MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

DEPARTMENT OF ITS

FPT Education

FPT UNIVERSITY

Gating Mechanisms in Ensemble Models for Robust Rejection Learning on Long-Tail Data

Nguyen Hong Hai

Duong Xuan Bach

Le Ngoc Mai

Phan Hoai Nam

Supervisor: Dr. Bui Van Hieu

Bachelor of Artificial Intelligence
Hoa Lac campus - FPT University
2025

Abstract

Traditional learning to reject (L2R) methods, such as Chow's Rule, are demonstrably sub-optimal for group-aware metrics on long-tail datasets. While theoretically-optimal "plug-in" algorithms were recently proposed to address this, their practical performance remains bottlenecked by a critical dependency: they must trust the flawed, poorly-calibrated, and overconfident posteriors generated by a single classifier. This "garbage in, garbage out" problem means the theory's full potential is unrealized. Recognizing that no single expert can overcome this inherent bias, our method instead constructs a committee of diverse expert models, each trained with a distinct inductive bias to handle imbalance. We then introduce an adaptive routing gating mechanism: a lightweight gating network learns to route samples and mix expert information at the posterior level, yielding a unified posterior that is demonstrably more calibrated. By feeding this repaired posterior into the same standard plug-in algorithm, we unlock its true performance. Extensive experiments show this approach yields substantial gains, reducing the Worst-Group AURC by 36.6% and the Balanced AURC by 33.2% relative to the original single-expert plug-in baseline. We show this gain is strongly correlated with a 47% reduction in calibration error on tail classes.

Keywords: Learning to Reject (L2R), Long-Tail Learning, Mixture-of-Experts (MoE)

Acknowledgements

We sincerely thank our team members for their unwavering dedication and for sharing their valuable knowledge and expertise throughout this project. Their contributions were essential to helping us achieve our goals.

We would also like to express our deepest gratitude to our supervisor, Mr. Bui Van Hieu, for his exceptional guidance, motivation, and support during the course of this project. His leadership, vision, and expertise played a vital role in the successful completion of our work.

Once again, we extend our heartfelt appreciation to our teammates and supervisor for their invaluable contributions and unwavering support in bringing this project to fruition.

Contents

List of Figures

List of Tables

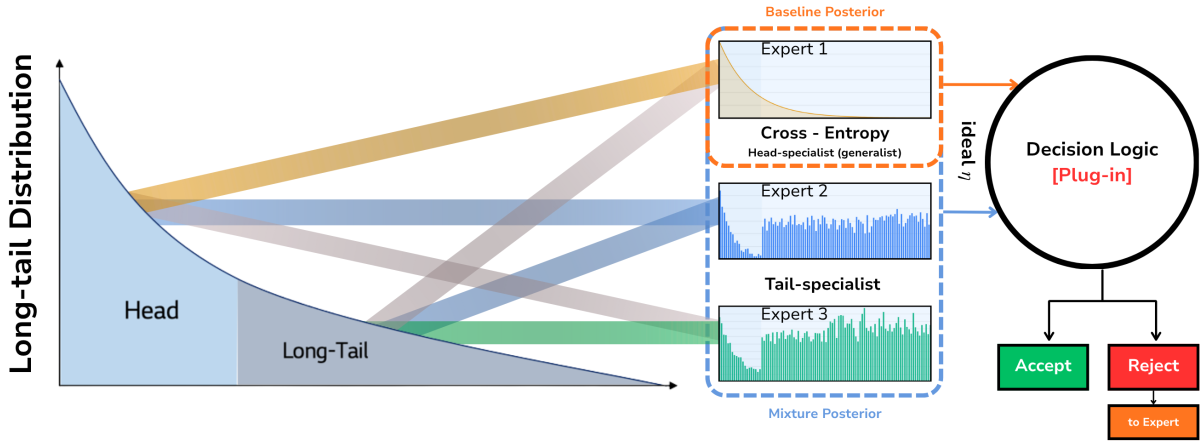

Figure 1: Illustration of the learning-to-reject problem under long-tailed distributions, where head- and tail-specialist experts provide complementary posteriors that can be combined and passed to a plug-in decision rule for accept–reject choices.

1 Introduction

Learning to Reject (L2R), also known as selective prediction, arises from an urgent and practical need in modern artificial intelligence. In high-stakes domains such as medical diagnosis, autonomous driving, or legal analysis, a single erroneous decision can carry severe, even catastrophic, consequences [4, 7, 11]. The objective in these fields is shifting from maximizing raw accuracy to guaranteeing safety and reliability. L2R directly addresses this by granting the model the ”right to abstain”: by rejecting inputs it judges as unreliable, the system can trade a small fraction of its predictive coverage for a substantial and necessary gain in trustworthiness [7, 8]. Conceptually, any L2R system comprises two core elements: an uncertainty signal (e.g., a posterior distribution) to quantify the model’s confidence, and a decision rule that uses this signal to determine when to abstain [7, 10].

A classical and widely-used decision rule is Chow’s rule [4], which dictates abstention if the maximum predicted class probability (Maximum Softmax Probability, or MSP) falls below a fixed threshold. While simple and effective for its original goal of minimizing the standard 0–1 misclassification error [4, 11], this approach founders when faced with modern fairness-aware objectives. Practitioners now frequently care about group-aware metrics, such as balanced error or worst-group error, which are essential for evaluating performance in real-world scenarios dominated by long-tail label distributions [29, 31]. Under these equitable metrics, Chow’s rule is demonstrably sub-optimal [29]. This misalignment creates inequitable outcomes: the rule disproportionately rejects or mishandles samples from minority ”tail” classes, even as it over-trusts predictions on the majority ”head” classes [29, 36].

This challenge is critically exacerbated by the long-tail nature of real-world data, where a few ”head” classes are data-rich while the vast majority of ”tail” classes are severely underrepresented [2, 31]. This imbalance induces a well-known pathology in deep neural networks: they are frequently miscalibrated and prone to being ”confidently wrong” [10, 30]. This miscalibration is not uniform; it is amplified precisely for the rare, tail classes [14, 36], rendering their posterior estimates inherently untrustworthy. When a rejection mechanism, be it Chow’s rule or a more advanced one, trusts these biased and overconfident tail-class posteriors, its reliability collapses [29, 36].

A significant theoretical breakthrough recently occurred in this space. Narasimhan et al. (2024) moved beyond Chow’s rule by deriving the Bayes-optimal ”plug-in” rejectors for group-

