331–339. Elsevier, 1995.

Appendix

A Related Works

A.1 Long-Tailed Recognition

LTR methods broadly fall into: (i) *data distribution re-balancing*, (ii) *class-balanced losses*, and (iii) *Head-to-Tail transfer learning* [21]. Re-balancing includes Tail over-sampling [8, 71], Head under-sampling [72], class-balanced sampling [73, 56], and dynamic label distribution shift [47]. Class-balanced losses reweight samples using class statistics [15, 16, 74], focal loss [41], or Bayesian uncertainty [75]. Transfer learning leverages Head features to improve Tail performance [76, 11, 48]. Vision-language models such as CLIP demonstrate promising performance in open-vocabulary settings [36]. However, when adapted to downstream tasks with highly skewed data, they exhibit a similar bias to conventional models: strong adaptation to head classes but poor performance on tail classes. To address this issue, recent studies have proposed methods for adapting vision-language models to the LTR regime. BALLAD introduces a simple two-stage CLIP-based framework that first continues contrastive pretraining on imbalanced data and then applies a balanced linear adapter to improve tail-class performance [68]. MMS aligns visual features with their corresponding text prototypes by adding class-aware prototype-distance margins to the softmax logits, thereby improving tail-class discrimination [69]. TASA further combines LLM-based semantic prototypes, dual-space image-text fusion, and MMS alignment to improve long-tailed vision-language classification across both head and tail classes [70]. However, these methods rely on external semantic supervision, such as text prototypes or LLM-generated descriptions, leaving the problem of adapting foundation models to LTR using only labeled data largely unaddressed. Recently, multi-stage Head/Tail training has emerged as an effective LTR strategy: BBN improves representation learning [18], while RIDE reduces model variance via diverse experts [22]. These methods implicitly adopt sequential training, but rely on heuristic LTR-specific architectures. In contrast, we provide a theoretical unification explaining *why* sequential methods work: we show such specialized methods are subsets of a broader phenomenon, and by framing LTR as a CL problem, we enable *any* off-the-shelf CL algorithm to address LTR without complex, LTR-specific architectural modifications. This also allows us to leverage a rich CL literature that, unlike existing multi-stage LTR solutions, tends to use a single network throughout training.

Although many works study LTR empirically, very few approach it theoretically [77, 78]. These works show the Head is learned faster than the Tail, focusing on training dynamics; in contrast, our theory studies the *convergence point* in LTR and unifies *why* sequential strategies work, establishing that specialized LTR methods are subsets of this broader phenomenon.

A.2 Continual Learning

CL methods are commonly categorized as *expansion-based*, *regularization-based*, and *memory-based*. Expansion-based methods allocate task-specific parameters [79, 80, 81], but can grow unbounded as tasks accumulate; FOSTER mitigates this with a two-stage expand-then-compress paradigm [25]. Regularization-based methods constrain changes to important parameters via loss penalties [26, 82, 83, 23, 20]. Memory-based approaches use replay buffers to reduce forgetting [84, 24, 85]. For class-incremental learning without test-time task IDs, TPL predicts task IDs via likelihood ratios using replay data and task-specific models within a shared network [39]. More recently, gradient surgery projects new-task gradients onto directions orthogonal to previous tasks to limit interference [26, 38], achieving state-of-the-art CL performance.

B Notation

Here, we summarize the key notations used throughout the paper. A few symbols are overloaded across sections; both meanings are recorded where relevant.

- $|\cdot|$: Cardinality of a set (number of elements).
- $\|\cdot\|$: Euclidean (L_2) norm of a vector.
- $\mathcal{O}(\cdot)$: Asymptotic upper bound (Big-O).
- $\mathrm{dist}(\theta, S)$ (also written $\mathrm{dist}(\theta, S)$) : Euclidean distance from a point θ to a set S (e.g., $\mathrm{dist}(\theta^*, S_H)$ in Theorem 3.5).
- $\mathbb{R}^d$, d : Input space and its dimension; each training input is written $\mathbf{x}_i \in \mathbb{R}^d$ (boldface $\mathbf{x}$ denotes a vector-valued input).
- y_i : Scalar class label of the ith sample; labels take values in $\{1, \ldots, k\}$ when the model has k classes (in partition arguments one often writes $k = k_H + k_T$).
- f_θ : Classifier/hypothesis used in mapping statements such as $f_\theta : x \mapsto y$; h_θ denotes the same model when emphasizing logits/outputs for Bregman terms.
- $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^n\}$: Training dataset with $n = |\mathcal{D}|$ samples (throughout the main text, n denotes sample count).
- $\mathcal{D}_c = \{(\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i = c\}$: Samples of class c; $|\mathcal{D}_c|$ is the per-class training count.
- c_k : Number of Head classes (cutoff label index for Head vs. Tail).
- $\mathcal{D}_H = \{(\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i \le c_k\}$: Head subset.
- $\mathcal{D}_T = \{(\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i > c_k\}$: Tail subset.
- $\mathcal{D}^i$ (or $\mathcal{D}^i$) : The ith cardinality-sorted partition when $\mathcal{D}$ is split into multiple subsets.
- N : Number of cardinality-sorted partitions in the LTR-as-CL view (Proposition 3.9, Algorithm 1); in Theorem 3.7 and its proof the same integer is denoted by lowercase n—*not* the sample count $|\mathcal{D}|$.
- $\mathcal{D}_{\mathcal{Y}_t} = \{(\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i \in \mathcal{Y}_t\}$: Samples whose labels fall in label set $\mathcal{Y}_t$ at CL timestep t.
- κ : Number of samples per class in the balanced test protocol (Definition 3.1).
- IF : Global imbalance factor (Eq. 1).
- $\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}$ (equivalently $\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}$) : Ratio $|\mathcal{D}^1|/|\mathcal{D}^2|$ between two partitions.
- $\mathrm{IF}_{\mathcal{D}^i}$: Local imbalance when partition $\mathcal{D}^i$ is viewed as its own problem, i.e. the ratio of the largest to the smallest per-class sample count among classes assigned to that partition (coincides with R_i in Appendix C.6 when $\mathcal{D}^i$ aggregates classes in $\mathcal{C}_i$). Used in Assumption 3.6 to express that each final partition is not long-tailed.
- $c^{\max} = \arg\max_c \ |\mathcal{D}_c|$: Majority (largest) class by cardinality.
- $c^{\min} = \arg\min_c \ |\mathcal{D}_c|$: Minority (smallest) class by cardinality.
- k_H, k_T : Numbers of Head and Tail classes under Assumption 3.2 (so $k_H + k_T = k$ for total classes k).
- n_H, n_T : Samples per class within Head and within Tail under Assumption 3.2. Together with Definition 3.1's requirement that $c^{\max}$ lies in the Head and $c^{\min}$ in the Tail, this implies $\mathrm{IF} = |\mathcal{D}_{c^{\max}}|/|\mathcal{D}_{c^{\min}}| = n_H/n_T$; $\gamma := |\mathcal{D}_H|/|\mathcal{D}|$ is the Head sample mass fraction.
- $\ell((\mathbf{x}_i, y_i), \theta)$: Per-example loss (also written $\ell_{(\mathbf{x}_i, y_i)}(\theta)$).
- $\mathcal{L}(\mathcal{D}_c, \theta) = \frac{1}{|\mathcal{D}_c|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_c} \ell((\mathbf{x}_i, y_i), \theta)$: Empirical average loss on $\mathcal{D}_c$.
- $\mathcal{L}_{\mathcal{D}_c}(\theta) = \mathcal{L}(\mathcal{D}_c, \theta)$: Shorthand for the same quantity.
- $\mathcal{L}(\theta) := \mathcal{L}(\mathcal{D}, \theta)$: Full-dataset empirical risk on $\mathcal{D}$; whenever proofs add $\frac{\mu}{2}\|\theta\|^2$, they work with $\mathcal{L}_{\text{total}}$ instead.
- $\mathcal{L}_H(\theta) := \mathcal{L}(\mathcal{D}_H, \theta)$, $\mathcal{L}_T(\theta) := \mathcal{L}(\mathcal{D}_T, \theta)$: Head and Tail average losses.
- $\mathcal{L}_{\text{bal}}(\theta) := \frac{1}{2}\mathcal{L}_T(\theta) + \frac{1}{2}\mathcal{L}_H(\theta)$: Balanced LTR objective.

- $\mathcal{L}_{\text{task}}(\theta)$: CL task loss; in CLTR stage-2, $\mathcal{L}_{\text{task}} \equiv \mathcal{L}_T$.
- $\mathcal{L}_{\text{CL}}(\theta)$: Unified continual-learning surrogate from Eq. 6.
- $\mathcal{L}_{\text{total}}(\mathcal{D}, \theta) := \mathcal{L}(\mathcal{D}, \theta) + \frac{\mu}{2}\|\theta\|^2$: L^2-regularized full objective minimized in the convex proof of Theorem 3.3 (Eq. 13). *Caution:* later lines of that proof reuse the symbol $\mathcal{L}_T(\mathcal{D})$ for this same regularized quantity; it is *not* the Tail average loss $\mathcal{L}(\mathcal{D}_T, \theta)$.
- $\mathcal{L}_c(\theta)$: Class-conditional average loss on $\mathcal{D}_c$ in the partitioning analysis (Appendix C.6).
- $\mathcal{L}_i^{\text{task}}$, $\mathcal{L}_i^{\text{bal}}$, $\mathcal{L}_i^{\text{CL}}$: Per-partition task, balanced, and CL objectives in Appendix C.6; $\theta_i^\star$ and $\theta_i^{\text{bal},\star}$ denote their respective minimizers.
- δ : $\max_\theta |\mathcal{L}(\mathcal{D}, \theta) - \mathcal{L}(\mathcal{D}_H, \theta)|$ in the convex proof; scales as $\mathcal{O}(1/\,\text{IF})$.
- λ, λ_H : Smallest eigenvalues of $\nabla^2\mathcal{L}_{\text{total}}(\mathcal{D}, \cdot)$ and $\nabla^2\mathcal{L}_{\text{total}}(\mathcal{D}_H, \cdot)$ in Lemma C.1; more generally λ_f, λ_g are smallest Hessian eigenvalues for auxiliary functions f, g in Lemma C.1.
- $D_\Phi(\cdot,\cdot)$, $D_\Psi(\cdot,\cdot)$: Bregman divergences generated by convex C^1 maps Φ (output space) and Ψ (parameter space).
- $D_{\mathcal{L}_H}(\theta, \theta_H^\star)$: Bregman divergence generated by the Head loss $\mathcal{L}_H$ (when used).
- z : Fixed Head reference for output regularization (e.g., stored logits); $z_H^\star$ is the calibrated Head target in Assumption 3.11.
- $D_{\Phi,i}$, $D_{\Psi,i}$, α_i, β_i, $c_{\Phi,i}$, z_{i-1} : Stage-i output/parameter regularizers and weights in Appendix C.6; z_{i-1} anchors divergences at the end of stage $i-1$.
- θ_{i-1} : Parameters at the end of stage $i-1$ (initial point for stage i CL objectives). Distinct from θ_i in the main-text figure, which denotes an initialization before long-tailed training, and from $\theta_i^\star$, the exact minimizer of $\mathcal{L}_i^{\text{CL}}$.
- $\Delta_i := \left[\mathcal{L}_i^{\text{task}}(\theta_{i-1}) - \mathcal{L}_i^{\text{task}}(\theta_i^\star)\right]_+$: Nonnegative "current-task learning pressure" after stage i (Appendix C.6).
- $\mathcal{C}_{<i} := \bigcup_{j<i} \mathcal{C}_j$: Union of class-index sets from partitions preceding i.
- $\mathcal{L}_{<i}^{\text{bal}}(\theta)$: Uniform average $\frac{1}{|\mathcal{C}_{<i}|}\sum_{c\in\mathcal{C}_{<i}} \mathcal{L}_c(\theta)$ over old classes (Appendix C.6).
- $\mathcal{L}_{i,\text{old}}^{\text{sm}}$, $m_{\Psi,i}$ (and uniform shorthands $L_{\text{old}}^{\text{sm}}$, m_Ψ) : Local smoothness of old-task balanced losses and strong-convexity constants for $D_{\Psi,i}$ in the multi-stage bounds (Appendix C.6).
- $\mathcal{L}_{\text{bal}}^\star$: Shorthand for the minimum attainable globally balanced risk in the idealized analysis of Appendix C.6.
- OptErr : Slack collecting finite-horizon optimization and approximation errors in Appendix C.6.
- $\frac{\mu}{2}\|\theta\|^2$: L^2 regularization with coefficient $\mu > 0$. *Important:* in Appendix C.5, $\mu(\cdot)$ also denotes a *measure* on $[0,1]$ (e.g., $d\mu(u) = w_0\,\text{IF}^{-u}\,du$); it is always clear from integration vs. penalty context.
- θ : Learnable parameters; θ^i (superscript) denotes parameters after the ith training iterate in the global optimization loop. The subscripted θ_i appearing in Fig. 2 denotes an initialization before long-tailed training (not a partition-stage parameter).
- θ^* / $\theta^\star$: Minimizer of the full-data objective being analyzed in each result (context-dependent).
- θ_H^* / $\theta_H^\star$: Head-only minimizer on $\mathcal{D}_H$.
- θ_T^* : Minimizer when training on $\mathcal{D}_T$ alone.
- θ_{HT}^* : Target solution with good Head and Tail behavior (intersection regime ψ_{HT}).
- $\theta_{\mathcal{D}^1}^*$, θ_1^* : Minimizer when training only on the largest partition $\mathcal{D}^1$ (Theorem 3.7).
- $\theta_{\mathcal{B}}^*$ for $\mathcal{B} \subseteq \mathcal{D}$: Minimizer of the empirical risk restricted to subset $\mathcal{B}$ (union-of-partitions notation in Appendix proofs).
- $\theta_{H,k}^\star$: k^{th} local Head minimizer in the non-convex setting (Theorem 3.5).
- θ_k : Generic iterate along an optimization trajectory (e.g., after k steps).
- $h_\theta(\cdot)$: Network map to logits/predictions used inside D_Φ.

