Optimal Query Allocation in Extractive QA with LLMs: A Learning-to-Defer Framework with Theoretical Guarantees

Yannis Montreuil[*,1,4,5] **Shu Heng Yeo**[*,1] **Axel Carlier**[2,4]

Lai Xing Ng[3,4] **Wei Tsang Ooi**[1,4]

[1]School of Computing, National University of Singapore, Singapore
[2]Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France
[3]Institute for Infocomm Research (A*STAR), Singapore
[4]IPAL, IRL 2955, Singapore
[5]CNRS@CREATE LTD, 1 Create Way, Singapore

Abstract

Large Language Models (LLMs) excel at generative language tasks but remain unreliable for structured prediction, particularly in extractive question answering (EQA), where success depends on precise span selection. These challenges are amplified in resource-constrained environments, such as mobile or embedded systems, where deploying high-capacity models is often infeasible. We propose a Learning-to-Defer framework that routes EQA queries across a pool of models with varying capabilities and costs to balance accuracy and efficiency. Our approach is grounded in statistical decision theory: we define a differentiable surrogate loss whose minimizer provably converges to the Bayes-optimal allocation policy. Experiments on SQuADv1, SQuADv2, and TriviaQA show that our method consistently improves the accuracy-efficiency trade-off relative to static baselines and prior routing heuristics. Overall, our framework provides a principled and scalable solution for EQA in both high-performance and on-device deployment settings.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong performance on a wide range of natural language processing tasks, including translation, summarization, and question answering (Touvron et al., 2023; Jiang et al., 2023; OpenAI et al., 2023). Their broad generalization ability, acquired through large-scale pretraining, enables fluent, context-aware responses across diverse inputs. However, these generative strengths do not necessarily transfer to high-precision structured prediction tasks. A notable example is *extractive question answering* (EQA), in which the model must identify an exact span from a given passage (Chen et al., 2017; Alqifari, 2019; Lan et al., 2020). In such settings, LLMs often produce plausible but unsupported answers, undermining reliability (Sadat et al., 2023).

Recent advances in compression and distillation have made it feasible to deploy lightweight LLM variants in resource-constrained environments, such as mobile or embedded systems, where memory, latency, and compute are limited (Sun et al., 2020; Merenda et al., 2020; Lin et al., 2024; Egashira et al., 2024). However, such models still struggle with fine-grained reasoning tasks like EQA, where faithful span selection is essential. One might instead deploy a specialized EQA model, but such models are inflexible for general queries, limiting their practical utility. This creates a dilemma: lightweight LLMs are versatile but error-prone on structured tasks, whereas EQA models are precise but narrow in scope. Deploying both on the same device is often infeasible under resource constraints. This trade-off motivates an adaptive hybrid approach that leverages the strengths of both model types without requiring them to be co-located.

To address this limitation, we propose a deferral-based strategy that adaptively routes queries between a lightweight on-device LLM and one or more off-device specialized EQA models. The lightweight model handles simple or low-risk inputs locally, while complex or uncertain cases are deferred to more accurate models. This balances the generality and efficiency of small LLMs with the precision of expert QA systems without requiring all models to reside on the same device (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020). We formalize this approach within a *Learning-to-Defer* (L2D) framework (Madras et al., 2018; Mozannar and Sontag, 2020; Verma et al., 2023; Mao et al., 2023a, 2024a; Montreuil et al., 2025b,a, 2026), where a learned policy assigns each input to the model offering the best accuracy-cost trade-off. Unlike prior heuristic or confidence-based routing, our policy minimizes a differentiable surrogate loss that provably converges to the Bayes-optimal allocation under mild conditions.

2 RELATED WORK

Model Cascades. Model cascades (Viola and Jones, 2001; Jitkrittum et al., 2023; Saberian and Vasconcelos, 2014) process a query through a sequence of models, forwarding it to the next stage only if a confidence-based criterion fails to meet a predefined threshold. These thresholds aim to balance predictive performance and computational cost. Although recent work has adapted cascades to LLMs (Kolawole et al., 2024; Yue et al., 2023), such methods are not tailored to the EQA setting. Moreover, cascade-based designs often struggle to accommodate heterogeneous models—e.g., mixing span-predicting EQA models with free-form generative LLMs—due to incompatible output formats (Varshney and Baral, 2022). As models are added to the cascade, inference latency increases and optimal predictions may be delayed. *Agreement-Based Cascading* (Narasimhan et al., 2025) uses ensemble agreement at each stage to decide whether to escalate the query. While this improves robustness, it still suffers from the limitations of sequential inference.