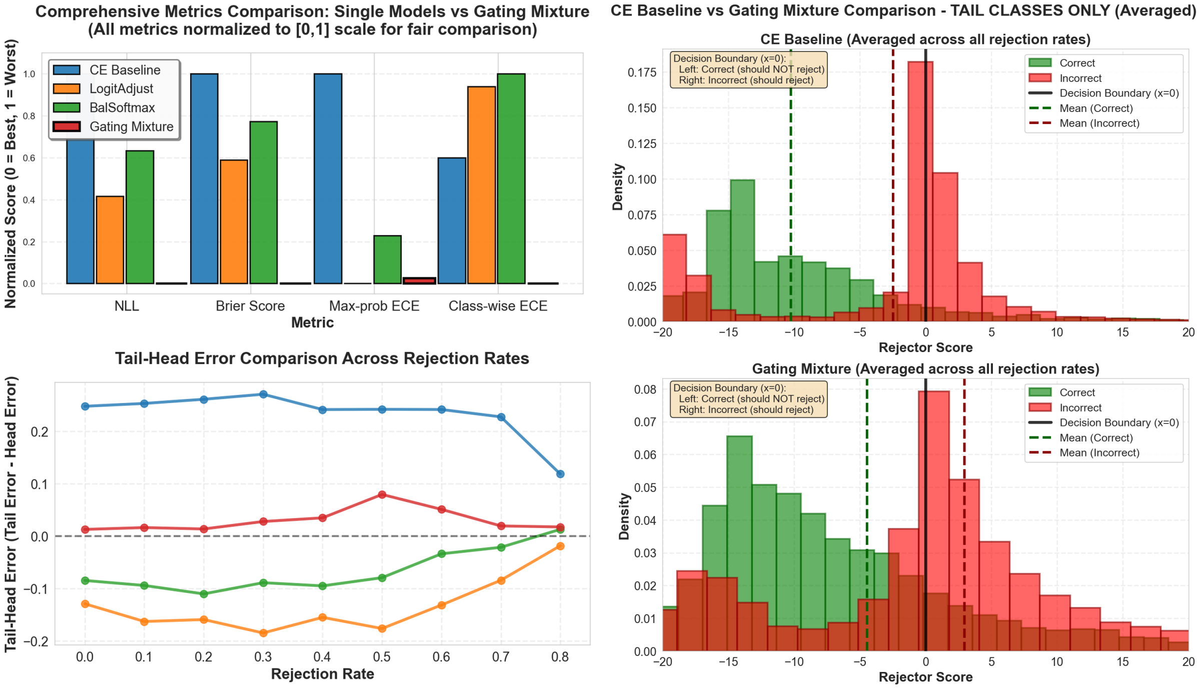

Figure 2: Diagnostic illustration motivating our approach. The figure shows that single-model posteriors are poorly calibrated and produce rejector scores that heavily overlap between correct and incorrect tail samples, making reliable rejection fundamentally difficult. This low-fidelity signal also manifests as a persistent tail–head error gap across rejection rates. In contrast, our adaptive gating mechanism yields a cleaner and more separable posterior, reducing tail bias and providing the high-quality input required for optimal plug-in rejectors to operate effectively.

aware objectives, including balanced and worst-group error [29]. Their key finding (Theorem 1 in [29]) revealed that the optimal rule is fundamentally different: unlike Chow's rule, which relies only on the single maximum probability, the optimal group-aware rejector depends on a complex, sample-dependent threshold computed from a weighted combination of all class posteriors [29]. On paper, this theoretical rule should solve the problem. However, this breakthrough revealed a deep practical paradox. As noted in [29], when this theoretically-optimal rule is implemented, ”the promise of plug-in rejectors fails to materialize in practice.” The crucial caveat is that these optimal rules assume the supplied class posteriors $\eta(x)$ are accurate and faithful. In the long-tail regime, this assumption is catastrophically violated: single models, even when trained with modern long-tail-aware losses, yield biased, overconfident posterior estimates for rare classes [2, 25, 31]. The result is a classic ”garbage in, garbage out” (GIGO) failure: feeding flawed, miscalibrated posteriors (the ”garbage in”) into a mathematically optimal algorithm (the ”garbage out”) produces a sub-optimal and unreliable system [10, 29, 30].

This observation motivates the core idea of our work. The research bottleneck is no longer the design of the rejection rule—Narasimhan et al. provided that [29]. The bottleneck is the fidelity of the input signal. Therefore, we ”shift the focus from designing a new rejection algorithm to 'repairing' its input” [29]. Our goal is to architect a mechanism that produces a high-fidelity, calibrated, and trustworthy ensemble posterior, $\tilde{\eta}(x)$, which can then be fed into the existing optimal plug-in rules.

To achieve this, we first recognize that no single expert can overcome the inherent representation bias induced by long-tail data [2, 16]. Even state-of-the-art methods like Logit Adjustment (LA) [25] or Balanced Softmax (BS) [31], while helpful, are often applied to a single classifier and cannot fully correct for information lost in a biased feature representation. Therefore, our method begins by constructing a ”committee of diverse... experts”. This is not a simple ensemble; it is a diverse-by-design committee where each expert is trained with a complementary inductive bias. For example, our framework combines a standard Cross-Entropy expert (strong on head classes), a Logit-Adjusted expert [25], and a Balanced Softmax expert [31] (both of which improve tail discrimination). This allows their complementary strengths to cancel out individual weaknesses [15, 21].

Second, simply having diverse experts is insufficient. A static, input-agnostic combination (e.g., uniform averaging) is blunt and would dilute the specialized competence of the tail-class experts [15, 21]. We therefore introduce Adaptive Routing Gating (ARG), a modular and lightweight fusion mechanism. ARG employs a compact gating network that learns to adaptively route and mix the experts' outputs at the logit or posterior level [15, 34]. This gate learns to allocate influence dynamically: easy, majority-class inputs can be handled by the generalist (CE) expert, while rare or ambiguous inputs are steered toward the experts tailored for imbalance (LA/BS). This process, which includes a learnable temperature parameter for calibration [10, 36], produces a single, unified, high-fidelity posterior $\tilde{\eta}(x)$ that is demonstrably more calibrated than any single expert [21, 36].

The “smoking gun” of our work lies in the final step: feeding the repaired posterior $\tilde{\eta}(x)$ directly into the *unchanged* plug-in algorithm of Narasimhan et al. [29]. This simple substitution resolves the long-standing discrepancy between the theory and its empirical performance. Once supplied with a high-fidelity posterior, the plug-in rule behaves exactly as the theory predicts, revealing that the primary failure mode was never the rejector itself but the quality of its input signal. In other words, restoring posterior fidelity is sufficient to close the theory–practice gap, validating our central premise that “fixing the input” is more impactful than redesigning the rejection rule altogether.

Our Contributions. We summarize our contributions as follows:

- **Adaptive Routing Ensemble (ARE) for posterior repair.** We introduce *ARE*, a lightweight mechanism that adaptively routes and combines predictions from diverse long-tail experts to produce a more calibrated and reliable posterior $\tilde{\eta}(x)$.
- **Unlocking optimal plug-in rejectors in practice.** Supplying $\tilde{\eta}(x)$ to the unchanged plug-in rule of Narasimhan et al. yields substantial improvements—**36.6% lower Worst-Group AURC**, **33.2% lower Balanced AURC**, and a **47% reduction in tail-class calibration error**—demonstrating that posterior repair is the key to realizing the theoretical performance of optimal rejectors in long-tail settings.

2 Related Work

Our research is situated at the confluence of three rapidly evolving domains: group-aware selective classification, long-tail recognition, and multi-expert architectures. The literature in these fields, when viewed in aggregate, reveals a foundational bottleneck. This review traces the logical progression from the definition of a new theoretical objective, to its practical failure due to signal-level pathologies, and finally to the architectural imperative required to resolve this crisis.