- ψ_H, ψ_T, ψ_{HT} : Informal "good-performance" regions in weight space for Head, Tail, and both.
- $\mathcal{Y}_t$: Label set at CL step t.
- $\mathcal{M}$: Replay memory buffer.
- F : Fisher information matrix (e.g., EWC).
- H_H : Positive-definite matrix in quadratic/EWC-style $\Psi(\theta) = \frac{1}{2}(\theta - \theta_H^\star)^\top H_H(\theta - \theta_H^\star)$.
- S_H : Set of minimizers of $\mathcal{L}_H$ on $\mathcal{D}_H$ (a singleton under strict convexity; possibly non-unique in Theorem 3.5).
- C : Positive constant in the local Hölder–$\frac{1}{2}$/KL-type bound $\operatorname{dist}(\theta^*, S_H) \leq C\sqrt{\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta_H^*)}$ from Theorem 3.5's proof (any fixed $\theta_H^* \in S_H$ on the right-hand side).
- M : In Theorems 3.3 and 3.5, $M := \max_\theta \left|\mathcal{L}(\mathcal{D}_T, \theta) - \mathcal{L}(\mathcal{D}_H, \theta)\right|$ (finite under the proof's bounded-loss assumption).
- $\mathcal{N} \subset \mathbb{R}^d$: Neighborhood of $\theta_H^\star$ where Assumption 3.11 holds.
- L_H^{sm}, m_Ψ, c_Φ : Local smoothness of $\mathcal{L}_H$, strong-convexity parameter for $D_\Psi(\cdot, \theta_H^\star)$, and output calibration constant.
- α, β : Weights in $\mathcal{L}_{\mathrm{CL}}$ (Eq. 6); $\alpha_{\min}$, $\beta_{\min}$ are the thresholds in Theorem 3.12.
- $\mathcal{A}$: Continual-learning algorithm in Proposition 3.10.
- $\eta_{\mathcal{A}} \geq 0$: Effective forgetting coefficient for algorithm $\mathcal{A}$ in Proposition 3.10; plain η denotes the uniform-stage surrogate grouped into $S(N)$ in Appendix C.6.
- $N_{\mathcal{A}}^\star$: Any member of $\arg\min_N$ of the bracket in Eq. 5 for algorithm $\mathcal{A}$ (optimal partition count under the stated tradeoff).
- $N^\star$: Same for the simplified score $S(N)$ when algorithm dependence is suppressed (Appendix C.6).
- B : Uniform upper bound on per-class losses $\mathcal{L}_c(\theta)$ over the optimization region in Appendix C.6.
- $\mathcal{C}_i$: Set of class indices assigned to cardinality partition i.
- R_i : Local imbalance for partition i. Discrete datasets: $R_i = \max_{c \in \mathcal{C}_i} n_c / \min_{c \in \mathcal{C}_i} n_c$ with $n_c = |\mathcal{D}_c|$. Under the continuous exponential profile $w(u) = w_0\,\mathrm{IF}^{-u}$ with boundaries $0 = u_0 < \cdots < u_N = 1$, also $R_i = w(u_{i-1})/w(u_i) = \mathrm{IF}^{u_i - u_{i-1}}$ (Proposition 3.10).
- p_c, q_c, r_c : Empirical class weights $p_c = n_c / \sum_{d \in \mathcal{C}_i} n_d$, uniform $q_c = 1/|\mathcal{C}_i|$, and ratio $r_c = p_c/q_c$ inside partition i (Appendix C.6); $\mathrm{TV}(p, q) = \frac{1}{2}\sum_{c \in \mathcal{C}_i} |p_c - q_c|$ is total variation.
- $S(N)$: Scalar tradeoff score $2B\,\frac{\mathrm{IF}^{1/(2N)} - 1}{\mathrm{IF}^{1/(2N)} + 1} + (N-1)\eta$ minimized to choose N when stage-wise constants are uniform (Appendix C.6).
- $\mathcal{U} = [0, 1]$: Continuous class-index domain.
- $w(u) = w_0\,\mathrm{IF}^{-u}$: Decaying class-frequency profile ($w_0 > 0$ a scale).
- $d\mu(u) = w(u)\,du$: Induced measure on $\mathcal{U}$ (here μ is a measure, not the L^2 weight).
- $L_A(\theta)$: Normalized risk on measurable $A \subseteq [0, 1]$, including shared regularizer $R(\theta)$.
- $\ell_u(\theta)$: Integrand loss at continuous index u.
- P_τ, T_τ : Head prefix $[0, \tau]$ and Tail suffix $(\tau, 1]$.
- $\varepsilon_\tau = \mu(T_\tau)/\mu(\mathcal{U})$: Tail mass fraction.
- S_{P_τ}, $L_{P_\tau}^\star$: Minimizer set of L_{P_τ} and its optimal value.
- $\bar{\theta}_{P_\tau}$: Any member of S_{P_τ} used as a witness in the exponential-decay proof (Appendix C.5).
- $\theta_{\mathcal{U}}^\star$: Minimizer of the continuum full-domain risk $L_{\mathcal{U}}$ in Appendix C.5.
- $\theta_{\mathcal{D}}^\star$: Full-dataset minimizer in the discrete remark statement (Remark 3.8); conceptually parallel to $\theta^\star$ and $\theta_{\mathcal{U}}^\star$.

- C_τ : KL/Hölder-type constant for prefix P_τ.
- M_τ : Uniform bound on $|L_{T_\tau}(\theta) - L_{P_\tau}(\theta)|$ over $\theta \in \mathcal{S}$ in Appendix C.5.
- $\mathrm{IF}_\tau := w(0)/w(\tau)$: Boundary imbalance factor (equals IF^τ under this profile).
- $\mathcal{S}$: Local sublevel set on which loss contrasts are uniformly bounded.

C Proofs

Here we provide proofs for all the theorems. For convenience, we restate each theorem at the start of its corresponding section before presenting the proof.

C.1 Proof of Theorem 3.3

Theorem (Theorem 3.3 (restated))**.** Given Assumption 3.2, if a model is trained in an LTR setting (Definition 3.1), then the weights of the model after training (θ^*) will lie within the bounded neighborhood of the model's weight if solely trained on Head (θ_H^*), with the scale of:

$$\|\theta^* - \theta_H^*\| = \mathcal{O}(\frac{1}{\sqrt{\text{IF}}}). \tag{12}$$

Proof Sketch. We start by expressing the total loss as a weighted combination of the Head and Tail set losses, where the weights depend on the dataset imbalance factor IF. As IF becomes larger, the contribution of the Tail set loss diminishes, and the total loss approaches the Head set loss. Next, we bound the difference between the total loss and the Head set loss in terms of IF. Using a lemma that connects loss differences to the proximity of their respective minimizers under Assumption 3.2, we translate this bound into a relationship between the model parameters obtained from the total dataset and those obtained from the Head set. Finally, we show that the loss difference scales as $\mathcal{O}(1/\text{IF})$, leading to the parameter difference scaling of $\mathcal{O}(1/\sqrt{\text{IF}})$.

Proof of Theorem 3.3. Following Assumption 3.2, the model is trained on the entire dataset $\mathcal{D}$ by minimizing the loss function $\mathcal{L}_T$, which can be formulated as:

$$\mathcal{L}_{total}(\mathcal{D}) = \mathcal{L}(\mathcal{D}) + \frac{\mu}{2}\|\theta\|^2, \tag{13}$$

The loss $\mathcal{L}(\mathcal{D})$ can be decomposed as:

$$\mathcal{L}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\left(\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_H} \ell((\mathbf{x}_i,y_i)) + \sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_T} \ell((\mathbf{x}_i,y_i))\right), \tag{14}$$

Using $\mathcal{L}(\mathcal{D}_H) = \frac{1}{|\mathcal{D}_H|}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_H} \ell((\mathbf{x}_i,y_i))$ and $\mathcal{L}(\mathcal{D}_T) = \frac{1}{|\mathcal{D}_T|}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_T} \ell((\mathbf{x}_i,y_i))$, we can rewrite $\mathcal{L}(\mathcal{D})$ as:

$$\mathcal{L}(\mathcal{D}) = \frac{|\mathcal{D}_H|}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}_H) + \frac{|\mathcal{D}_T|}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}_T). \tag{15}$$

Let $\gamma = \frac{|\mathcal{D}_H|}{|\mathcal{D}|}$. Since $|\mathcal{D}| = |\mathcal{D}_H| + |\mathcal{D}_T|$, we can derive that $1-\gamma = \frac{|\mathcal{D}_T|}{|\mathcal{D}|}$. Plugging γ into Eq. 15 yields:

$$\mathcal{L}(\mathcal{D}) = \gamma\mathcal{L}(\mathcal{D}_H) + (1-\gamma)\mathcal{L}(\mathcal{D}_T). \tag{16}$$

We now express γ in terms of the imbalance factor IF in Definition 3.1 under the LTR setting. Let k_H and k_T denote the number of Head and Tail classes, respectively ($k_H + k_T = k$). Under Assumption 3.2 (LTR setting with approximately constant per-class sizes within Head and within Tail), let n_H be the number of samples per Head class and n_T be the number of samples per Tail class. Then $|\mathcal{D}_H| = k_H n_H$ and $|\mathcal{D}_T| = k_T n_T$. Moreover, by Definition 3.1, $\text{IF} = \frac{n_H}{n_T}$, hence

$$\gamma = \frac{|\mathcal{D}_H|}{|\mathcal{D}_H| + |\mathcal{D}_T|} = \frac{k_H n_H}{k_H n_H + k_T n_T} = \frac{k_H\,\text{IF}}{k_H\,\text{IF} + k_T}, \tag{17}$$

and therefore

$$1-\gamma = \frac{k_T}{k_H\,\text{IF} + k_T}. \tag{18}$$

Since in LTR we have $\text{IF} \gg 1$ (Definition 3.1), $\gamma \to 1$ and $1-\gamma = \mathcal{O}(1/\text{IF})$. Consequently, $\mathcal{L}(\mathcal{D})$ approaches $\mathcal{L}(\mathcal{D}_H)$ for all θ values.

Let δ represent the maximum difference between the losses:

$$\delta = \max_\theta |\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)|. \tag{19}$$

From Eq. 16, it follows that as IF $\to \infty$, $\delta \to 0$. To find the asymptotic relationship of δ with IF, we derive:

$$|\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)| = |(1-\gamma)(\mathcal{L}(\mathcal{D}_T) - \mathcal{L}(\mathcal{D}_H))|, \tag{20}$$

Replacing $1-\gamma$ with $\frac{k_T}{k_H \,\mathrm{IF} + k_T}$ from Eq. 18 and simplifying:

$$|\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)| = \frac{k_T}{k_H \,\mathrm{IF} + k_T} |\mathcal{L}(\mathcal{D}_T) - \mathcal{L}(\mathcal{D}_H)|. \tag{21}$$

By applying maximization over θ, we derive:

$$\max_\theta |\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)| = \frac{k_T}{k_H \,\mathrm{IF} + k_T} \max_\theta |\mathcal{L}(\mathcal{D}_T) - \mathcal{L}(\mathcal{D}_H)| \,. \tag{22}$$

Since IF is independent of θ:

$$\max_\theta |\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)| = M \frac{k_T}{k_H \,\mathrm{IF} + k_T}, \tag{23}$$

where $M = \max_\theta(|\mathcal{L}(\mathcal{D}_T) - \mathcal{L}(\mathcal{D}_H)|)$. We assume $M < \infty$ (e.g., ℓ is bounded/clipped or the optimization domain is restricted). Since $\mathcal{L}(\mathcal{D}_T)$ and $\mathcal{L}(\mathcal{D}_H)$ are average loss over Head and Tail, respectively, they are independent of IF. Moreover, number of classes in the Head (k_H) and number of classes in the Tail (k_T) are also independent of IF. Consequently, replacing δ from Eq. 19 in Eq. 23 yields:

$$\delta = \mathcal{O}\left(\frac{1}{\mathrm{IF}}\right). \tag{24}$$

Adding and subtracting the regularizer $\frac{\mu}{2}\|\theta\|^2$ in Eq. 19 yields:

$$\delta = \max_\theta |\mathcal{L}_T(\mathcal{D}) - \mathcal{L}_T(\mathcal{D}_H)|. \tag{25}$$

We now use the following lemma to relate the difference in losses to the difference in the parameter vectors:

Lemma C.1. *Let $f, g : \mathbb{R}^d \to \mathbb{R}$ satisfy $|f(x) - g(x)| \le \delta$ for all x, and let $x_f \in \arg\min f$, $x_g \in \arg\min g$. Assume quadratic growth around the minimizers with constants $\lambda_f, \lambda_g \ge 0$, $\lambda_f + \lambda_g > 0$: $f(y) \ge f(x_f) + \frac{\lambda_f}{2}\|y - x_f\|^2$, $\quad g(y) \ge g(x_g) + \frac{\lambda_g}{2}\|y - x_g\|^2 \quad \forall y$. (This holds in particular if f is λ_f-strongly convex and g is λ_g-strongly convex.) Then $\|x_f - x_g\|^2 \le \frac{4\delta}{\lambda_f + \lambda_g}$.*

Applying Lemma C.1 to Eq. 25, we get:

$$\|\theta^* - \theta_H^*\|^2 \le \frac{4\delta}{\lambda + \lambda_H}, \tag{26}$$

where λ and λ_H are the minimum eigenvalues of the Hessians of $\mathcal{L}_T(\mathcal{D})$ and $\mathcal{L}_T(\mathcal{D}_H)$, respectively.

To find λ and λ_H, note that:

$$\nabla^2 \mathcal{L}_T(\mathcal{D}) = \nabla^2 \mathcal{L}(\mathcal{D}) + \mu \boldsymbol{I}, \quad \nabla^2 \mathcal{L}_T(\mathcal{D}_H) = \nabla^2 \mathcal{L}(\mathcal{D}_H) + \mu \boldsymbol{I}, \tag{27}$$

Since $\nabla^2 \mathcal{L}(\mathcal{D}) \succeq 0$ and $\nabla^2 \mathcal{L}(\mathcal{D}_H) \succeq 0$ due to the fact that $\mathcal{L}(.)$ is convex, we have $\lambda, \lambda_H \ge \mu$.

Substituting this back into Eq. 26 yields:

$$\|\theta^* - \theta_H^*\|^2 \le \frac{2\delta}{\mu}. \tag{28}$$

From Eq. 24, $\delta = \mathcal{O}(\frac{1}{\mathrm{IF}})$, so:

$$\|\theta^* - \theta_H^*\|^2 = \mathcal{O}\left(\frac{1}{\mathrm{IF}}\right). \tag{29}$$

According to Definition 3.1, IF $\gg 1$, consequently:

$$\|\theta^* - \theta_H^*\| = \mathcal{O}(\frac{1}{\sqrt{\mathrm{IF}}}). \tag{30}$$

Eq. 30 and Eq. 28 complete the proof. □

C.2 Proof of Lemma C.1

Proof Sketch. We evaluate the quadratic growth of f at the minimizer of g and the quadratic growth of g at the minimizer of f. Adding the two inequalities bounds $\|x_f - x_g\|^2$ by two differences between f and g, each of which is at most δ. Finally, we show that strong convexity implies the required quadratic growth.

Proof. By assumption, f and g grow at least quadratically around their minimizers x_f and x_g, respectively:

$$f(y) \geq f(x_f) + \frac{\lambda_f}{2}\|y - x_f\|^2, \qquad \forall y \in \mathbb{R}^d, \tag{31}$$

$$g(y) \geq g(x_g) + \frac{\lambda_g}{2}\|y - x_g\|^2, \qquad \forall y \in \mathbb{R}^d. \tag{32}$$

Evaluating Eq. 31 at $y = x_g$ and Eq. 32 at $y = x_f$, we obtain:

$$f(x_g) - f(x_f) \geq \frac{\lambda_f}{2}\|x_g - x_f\|^2, \tag{33}$$

$$g(x_f) - g(x_g) \geq \frac{\lambda_g}{2}\|x_f - x_g\|^2. \tag{34}$$

Since $\|x_g - x_f\| = \|x_f - x_g\|$, adding Eq. 33 and Eq. 34 together and regrouping the terms by their evaluation points yields:

$$\frac{\lambda_f + \lambda_g}{2}\|x_f - x_g\|^2 \leq \big(f(x_g) - g(x_g)\big) + \big(g(x_f) - f(x_f)\big). \tag{35}$$

Using $|f(x) - g(x)| \leq \delta$ for all x, each of the two terms on the right-hand side of Eq. 35 is bounded by δ:

$$f(x_g) - g(x_g) \leq \delta, \qquad g(x_f) - f(x_f) \leq \delta. \tag{36}$$

Substituting Eq. 36 into Eq. 35, we get:

$$\frac{\lambda_f + \lambda_g}{2}\|x_f - x_g\|^2 \leq 2\delta. \tag{37}$$

Since $\lambda_f + \lambda_g > 0$, Eq. 37 can be rewritten as:

$$\|x_f - x_g\|^2 \leq \frac{4\delta}{\lambda_f + \lambda_g}. \tag{38}$$

It remains to show that the quadratic growth in Eq. 31 holds whenever f is λ-strongly convex (the argument for g is identical). By the definition of λ-strong convexity, for all $x, y \in \mathbb{R}^d$ and $t \in [0, 1]$:

$$f\big((1-t)x + ty\big) \leq (1-t)f(x) + tf(y) - t(1-t)\frac{\lambda}{2}\|y - x\|^2. \tag{39}$$

Setting $x = x_f$ in Eq. 39 and using the optimality of x_f, i.e. $f(x_f) \leq f\big((1-t)x_f + ty\big)$, we have for all $t \in (0, 1)$:

$$f(x_f) \leq (1-t)f(x_f) + tf(y) - t(1-t)\frac{\lambda}{2}\|y - x_f\|^2. \tag{40}$$

Subtracting $(1-t)f(x_f)$ from both sides of Eq. 40 and dividing by $t > 0$ yields:

$$f(x_f) + (1-t)\frac{\lambda}{2}\|y - x_f\|^2 \leq f(y). \tag{41}$$

Letting $t \to 0^+$ in Eq. 41, we obtain:

$$f(x_f) + \frac{\lambda}{2}\|y - x_f\|^2 \leq f(y), \tag{42}$$

which is Eq. 31 with $\lambda_f = \lambda$. This completes the proof. □

C.3 Proof of Theorem 3.5

Theorem (Theorem 3.5 (restated))**.** Given Assumption 3.4, there exists a bound for the distance between the weights of the model after training (θ^*) and one of the minima of the loss landscape when training on the head solely, scaling inversely with the square root of the imbalance factor:

$$\text{dist}(\theta^*, S_H) = \mathcal{O}(\frac{1}{\sqrt{\text{IF}}}). \tag{43}$$

Proof Sketch. The proof first establishes a Hölder–1/2 error bound based on the loss's KL property and piecewise strong convexity (Assumption 3.4), which links the parameter distance $\text{dist}(\theta^*, S_H)$ to the square root of the Head-set loss difference. To bound this loss difference in terms of IF, we decompose it by adding and subtracting the total loss $\mathcal{L}$ at the optimal points θ^* and θ^*_H. Using the optimality of θ^* for $\mathcal{L}$, we simplify the inequality by removing a non-positive term. Finally, we apply the bound from Eq. 23, which limits the deviation between $\mathcal{L}$ and $\mathcal{L}_H$ to $\mathcal{O}(1/\text{IF})$, to the remaining terms. This shows the loss difference $\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H)$ scales as $\mathcal{O}(1/\text{IF})$, and substituting this back into the Hölder bound (Eq. 44) gives the final parameter distance scaling of $\mathcal{O}(1/\sqrt{\text{IF}})$.

Proof. Following assumption 3.4, $\mathcal{L}$ has the KL property with an exponent of 1/2. Since the loss is regularized with L^2 regularization, the total loss function is proved to be piecewise strongly convex [31]. Hence, a Hölder–1/2 error bound holds for the loss function $\mathcal{L}_H$:

$$\text{dist}(\theta^*, S_H) \leq C\sqrt{\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H)} \qquad \text{for all } \theta^*_H \in S_H, \tag{44}$$

where C is a constant. Let θ^*_H be any element in S_H. Since θ^* is the minimizer of $\mathcal{L}$, by optimality we have:

$$\mathcal{L}(\theta^*) \leq \mathcal{L}(\theta^*_H) \tag{45}$$

We seek to bound the term $\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H)$. We can add and subtract terms to decompose it:

$$\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H) = (\mathcal{L}_H(\theta^*) - \mathcal{L}(\theta^*)) + (\mathcal{L}(\theta^*) - \mathcal{L}(\theta^*_H)) + (\mathcal{L}(\theta^*_H) - \mathcal{L}_H(\theta^*_H)) \tag{46}$$

From the optimality of θ^*, we know the middle term is non-positive: $(\mathcal{L}(\theta^*) - \mathcal{L}(\theta^*_H)) \leq 0$. We can thus bound the expression by dropping this term:

$$\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H) \leq (\mathcal{L}_H(\theta^*) - \mathcal{L}(\theta^*)) + (\mathcal{L}(\theta^*_H) - \mathcal{L}_H(\theta^*_H)) \tag{47}$$

We now apply the bound from Eq. 23 ($\max_\theta |\mathcal{L}(\mathcal{D}) - \mathcal{L}(\mathcal{D}_H)| = M\frac{k_T}{k_H\,\text{IF}+k_T}$) to both terms on the right-hand side:

$$(\mathcal{L}_H(\theta^*) - \mathcal{L}(\theta^*)) \leq |\mathcal{L}_H(\theta^*) - \mathcal{L}(\theta^*)| \leq M\frac{k_T}{k_H\,\text{IF}+k_T} \tag{48}$$

$$(\mathcal{L}(\theta^*_H) - \mathcal{L}_H(\theta^*_H)) \leq |\mathcal{L}(\theta^*_H) - \mathcal{L}_H(\theta^*_H)| \leq M\frac{k_T}{k_H\,\text{IF}+k_T} \tag{49}$$

Substituting these bounds back gives:

$$\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H) \leq M\frac{k_T}{k_H\,\text{IF}+k_T} + M\frac{k_T}{k_H\,\text{IF}+k_T} = 2M\frac{k_T}{k_H\,\text{IF}+k_T} \tag{50}$$

Replacing the result from Eq. 50 into the error bound Eq. 44 yields:

$$\text{dist}(\theta^*, S_H) \leq C\sqrt{\mathcal{L}_H(\theta^*) - \mathcal{L}_H(\theta^*_H)} \leq C\sqrt{2M\frac{k_T}{k_H\,\text{IF}+k_T}} \tag{51}$$

Consequently:

$$\text{dist}(\theta^*, S_H) = \mathcal{O}\left(\frac{1}{\sqrt{\text{IF}}}\right), \tag{52}$$

which completes the proof. □

C.4 Proof of Theorem 3.7

Theorem (Theorem 3.7 (restated))**.** Following Assumption 3.6, let $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}^i$ be divided into $n \geq 2$ disjoint partitions, ordered such that $|\mathcal{D}^1| \geq \cdots \geq |\mathcal{D}^n|$. Taking $\mathcal{D}^1$ as the Head subset $\mathcal{D}_H$, define the aggregated Tail as $\mathcal{D}_T = \bigcup_{i=2}^{n} \mathcal{D}^i$

and the Head minimizer set as $S_{\mathcal{D}^1} = \arg\min_\theta F_{\mathcal{D}^1}(\theta)$, where $F_A(\theta) = \mathcal{L}(A, \theta) + \mathcal{R}(\theta)$ denotes the training objective with a shared regularizer. Let θ^* minimize $F_{\mathcal{D}}$. Under the local error-bound and bounded loss-contrast conditions of Theorem 3.5, applied to the Head and aggregated Tail, and assuming θ^* lies within the corresponding neighborhood, the following bound holds for fixed n:

$$\operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) \leq C\sqrt{2M\frac{|\mathcal{D}_T|}{|\mathcal{D}|}} \leq C\sqrt{2M(n-1)\frac{|\mathcal{D}^2|}{|\mathcal{D}^1|}} = \mathcal{O}_n\left(\frac{1}{\sqrt{\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}}}\right), \tag{53}$$

where $\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2} = |\mathcal{D}^1|/|\mathcal{D}^2|$, and $C, M > 0$ are constants defined by the local error bound and loss-contrast condition, respectively.

When the Head objective has a unique minimizer $\theta^*_{\mathcal{D}^1}$, the set-valued distance reduces to $\|\theta^* - \theta^*_{\mathcal{D}^1}\|$.

Proof Sketch. We extend the perturbation argument of Theorem 3.5 to multiple partitions by treating the largest partition $\mathcal{D}^1$ as the Head and aggregating all remaining partitions into a single Tail set $\mathcal{D}_T = \bigcup_{i=2}^{n} \mathcal{D}^i$. Using the optimality of the full-dataset solution and a bounded Head–Tail loss contrast, we show that the increase in Head loss is proportional to the relative size of the aggregated Tail. The local Hölder-$1/2$ error bound then converts this loss difference into a bound on the distance to the Head minimizer set. Finally, the cardinality ordering of the partitions yields the inverse-square-root scaling with respect to the imbalance factor between the two largest partitions.

Proof of Theorem 3.7. Let the long-tailed dataset be divided into n disjoint partitions ordered by cardinality:

$$\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}^i, \qquad |\mathcal{D}^1| \geq |\mathcal{D}^2| \geq \cdots \geq |\mathcal{D}^n|. \tag{54}$$

We treat $\mathcal{D}^1$ as the Head partition and aggregate the remaining partitions into the Tail, $\mathcal{D}_T = \bigcup_{i=2}^{n} \mathcal{D}^i$. The relative Tail mass is defined as:

$$\varepsilon = \frac{|\mathcal{D}_T|}{|\mathcal{D}|} = \frac{\sum_{i=2}^{n} |\mathcal{D}^i|}{\sum_{i=1}^{n} |\mathcal{D}^i|}. \tag{55}$$

For any nonempty subset $A \subseteq \mathcal{D}$, let $F_A(\theta) = \mathcal{L}(A, \theta) + \mathcal{R}(\theta)$ denote the training objective, where $\mathcal{L}(A, \theta)$ is the average empirical loss and $\mathcal{R}(\theta)$ is the shared regularization term. Since the partitions are disjoint, the full-dataset objective can be decomposed as:

$$F_{\mathcal{D}}(\theta) = (1 - \varepsilon)F_{\mathcal{D}^1}(\theta) + \varepsilon F_{\mathcal{D}_T}(\theta). \tag{56}$$

Equivalently,

$$F_{\mathcal{D}}(\theta) = F_{\mathcal{D}^1}(\theta) + \varepsilon\Delta(\theta), \tag{57}$$

where $\Delta(\theta) = F_{\mathcal{D}_T}(\theta) - F_{\mathcal{D}^1}(\theta)$ represents the Head–Tail loss contrast.

Define the Head minimizer set as $S_{\mathcal{D}^1} = \arg\min_\theta F_{\mathcal{D}^1}(\theta)$ and its minimum loss as $F^*_{\mathcal{D}^1} = \min_\theta F_{\mathcal{D}^1}(\theta)$. Following the local geometric conditions employed in Theorem 3.5, assume that there exists a neighborhood $\mathcal{U}$ of $S_{\mathcal{D}^1}$ containing the full-dataset minimizer θ^*, on which the Hölder-$1/2$ error bound holds:

$$\operatorname{dist}(\theta, S_{\mathcal{D}^1}) \leq C\sqrt{F_{\mathcal{D}^1}(\theta) - F^*_{\mathcal{D}^1}}, \qquad \forall\theta \in \mathcal{U}, \tag{58}$$

for some constant $C > 0$. We further assume that the Head–Tail loss contrast is uniformly bounded on this neighborhood:

$$\sup_{\theta\in\mathcal{U}} |\Delta(\theta)| \leq M, \tag{59}$$

where $M < \infty$.

Let $\bar{\theta} \in S_{\mathcal{D}^1}$ be a Head minimizer. Since θ^* minimizes $F_{\mathcal{D}}$, its optimality implies:

$$F_{\mathcal{D}}(\theta^*) \leq F_{\mathcal{D}}(\bar{\theta}). \tag{60}$$

Substituting Eq. 57 into Eq. 60, we obtain:

$$F_{\mathcal{D}^1}(\theta^*) + \varepsilon\Delta(\theta^*) \le F_{\mathcal{D}^1}(\bar{\theta}) + \varepsilon\Delta(\bar{\theta}). \tag{61}$$

Rearranging and using $F_{\mathcal{D}^1}(\bar{\theta}) = F^*_{\mathcal{D}^1}$, we have:

$$\begin{aligned} F_{\mathcal{D}^1}(\theta^*) - F^*_{\mathcal{D}^1} &\le \varepsilon\left[\Delta(\bar{\theta}) - \Delta(\theta^*)\right] \\ &\le 2M\varepsilon, \end{aligned} \tag{62}$$

where the last inequality follows from Eq. 59. Applying the local error bound in Eq. 58 at θ^* gives:

$$\begin{aligned} \operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) &\le C\sqrt{F_{\mathcal{D}^1}(\theta^*) - F^*_{\mathcal{D}^1}} \\ &\le C\sqrt{2M\varepsilon}. \end{aligned} \tag{63}$$

We now bound the relative Tail mass using the cardinality ordering of the partitions. Since $|\mathcal{D}^i| \le |\mathcal{D}^2|$ for all $i \ge 2$, we have:

$$|\mathcal{D}_T| = \sum_{i=2}^{n} |\mathcal{D}^i| \le (n-1)|\mathcal{D}^2|. \tag{64}$$

Consequently,

$$\varepsilon = \frac{|\mathcal{D}_T|}{|\mathcal{D}|} \le \frac{(n-1)|\mathcal{D}^2|}{|\mathcal{D}^1|}. \tag{65}$$

Substituting Eq. 65 into Eq. 63, we obtain:

$$\operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) \le C\sqrt{2M(n-1)\frac{|\mathcal{D}^2|}{|\mathcal{D}^1|}}. \tag{66}$$

Defining the imbalance factor between the two largest partitions as $\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2} = |\mathcal{D}^1|/|\mathcal{D}^2|$, the final bound becomes:

$$\begin{aligned} \operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) &\le C\sqrt{\frac{2M(n-1)}{\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}}} \\ &= \mathcal{O}_n\left(\frac{1}{\sqrt{\mathrm{IF}_{\mathcal{D}^1,\mathcal{D}^2}}}\right), \end{aligned} \tag{67}$$

for fixed n and constants C, M independent of the imbalance factor.

Finally, when the Head objective admits a unique minimizer $\theta^*_{\mathcal{D}^1}$, the minimizer set reduces to a singleton, and therefore:

$$\operatorname{dist}(\theta^*, S_{\mathcal{D}^1}) = \|\theta^* - \theta^*_{\mathcal{D}^1}\|. \tag{68}$$

Thus, the result extends the Head-dominance analysis to multiple partitions without requiring uniqueness of the Head minimizer, while preserving the inverse-square-root scaling with respect to the partition imbalance factor. □

C.5 Exponential-Decay Extension of Theorem 3.7

We now give the full statement and proof of Remark 3.8.

Remark (Remark 3.8 (Full version))**.** The partition-based statement in Theorem 3.7 also extends to the standard exponential long-tail model used in CIFAR10-LT and CIFAR100-LT. Let sorted class frequencies follow $w(u) = w_0\,\mathrm{IF}^{-u}$ for $u \in [0,1]$, where $\mathrm{IF} = w(0)/w(1)$. For a Head prefix $P_\tau = [0,\tau]$ and Tail suffix $T_\tau = (\tau,1]$, the Tail mass fraction is $\varepsilon_\tau = \frac{\mu(T_\tau)}{\mu([0,1])} = \frac{\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1}}{1-\mathrm{IF}^{-1}}$. Under the same local error-bound/KL-$1/2$ condition used in Theorem 3.5, the full-data optimum satisfies

$$\operatorname{dist}(\theta^\star_{\mathcal{D}}, S_{P_\tau}) \le C_\tau\sqrt{2M_\tau\frac{\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1}}{1-\mathrm{IF}^{-1}}} = \mathcal{O}(\mathrm{IF}^{-\tau/2}). \tag{69}$$

Equivalently, defining the boundary imbalance factor $\mathrm{IF}_\tau = w(0)/w(\tau) = \mathrm{IF}^\tau$, we recover the same inverse-square-root form

$$\mathrm{dist}(\theta^\star_{\mathcal{D}}, S_{P_\tau}) = \mathcal{O}(\mathrm{IF}_\tau^{-1/2}). \tag{70}$$

Thus, the geometric conclusion of Theorem 3.7 is not an artifact of uniform within-partition cardinalities; it also holds for exponentially decaying long-tailed datasets.

Proof. Let $\mathcal{U} = [0, 1]$ denote the full sorted class-index domain. This continuous domain is the measure-theoretic analogue of the full dataset $\mathcal{D}$; hence $L_{\mathcal{U}}$ and $\theta^\star_{\mathcal{U}}$ correspond to the full-data loss and full-data optimum in the exponential-decay model. Recall that the Head prefix and Tail suffix are defined as $P_\tau = [0, \tau]$ and $T_\tau = (\tau, 1]$, so that $\mathcal{U} = P_\tau \cup T_\tau$ and $P_\tau \cap T_\tau = \emptyset$.