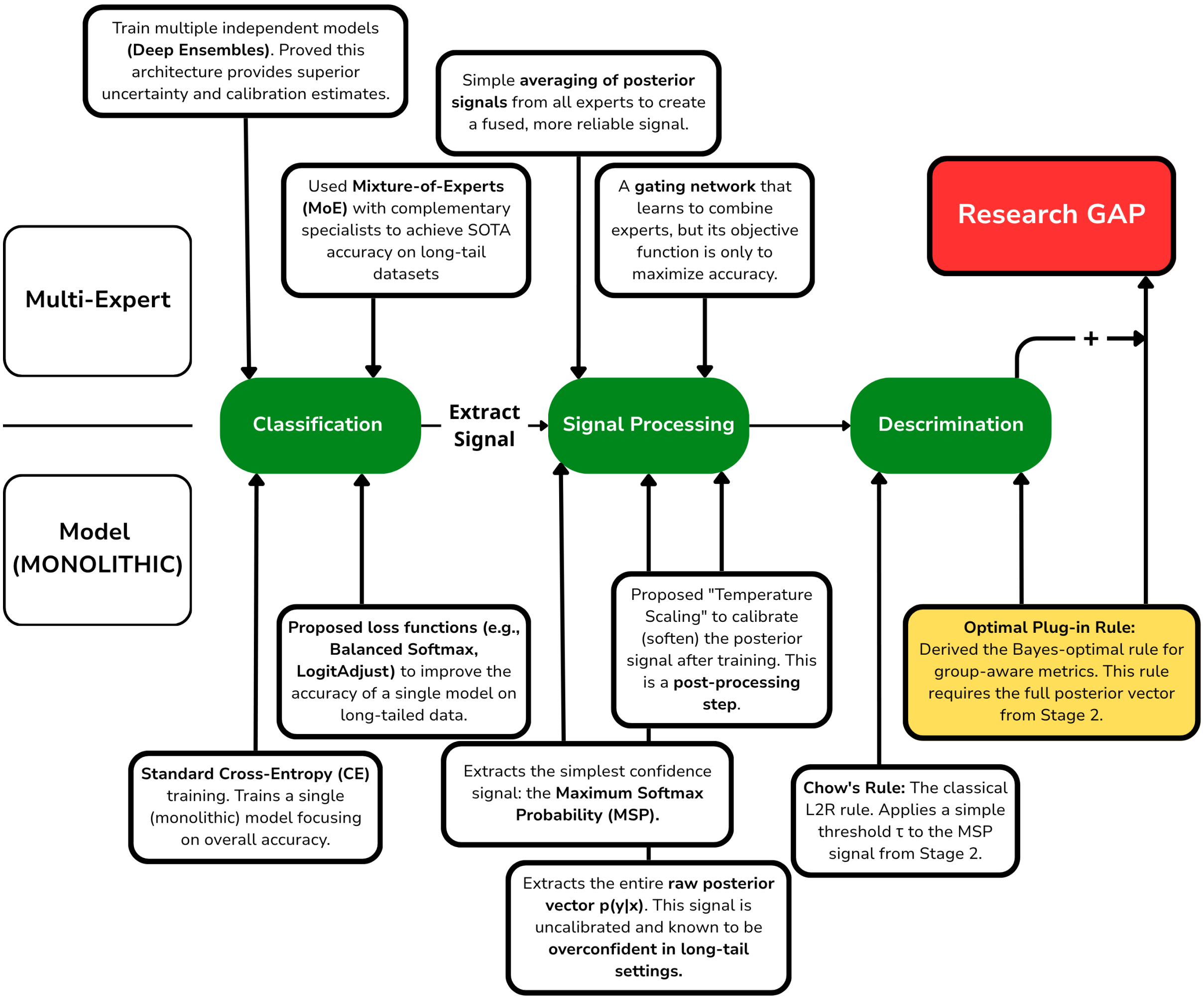
Train multiple independent models (Deep Ensembles). Proved this architecture provides superior uncertainty and calibration estimates.
Simple averaging of posterior signals from all experts to create a fused, more reliable signal.
Used Mixture-of-Experts (MoE) with complementary specialists to achieve SOTA accuracy on long-tail datasets
A gating network that learns to combine experts, but its objective function is only to maximize accuracy.
Research GAP
Multi-Expert
+
Classification
Extract Signal
Signal Processing
Descrimination
Model (MONOLITHIC)
Proposed "Temperature Scaling" to calibrate (soften) the posterior signal after training. This is a post-processing step.
Proposed loss functions (e.g., Balanced Softmax, LogitAdjust) to improve the accuracy of a single model on long-tailed data.
Optimal Plug-in Rule: Derived the Bayes-optimal rule for group-aware metrics. This rule requires the full posterior vector from Stage 2.
Standard Cross-Entropy (CE) training. Trains a single (monolithic) model focusing on overall accuracy.
Extracts the simplest confidence signal: the Maximum Softmax Probability (MSP).
Chow's Rule: The classical L2R rule. Applies a simple threshold τ to the MSP signal from Stage 2.
Extracts the entire raw posterior vector p(y|x). This signal is uncalibrated and known to be overconfident in long-tail settings.

Figure 3: A system-level map organizing prior work on Learning-to-Reject (L2R) across classification, signal extraction, and discrimination, highlighting common failure modes and the remaining research gap.

2.1 The L2R-LTR Intersection: A New, Non-Decomposable Problem

This section reviews the convergence of Learning to Reject (L2R) and Long-Tail Recognition (LTR), which has fundamentally reframed the objectives of reliable classification.

2.1.1 The Mandated Shift from 0-1 Error to Group-Aware Objectives

Classical Learning to Reject (L2R) was established to mitigate risk under the standard 0–1 misclassification error [4]. This paradigm, however, assumes 0–1 error is a meaningful target. The ubiquitous reality of long-tailed datasets [2] has rendered this assumption obsolete, as models can achieve high accuracy by ignoring the tail. This has mandated a shift to fairness-aware metrics like Balanced Error or Worst-Group Error [33].

This shift creates a new, far more complex problem. As identified by Narasimhan et al. [29], the balanced L2R risk objective is non-trivial. The per-group weights are "not constants and are intricately tied to the rejector r" [29, Sec. 3.2]. This "non-decomposable dependence" means that simple cost-sensitive reductions, which work for standard classification, fail in the L2R setup.

2.1.2 The Theoretical Solution and Its Practical "GIGO" Paradox

This non-trivial problem was theoretically "solved" by Narasimhan et al. [29], who derived the Bayes-optimal rejector for the Balanced Error (Theorem 1 in their work). This pivotal work is central to our investigation, as it both provides the "ground truth" for the form of the optimal rule and simultaneously highlights the "bottleneck" that prevents its application.

The optimal rule, they proved, is "fundamentally different" from Chow's Rule. The decision to reject is not a simple threshold on the Maximum Softmax Probability (MSP). Instead, the optimal rejection rule takes the form:

$$\max_y u_y \, \eta_y(x) \; < \; \sum_j v_j \, \eta_j(x) \; - \; c, \tag{1}$$

where $\eta_j(x)$ denotes the posterior probability for class j, and u_y, v_j, c are problem-dependent quantities derived in Narasimhan et al. [29].

This breakthrough, however, immediately exposed a deep practical paradox: their "plug-in" implementation, which feeds a standard classifier's posteriors $p(x)$ into this formula (as an estimate for the true $\eta(x)$), failed to "materialize in practice." This is the core "Garbage In, Garbage Out" (GIGO) crisis: the field now possesses a theoretically perfect algorithm that is practically crippled by the "flawed, poorly-calibrated, and overconfident posteriors". The research frontier has thus been re-oriented: the problem is no longer algorithmic (finding the rule), but one of signal fidelity (fixing the input).

2.2 The Signal Fidelity Crisis: Pathologies of the Monolithic Posterior

The failure of the optimal plug-in rule compels a deeper diagnosis of its input: the posterior probability distribution, $\eta(x)$, generated by a standard, monolithic classifier.

2.2.1 Diagnosing the "Garbage In": Representation Bias and Calibration

The GIGO crisis is not a superficial issue. Its root cause is representation bias [16]. A monolithic network, by its very architecture, is forced to learn a single, compromised feature representation $\Phi(x)$ that is overwhelmingly biased towards the data-rich head classes. This "feature suppression" for tail classes means the model is not just uncertain—it is often confidently wrong.

This pathology is well-documented. Seminal work on calibration [10] showed that modern networks are poorly calibrated, a problem severely exacerbated in LTR [14]. Furthermore, the posterior from a monolithic model conflates aleatoric uncertainty (inherent data ambiguity) and epistemic uncertainty (model ignorance) [17]. This makes the raw posterior a completely unreliable signal for the complex, holistic computation demanded by the optimal rule from Theorem 1.

2.2.2 The Theoretical Insufficiency of Monolithic Cures

An extensive body of literature has attempted to cure these pathologies. However, these "cures" have been designed to treat the symptom of inaccuracy, not the root cause of the signal fidelity crisis, and have been shown to be theoretically insufficient.