We begin by writing the training measure induced by the exponential class-frequency profile. Since $w(u) = w_0 \,\mathrm{IF}^{-u}$, define

$$d\mu(u) = w(u)du = w_0 \,\mathrm{IF}^{-u}\, du. \tag{71}$$

For any measurable set $A \subseteq [0, 1]$, define the normalized risk over A as

$$L_A(\theta) = \frac{1}{\mu(A)} \int_A \ell_u(\theta) d\mu(u) + R(\theta), \tag{72}$$

where $R(\theta)$ is the same regularization term for all subsets.

The first step is to decompose the full-data risk into a weighted combination of the Head-prefix risk and the Tail-suffix risk. By definition,

$$L_{\mathcal{U}}(\theta) = \frac{1}{\mu(\mathcal{U})} \int_{\mathcal{U}} \ell_u(\theta) d\mu(u) + R(\theta). \tag{73}$$

Since $\mathcal{U} = P_\tau \cup T_\tau$ and the two sets are disjoint, we have

$$\int_{\mathcal{U}} \ell_u(\theta) d\mu(u) = \int_{P_\tau} \ell_u(\theta) d\mu(u) + \int_{T_\tau} \ell_u(\theta) d\mu(u). \tag{74}$$

Therefore,

$$L_{\mathcal{U}}(\theta) = \frac{1}{\mu(\mathcal{U})} \left(\int_{P_\tau} \ell_u(\theta) d\mu(u) + \int_{T_\tau} \ell_u(\theta) d\mu(u) \right) + R(\theta). \tag{75}$$

Using the definitions of L_{P_τ} and L_{T_τ}, we can rewrite the two integrals as

$$\int_{P_\tau} \ell_u(\theta) d\mu(u) = \mu(P_\tau) \left(L_{P_\tau}(\theta) - R(\theta) \right), \tag{76}$$

and

$$\int_{T_\tau} \ell_u(\theta) d\mu(u) = \mu(T_\tau) \left(L_{T_\tau}(\theta) - R(\theta) \right). \tag{77}$$

Substituting these identities into the expression for $L_{\mathcal{U}}(\theta)$ gives

$$L_{\mathcal{U}}(\theta) = \frac{\mu(P_\tau)}{\mu(\mathcal{U})} \left(L_{P_\tau}(\theta) - R(\theta) \right) + \frac{\mu(T_\tau)}{\mu(\mathcal{U})} \left(L_{T_\tau}(\theta) - R(\theta) \right) + R(\theta). \tag{78}$$

Since $\mu(\mathcal{U}) = \mu(P_\tau) + \mu(T_\tau)$, the regularization terms combine as

$$-\frac{\mu(P_\tau)}{\mu(\mathcal{U})} R(\theta) - \frac{\mu(T_\tau)}{\mu(\mathcal{U})} R(\theta) + R(\theta) = 0. \tag{79}$$

Thus,

$$L_{\mathcal{U}}(\theta) = \frac{\mu(P_\tau)}{\mu(\mathcal{U})} L_{P_\tau}(\theta) + \frac{\mu(T_\tau)}{\mu(\mathcal{U})} L_{T_\tau}(\theta). \tag{80}$$

Define the Tail mass fraction as

$$\varepsilon_\tau = \frac{\mu(T_\tau)}{\mu(\mathcal{U})}. \tag{81}$$

Then

$$\frac{\mu(P_\tau)}{\mu(\mathcal{U})} = 1 - \varepsilon_\tau, \tag{82}$$

and hence

$$L_{\mathcal{U}}(\theta) = (1 - \varepsilon_\tau) L_{P_\tau}(\theta) + \varepsilon_\tau L_{T_\tau}(\theta). \tag{83}$$

We now bound how far the full-data optimum can be from the Head-prefix minimizer set. Let

$$S_{P_\tau} = \arg\min_\theta L_{P_\tau}(\theta), \qquad L^\star_{P_\tau} = \min_\theta L_{P_\tau}(\theta), \tag{84}$$

and let

$$\theta^\star_{\mathcal{U}} \in \arg\min_\theta L_{\mathcal{U}}(\theta). \tag{85}$$

Choose any $\bar{\theta}_{P_\tau} \in S_{P_\tau}$. Then

$$L_{P_\tau}(\bar{\theta}_{P_\tau}) = L^\star_{P_\tau}. \tag{86}$$

We want to bound the Head-prefix excess risk

$$L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{P_\tau}(\bar{\theta}_{P_\tau}). \tag{87}$$

Add and subtract $L_{\mathcal{U}}(\theta^\star_{\mathcal{U}})$ and $L_{\mathcal{U}}(\bar{\theta}_{P_\tau})$:

$$\begin{aligned} L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{P_\tau}(\bar{\theta}_{P_\tau}) = {} & [L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{\mathcal{U}}(\theta^\star_{\mathcal{U}})] \\ & + \left[L_{\mathcal{U}}(\theta^\star_{\mathcal{U}}) - L_{\mathcal{U}}(\bar{\theta}_{P_\tau})\right] \\ & + \left[L_{\mathcal{U}}(\bar{\theta}_{P_\tau}) - L_{P_\tau}(\bar{\theta}_{P_\tau})\right]. \end{aligned} \tag{88}$$

Since $\theta^\star_{\mathcal{U}}$ minimizes $L_{\mathcal{U}}$, we have

$$L_{\mathcal{U}}(\theta^\star_{\mathcal{U}}) - L_{\mathcal{U}}(\bar{\theta}_{P_\tau}) \le 0. \tag{89}$$

Therefore,

$$\begin{aligned} L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{P_\tau}(\bar{\theta}_{P_\tau}) \le {} & [L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{\mathcal{U}}(\theta^\star_{\mathcal{U}})] \\ & + \left[L_{\mathcal{U}}(\bar{\theta}_{P_\tau}) - L_{P_\tau}(\bar{\theta}_{P_\tau})\right]. \end{aligned} \tag{90}$$

Next, using the decomposition of $L_{\mathcal{U}}$, for any θ we have

$$\begin{aligned} L_{\mathcal{U}}(\theta) - L_{P_\tau}(\theta) &= (1 - \varepsilon_\tau) L_{P_\tau}(\theta) + \varepsilon_\tau L_{T_\tau}(\theta) - L_{P_\tau}(\theta) \\ &= \varepsilon_\tau \left(L_{T_\tau}(\theta) - L_{P_\tau}(\theta)\right). \end{aligned} \tag{91}$$

Equivalently,

$$L_{P_\tau}(\theta) - L_{\mathcal{U}}(\theta) = \varepsilon_\tau \left(L_{P_\tau}(\theta) - L_{T_\tau}(\theta)\right). \tag{92}$$

By the local bounded loss-contrast condition on the relevant sublevel set $\mathcal{S}$, there exists $M_\tau < \infty$ such that

$$\sup_{\theta \in \mathcal{S}} |L_{T_\tau}(\theta) - L_{P_\tau}(\theta)| \le M_\tau. \tag{93}$$

Assuming $\theta^\star_{\mathcal{U}}, \bar{\theta}_{P_\tau} \in \mathcal{S}$, this implies

$$|L_{\mathcal{U}}(\theta) - L_{P_\tau}(\theta)| \le \varepsilon_\tau M_\tau \tag{94}$$

for both $\theta = \theta^\star_{\mathcal{U}}$ and $\theta = \bar{\theta}_{P_\tau}$. Hence,

$$L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{\mathcal{U}}(\theta^\star_{\mathcal{U}}) \le \varepsilon_\tau M_\tau, \tag{95}$$

and

$$L_{\mathcal{U}}(\bar{\theta}_{P_\tau}) - L_{P_\tau}(\bar{\theta}_{P_\tau}) \le \varepsilon_\tau M_\tau. \tag{96}$$

Substituting these two bounds into the previous inequality gives

$$L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L_{P_\tau}(\bar{\theta}_{P_\tau}) \le 2\varepsilon_\tau M_\tau. \tag{97}$$

Since $\bar{\theta}_{P_\tau} \in S_{P_\tau}$, we have

$$L_{P_\tau}(\bar{\theta}_{P_\tau}) = L^\star_{P_\tau}. \tag{98}$$

Therefore,

$$L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L^\star_{P_\tau} \le 2\varepsilon_\tau M_\tau. \tag{99}$$

We now invoke the same local error-bound/KL-$1/2$ condition used in Theorem 3.5. Namely, there exists a constant $C_\tau > 0$ such that

$$\operatorname{dist}(\theta, S_{P_\tau}) \le C_\tau \sqrt{L_{P_\tau}(\theta) - L^\star_{P_\tau}} \tag{100}$$

for all θ in the relevant local sublevel set. Applying this inequality at $\theta = \theta^\star_{\mathcal{U}}$ yields

$$\operatorname{dist}(\theta^\star_{\mathcal{U}}, S_{P_\tau}) \le C_\tau \sqrt{L_{P_\tau}(\theta^\star_{\mathcal{U}}) - L^\star_{P_\tau}}. \tag{101}$$

Using the excess-risk bound above, we obtain

$$\operatorname{dist}(\theta^\star_{\mathcal{U}}, S_{P_\tau}) \le C_\tau \sqrt{2\varepsilon_\tau M_\tau}. \tag{102}$$

It remains to compute ε_τ under the exponential decay measure. For any interval $[a, b] \subseteq [0, 1]$,

$$\mu([a, b]) = \int_a^b w_0 \, \mathrm{IF}^{-u} \, du. \tag{103}$$

Since $\mathrm{IF}^{-u} = e^{-u \log \mathrm{IF}}$, we have

$$\begin{aligned} \mu([a, b]) &= w_0 \int_a^b e^{-u \log \mathrm{IF}} du \\ &= w_0 \left[-\frac{1}{\log \mathrm{IF}} e^{-u \log \mathrm{IF}} \right]_a^b \\ &= \frac{w_0}{\log \mathrm{IF}} \left(\mathrm{IF}^{-a} - \mathrm{IF}^{-b} \right). \end{aligned} \tag{104}$$

Applying this formula to the full domain $\mathcal{U} = [0, 1]$ gives

$$\mu(\mathcal{U}) = \mu([0, 1]) = \frac{w_0}{\log \mathrm{IF}} \left(1 - \mathrm{IF}^{-1} \right). \tag{105}$$

Applying the same formula to the Tail suffix $T_\tau = (\tau, 1]$ gives

$$\mu(T_\tau) = \frac{w_0}{\log \mathrm{IF}} \left(\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1} \right). \tag{106}$$

Therefore,

$$\varepsilon_\tau = \frac{\mu(T_\tau)}{\mu(\mathcal{U})} = \frac{\frac{w_0}{\log \mathrm{IF}} \left(\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1} \right)}{\frac{w_0}{\log \mathrm{IF}} \left(1 - \mathrm{IF}^{-1} \right)}. \tag{107}$$

Canceling the common factor $\frac{w_0}{\log \mathrm{IF}}$ gives

$$\varepsilon_\tau = \frac{\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1}}{1 - \mathrm{IF}^{-1}}. \tag{108}$$

Substituting this expression for ε_τ into the distance bound yields

$$\operatorname{dist}(\theta^\star_{\mathcal{U}}, S_{P_\tau}) \le C_\tau \sqrt{2M_\tau \frac{\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1}}{1 - \mathrm{IF}^{-1}}}. \tag{109}$$

This proves the first claimed bound.

We now derive the asymptotic form. For a fixed $\tau \in (0, 1)$ and large IF, we have

$$1 - \mathrm{IF}^{-1} \to 1. \tag{110}$$

Moreover, since $\tau < 1$, the term IF^{-1} decays faster than $\mathrm{IF}^{-\tau}$, so

$$\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1} = \mathcal{O}(\mathrm{IF}^{-\tau}). \tag{111}$$

Hence,

$$\frac{\mathrm{IF}^{-\tau} - \mathrm{IF}^{-1}}{1 - \mathrm{IF}^{-1}} = \mathcal{O}(\mathrm{IF}^{-\tau}). \tag{112}$$

Substituting this into the previous bound gives

$$\operatorname{dist}(\theta^\star_{\mathcal{U}}, S_{P_\tau}) = \mathcal{O}\left(\sqrt{\mathrm{IF}^{-\tau}}\right) = \mathcal{O}(\mathrm{IF}^{-\tau/2}). \tag{113}$$

Finally, define the boundary imbalance factor as

$$\mathrm{IF}_\tau = \frac{w(0)}{w(\tau)}. \tag{114}$$

Since $w(0) = w_0$ and $w(\tau) = w_0\,\mathrm{IF}^{-\tau}$, we get

$$\mathrm{IF}_\tau = \frac{w_0}{w_0\,\mathrm{IF}^{-\tau}} = \mathrm{IF}^{\tau}\,. \tag{115}$$

Therefore,

$$\mathrm{IF}^{-\tau/2} = (\mathrm{IF}^{\tau})^{-1/2} = \mathrm{IF}_\tau^{-1/2}\,. \tag{116}$$

Thus,

$$\operatorname{dist}(\theta^\star_{\mathcal{U}}, S_{P_\tau}) = \mathcal{O}(\mathrm{IF}_\tau^{-1/2}). \tag{117}$$

This shows that the same inverse-square-root geometry from Theorem 3.7 also holds under the exponential long-tail model. Finally, the same perturbation argument can be applied sequentially to multiple cardinality-sorted exponential partitions; summing the resulting prefix-to-prefix bounds via the triangle inequality yields a telescoping extension analogous to Theorem 3.7. □

C.6 Proof of Proposition 3.10

We now provide the full proof of the partitioning bound. The proof has three steps. First, we bound the mismatch between the imbalanced training risk and the balanced risk inside each partition. Second, we bound the forgetting introduced by one continual-learning transition. Third, we combine these terms and derive the equal-local-imbalance rule for exponentially decaying class counts.

Proof. Let the classes be sorted by decreasing cardinality. For a partition $\mathcal{C}_i$, define the corresponding dataset partition as

$$\mathcal{D}^i = \{(x, y) \in \mathcal{D} : y \in \mathcal{C}_i\}. \tag{118}$$

Let $n_c = |\mathcal{D}_c|$ denote the number of samples in class c. The local imbalance factor inside partition i is

$$R_i = \frac{\max_{c\in\mathcal{C}_i} n_c}{\min_{c\in\mathcal{C}_i} n_c}. \tag{119}$$

We first compare the imbalanced training risk and the balanced risk inside a fixed partition. Define the empirical class distribution inside partition i by

$$p_c = \frac{n_c}{\sum_{d\in\mathcal{C}_i} n_d}, \qquad c \in \mathcal{C}_i. \tag{120}$$

Define the balanced class distribution inside the same partition by

$$q_c = \frac{1}{|\mathcal{C}_i|}, \qquad c \in \mathcal{C}_i. \tag{121}$$

The task loss used for training on partition i can be written as

$$\mathcal{L}_i^{\mathrm{task}}(\theta) = \sum_{c\in\mathcal{C}_i} p_c \mathcal{L}_c(\theta), \tag{122}$$

where $\mathcal{L}_c(\theta)$ denotes the class-conditional loss. The balanced risk inside partition i is

$$\mathcal{L}_i^{\mathrm{bal}}(\theta) = \sum_{c\in\mathcal{C}_i} q_c \mathcal{L}_c(\theta). \tag{123}$$

Therefore,

$$\mathcal{L}_i^{\mathrm{task}}(\theta) - \mathcal{L}_i^{\mathrm{bal}}(\theta) = \sum_{c\in\mathcal{C}_i} (p_c - q_c)\mathcal{L}_c(\theta). \tag{124}$$

Assume that class losses are locally bounded on the optimization region, i.e.,

$$0 \le \mathcal{L}_c(\theta) \le B \tag{125}$$

for all c and all relevant θ. Then

$$\left|\mathcal{L}_i^{\mathrm{task}}(\theta) - \mathcal{L}_i^{\mathrm{bal}}(\theta)\right| \le B \sum_{c \in \mathcal{C}_i} |p_c - q_c|. \tag{126}$$

Using the convention

$$\mathrm{TV}(p, q) = \frac{1}{2} \sum_{c \in \mathcal{C}_i} |p_c - q_c|, \tag{127}$$

we obtain

$$\left|\mathcal{L}_i^{\mathrm{task}}(\theta) - \mathcal{L}_i^{\mathrm{bal}}(\theta)\right| \le 2B\,\mathrm{TV}(p, q). \tag{128}$$

Equivalently, if total variation is defined without the factor $1/2$, the same bound is written with $B\,\mathrm{TV}(p, q)$. In the statement of the proposition, we use the normalized form. We now bound this total variation in terms of the local imbalance factor R_i. Let

$$r_c = \frac{p_c}{q_c} = |\mathcal{C}_i| p_c. \tag{129}$$

Since p_c is proportional to n_c and q_c is uniform, we have

$$\frac{\max_{c \in \mathcal{C}_i} r_c}{\min_{c \in \mathcal{C}_i} r_c} = \frac{\max_{c \in \mathcal{C}_i} n_c}{\min_{c \in \mathcal{C}_i} n_c} = R_i. \tag{130}$$

Moreover,

$$\sum_{c \in \mathcal{C}_i} q_c r_c = \sum_{c \in \mathcal{C}_i} p_c = 1. \tag{131}$$

For distributions whose likelihood ratio $r_c = p_c/q_c$ has range at most R_i, the total variation satisfies the standard bound

$$\mathrm{TV}(p, q) \le \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1}. \tag{132}$$

For completeness, we recall the short argument. Since $\mathbb{E}_q[r_c] = 1$ and $\max r_c / \min r_c \le R_i$, the largest possible absolute deviation from 1 is attained by a two-level likelihood ratio, with one level below 1 and one level above 1. Balancing the two sides around 1 gives lower and upper levels proportional to $1/\sqrt{R_i}$ and $\sqrt{R_i}$, which yields

$$\mathrm{TV}(p, q) \le \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1}. \tag{133}$$

Thus, for all relevant θ,

$$\left|\mathcal{L}_i^{\mathrm{task}}(\theta) - \mathcal{L}_i^{\mathrm{bal}}(\theta)\right| \le 2B \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1}. \tag{134}$$

Absorbing the factor convention for total variation into B, we write the partition bias term as

$$B \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1}. \tag{135}$$

Let

$$\theta_i^\star \in \arg\min_\theta \mathcal{L}_i^{\mathrm{task}}(\theta), \qquad \theta_i^{\mathrm{bal},\star} \in \arg\min_\theta \mathcal{L}_i^{\mathrm{bal}}(\theta). \tag{136}$$

Then

$$\begin{aligned} \mathcal{L}_i^{\mathrm{bal}}(\theta_i^\star) &\le \mathcal{L}_i^{\mathrm{task}}(\theta_i^\star) + B \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1} \\ &\le \mathcal{L}_i^{\mathrm{task}}(\theta_i^{\mathrm{bal},\star}) + B \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1} \\ &\le \mathcal{L}_i^{\mathrm{bal}}(\theta_i^{\mathrm{bal},\star}) + 2B \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1}. \end{aligned} \tag{137}$$

The first and third inequalities use the risk-mismatch bound, and the second inequality uses the optimality of $\theta_i^\star$ for $\mathcal{L}_i^{\text{task}}$. Therefore, the balanced-risk penalty caused by training on an imbalanced partition is bounded by

$$2B\frac{\sqrt{R_i}-1}{\sqrt{R_i}+1}. \tag{138}$$

Taking the worst partition gives the within-partition LTR bias term

$$2B\max_{i\le N}\frac{\sqrt{R_i}-1}{\sqrt{R_i}+1}. \tag{139}$$

We now bound the forgetting introduced by one CL transition. At stage $i \ge 2$, CLTR optimizes (Note that i in this part of the proof refers to the CL training step, whereas in the previous part it denotes the class index.)

$$\mathcal{L}_i^{\text{CL}}(\theta) = \mathcal{L}_i^{\text{task}}(\theta) + \alpha_i D_{\Phi,i}(h_\theta, z_{i-1}) + \beta_i D_{\Psi,i}(\theta, \theta_{i-1}). \tag{140}$$

Assume first that the stage is optimized exactly, so that

$$\theta_i^\star \in \arg\min_\theta \mathcal{L}_i^{\text{CL}}(\theta). \tag{141}$$

Because both divergences vanish at θ_{i-1}, we have

$$D_{\Phi,i}(h_{\theta_{i-1}}, z_{i-1}) = 0, \qquad D_{\Psi,i}(\theta_{i-1}, \theta_{i-1}) = 0. \tag{142}$$

By optimality of $\theta_i^\star$,

$$\mathcal{L}_i^{\text{CL}}(\theta_i^\star) \le \mathcal{L}_i^{\text{CL}}(\theta_{i-1}) = \mathcal{L}_i^{\text{task}}(\theta_{i-1}). \tag{143}$$

Expanding the left-hand side gives

$$\mathcal{L}_i^{\text{task}}(\theta_i^\star) + \alpha_i D_{\Phi,i}(h_{\theta_i^\star}, z_{i-1}) + \beta_i D_{\Psi,i}(\theta_i^\star, \theta_{i-1}) \le \mathcal{L}_i^{\text{task}}(\theta_{i-1}). \tag{144}$$

Rearranging,

$$\alpha_i D_{\Phi,i}(h_{\theta_i^\star}, z_{i-1}) + \beta_i D_{\Psi,i}(\theta_i^\star, \theta_{i-1}) \le \mathcal{L}_i^{\text{task}}(\theta_{i-1}) - \mathcal{L}_i^{\text{task}}(\theta_i^\star). \tag{145}$$

Define the current-task learning pressure as

$$\Delta_i = \left[\mathcal{L}_i^{\text{task}}(\theta_{i-1}) - \mathcal{L}_i^{\text{task}}(\theta_i^\star)\right]_+. \tag{146}$$

Then

$$\alpha_i D_{\Phi,i}(h_{\theta_i^\star}, z_{i-1}) + \beta_i D_{\Psi,i}(\theta_i^\star, \theta_{i-1}) \le \Delta_i. \tag{147}$$

Since both divergence terms are nonnegative, we obtain

$$D_{\Phi,i}(h_{\theta_i^\star}, z_{i-1}) \le \frac{\Delta_i}{\alpha_i}, \tag{148}$$

and

$$D_{\Psi,i}(\theta_i^\star, \theta_{i-1}) \le \frac{\Delta_i}{\beta_i}. \tag{149}$$

Let $\mathcal{C}_{<i} = \bigcup_{j<i} \mathcal{C}_j$ denote the set of previously learned classes, and define the old-task balanced loss as

$$\mathcal{L}_{<i}^{\text{bal}}(\theta) = \frac{1}{|\mathcal{C}_{<i}|}\sum_{c\in\mathcal{C}_{<i}} \mathcal{L}_c(\theta). \tag{150}$$

Assume the output-route control condition

$$\mathcal{L}_{<i}^{\text{bal}}(\theta) - \mathcal{L}_{<i}^{\text{bal}}(\theta_{i-1}) \le c_{\Phi,i} D_{\Phi,i}(h_\theta, z_{i-1}) \tag{151}$$

on the relevant local region. Applying this at $\theta = \theta_i^\star$ gives

$$\mathcal{L}_{<i}^{\text{bal}}(\theta_i^\star) - \mathcal{L}_{<i}^{\text{bal}}(\theta_{i-1}) \le c_{\Phi,i} D_{\Phi,i}(h_{\theta_i^\star}, z_{i-1}) \le \Delta_i \frac{c_{\Phi,i}}{\alpha_i}. \tag{152}$$

Similarly, assume the parameter-route control condition. Specifically, suppose $\mathcal{L}_{<i}^{\mathrm{bal}}$ is locally $\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}$-smooth around θ_{i-1} and that $D_{\Psi,i}$ is locally $m_{\Psi,i}$-strongly convex:

$$D_{\Psi,i}(\theta, \theta_{i-1}) \geq \frac{m_{\Psi,i}}{2}\|\theta - \theta_{i-1}\|^2. \tag{153}$$

Then the same argument used in Theorem 3.12 gives

$$\mathcal{L}_{<i}^{\mathrm{bal}}(\theta) - \mathcal{L}_{<i}^{\mathrm{bal}}(\theta_{i-1}) \leq \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}} D_{\Psi,i}(\theta, \theta_{i-1}). \tag{154}$$

Applying this at $\theta = \theta_i^\star$ gives

$$\mathcal{L}_{<i}^{\mathrm{bal}}(\theta_i^\star) - \mathcal{L}_{<i}^{\mathrm{bal}}(\theta_{i-1}) \leq \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}} D_{\Psi,i}(\theta_i^\star, \theta_{i-1}) \leq \Delta_i \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}\beta_i}. \tag{155}$$

Either the output route or the parameter route is sufficient to control old-task loss increase. Therefore, the sharper bound is

$$\mathcal{L}_{<i}^{\mathrm{bal}}(\theta_i^\star) - \mathcal{L}_{<i}^{\mathrm{bal}}(\theta_{i-1}) \leq \Delta_i \min\left\{ \frac{c_{\Phi,i}}{\alpha_i}, \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}\beta_i} \right\}. \tag{156}$$

This is the forgetting cost of transition $i-1 \to i$. Summing over all transitions gives the cumulative forgetting bound

$$\sum_{i=2}^{N} \Delta_i \min\left\{ \frac{c_{\Phi,i}}{\alpha_i}, \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}\beta_i} \right\}. \tag{157}$$

Combining the within-partition LTR bias term and the cumulative forgetting term yields

$$\mathcal{L}_{\mathrm{bal}}(\theta_N) - \mathcal{L}_{\mathrm{bal}}^\star \leq 2B \max_{i \leq N} \frac{\sqrt{R_i} - 1}{\sqrt{R_i} + 1} + \sum_{i=2}^{N} \Delta_i \min\left\{ \frac{c_{\Phi,i}}{\alpha_i}, \frac{\mathcal{L}_{i,\mathrm{old}}^{\mathrm{sm}}}{m_{\Psi,i}\beta_i} \right\} + \mathrm{OptErr}. \tag{158}$$

Here OptErr collects optimization errors, approximation errors, and constants arising from local boundedness assumptions. This proves the first claim.

We now derive the boundary rule for exponentially decaying class counts. In the continuous approximation, let the sorted class-frequency profile be

$$w(u) = w_0 \,\mathrm{IF}^{-u}, \qquad u \in [0, 1]. \tag{159}$$

Let the partition boundaries be

$$0 = u_0 < u_1 < \cdots < u_N = 1. \tag{160}$$

The local imbalance factor in partition i is

$$R_i = \frac{w(u_{i-1})}{w(u_i)}. \tag{161}$$

Substituting $w(u) = w_0 \,\mathrm{IF}^{-u}$ gives

$$R_i = \frac{w_0 \,\mathrm{IF}^{-u_{i-1}}}{w_0 \,\mathrm{IF}^{-u_i}} = \mathrm{IF}^{u_i - u_{i-1}}. \tag{162}$$

For fixed N, the forgetting term has a fixed number of transitions. Thus, the partition-boundary-dependent part of the first term is

$$\max_{i \leq N} R_i = \max_{i \leq N} \mathrm{IF}^{u_i - u_{i-1}}. \tag{163}$$

Since IF^x is increasing in x, minimizing $\max_i R_i$ is equivalent to minimizing

$$\max_{i \leq N} (u_i - u_{i-1}) \tag{164}$$

subject to

$$\sum_{i=1}^{N} (u_i - u_{i-1}) = 1. \tag{165}$$

The minimizer is obtained by equalizing all interval lengths:

$$u_i - u_{i-1} = \frac{1}{N} \tag{166}$$

for all $i = 1, \ldots, N$. Hence

$$u_i = \frac{i}{N}. \tag{167}$$

Substituting this into the expression for R_i gives

$$R_i = \mathrm{IF}^{1/N} \tag{168}$$

for every i. Therefore,

$$R_1 = R_2 = \cdots = R_N = \mathrm{IF}^{1/N}. \tag{169}$$

This proves that, for exponentially decaying class counts and fixed N, the optimal boundaries equalize the local imbalance factors.

Finally, suppose the CL constants are approximately uniform across stages:

$$\Delta_i \approx \Delta, \qquad c_{\Phi,i} \approx c_\Phi, \qquad L^{\mathrm{sm}}_{i,\mathrm{old}} \approx L^{\mathrm{sm}}_{\mathrm{old}}, \qquad m_{\Psi,i} \approx m_\Psi, \qquad \alpha_i \approx \alpha, \qquad \beta_i \approx \beta. \tag{170}$$

Then

$$\sum_{i=2}^{N} \Delta_i \min\left\{ \frac{c_{\Phi,i}}{\alpha_i}, \frac{L^{\mathrm{sm}}_{i,\mathrm{old}}}{m_{\Psi,i}\beta_i} \right\} \approx (N-1)\Delta \min\left\{ \frac{c_\Phi}{\alpha}, \frac{L^{\mathrm{sm}}_{\mathrm{old}}}{m_\Psi \beta} \right\}. \tag{171}$$

Using the fixed-N optimal boundary rule $R_i = \mathrm{IF}^{1/N}$ and representing $\Delta \min\left\{ \frac{c_\Phi}{\alpha}, \frac{L^{\mathrm{sm}}_{\mathrm{old}}}{m_\Psi \beta} \right\}$ as effective forgetting coefficient η, the partition-selection score becomes

$$S(N) = 2B \frac{\mathrm{IF}^{1/(2N)} - 1}{\mathrm{IF}^{1/(2N)} + 1} + (N-1)\eta. \tag{172}$$

Thus,

$$N^\star \in \arg\min_N S(N). \tag{173}$$

This expression captures the bias-forgetting tradeoff: increasing N decreases the local LTR-bias term but increases the cumulative CL-forgetting term. Therefore, the practical rule is to use the largest number of partitions that the chosen CL method can handle without excessive forgetting. □

C.7 Proof of Theorem 3.12

Theorem (Theorem 3.12 (restated))**.** Suppose Assumption 3.11 holds on $\mathcal{N}$. Define the thresholds $\beta_{\min} := \frac{L^{\mathrm{sm}}_H}{2\, m_\Psi}$, $\alpha_{\min} := \frac{c_\Phi}{2}$, where L^{sm}_H is local smoothness constant of $\mathcal{L}_H$ at $\theta^\star_H$, m_Ψ is strong-convexity constant of $D_\Psi(\cdot, \theta^\star_H)$, and c_Φ is calibration constant relating $\mathcal{L}_H$ to D_Φ. If either $\beta \geq \beta_{\min}$ (Parameter route) or $\alpha \geq \alpha_{\min}$ (Output route), then for any $\theta \in \mathcal{N}$,

$$\mathcal{L}_{\mathrm{bal}}(\theta) \;\leq\; \mathcal{L}_{\mathrm{CL}}(\theta) \;+\; \tfrac{1}{2}\, \mathcal{L}_H(\theta^\star_H). \tag{174}$$

In words: if either $\beta \geq \beta_{\min}$ or $\alpha \geq \alpha_{\min}$, then the CL objective $\mathcal{L}_{\mathrm{CL}}(\theta)$ upper-bounds the balanced loss $\mathcal{L}_{\mathrm{bal}}(\theta)$ up to an additive θ-independent constant.

Proof Sketch. The proof first rewrites the balanced loss as the sum of a Tail term, a Head-loss difference term, and a constant by adding and subtracting $\mathcal{L}_H(\theta^\star_H)$. The Tail term $\frac{1}{2}\mathcal{L}_T(\theta)$ is upper-bounded by the task loss $\mathcal{L}_{\mathrm{task}}(\theta)$ and hence by $\mathcal{L}_{\mathrm{CL}}(\theta)$. For the parameter route, local smoothness of $\mathcal{L}_H$ and strong convexity of D_Ψ are combined to bound the Head-loss difference in terms of D_Ψ, which appears in $\mathcal{L}_{\mathrm{CL}}$ when $\beta \geq \beta_{\min}$. For the output route, the calibration inequality bounds the same Head-loss difference in terms of D_Φ, controlled by $\mathcal{L}_{\mathrm{CL}}$ when $\alpha \geq \alpha_{\min}$. Adding these bounds and the constant yields the dominance relation in Eq. 11.