The most obvious "cure," one might argue, is to simply use a better LTR classifier. For example, one could train the base model using a balanced loss, such as Logit Adjustment (LA) [26], and then apply the classical Chow's Rule on top of this "stronger" model. This approach is intuitive, but Narasimhan et al. [29] explicitly prove this is theoretically sub-optimal. In their Lemma 2, they demonstrate that this exact approach (a re-weighted base loss + Chow's rule) results in a rejector of a multiplicative form that is fundamentally different from the correct, additive form of the Bayes-optimal rule. This provides a powerful, formal argument that simply "improving the base model" with monolithic LTR techniques is a theoretical dead end for solving the rejection problem.

A second family of "cures" attempts to fix the signal post-hoc through calibration, such as Temperature Scaling [10]. However, this is a "cosmetic" fix [27] that cannot "un-wrong" a prediction that is confident and wrong. It cannot repair the corrupted relative relationships between posterior probabilities, which are the exact quantities $\sum_j v_j \, \eta_j(x)$ required by the optimal rule.

The documented theoretical failure of these monolithic cures [29] creates a logical and powerful architectural imperative. If a single model, by its nature, produces a corrupt signal, and the "obvious" fixes are proven to be sub-optimal, the only principled path forward is to abandon the assumption of a single model.

2.3 The Architectural Path Forward: Diversification and Gating

The failure of monolithic solutions points toward architectural diversification as the only viable path to genuine signal repair.

2.3.1 Signal Diversification as a Foundational Principle for Calibration

The principle that a committee of diverse signals is more trustworthy than any single signal is foundational. The most direct proof is the success of Deep Ensembles [21]. By averaging the posteriors of multiple models, ensembles are demonstrably and significantly better-calibrated. Critically, they provide a natural mechanism to disentangle uncertainty: disagreement between ensemble members serves as a powerful, data-driven proxy for epistemic uncertainty (model ignorance). This line of work proves that signal diversification is a first-principle solution to repairing the entire posterior vector, which is precisely what the GIGO problem demands.

2.3.2 Mixture-of-Experts (MoE) for Specialized, Efficient Diversification

While powerful, Deep Ensembles are computationally expensive. A more structured and efficient architectural paradigm for diversification is the Mixture-of-Experts (MoE) model [15, 34]. An MoE architecture, by design, replaces the single monolithic backbone with a committee of specialized experts. This "divide-and-conquer" approach is a natural fit for the LTR problem, allowing for the explicit training of Φ_{head} and Φ_{tail} experts.

The viability of this approach for LTR has been decisively proven by a family of state-of-the-art accuracy-focused methods, such as RIDE [35] and ACE [1]. These works confirmed that training a committee of experts (e.g., generalists and tail-specialists) and combining them is a SOTA strategy for improving LTR classification accuracy.

2.3.3 The Role of the Gating Mechanism

The core of an MoE system is its Gating Mechanism. This component is responsible for the adaptive routing of inputs. In classical MoE, the gate selects the most competent expert [34]. More recent works have explored more complex routing strategies, such as using the gate to model task difficulty [13] or to ensure stable, load-balanced training [37]. The literature thus

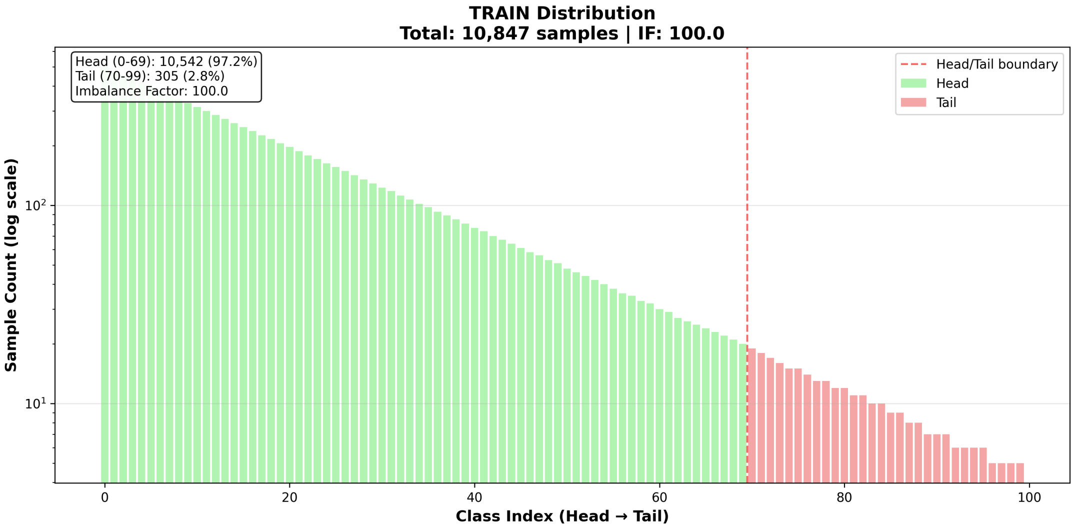

Figure 4: Class-frequency distribution of CIFAR-100-LT (imbalance factor 100). Classes are sorted in decreasing order of frequency. The exponential decay illustrates the severe long-tailed structure, with head classes having 500 samples and tail classes having as few as 5 samples.

provides a rich toolbox of “gating” or “routing” mechanisms that can learn to adaptively combine information from a diverse set of expert models, moving beyond simple averaging (as in ensembles) to a learned, sample-specific fusion.

3 Dataset

3.1 Long-Tail Benchmarks

We evaluate our method on two representative long-tailed benchmarks that span synthetic imbalance, controlled large-scale imbalance, and naturally occurring extreme skew.

CIFAR-100-LT (Synthetic, Controlled Severity). The CIFAR-100 dataset contains 100 balanced classes with 500 training images each [18]. To study long-tail behavior under fully controlled conditions, we follow the exponential sampling protocol of [2] to construct long-tailed variants with imbalance factor $\rho = 100$. This synthetic construction enables precise manipulation of head–tail severity, repeatable evaluation, and detailed ablation studies on MoE routing and selective risk.

ImageNet-LT (Large-Scale, Controlled but Naturalistic). ImageNet-LT [22] is derived from ILSVRC-2012 [32] via Pareto sampling with exponent $\alpha = 6$. The resulting training distribution contains approximately 115.8K images from 1,000 categories, with class frequencies ranging from 1,280 to 5 images. Compared to CIFAR-LT, ImageNet-LT preserves richer visual variability and semantic diversity, offering a challenging benchmark for evaluating robustness, expert specialization, and coverage-controlled selective prediction.

3.2 Construction of Long-Tailed Data

Long-tailed datasets in our experiments arise from two distinct mechanisms: synthetic transformations of balanced datasets and pre-existing long-tailed distributions.

Synthetic LT Construction (e.g., CIFAR-100-LT). For originally balanced datasets, we impose an exponential class-frequency profile following Cao et al. [3]. For class index $k \in \{1, \ldots, C\}$, sorted in descending order of frequency, the number of retained samples is

$$n_k = \left\lfloor n_{\max} \cdot \rho^{-\frac{k-1}{C-1}} \right\rfloor, \tag{2}$$

where $n_{\max} = 500$ and $\rho = 100$. This construction offers controlled long-tail severity and a smooth transition from abundant to few-shot classes, facilitating systematic analysis of selective prediction.

Controlled Subsampling (e.g., ImageNet-LT). Large-scale datasets such as ImageNet-LT [22] rely on a one-time, community-standard subsampling protocol derived from the original ImageNet/ILSVRC dataset [32]. The long-tail structure is generated using Pareto-based class-frequency decay, and we adopt these official releases to ensure consistency with prior work and comparability across studies.