Proof. We first prove the dominance inequality Eq. 11. Recall

$$\mathcal{L}_{\mathrm{bal}}(\theta) := \tfrac{1}{2}\mathcal{L}_T(\theta) + \tfrac{1}{2}\mathcal{L}_H(\theta), \tag{175}$$

and add/subtract $\mathcal{L}_H(\theta_H^\star)$:

$$\mathcal{L}_{\text{bal}}(\theta) = \tfrac{1}{2}\mathcal{L}_T(\theta) + \tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star). \tag{176}$$

Since $\mathcal{L}_{\text{task}} \equiv \mathcal{L}_T$ and the loss is non-negative, we have

$$\tfrac{1}{2}\mathcal{L}_T(\theta) \;\le\; \mathcal{L}_{\text{task}}(\theta), \tag{177}$$

and combining Eq. 176 and Eq. 177 gives

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{task}}(\theta) + \tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star). \tag{178}$$

We now bound the term $\frac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big)$ in the two cases of the theorem.

Parameter route. Assume $\beta \ge \beta_{\min}$ and use Eq. 7 and Eq. 8. For all $\theta \in \mathcal{N}$,

$$\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star) \le \tfrac{L_H^{\text{sm}}}{2}\,\|\theta - \theta_H^\star\|^2 \tag{179}$$

$$\le \tfrac{L_H^{\text{sm}}}{2}\,\frac{2}{m_\Psi}\,D_\Psi(\theta, \theta_H^\star) \tag{180}$$

$$\le \frac{L_H^{\text{sm}}}{m_\Psi} D_\Psi(\theta, \theta_H^\star), \tag{181}$$

where the second inequality uses $\|\theta - \theta_H^\star\|^2 \le \frac{2}{m_\Psi} D_\Psi(\theta, \theta_H^\star)$ from Eq. 8. Multiplying by $\frac{1}{2}$ yields

$$\tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) \;\le\; \frac{L_H^{\text{sm}}}{2m_\Psi}\, D_\Psi(\theta, \theta_H^\star). \tag{182}$$

Since $\beta \ge \beta_{\min} = L_H^{\text{sm}}/(2m_\Psi)$, we deduce

$$\tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) \;\le\; \beta\, D_\Psi(\theta, \theta_H^\star). \tag{183}$$

Substituting Eq. 183 into Eq. 178 gives

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{task}}(\theta) + \beta\, D_\Psi(\theta, \theta_H^\star) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star). \tag{184}$$

By definition, $\mathcal{L}_{\text{CL}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \alpha D_\Phi\big(h_\theta(\cdot), z\big) + \beta D_\Psi(\theta, \theta_H^\star) \;\ge\; \mathcal{L}_{\text{task}}(\theta) + \beta D_\Psi(\theta, \theta_H^\star)$, since $\alpha D_\Phi \ge 0$. Combining this with Eq. 184 yields

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{CL}}(\theta) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star), \tag{185}$$

, which is Eq. 11 in the parameter route.

Output route. Assume $\alpha \ge \alpha_{\min}$ and use Eq. 9. For all $\theta \in \mathcal{N}$,

$$\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star) \;\le\; c_\Phi\, D_\Phi\big(h_\theta(\cdot), z_H^\star\big). \tag{186}$$

Multiplying by $\frac{1}{2}$ gives

$$\tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) \;\le\; \tfrac{c_\Phi}{2}\, D_\Phi\big(h_\theta(\cdot), z_H^\star\big). \tag{187}$$

Since $\alpha \ge \alpha_{\min} = c_\Phi/2$, we have

$$\tfrac{1}{2}\big(\mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)\big) \;\le\; \alpha\, D_\Phi\big(h_\theta(\cdot), z_H^\star\big). \tag{188}$$

Substituting Eq. 188 into Eq. 178 yields

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{task}}(\theta) + \alpha\, D_\Phi\big(h_\theta(\cdot), z_H^\star\big) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star). \tag{189}$$

Again, $\mathcal{L}_{\text{CL}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \alpha D_\Phi\big(h_\theta(\cdot), z\big) + \beta D_\Psi(\theta, \theta_H^\star) \;\ge\; \mathcal{L}_{\text{task}}(\theta) + \alpha D_\Phi\big(h_\theta(\cdot), z_H^\star\big)$, so Eq. 189 implies

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{CL}}(\theta) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star), \tag{190}$$

which is again Eq. 11, now for the output route.

Combining Eq. 185 and Eq. 190 shows that if either $\beta \ge \beta_{\min}$ or $\alpha \ge \alpha_{\min}$, then for all $\theta \in \mathcal{N}$,

$$\mathcal{L}_{\text{bal}}(\theta) \;\le\; \mathcal{L}_{\text{CL}}(\theta) + \tfrac{1}{2}\mathcal{L}_H(\theta_H^\star), \tag{191}$$

which completes the proof. □

Table A1: Implementation details of the considered algorithms for the LTR benchmark.

Algorithm	LR	Opt.	Momentum	LR Scheduler	CL Loss Weight	Epochs	Steps
LwF	0.001	SGD	0.9	–	0.01	5	2
EWC	0.01	SGD	0.9	–	10	90	2
Modified EWC	0.01	SGD	0.9	–	1000	90	2
GPM	0.001	SGD	0	Cosine Anneal LR	–	100	2
SGP	0.001	SGD	0.9	Cosine Anneal LR	–	100	2
FOSTER (phase 1)	0.1	SGD	0.9	Cosine Anneal LR	–	170	4
FOSTER (phase 2)	0.1	SGD	0	Cosine Anneal LR	–	130	4
TPL	0.01	SGD	0.8	Cosine Anneal LR	–	50	4

C.8 Stronger Statement (Special Case): Exact Minimizer Equivalence

For completeness, we recall a special case (useful for exposition) where CL *exactly* optimizes the balanced objective.

Proposition C.2 (Exact equality up to a constant)**.** *Let $\Psi = \mathcal{L}_H$ and assume $\mathcal{L}_H$ is convex on $\mathcal{N}$. Set $\alpha = 0$, $\beta = \frac{1}{2}$, $\mathcal{L}_{\text{task}} = \frac{1}{2}\mathcal{L}_T$, and z arbitrary. Then $\mathcal{L}_{\text{CL}}(\theta) = \frac{1}{2}\mathcal{L}_T(\theta) + \frac{1}{2}D_{\mathcal{L}_H}(\theta, \theta_H^\star) = \mathcal{L}_{\text{bal}}(\theta) - \frac{1}{2}\mathcal{L}_H(\theta_H^\star)$, so $\mathcal{L}_{\text{CL}}$ and $\mathcal{L}_{\text{bal}}$ have identical gradients and the same minimizers.*

Proof. Since $\nabla\mathcal{L}_H(\theta_H^\star) = 0$, $D_{\mathcal{L}_H}(\theta, \theta_H^\star) = \mathcal{L}_H(\theta) - \mathcal{L}_H(\theta_H^\star)$. Add and subtract $\frac{1}{2}\mathcal{L}_H(\theta_H^\star)$ and regroup terms. □

D Implementation Details

All our experiments were conducted using the PyTorch framework. We build upon publicly available implementations of Learning without Forgetting (LwF), Elastic Weight Consolidation (EWC), Gradient Projection Memory (GPM), Scaled Gradient Projection (SGP), FOSTER, TPL, and DualPrompt, adapting their training protocols where necessary for long-tailed recognition. The corresponding method-specific modifications and hyperparameters are detailed below. [1] The specifics of each algorithm's implementation are summarized in Table A1. The parameters for each algorithm, such as learning rate (LR), optimizer, momentum, LR scheduler, CL loss weight, number of epochs, and number of steps, are detailed there. We mainly follow the original hyperparameter settings recommended for each method, including their training schedules (e.g., number of epochs), and only deviate when strictly necessary (e.g., for algorithm-specific grid search), in order to ensure fair and faithful comparisons.

Note that for SGP, the algorithm-specific hyperparameters are obtained through grid search as follows: $gpm_{\text{eps}} = 0.9$ and $gpm_{\text{eps-inc}} = 0.004$.

We evaluate six adaptation strategies for a frozen ViT-B/32 CLIP backbone (OpenAI pretrained weights) on CIFAR100-LT under imbalance factors $\text{IF} \in \{10, 50, 100\}$.

Initialization Strategy. Algorithm 1 permits initialization using either the Head partition $\mathcal{D}^1$ or the full long-tailed dataset $\mathcal{D}$. This flexibility is motivated by Theorems 3.3, 3.5, and 3.7, which establish that, under their respective assumptions, the full-dataset minimizer lies within an $O(1/\sqrt{IF})$ neighborhood of the Head-only solution or minimizer set. Thus, full-dataset training offers a theoretically motivated Head-dominated initialization.

Zero-shot CLIP requires no training: we compute cosine similarity between image embeddings and text embeddings produced by the fixed template "`a photo of a {class}`", and classify by nearest neighbor.

[1] The code for the algorithms was obtained and modified from various open-source repositories.
https://github.com/ngailapdi/LWF
https://github.com/shivamsaboo17/Overcoming-Catastrophic-forgetting-in-Neural-Networks
https://github.com/sahagobinda/GPM
https://github.com/sahagobinda/SGP
https://github.com/G-U-N/ECCV22-FOSTER
https://github.com/linhaowei1/TPL
https://github.com/google-research/l2p

Linear probe freezes the ViT-B/32 encoder and trains only a linear $512 \to 100$ classification head with AdamW (lr $= 10^{-3}$, $\lambda = 10^{-2}$, 50 epochs, cosine schedule).

Full fine-tuning optimizes all ViT-B/32 parameters jointly with a linear head using AdamW (lr $= 3 \times 10^{-5}$, $\lambda = 10^{-2}$, 20 epochs, cosine schedule with 200-step warmup).

Prompt learning follows the clip-zs-prompting style [86]: a learnable 30-pixel visual-prompt border is prepended to every image while the ViT encoder and text branch remain frozen, and is trained with SGD (lr $= 40$, momentum 0.9, cosine schedule, 1000-step warmup, 10 epochs).

Naive sequential applies the same full fine-tuning procedure sequentially: first on the Head partition (60 classes), then on the Tail partition (40 classes), with no replay buffer, intentionally inducing catastrophic forgetting.

CLTR (DualPrompt). We adapt the DualPrompt architecture [40] using the OpenAI-pretrained CLIP ViT-B/32 backbone. The visual encoder is frozen except for its LayerNorm parameters. We employ G-prompts (length 5, layers 0–1) and E-prompts (pool size 10, length 5, top-$k = 1$, layers 2–4), together with a trainable cosine classifier initialized from CLIP text embeddings of class names. Optimization uses AdamW with cosine learning-rate scheduling and a 200-step warmup, balanced-softmax classification loss, and prompt-similarity regularization. During Tail adaptation, the method incorporates exemplar replay from the Head partition and knowledge distillation from a frozen Head-stage teacher. At inference, logits are averaged over the original and horizontally flipped images, followed by validation-calibrated logit adjustment. Table A2 summarizes the key hyperparameters.

Table A2: Implementation details for foundation-model adaptation experiments on CIFAR100-LT.

Method	Backbone	Trainable params	Opt.	LR	Epochs	Tasks
Zero-shot CLIP	ViT-B/32 (frozen)	0	–	–	–	–
Linear probe	ViT-B/32 (frozen)	head only	AdamW	10^{-3}	50	1
Full fine-tuning	ViT-B/32	all	AdamW	3×10^{-5}	20	1
Prompt learning	ViT-B/32 (frozen)	visual prompts	SGD	40	10	1
Naive sequential	ViT-B/32	all	AdamW	3×10^{-5}	20	2
DualPrompt	ViT-B/32 (LN adapted)	G/E prompts + LN + cosine head	AdamW	3×10^{-3}	40–60	2

Formal Verification and Resources. Lean 4 formalizations of our main theoretical results, together with implementations of selected top-performing CLTR configurations (SGP and DualPrompt-based), are available at `https://github.com/MahdiyarMM/CLTR`.

E Datasets

Fig. A1 illustrates the distribution of samples among different classes and the division of the dataset into the Head and Tail sections. For exponentially decaying class distributions, we follow Proposition 3.10 to partition each dataset into Head and Tail subsets.

F Additional Discussions

F.1 Comparison with Methods Using Additional Techniques

Table A3 provides an additional comparison with LTR methods that use extra techniques such as aggressive data augmentation, ensemble learning, multi-expert architectures, or self-supervised pretraining. Since these methods do not follow the same protocol as our main comparison, we report them separately in the appendix for reference rather than including them in the main benchmark table.

F.2 Long-Tailed Class-Incremental Learning

Few prior works have attempted to address the problem of class-incremental learning when the data is heavily imbalanced. A novel replay method called Partitioning Reservoir Sampling (PRS) is proposed in [87]. This method dedicates a sufficient amount of memory to Tail classes in order to

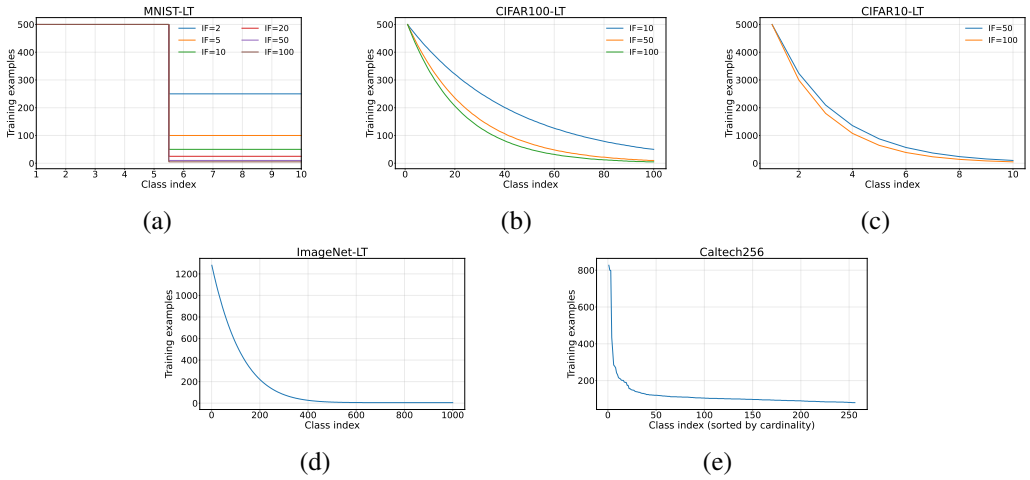

Figure A1: Class cardinality of (a) MNIST-LT, (b) CIFAR100-LT, (c) CIFAR10-LT, (d) ImageNet-LT, and (e) Caltech256.

Table A3: Performance comparison on CIFAR100-LT and ImageNet-LT. The blue rows show CLTR variants. The gray block reports methods with additional bells and whistles and is included for context, since these methods may not follow the same protocol.

Method	ImageNet-LT	CIFAR100-LT		
		IF = 100	IF = 50	IF = 10
CLTR (LwF)	47.6	45.1	49.3	58.7
CLTR (EWC)	48.9	44.4	50.3	58.8
CLTR (Modified EWC)	49.1	45.9	51.0	60.7
CLTR (GPM)	51.7	48.3	54.7	64.7
CLTR (FOSTER)	52.7	48.7	54.4	63.6
CLTR (SGP)	<u>53.2</u>	<u>50.7</u>	**58.0**	<u>67.2</u>
CLTR (TPL)	**53.9**	48.4	54.0	62.1
Methods with bells and whistles				
RIDE [22]	56.1	49.1	NR	NR
ACE [66]	56.6	49.4	50.7	NR
SSD [44]	56.0	46.0	50.5	62.3
PaCo [67]	54.0	52.0	56.0	64.2

avoid catastrophic forgetting in minority classes. In [88], this problem is addressed in two different setups, ordered and shuffled. In the ordered scenario the number of samples in each new task is less than in previous tasks, while in the shuffled scenario, the size of classes is completely random. They propose a two-stage learning method utilizing a learnable weight scaling layer for reducing the bias due to data imbalance. Finally, in [89], OLTR++ is proposed which is a unified algorithm that integrates imbalanced classification, few-shot learning, open-set recognition, and active learning through dynamic meta-embedding and memory association. Note that none of the above works attempt to employ CL as a solution for LTR scenarios.

Here, we compare the performance of CLTR in addressing the LT-CIL problem with the prior state-of-the-art solutions in the area. Following the experimental setup in [88, 90, 91], the models are first trained on the largest 50 classes (Head), then, the remaining classes are learned incrementally in 5 or 10 consecutive tasks (Tail) with an equal number of new classes in each new task, from the largest subset to the smallest subset of the dataset. We apply our method in this setting on the CIFAR100-LT dataset and compare its performance with the prior works, as presented in Table A4. The results demonstrate that CLTR outperforms prior methods in both 5- and 10-task settings by considerable margins of 2.5% and 1.6%, respectively.

Table A4: The performance of CLTR on Ordered LT-CIL Benchmark for CIFAR100-LT

Method	Tasks	
	5	10
EEIL [92]	38.5	37.5
EEIL+2sLWS [88]	39.0	37.6
LUCIR [90]	42.7	42.2
PODNET [91]	44.1	44.0
PODNET+2sLWS [88]	44.4	44.4
LUCIR+2sLWS [88]	<u>45.9</u>	<u>45.7</u>
CLTR (TPL)	**48.4**	**47.3**

F.3 Naturally Skewed Dataset

In LTR benchmarks, datasets are modified to exhibit a skewed distribution of samples among various classes. However, such imbalanced class distributions are naturally observed in real-world data as well [5]. To evaluate the efficacy of CL techniques on non-LTR benchmark datasets, we utilize the Caltech256 dataset [33], which consists of 256 distinct classes representing everyday objects. The largest class comprises 827 samples, while the smallest class contains only 80 samples, exhibiting an IF of over 10. Here, we employ the CLTR and compare its performance to the state-of-the-art methods on this dataset for object classification. The results are presented in Table A5. We observe that CLTR outperforms the previous method on this dataset, demonstrating the strong potential of using CL in dealing with long-tailed real-world datasets.

Table A5: The performance of CLTR on Caltech256.

Method	Backbone	
	Inc.V4	Res.101
$L^2 - FE$ [93]	84.1	85.3
L^2 [93]	85.8	87.2
$L^2 - SP$ [93]	85.3	87.2
DELTA [93]	<u>86.8</u>	<u>88.7</u>
GBN [94]	-	86.9
TransTailor [94]	-	87.3
CLTR (SGP)	**88.6**	**89.8**

F.4 Training Order

Our theoretical analysis motivates a specific ordering for CLTR: the model should first learn the largest partition, i.e., the Head classes, and then progressively incorporate smaller Tail partitions while preventing forgetting. This ordering follows directly from Theorems 3.3–3.7, which show that standard training on a long-tailed dataset yields parameters close to the solution obtained by training on the Head classes alone. Therefore, the Head solution provides a natural anchor point for the subsequent CL stage. Learning the Tail after the Head allows the CL objective to move the model toward Tail-sensitive solutions while preserving Head knowledge.

To verify this prediction empirically, we evaluate CLTR(SGP) on CIFAR100-LT with $\mathrm{IF} = 100$ under three task orders: Head→Tail, Tail→Head, and Random. The results are reported in Table A6. These results confirm that the ordering is not an arbitrary implementation detail: it is a key part of the CLTR formulation and is consistent with the geometry predicted by our theory.

F.5 Practical estimation of $\beta_{\min}$ and $\alpha_{\min}$

To illustrate that the conditions in Theorem 3.12 are mild and typically satisfied by standard CL methods, we provide a concrete empirical estimation of $\beta_{\min}$ and $\alpha_{\min}$ in a real-world setting. We

Table A6: Effect of task order for CLTR(SGP) on CIFAR100-LT with IF $= 100$. Head→Tail follows the cardinality-sorted ordering motivated by our theory and achieves the best performance.

Training order	Top-1 accuracy
Tail→Head	36.0
Random	32.3
Head→Tail (CLTR order)	**50.7**

consider a ResNet-18 trained on the Head classes of CIFAR-100-LT, matching our experimental setup.

Estimating $\beta_{\min}$: Recall that $\beta_{\min} = \frac{L_H^{\mathrm{sm}}}{2\, m_\Psi}$, where L_H^{sm} is the local smoothness constant of the Head loss and m_Ψ is the strong-convexity parameter of the parameter-space divergence D_Ψ. We approximate L_H^{sm} by the top eigenvalue of the Hessian of L_H at $\theta_H^\star$ using Hessian–vector products and power iteration. For the trained ResNet-18, we obtain $L_H^{\mathrm{sm}} \approx 0.2$. When D_Ψ is the standard ℓ_2 divergence (the default choice in many EWC-style methods), the corresponding Bregman divergence is 2-strongly convex, implying $m_\Psi \approx 1$. This gives the practical threshold $\beta_{\min} \approx \frac{0.2}{2} = 0.1$. This value is substantially *below* the regularization strengths typically used in off-the-shelf CL methods (e.g., $\beta = 10$–10^3 in EWC variants), indicating that the theoretical requirement is easily satisfied in practice.

Estimating $\alpha_{\min}$: In the unified CL formulation, the output-space regularizer takes the form $\alpha\, D_\Phi(h_\theta, z_H^\star)$, where D_Φ is a Bregman divergence in logit space. In Theorem 3.12, $\alpha_{\min}$ is defined as $\alpha_{\min} = \frac{c_\Phi}{2}$, where c_Φ is the smallest constant satisfying $L_H(\theta) - L_H(\theta_H^\star) \le c_\Phi\, D_\Phi(h_\theta, z_H^\star)$ for all θ in a local neighborhood. For Bregman divergences D_Φ induced by a strongly convex potential in logit space (e.g., squared-logit or KL divergences), this inequality is equivalent to a standard smoothness–strong-convexity relation in the logit domain. In particular, if L_H is L_{out}-smooth with respect to logits and D_Φ is m_Φ-strongly convex, then $c_\Phi = \frac{L_{\text{out}}}{m_\Phi}$. Substituting this expression into $\alpha_{\min} = c_\Phi/2$ yields the empirically estimable form $\alpha_{\min} = \frac{L_{\text{out}}}{2m_\Phi}$, which we use below. L_{out} quantifies the local smoothness of L_H with respect to logits and m_Φ is the strong-convexity constant of D_Φ. To estimate L_{out}, we compute the largest eigenvalue of the logit-space Hessian $\nabla_z^2 L_H$ at $\theta_H^\star$, again using power iteration on batches of Head samples. Since L_H is a batch-averaged loss in our theory, this batch-averaged Hessian is the correct object to analyze. In our ResNet-18 / CIFAR-100-LT Head-class setting, we observe $L_{\text{out}} \approx 2 \times 10^{-4}$. For squared-logit or KL-based output divergences, the associated Bregman divergence is 1-strongly convex in the relevant output domain, giving $m_\Phi \approx 1$. Hence, $\alpha_{\min} \approx \frac{2\times 10^{-4}}{2} = 10^{-4}$. Again, this threshold is far smaller than the regularization strengths commonly used in output-regularized CL methods (e.g., LwF typically uses α between 10^{-2} and 1). Thus, even modest output-space regularization comfortably exceeds $\alpha_{\min}$.

Both estimated thresholds, $\beta_{\min} \approx 0.1$ and $\alpha_{\min} \approx 10^{-4}$, lie *orders of magnitude below* the α and β values routinely used in practical CL methods. This empirical check supports the conclusion that the theoretical conditions of Theorem 3.12 are not restrictive and are naturally satisfied by off-the-shelf continual learning algorithms.

Empirical Verification of the Upper Bound. To investigate whether the upper-bound inequality in Theorem 3.12 remains satisfied during Tail learning, we evaluate its empirical margin at intermediate training checkpoints. Because the radius of the neighborhood $\mathcal{N}$ is not directly identifiable, parameter distances alone cannot certify neighborhood membership. Instead, for random seed s and checkpoint t, we measure:

$$M_{s,t} = \mathcal{L}_{\mathrm{CL}}(\theta_{s,t}) + \frac{1}{2}\mathcal{L}_H(\theta_{H,s}^*) - \mathcal{L}_{\mathrm{bal}}(\theta_{s,t}), \tag{192}$$

where $M_{s,t} \ge 0$ indicates that the upper-bound inequality holds at the corresponding checkpoint. We evaluate CIFAR100-LT ($IF = 100$, 10,847 training samples) using ResNet-32 trained from scratch, with three random seeds $s \in \{0, 1, 2\}$. During the 100-epoch Tail stage, we record checkpoints at epochs $\{1, 11, 21, 31, 41, 51, 61, 71, 81, 91, 100\}$, yielding 33 measurements per method. Table A7 reports the worst-case margin $\min_{s,t} M_{s,t}$ across all seeds and checkpoints, rather than an average.

Table A7: Empirical verification of the upper-bound inequality on CIFAR100-LT ($IF = 100$) across 33 measurements per method.

Method	Min. margin	Satisfied
Naive sequential	-12.15	$0/33$
CLTR (LwF)	$+2.25$	$33/33$
CLTR (EWC)	$+0.07$	$33/33$

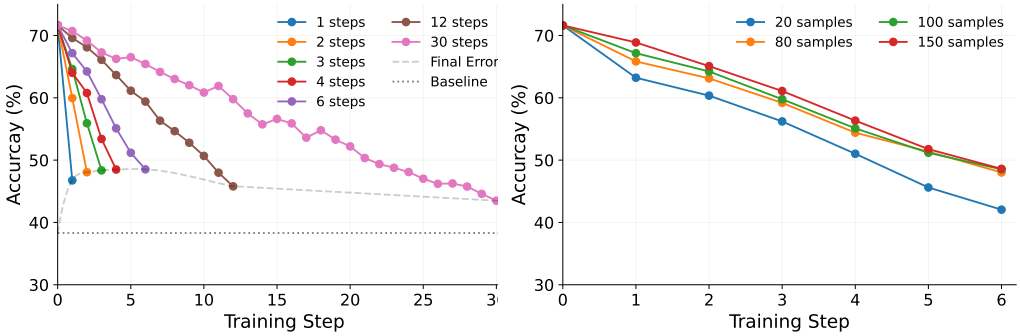

Figure A2: CLTR performance for various (left) number of incremental steps and (right) replay memory ($\mathcal{M}$) size.

Both CLTR variants satisfy the evaluated inequality at all recorded checkpoints, while naive sequential training, for which the CL regularization coefficients are zero, violates it throughout. The worst-case EWC margin is close to zero, whereas the LwF certificate is looser. Furthermore, the measured Head-stage residual loss is 0.0170 ± 0.0009, corresponding to an additive constant of only 0.0085 ± 0.00045. Thus, the observed satisfaction of the inequality is not attributable to a large residual Head-loss constant. These results provide empirical support for the upper-bound certificate along the evaluated trajectories but do not establish that every iterate remains inside the neighborhood required by Theorem 3.12.

F.6 Multiple Incremental Steps

Recall that Theorem 3.7 highlights CLTR's capability for extending beyond two incremental steps. Increasing the number of partitions leads to smaller IF within each partition at the cost of an increase in forgetting. To explore the effect of varying partition numbers on CLTR's final performance, we adhere to the experimental protocol outlined in [88]. Initially, the model is trained on the first 60 classes, followed by sequential learning of the remaining classes, divided into different numbers of partitions. The results of this experiment are presented in Fig. A2 (left). Our results reveal an optimal value for CLTR (FOSTER) (4 steps), yet the performance margin remains slim even with up to 30 tasks. This highlights the effectiveness of CL methods employed within CLTR. The optimal value for the number of incremental steps for each CL algorithm can be found in Appendix D.

To further evaluate the boundary-selection criterion of Proposition 3.10, we vary the Head–Tail class boundary while keeping the remaining training configuration fixed. Specifically, we evaluate CLTR (SGP) with two partitions on CIFAR100-LT ($IF = 100$) using three class splits: 20:80, 50:50, and 80:20. As shown in Table A8, the 50:50 split approximately equalizes the local imbalance factors and achieves the highest accuracy among the evaluated boundaries, outperforming the alternative splits by 3.7 and 5.0 percentage points. These results empirically support the boundary-selection criterion in Proposition 3.10.

F.7 Replay Memory

Several CL algorithms incorporate mechanisms to retain partial information from the previous task, aiming to mitigate catastrophic forgetting. For example, EWC maintains prior model parameters along with their Fisher values, whereas both GPM and SGP safeguard the Core Gradient Space of previous tasks. FOSTER, on the other hand, utilizes a replay memory for this purpose. Within the LTR context, the presence of a buffer memory doesn't require additional storage, as access to the

Table A8: Sensitivity of CLTR (SGP) to Head–Tail class boundaries on CIFAR100-LT ($IF = 100$) with two partitions.

Head:Tail split	Local IFs	Top-1 (%)
20:80	[2.43, 39.40]	46.0
50:50	**[9.8, 9.6]**	**49.7**
80:20	[41.67, 2.40]	44.7

Table A9: Computational overhead on CIFAR100-LT using ResNet-32 and a single RTX 3090 GPU. Wall-clock time is averaged over three runs. Relative values are normalized to WD+MaxNorm.

Method	Time (s)	Rel.	FLOPs (T)	Rel.	Mem. (MB)
LTR (WD)	205.5 ± 1.7	0.986	227.0	1.000	334
LTR (WD+MaxNorm)	208.3 ± 7.5	1.000	227.0	1.000	334
CLTR (no CL term)	238.6 ± 1.9	1.145	252.1	1.111	334
CLTR (LwF)	238.2 ± 1.4	1.143	264.4	1.164	374
CLTR (EWC)	260.4 ± 13.0	1.250	254.3	1.120	340
CLTR (SGP)	280.9 ± 2.7	1.349	252.5	1.112	343
CLTR (SGP, GPU SVD)	257.9 ± 8.3	1.238	252.5	1.112	351

full dataset is already available. Nonetheless, to prevent hindering the model's capacity to learn Tail distributions, we deliberately avoid replaying all Head samples when learning the Tail (evidenced by Eq. 16). Accordingly, our analysis extends to how the number of Head samples replayed while learning the Tail impacts the model's performance, as illustrated in Fig. A2 (right). The replay memory's size serves as a mediator between forgetting previous information and worsening class imbalance, e.g. a larger replay memory reduces forgetting but increases imbalance. Therefore, identifying an optimal balance in this trade-off is crucial. Our results demonstrate the significance of an appropriate replay memory size; however, there exists a threshold beyond which additional samples per class do not further improve the performance and the performance levels off.

F.8 Runtime

We evaluate the computational overhead of CLTR against standard LTR training on CIFAR100-LT using ResNet-32, a batch size of 128, and a single NVIDIA RTX 3090 GPU. All methods use identical data-loading and evaluation settings, and the measurements include method-specific computations such as Fisher information estimation for EWC and singular value decomposition (SVD) for SGP. We report wall-clock training time averaged over three runs, total training FLOPs, and peak GPU memory in Table A9. Using WD+MaxNorm as a computationally efficient LTR reference, CLTR introduces a moderate, method-dependent overhead. Specifically, LwF, EWC, and SGP increase wall-clock training time by 14.3%, 25.0%, and 34.9%, respectively, while increasing training FLOPs by 11.2–16.4%. The larger runtime overhead of SGP is primarily associated with its SVD computation; performing SVD on the GPU reduces its overhead from 34.9% to 23.8%. Peak GPU memory increases by at most 40 MB across the evaluated CLTR variants. These results demonstrate that CLTR incurs additional training costs associated with the selected CL algorithm, rather than being computationally equivalent to standard LTR training. However, for the evaluated methods, the overhead remains relatively modest, and inference complexity is unchanged when the same backbone architecture is retained. The measurements in Table A9 are obtained under a consistent CIFAR100-LT/ResNet-32 benchmarking setup on an RTX 3090, providing a comparative assessment of computational overhead across methods. The reported wall-clock times correspond to this benchmark configuration, while experiment-specific training schedules, including that of the released CIFAR-SGP implementation, are documented separately in Appendix D.