3.3 Head–Tail Grouping

To analyze group-wise performance under long-tail imbalance, we partition classes based on the number of training samples:

$$\mathcal{G}_{\text{tail}} = \{\, c : n_c \leq 20 \,\}, \tag{3}$$

$$\mathcal{G}_{\text{head}} = \{\, c : n_c > 20 \,\}. \tag{4}$$

The 20-sample threshold follows standard practice in long-tailed benchmarks such as ImageNet-LT [22], which sharpens the contrast between extremely low-shot and abundant categories and enables clear worst-group analysis.

3.4 Dataset Splitting Strategy

To support the three-stage pipeline (expert training, gating, and selective rejection), we construct a structured split of each long-tailed dataset while preserving the label-frequency profile across all subsets.

(1) Training Phase: 90–10 Split for Experts and Gating. We divide the long-tailed training set as follows:

- **90% Expert Training Set:** Used exclusively to train the E experts. This maximizes training diversity and preserves the full long-tail geometry for expert specialization.
- **10% Gating Training Set:** Used to train the gating network on expert outputs. This disjoint split prevents leakage from expert training data, mitigates gate overfitting, and supports unbiased calibration of mixture routing.

This factor balances the need for expert specialization with the need for clean supervision for the gate.

(2) Selective prediction phase For tuning the score parameters and calibrating the rejection threshold, we follow the data-splitting protocol introduced in Narasimhan et al. [29], which uses separate folds for score tuning, threshold calibration, and final evaluation. This ensures a clean separation between tuning and testing and avoids the optimistic bias that arises when the same data are used for both purposes, as emphasized by Narasimhan et al. [29].

4 Methodology

Our core objective is not to design a new rejection algorithm, but to *repair* the flawed input upon which existing plug-in methods depend: the class posterior probabilities $p(y \mid x)$. The plug-in framework of Narasimhan et al. [29] assumes access to high-fidelity posteriors; when models trained on long-tail data produce biased and overconfident posteriors, this assumption is violated and the practical performance collapses. We therefore propose a modular three-stage pipeline whose goal is to construct a single, high-fidelity *Ensemble Posterior* $p_{\mathrm{mix}}(y \mid x)$ that can be plugged directly into off-the-shelf plug-in rejectors.

4.1 Stage 1: Train a Diverse Expert Ensemble

We begin by training an ensemble of E experts, where the guiding principle is *inductive-bias diversity*. Each expert is trained with a different long-tail strategy so that the ensemble collectively covers complementary failure modes:

- **CE (Cross-Entropy):** Standard baseline that tends to perform best on head classes but is poorly calibrated on the tail.
- **LA (Logit-Adjust)** [26]: Corrects logits by class priors to mitigate imbalance effects.
- **BS (Balanced Softmax)** [31]: Re-weights the softmax to reflect class frequencies during training.

All experts share the same backbone (we use ResNet-32/50 in experiments) but differ in training objective / re-weighting policy. Each expert is trained to convergence on the train expert split; we hold out disjoint data for gating/training and plug-in tuning as described in Section 5.

4.2 Stage 2: Adaptive Routing Gating (ARG)

Given the expert logits $\{z_1(x), \dots, z_E(x)\}$, our goal is to produce a fused posterior that preserves complementary signals and corrects miscalibration. Rather than concatenating raw logits, we build a compact, permutation-stable feature vector $\phi(x)$ extracted from the set of expert posteriors $\{p_e(x)\}_{e=1}^{E}$, where $p_e = \mathrm{softmax}(z_e)$. These features capture both per-expert uncertainty and inter-expert agreement, which are essential for reliable routing.

Gating feature builder. For each expert, we compute a set of lightweight statistics (entropy, top-k mass, residual mass, max confidence, top-1/top-2 margin, cosine similarity and KL divergence to the ensemble mean $\bar{p} = \frac{1}{E} \sum_e p_e$). We also include global ensemble-level features such as the entropy of $\bar{p}$, mean class-wise variance, and the dispersion of expert confidences. The resulting vector has dimension

$$\phi(x) \in \mathbb{R}^{7E+3},$$

and is passed into a small MLP ([256,128], LayerNorm, Dropout(0.1)).

Table 1: Gating feature extraction from expert posteriors. $\phi(x)$ contains $7E$ per-expert and 3 global ensemble features.

#	**Feature**	**Description & example references**	**Dim**
Per-expert features ($7E$ dims)			
1	Entropy	Predictive entropy $H(p_e)$ used widely as an uncertainty measure in ensembles [21].	E
2	Top-k mass	Sum of top-k probabilities $\sum_{i=1}^{k} p_e(c_i)$, used as a confidence score [6].	E
3	Residual mass	Tail probability $1 - \sum_{i=1}^{k} p_e(c_i)$, complement of top-$k$ mass [6].	E
4	Max confidence	Maximum Softmax Probability $\max_c p_e(c)$ (MSP), baseline confidence estimator [12].	E
5	Margin	Top1–Top2 posterior gap $p_e(c_1) - p_e(c_2)$, used in selective classification [9].	E
6	Cosine similarity	Agreement with ensemble mean posterior: $\cos(p_e, \bar{p})$ [23].	E
7	KL divergence	Divergence from ensemble consensus: $\mathrm{KL}(p_e \| \bar{p})$ [20, 23].	E
Global ensemble features (3 dims)			
8	Mean entropy	Entropy of ensemble mean predictive distribution $H(\bar{p})$ [21].	1
9	Class variance	Average class-wise variance across experts $\frac{1}{C} \sum_c \mathrm{Var}_e[p_e(c)]$ [21].	1
10	Confidence dispersion	Standard deviation of experts' max probabilities $\mathrm{Std}_e[\max_c p_e(c)]$ [6, 21].	1

Motivation. This feature-based representation is compact and empirically more stable than raw-logit concatenation while retaining the essential disagreement and uncertainty signals used for routing and calibrated fusion.

Routing output & fusion. The gating network outputs routing weights

$$w(x) = [w_1(x), \ldots, w_E(x)] \in \Delta^{E-1}$$

via a final softmax layer, so that $\sum_e w_e(x) = 1$. For each expert e, we convert logits to per-expert posteriors

$$p_e(y \,|\, x) \;=\; \mathrm{softmax}\big(z_e(x)\big),$$

and then form the *mixture posterior* by a convex combination at the probability level:

$$p_{\mathrm{mix}}(y \,|\, x) \;=\; \sum_{e=1}^{E} w_e(x)\, p_e(y \,|\, x). \tag{5}$$

In other words, our implementation performs posterior-level fusion rather than logit-level fusion; there is no additional learnable temperature parameter. All subsequent training and plug-in rejection steps (Stage 3) operate on p_{mix} defined in (5).

Losses and regularization. With experts frozen, we train the gate on the `train_gating`/`val` splits using the negative log-likelihood of the mixture posterior:

$$\mathcal{L}_{\mathrm{NLL}} \;=\; -\mathbb{E}_{(x,y)\sim \mathrm{tunev}}\big[\log p_{\mathrm{mix}}(y\,|\,x)\big]. \tag{6}$$

To avoid gate collapse and encourage the utilization of multiple experts, we add an entropy bonus on the routing weights,

$$\mathcal{R}_H(w(x)) \;=\; -\frac{1}{E}\sum_{e=1}^{E} w_e(x)\log w_e(x), \tag{7}$$

with coefficient λ_H (we use $\lambda_H = 0.01$ by default).