F.9 Weight Imbalance

An interesting phenomenon observed when training models on highly imbalanced data is the presence of artificially large weights in neurons corresponding to the Head classes [5]. The LTR solution, WD, addresses this problem by penalizing weight growth using weight decay. One way to assess the network's ability to handle LTR is by analyzing the bias in per-class weight norms. To this end, we present the per-class weight norms of the Baseline, WD, and CLTR (SGP) models in Fig. A3.

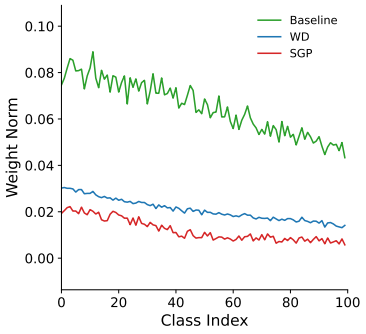

Figure A3: Per-class weight norms of the baseline, SGP, and WD.

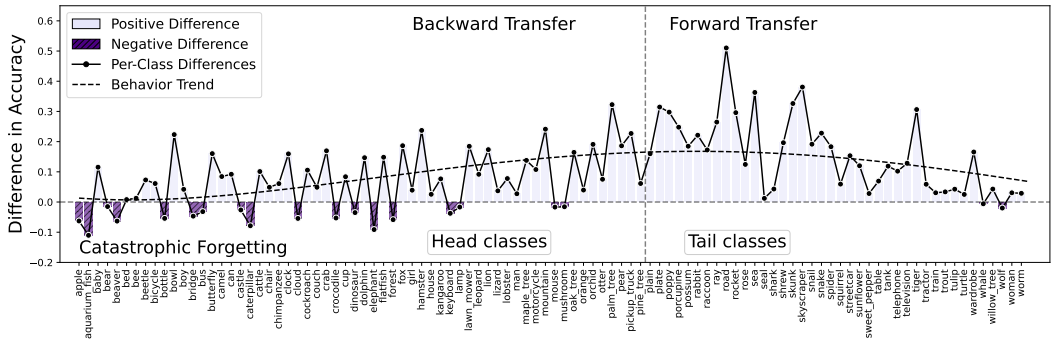

Figure A4: The difference in per-class accuracy of CLTR (SGP) and the baseline model.

The figure reveals a significant imbalance in the weight norms of the Baseline model, which is naively trained on the imbalanced dataset. In contrast, the WD and CLTR (SGP) models exhibit more uniform weight norms across different classes. Interestingly, although CLTR (SGP) starts with the heavily imbalanced weights of the Baseline model, it converges towards a more uniform weight distribution without any explicit penalty on weight growth. Unlike many other CL methods that restrict the plasticity of crucial weights, SGP only constrains the direction of the weight update in the weight space, enabling the model to converge to a more balanced weight distribution. This further demonstrates the effectiveness of CL in addressing LTR problem.

F.10 Backward/Forward Transfer and Catastrophic Forgetting

Prior works discuss three key concepts in the context of CL: catastrophic forgetting, backward transfer, and forward transfer [95]. As mentioned earlier, catastrophic forgetting occurs when the performance of a class declines after retraining. Despite the use of CL methods, which are designed to mitigate this forgetting, a certain degree of forgetting is still inevitable. Forward transfer is the improvement in performance on a new task after employing CL, which is the central aim of retraining in CL. Finally, backward transfer is a beneficial side-effect where retraining on new samples can actually enhance the model's performance on the previous tasks. This interesting phenomenon in CL has been extensively discussed in previous works in theory and practice [96]. Now, let us discuss Fig. A4, which presents the difference in per-class accuracy of CLTR (SGP) versus the baseline network. The analysis is based on CIFAR100-LT with an IF of 100. The figure is divided into three regions corresponding to the scenarios discussed above: catastrophic forgetting (bottom), backward transfer (top-left), and forward transfer (top-right). The bottom region in the figure represents classes that undergo catastrophic forgetting, while the top-right region represents the Tail samples (with a class index larger than 60), which demonstrate improved performance, or forward transfer. We observe that using SGP as a CL solution for LTR results in very effective improvements in the per-class accuracy of the Tail (forward transfer). Interestingly, despite the absence of Head data in the retraining process, 42 out of 60 Head classes see some level of improvement after the model is exposed to the Tail

samples (backward transfer). This result emphasizes the remarkable potential of CL methods in enhancing performance in both new and previous tasks.

F.11 CLTR performance on ViT backbone

Our choice of ResNet-32 on CIFAR and ResNeXt-50 on ImageNet followed standard LTR benchmark protocols to ensure fair comparison with prior baselines [5, 4]. Theorem 3.12 is an optimization-level guarantee: it depends only on local loss properties, not on a specific architecture. Therefore, it applies to any differentiable model, provided Assumption 3.11 and the coefficient conditions of Theorem 3.12 hold . To empirically verify this, we have also conducted additional experiments with a ViT-B backbone on CIFAR100-LT:

Table A10: Performance of CLTR with a ViT-B backbone on CIFAR100-LT under different imbalance factors.

Method (ViT-B)	IF		
	10	50	100
Baseline [16]	66.02	54.78	50.59
BCE [16]	64.63	50.65	45.50
CB [16]	<u>66.30</u>	<u>56.18</u>	<u>50.63</u>
CLTR (TPL)	**69.74**	**59.71**	**55.94**

CLTR (TPL) outperforms the baselines across every imbalance factor, confirming that the framework transfers directly to modern architectures without any modification.

F.12 Generalization to Long-Tailed Text Classification

To investigate whether CLTR generalizes beyond vision, we evaluate it on 20 Newsgroups-LT, constructed from the 20 Newsgroups dataset using the standard `bydate` split, with headers, footers, and quotes removed. The 20 classes are sorted by training frequency and subsampled according to the exponential profile $n_i = n_{\max} IF^{-i/(C-1)}$, with $IF = 100$, $C = 20$, and $n_{\max} = 377$. This produces 1,729 training documents, divided into two class-disjoint partitions of 10 Head classes (1,593 samples) and 10 Tail classes (136 samples), following Proposition 3.10. Evaluation uses the original test split containing 7,532 documents.

We fine-tune the `sentence-transformers/all-mpnet-base-v2`[2] encoder end-to-end, using mean token pooling followed by dropout (0.1) and a linear classifier. Input sequences are truncated or padded to 128 tokens. Training uses AdamW with weight decay 0.01, batch size 32, gradient clipping at 1.0, bf16 mixed precision, and a OneCycle learning-rate schedule with 10% linear warmup and linear decay. The initial learning rate is 3×10^{-5}, increased fourfold to 1.2×10^{-4} during Tail training for all sequential methods. Each partition is trained for 15 epochs, while joint training uses 15 epochs on the full dataset. All sequential variants share the same Head-stage initialization. For CLTR (LwF), we use a frozen Head-stage teacher with distillation weight $\alpha = 0.03$ and temperature $T = 2$. For CLTR (GPM), we use an energy threshold $\epsilon = 0.95$ and hard gradient projection across all encoder linear layers, excluding the classifier, with up to 4,000 sampled activation vectors per layer. CLTR variants additionally retain 50 exemplars per Head class for replay.

We compare CLTR against joint cross-entropy training on the long-tailed dataset, naive Head-to-Tail sequential training without a CL mechanism, and balanced training ($IF = 1$) as a reference. We evaluate Head, Tail, and balanced accuracy, with balanced accuracy defined as the mean per-class recall across all 20 classes.

As shown in Table A11, joint long-tailed training exhibits substantial Head bias, while naive sequential training suffers catastrophic forgetting. CLTR mitigates this issue, improving Tail accuracy over joint training by 8.49 and 11.52 percentage points with LwF and GPM, respectively, while achieving balanced accuracy gains of 3.08 and 3.69 points. These results provide preliminary evidence that the benefits of Head-to-Tail continual learning extend to text classification, although broader validation across additional NLP benchmarks and random seeds remains future work.

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Table A11: Performance on 20 Newsgroups-LT ($IF = 100$). Head, Tail, and Balanced denote macro-averaged classification recall (%).

Method	Head	Tail	Balanced
Balanced training ($IF = 1$)	77.42	66.18	71.80
Long-tailed joint training	81.53	33.20	57.37
Naive sequential	0.00	32.78	16.39
CLTR (LwF)	**79.21**	41.69	60.45
CLTR (GPM)	77.39	**44.72**	**61.06**

G Broader Impact

LTR is fundamental to applications where data acquisition is expensive or naturally imbalanced like medical imaging (rare disease classification), ecological monitoring (rare species), industrial inspection (rare defects), and satellite imagery (rare land-cover classes). Improving long-tail performance directly benefits these high-impact applications by enabling reliable recognition of underrepresented categories. CLTR is method-agnostic and can be applied with any off-the-shelf CL technique, lowering the barrier to deploying LTR in practice. We do not foresee significant negative societal impact specific to this work; however, like all classification techniques, CLTR could be misapplied in surveillance or biometric contexts where rare-category recognition raises privacy concerns. The pretrained CLIP backbone we use inherits any biases from its pretraining data; downstream practitioners should audit foundation model biases before deployment.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the three main contributions: (1) the $\mathcal{O}(1/\sqrt{\mathrm{IF}})$ convergence bound for both uniform and exponentially decayed cardinality distributions (Theorems 3.3, 3.5, and 3.7); (2) the objective-level reformulation of LTR as a CL problem and the CL upper-bound guarantee on the balanced LTR loss (Proposition 3.9 and Theorem 3.12); and (3) the empirical validation including strong foundation-model adaptation results (the best results among methods not using additional semantic supervision) in Section 4. All quantitative claims in the abstract are backed by the corresponding tables and figures in the paper. Scope limitations (primarily image classification, preliminary text-classification evidence, and local theoretical conditions) are explicitly acknowledged in the Limitations section.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations discussion identifies the local assumptions underlying the theoretical guarantees, the incomplete characterization of the stability–plasticity trade-off, and the scope of our empirical evaluation. Experiments primarily cover long-tailed image classification, with additional evaluation using a CLIP foundation model and preliminary text-classification results on 20 Newsgroups-LT. Broader NLP tasks, joint multimodal adaptation, detection, segmentation, and regression remain directions for future work.

   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: [Yes]

   Justification: Each theoretical result states its assumptions explicitly or refers to the relevant preceding assumptions. Complete proofs for all results: Theorem 3.3 (convex case), Theorem 3.5 (non-convex / KL framework), Theorem 3.7 (multi-partition), Remark 3.8 (exponential decay), Proposition 3.9 (objective-level reformulation), Proposition 3.10 (optimal partitioning), and Theorem 3.12 (CL upper bound), are provided in Appendix C with cross-references to the corresponding assumptions. Lemmas relied upon are stated and proved in Appendix C.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 4 and Appendix D document the experimental datasets, backbone architectures, training protocols, method-specific hyperparameters, partitioning procedures, and evaluation settings. Appendix F.12 documents the text-classification experiment. Public implementations of the remaining CL methods are referenced in Appendix D.

   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 are public benchmarks, and Appendix D documents dataset construction, model architectures, training protocols, hyperparameters, and evaluation procedures. The CL methods employed in CLTR build upon publicly available implementations, referenced in Appendix D. Implementations of selected CLTR configurations (CIFAR-LT with SGP and DualPrompt-based foundation-model adaptation), together with Lean 4 formalizations of the main theoretical results, are available at `https://github.com/MahdiyarMM/CLTR`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4 specifies backbones, datasets, evaluation protocol (balanced test set, classification accuracy), and partition strategy. Appendix D provides full hyperparameter

tables for all CL methods (Table A1) and for all foundation-model adaptation baselines (Table A2), including optimizer type, learning rate, momentum, scheduler, epochs, and method-specific parameters. For SGP, algorithm-specific hyperparameters (gpm_{ε}, $gpm_{\varepsilon\text{-inc}}$) obtained via grid search are explicitly stated. Compute hardware (NVIDIA RTX 3090, 24 GB VRAM) is reported in Section 4.

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: [No]

   Justification: Following the established reporting convention in the long-tailed recognition literature [5, 4, 70], main results in Tables 1 and 2 are reported as single-run accuracy without error bars; baseline results are taken directly from their respective publications under the same protocol. We do not report multi-seed error bars because repeating all CLTR variants across multiple datasets, imbalance factors, and foundation-model settings would require substantial additional compute. Supporting analyses use three random seeds: the upper-bound verification reports the worst case over all seeds and checkpoints (Appendix F.5), the runtime measurements report mean $\pm$ standard deviation over three runs (Appendix F.8), and the 20 Newsgroups-LT results are averaged over three seeds (Appendix F.12). The theoretical results are deterministic and require no statistical treatment.

   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: Section 4 reports the compute hardware used for all experiments, namely a single NVIDIA RTX 3090 GPU with 24 GB VRAM. Appendix F.8 reports wall-clock training

time, total training FLOPs, and peak GPU memory for representative CLTR variants and LTR baselines. The experiments do not require specialized clusters, proprietary hardware, or cloud-only infrastructure.

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 work studies supervised classification on public benchmark datasets: long-tailed image classification (CIFAR-10/100-LT, ImageNet-LT, Caltech-256, MNIST-LT) and text classification (20 Newsgroups-LT). We collected no new data, ran no studies with human subjects, and used no personal data beyond what is already in these standard public benchmarks. All authors have reviewed the NeurIPS Code of Ethics, and the research follows it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The Broader Impact section (Appendix G) discusses positive societal impacts (improved recognition of rare categories in medical imaging, ecological monitoring, industrial inspection, and satellite imagery). It also acknowledges a potential negative impact: like all recognition methods, CLTR could be applied in surveillance or biometric contexts where rare-category recognition raises privacy concerns, and that pretrained foundation-model backbones may inherit biases from their pretraining data that practitioners should audit before deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out

that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not release any pretrained language model, image generator, scraped dataset, model weights, or other high-risk asset. CLTR is evaluated as a training framework on standard image and text classification benchmarks. No special safeguards beyond respecting the licenses and terms of use of the existing datasets, pretrained backbones, and public codebases are required.

    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 assets are cited, and their licenses and terms of use are respected. *Datasets:* CIFAR-10/100 [97] (released for research use, no explicit license); MNIST [32] (CC BY-SA 3.0); ImageNet-LT [11] (subset of ImageNet, used under the ImageNet terms of access for non-commercial research; split files from the OLTR repository, BSD-3-Clause); Caltech-256 [33] (CC BY 4.0); 20 Newsgroups [98] (publicly distributed for research, no explicit license; loaded via scikit-learn, BSD-3-Clause). *Pretrained models:* CLIP ViT-B/32 [36] (MIT); all-mpnet-base-v2 from Sentence-Transformers (Apache-2.0). *Code:* public implementations cited in Appendix D: LwF (Apache-2.0); GPM, SGP, and FOSTER (MIT); DualPrompt via google-research/l2p (Apache-2.0). The EWC and TPL repositories state no license; we use them for non-commercial research only and cite the original papers. The long-tailed versions of the benchmarks are constructed following [5, 4] and do not alter the original 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: [Yes]

    Justification: The only new asset is research code (Lean 4 formalizations of the main theoretical results and CLTR implementations for selected configurations), documented with reproduction instructions in its repository (Appendix D). No new datasets or pretrained models are released.

    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: This paper involves no crowdsourcing and no research with human subjects. All experiments are computational and use publicly available image and text classification datasets.

    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: This paper involves no human subjects; therefore, IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as any component of the core methods in this research. The paper's contributions are a theoretical framework (mathematical proofs) and an empirical framework (classification experiments); no LLM is involved in either.

    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.