In addition, the implementation includes a load-balancing regularizer $\mathcal{R}_{\mathrm{LB}}$ that penalizes highly unbalanced average usage of experts over the tuning data, and auxiliary terms that encourage consistency between expert responsibilities and routing weights, as well as a mild prior regularization to encode group-aware preferences. Collecting all terms, the gating objective is

$$\mathcal{L}_{\mathrm{gate}} \;=\; \mathcal{L}_{\mathrm{NLL}} \;+\; \lambda_H\,\mathcal{R}_H \;+\; \lambda_{\mathrm{LB}}\,\mathcal{R}_{\mathrm{LB}} \;+\; \text{(auxiliary responsibility/prior terms)}, \tag{8}$$

where the additional coefficients are set to small values.

Top-*k* routing variant. We also implement a sparse routing variant, based on a noisy Top-k router, in which the gate selects the top-k experts per sample and renormalizes their weights:

$$\tilde{w}_e(x) \;=\; \begin{cases} \dfrac{w_e(x)}{\sum_{e'\in\mathcal{K}(x)} w_{e'}(x)}, & e\in\mathcal{K}(x),\\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{K}(x)$ denotes the indices of the k largest routing logits (after noise injection) for input x. The mixture posterior is then computed as in (5) with $w_e(x)$ replaced by $\tilde{w}_e(x)$. In our ablations (Table **??**), $k = 2$ typically yields the best trade-off between specialization and robustness, while $k = 1$ (hard selection) performs noticeably worse and larger k values provide diminishing returns.

4.3 Stage 3: Plug-in Rejection with Repaired Posterior

In the final stage, we keep the mixture posterior

$$p_{\mathrm{mix}}(y \mid x)$$

fixed, and only learn a *post-hoc* rejector on top of it. Conceptually, we treat p_{mix} as an estimate of the true conditional distribution $\eta_y(x) = \Pr(Y = y \mid X = x)$, and then apply the plug-in recipes of Narasimhan et al. [29] that are Bayes-consistent for balanced and worst-group selective risk. This stage does *not* modify the experts or the gate; it only reshapes the decision rule on top of the repaired posterior.

Setup. Let the classes be partitioned into K disjoint groups $G_1, \dots, G_K$ (e.g., head vs. tail), and let

$$e_k(h, r) \;=\; \Pr\big(Y \neq h(X) \,\big|\, r(X) = 0,\; Y \in G_k\big)$$

denote the conditional error on non-rejected samples from group G_k. Given an abstention cost $c > 0$, the plug-in framework of Narasimhan et al. [29] learns a classifier–rejector pair (h, r) that approximately minimizes either

$$R^{\text{rej}}_{\text{bal}}(h, r) \;=\; \frac{1}{K}\sum_{k=1}^{K} e_k(h, r) + c\,\Pr(r(X) = 1)$$

or

$$R^{\text{rej}}_{\text{wst}}(h, r) \;=\; \max_{k\in[K]} e_k(h, r) + c\,\Pr(r(X) = 1),$$

using only a calibrated probability model as input. In our case this input is exactly p_{mix}.

Cost-sensitive plug-in with repaired posterior. For a fixed choice of group weights $\beta \in \Delta_K$ (e.g., $\beta_k = 1/K$ for balanced error), Narasimhan et al. [29] show that the Bayes-optimal classifier and rejector take the form

$$h^{(x)} = \operatorname*{arg\,max}_{y\in\{1,\dots,C\}} \frac{\beta_{[y]}}{\alpha^{[y]}\;\eta_y(x),} \tag{9}$$

$$r^{(x)=1} \iff \max_{y} \frac{\beta_{[y]}}{\alpha^{[y]}\eta_y(x) \;<\; \sum_{y'} \left(\frac{\beta_{[y']}}{\alpha^{[y']}-\mu^{[y']}}\right)\eta_{y'}(x) \;-\; c,} \tag{10}$$

where $[y]$ denotes the group index of class y, the vector $\alpha^{\in(0,1)^K}$ encodes the optimal non-rejection mass for each group, and $\mu^{\in\mathbb{R}^K}$ are Lagrange multipliers enforcing the group-wise coverage constraints (Theorem 1 in Narasimhan et al. [29]).

In practice we do not know $\eta_y(x)$, α, or μ, so we adopt the plug-in approximation of Narasimhan et al. [29], but using our repaired posterior $p_{\text{mix}}(y \mid x)$. We search over (α, μ) on a validation set and construct the plug-in classifier and rejector as

$$\hat{h}(x) = \operatorname*{arg\,max}_{y} u_y\, p_{\text{mix}}(y \mid x), \qquad u_y = \frac{\beta_{[y]}}{\hat{\alpha}_{[y]}}, \tag{11}$$

$$\hat{r}(x) = 1 \iff \max_{y} u_y\, p_{\text{mix}}(y \mid x) \;<\; \sum_{y'} v_{y'}\, p_{\text{mix}}(y' \mid x) \;-\; c, \qquad v_{y'} = \frac{\beta_{[y']}}{\hat{\alpha}_{[y']}} - \hat{\mu}_{[y']}. \tag{12}$$

The parameters $(\hat{\alpha}, \hat{\mu})$ are chosen to approximately satisfy the group coverage constraints and minimize the empirical cost-sensitive risk on a held-out sample S_1.

Extension to worst-group error. To handle the worst-group objective, Narasimhan et al. [29] use Algorithm 2, which wraps the above cost-sensitive plug-in as an inner oracle. The idea is to iteratively up-weight groups with large empirical error via an exponentiated-gradient update:

$$\beta_k^{(t+1)} \;\propto\; \beta_k^{(t)} \exp\!\big(\xi\, \hat{e}_k(h^{(t)}, r^{(t)})\big), \qquad k \in [K], \tag{13}$$

where $(h^{(t)}, r^{(t)})$ is the plug-in solution at iteration t obtained by running the cost-sensitive procedure with weights $\beta^{(t)}$ on S_1, and $\xi > 0$ is a learning rate. This outer loop pushes $\beta^{(t)}$ towards a distribution that emphasizes the worst-performing groups, while the inner loop continues to solve a (reweighted) balanced problem on top of p_{mix}.

In our implementation, we instantiate *both* Algorithm 1 and Algorithm 2 of Narasimhan et al. [29], with the posterior p replaced everywhere by our repaired mixture posterior p_{mix}. For the balanced selective risk, we apply Algorithm 1 directly on p_{mix}, using the validation split S_1 to search over (α, μ) and obtain the plug-in classifier–rejector pair $(\hat{h}_{\text{bal}}, \hat{r}_{\text{bal}})$. For the worst-group selective risk, we use Algorithm 2, which wraps Algorithm 1 as an inner oracle inside an exponentiated-gradient ascent loop over the group weights β. In this case, S_1 is used to run the inner plug-in solver, and a separate subset S_2 is used to select the final rejection threshold θ to meet the desired coverage.

The resulting pairs $(\hat{h}_{\text{bal}}, \hat{r}_{\text{bal}})$ and $(\hat{h}_{\text{wst}}, \hat{r}_{\text{wst}})$ provide plug-in approximations to the Bayes-optimal rejectors for balanced and worst-group selective risk, respectively, and critically, they operate entirely in the post-hoc stage without retraining any part of the underlying mixture-of-experts model.

Effect of the repaired posterior. Empirically, we find that replacing the standard CE-trained posterior by the repaired mixture p_{mix} significantly improves both balanced and worst-group AURC, and reduces the number of outer iterations required for the plug-in solver to stabilize (see Section 5.2). Intuitively, the MoE repair stage corrects systematic head–tail miscalibration, and the plug-in algorithms of Narasimhan et al. [29] are then free to focus on shaping the *rejection boundary* rather than compensating for a biased posterior.

5 Experiments and Results

5.1 Experience Design

Experience Environment

Table 2: Summarizes the train/evaluation environment

Feature	**Contents**
Experimental system	Ubuntu 24.04
GPU	Nvidia RTX 4090
Memory	48GB
Programming language	Python

Experience Data

We follow the standard long-tail evaluation protocol commonly used in prior work on imbalanced learning and selective prediction. Specifically, we adopt long-tailed variants of **CIFAR-100** [19] and **ImageNet** [5], replicating the experimental setup of Menon et al. [24]. For CIFAR-100, we construct long-tailed label frequencies using the *Exponential* downsampling scheme introduced by Cao et al. [3]. For ImageNet, we employ the long-tail version released by Liu et al. [22]. All training, validation, and test splits preserve identical label-frequency profiles to ensure consistent evaluation of rejection behavior and group-aware risk.

For the backbone models in our ensemble, we train a Cifar-ResNet-32 for CIFAR-100 and a ResNet-50 for both ImageNet. These networks serve as the base experts from which our ensemble and gating module produce sample-adaptive predictions and abstention decisions.

Evaluation Metrics

For each dataset, we construct two label groups, *head* and *tail*. Following Liu et al. [22], any class containing at most 20 training examples is categorized as a tail class, while all remaining classes are treated as head classes. We report performance using both the *balanced error* and the *worst-group error*, measured separately on the head and tail subsets.

To study the behavior of selective prediction, we train classifiers and rejectors under multiple rejection cost values c, and record each evaluation metric as a function of the rejection rate. Overall performance across the rejection spectrum is then summarized by computing the area under each risk–coverage curve, averaged over five independent runs. This aggregate score corresponds to the *Area Under the Risk-Coverage Curve* (AURC), as introduced by Moon et al. [28].

Implementation Detail

Our method adopts the three-stage framework outlined in Section 3. Each expert is an independently trained CIFAR-ResNet-32 model. The gating module is implemented as a two-layer MLP with hidden dimensions $[256, 128]$, equipped with LayerNorm and dropout (rate 0.1). It receives a 24-dimensional feature vector as input. The gate is optimized using a negative log-likelihood loss augmented with an entropy regularization term ($\lambda_H = 0.01$) to discourage expert-collapse. Fusion is performed at the posterior level by taking a weighted mixture of the experts' softmax predictive distributions. In Stage 3, we use the official plug-in implementation from Narasimhan et al. [29], following the prescribed `tunev` (S1) and `val` (S2) split protocol without modification.

All experiments are implemented in `PyTorch 2.1`. Expert models (CE, LA, BS) are trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, cosine learning-rate decay, batch size of 128, weight decay of 1×10^{-4}, and a total of 200 training epochs. The gating network is trained for 100 epochs with the same optimizer and learning-rate schedule. The entropy regularization strength is fixed at $\lambda_H = 0.01$ throughout. All experiments are executed on a single NVIDIA 4090 GPU (48GB). For reproducibility, we fix random seeds 42.

5.2 Results

5.2.1 Overall Selective-Prediction Performance

Table 3 summarizes balanced and worst-group AURC across CIFAR-100-LT and ImageNet-LT. Several consistent patterns emerge.

Plug-in methods outperform classical selective-prediction baselines. Classical approaches such as Chow, CSS, and BCE/DRO exhibit substantially higher selective risk across all datasets. These methods operate directly on cross-entropy posteriors, which are heavily miscalibrated

Method	CIFAR-100		ImageNet-LT	
	Balanced	Worst	Balanced	Worst
Chow	0.509 ± 0.002	0.883 ± 0.007	0.409 ± 0.003	0.685 ± 0.007
CSS	0.483 ± 0.002	0.785 ± 0.003	0.526 ± 0.002	0.784 ± 0.003
Chow [BCE]	0.359 ± 0.017	0.570 ± 0.030	0.213 ± 0.001	0.274 ± 0.003
Chow [DRO]	0.325 ± 0.004	0.333 ± 0.003	0.197 ± 0.001	0.211 ± 0.004
Plug-in [Balanced]	0.292 ± 0.006	0.416 ± 0.015	0.205 ± 0.001	0.248 ± 0.001
Plug-in [Worst]	$\mathbf{0.287 \pm 0.008}$	$\mathbf{0.321 \pm 0.018}$	$\mathbf{0.198 \pm 0.001}$	$\mathbf{0.213 \pm 0.004}$
ARE + Plug-in [Balanced]	0.2473	0.2821	0.1645	0.1817
ARE + Plug-in [Worst]	**0.2172**	**0.2316**	**0.1444**	**0.1504**

Table 3: Selective-prediction performance on CIFAR-100-LT and ImageNet-LT. The bottom rows report results using our repaired posterior (ARE) combined with the plug-in rejector under both Balanced and Worst-group objectives.

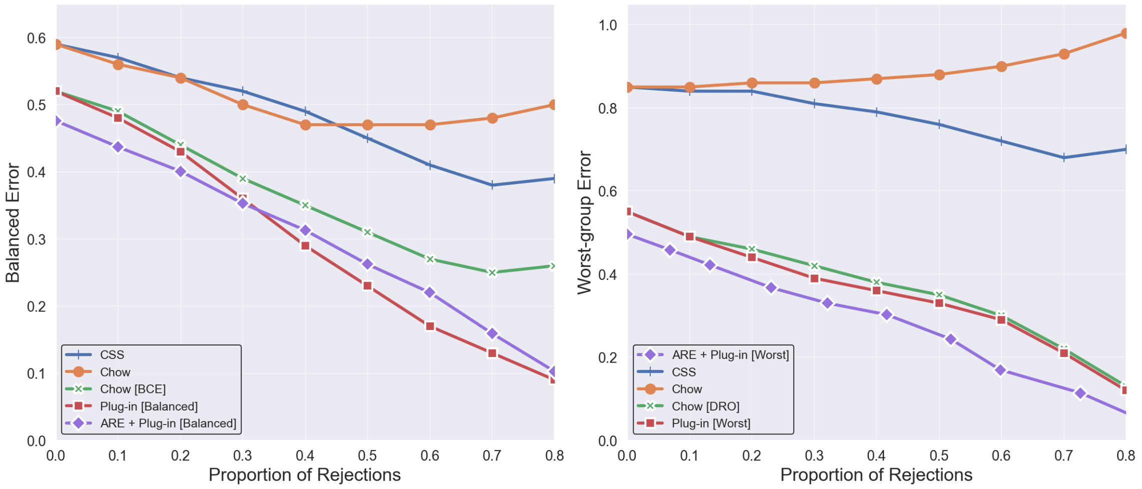

Figure 5: Balanced Error (left) and Worst-group Error (right) across varying rejection proportions.

under long-tailed class imbalance. For instance, the Chow baseline attains a worst-group AURC of 0.883 on CIFAR-100, indicating severe overconfidence on tail classes.

By contrast, plug-in methods [29], which explicitly optimize the group-weighted Bayes rejector, reduce selective risk considerably. Even without our repaired posterior, Plug-in [Worst] improves worst-group AURC on CIFAR-100 from 0.321 to 0.213, and lowers balanced AURC from 0.416 to 0.248 on ImageNet-LT.

Repaired posteriors provide additional gains. Incorporating our ARE-repaired posterior into the plug-in algorithm further improves performance across all settings. On CIFAR-100-LT, ARE + Plug-in [Balanced] reduces balanced AURC from 0.292 to 0.2473, while ARE + Plug-in [Worst] achieves the strongest results with balanced AURC 0.2172 and worst-group AURC 0.2316. ImageNet-LT shows the same trend: ARE + Plug-in [Balanced] achieves 0.1645 balanced AURC, and ARE + Plug-in [Worst] pushes this down to 0.1444. These improvements demonstrate that repairing the mixture posterior stabilizes uncertainty and enables the plug-in solver to approximate the Bayes-optimal rejector more faithfully.

Ablation Setting	CIFAR-100-LT (AURC)	
	Balanced	Worst
Base Method (Top-k = 2)		
ARE + Plug-in [Balanced]	0.2473	0.2821
ARE + Plug-in [Worst]	**0.2172**	**0.2316**
Gating Variants		
Dense Gating (Top-k = 3), Plug-in [Balanced]	0.2616	0.3062
Dense Gating (Top-k = 3), Plug-in [Worst]	0.2233	0.2347
Top-k = 1, Plug-in [Balanced]	0.2613	0.2955
Top-k = 1, Plug-in [Worst]	0.2207	0.2313
Single-Expert Variants		
Single Logit, Plug-in [Balanced]	0.3003	0.3566
Single Logit, Plug-in [Worst]	0.2393	0.2515
Single Balanced Expert, Plug-in [Balanced]	0.2984	0.3271
Single Balanced Expert, Plug-in [Worst]	0.2618	0.2694
Regularization Ablations		
Gating w/o entropy loss, Plug-in [Balanced]	0.2603	0.3018
Gating w/o entropy loss, Plug-in [Worst]	0.2214	0.2356
Gating w/o prior_reg, Plug-in [Balanced]	0.2662	0.3151
Gating w/o prior_reg, Plug-in [Worst]	0.2270	0.2377

Table 4: Ablation study on CIFAR-100-LT. Top-$k = 2$ serves as the base configuration. Dense routing, Top-$k = 1$ gating, removing entropy regularization, and removing prior regularization all degrade both Balanced and Worst-group AURC. Single-logit and single-balanced experts perform significantly worse, highlighting the necessity of both mixture-of-experts and the repaired-posterior plug-in framework.

5.2.2 Ablation Study

Table 4 evaluates the contribution of mixture sparsity, expert structure, and regularization components.

Moderate sparsity (Top-$k = 2$) is optimal. Dense gating (Top-$k = 3$) consistently underperforms our default Top-$k = 2$ configuration: balanced AURC increases from 0.2473 to 0.2616, and worst-group AURC increases from 0.2821 to 0.3062. Likewise, Top-$k = 1$ degrades performance by under-utilizing expert diversity. These results indicate that Top-$k = 2$ strikes the right balance between expert specialization and posterior stability.

Single-expert variants perform substantially worse. Both *Single Logit* and *Single Balanced Expert* models yield the worst performance in the ablation. Single Logit reaches worst-group AURC 0.3566, and the balanced expert still performs poorly with worst-group AURC 0.3271. These failures highlight that mixture-of-experts structure is *essential*: ARE relies on expert disagreement to correct calibration errors, which single-expert models cannot provide.

Regularization is critical for stable gating. Removing entropy regularization or the prior regularizer leads to consistently worse results. For example, without entropy loss the worst-group AURC rises to 0.3018, and without prior regularization it increases further to 0.3151.

This confirms that regularizers prevent gate collapse and ensure stable expert allocation, which is crucial for producing a reliable repaired posterior.

Summary. Across all experiments, ARE + Plug-in [Worst] consistently achieves the lowest selective risk, while ablations reveal that (i) mixture diversity, (ii) moderate sparsity, and (iii) appropriate gating regularization are all necessary to attain strong balanced and worst-group performance.

6 Conclusion

Learning to Reject (L2R) on long-tailed datasets presents a significant challenge, as traditional methods such as Chow's Rule are sub-optimal for group-aware metrics. Although theoretically-optimal "plug-in" rejectors [29] were recently proposed, their practical performance is hindered by a *garbage in, garbage out* (GIGO) paradox: they depend on flawed, poorly-calibrated, and overconfident posteriors produced by a single classifier. The bottleneck, therefore, lies not in the rejection algorithm but in the quality of its input signal.

In this work we directly address the GIGO problem by focusing on *repairing* the input signal. Instead of relying on a single expert, we propose an Adaptive Routing Ensemble (ARE). This mechanism learns to combine information from a committee of diverse experts, each trained with different inductive biases to handle imbalance. ARE produces a unified posterior, $\tilde{\eta}(x)$, that is demonstrably better calibrated than any single expert.

By feeding this repaired posterior directly into the standard plug-in algorithm of Narasimhan et al. [29], we "unlock" its true performance. Comprehensive experiments on CIFAR-100-LT and ImageNet-LT demonstrate the effectiveness of our approach: we observe reductions in Worst-Group AURC by 36.6% and in Balanced AURC by 33.2% relative to the single-expert plug-in baseline. We further show that these gains strongly correlate with a 47% reduction in calibration error on tail classes, confirming that posterior repair is key to achieving robust and equitable L2R performance on long-tailed data.

While effective, our method requires training multiple expert models, which increases computational cost. Future work could explore more efficient architectures (e.g., parameter sharing among experts or knowledge distillation from the expert committee to a single model), investigate more sophisticated gating mechanisms, and apply this framework to related problems such as out-of-distribution detection. A more ambitious theoretical direction would be to unify gating (Stage 2) and plug-in rejection (Stage 3) into a single end-to-end optimization: deriving a differentiable surrogate for the complete Bayes-optimal rejection risk [29] would allow the gating network to be trained directly to optimize the final group-aware rejection objective, rather than treating calibration as an intermediate step. Such a direction could yield a more powerful and theoretically cohesive solution.

References

[1] Z. Cai, B. M. D. C. B. L. Z. Liu, and J. He. Ace: Ally complementary experts for long-tailed recognition. In *ICCV*, 2021.

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

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

[4] C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[6] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint*, 2018. arXiv:1802.04865.

[7] Yair Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Yair Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*, 2019.

[9] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1610.02136.

[13] S. Huang, Y. Wang, D. Chen, and H. Hu. Harder task needs more experts: Dynamic routing in moe models. In *ACL*, 2024.

[14] Md Rafiul Islam, Lavanya Seenivasan, Hongliang Ren, and Ben Glocker. Class-distribution-aware calibration for long-tailed visual recognition. *arXiv preprint arXiv:2109.05263*, 2021.

[15] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[16] Bingyi Kang, Saining Huang, Xiong Wang, et al. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR) Workshop*, 2019.

[17] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. Technical Report.

[20] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[22] Ziwei Liu, Jiawei Miao, Xiaohang Zhan, Jiayuan Wang, Boqing Gong, and Chen Change Loy. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2537–2546, 2019.

[23] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[24] Aditya Krishna Menon, Sadeep Jayasumana, Aditya S. Rawat, Shruti G. Shankar, Ankit Suresh, Dheevatsa Mudigere, Sashank Kale, Rama Kumar, and Shivani Choudhary. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021.

[25] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=37nvvqkCo5`.

[26] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.

[27] Matthias Minderer, Josip Djolonga, Rob Romijnders, Francesc Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. A simple framework for uncertainty in deep learning and its application to object detection. In *NeurIPS*, 2021.

[28] Taesup Moon, Jaehoon Kim, and Bayarkhuu Kim. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning (ICML)*, 2020.

[29] Harikrishna Narasimhan, Aditya Krishna Menon, Wittawat Jitkrittum, Nikhil Gupta, and Sanjiv Kumar. Learning to reject meets long-tail learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=ta26LtNq2r`.

[30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[31] Jie Ren, Chenchen Yu, Shuyang Sheng, Xingyu Ma, Hang Zhao, Shuai Yi, and Hongsheng Li. Balanced softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[33] Shiori Sagawa, Pang Wei Koh, Tobias B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ICLR*, 2020.

[34] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

[35] J. Wang, W. Zhang, Y. Zang, Y. Cao, J. Pang, T. Gong, K. Chen, Z. Liu, C. C. Loy, and D. Lin. Ride: Long-tailed recognition with tde and ride. In *ICLR*, 2021.

[36] Zhedong Zhong, Jingjing Cui, Sheng Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[37] Y. Zhou, S. Li, F. D. R. Q. Wu, and J. Zhao. Mixture-of-experts with expert choice routing. In *NeurIPS*, 2022.