Query Routing. Query routing (Ding et al., 2024; Ong et al., 2025; Kag et al., 2023; Ding et al., 2022; Stojkovic et al., 2025; Chen et al., 2024) aims to improve efficiency by learning to dispatch each query among models, trading off pools or instances of fast, low-capacity models against a slower, higher-accuracy alternative (Chen et al., 2025). Routing decisions are guided by estimates of input difficulty or task-specific quality requirements, and are particularly relevant in edge and resource-constrained deployments where latency and energy consumption are critical (Qu et al., 2025). Recent work extends this paradigm beyond binary routing to allow selection among multiple candidate models (Lu et al., 2024; Ding et al., 2025), which better reflects real-world deployment choices. However, existing approaches either lack Bayes-consistent guarantees, do not address structured span prediction settings, or neglect practical deployment costs such as latency and token expenditure.

Structured Output Abstention. Structured Output Abstention (Garcia et al., 2018) allows a model to withhold predictions on components of a structured output while incurring a pre-specified abstention cost. While seemingly related to query allocation, it addresses a different decision problem. Specifically, abstention focuses on *when* a model should refrain from predicting. In contrast, query allocation determines which model or expert should produce the prediction, explicitly routing the input to another decision maker under a cost-quality trade-off.

Contributions. We advance EQA under resource constraints by unifying statistical decision theory with dynamic model selection. We propose a hybrid setting that combines the multi-expert flexibility of cascades with the direct-dispatch paradigm of routing. This generalization enables richer trade-offs by allowing queries to be dispatched directly across several experts with potentially different architectures.

(i) We introduce a new principled framework for cost-sensitive model selection in EQA, deriving a consistent, end-to-end trainable loss tailored to deferral-based architectures. (ii) We establish formal guarantees showing that our learned deferral policy provably converges to the Bayes-optimal allocation, offering theoretical insight into the limits of adaptive routing. (iii) Through extensive experiments on SQuADv1, SQuADv2, and TriviaQA, we demonstrate that our method consistently outperforms existing routing strategies and heuristic baselines.

3 PRELIMINARIES

Extractive QA. We consider EQA, where the answer a must be returned as a contiguous span in a context c given a question q. Let the random pair $(X, Y) \sim \mathcal{D}$ denote a draw from an unknown data-generating distribution. Throughout, upper-case symbols (e.g. X) denote random variables, while lower-case symbols (e.g. x) denote their fixed realizations. A specific instance is $x = (q, c) \in \mathcal{X}$. Its label is the span $y = (y^{\text{start}}, y^{\text{end}}) \in \mathcal{Y}$ with $0 \leq y^{\text{start}} \leq y^{\text{end}} < |c|$; hence $\mathcal{Y}$ factorises as $\mathcal{Y}^{\text{start}} \times \mathcal{Y}^{\text{end}}$. Following prior work (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020), we assume the start and end indices are conditionally independent given the input. All samples are i.i.d. according to $\mathcal{D}$ (Mohri et al., 2012). For later use, let $\mathcal{D}^i$ be the marginal of $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}^i$; i.e., $(X, Y^i) \sim \mathcal{D}^i$.

The EQA model is defined as a parametric function $g \in \mathcal{G}$, composed of a feature extractor $w \in \mathcal{W}$ and a span predictor $h \in \mathcal{H}$. The extractor maps inputs to latent representations via $w : \mathcal{X} \to \mathcal{T}$, with $t = w(x) \in \mathcal{T}$. These representations are scored by the classifier $h = (h^{\text{start}}, h^{\text{end}})$, where each head $h^i : \mathcal{T} \times \mathcal{Y}^i \to \mathbb{R}$ defines a position-wise scoring function. Predictions are then made according to the maximization rule $g^i(x) = \arg\max_{y \in \mathcal{Y}^i} h^i(w(x), y)$, where $w \in \mathcal{W}$ and $h^i \in \mathcal{H}$ denote a shared representation map and task-specific scoring function, respectively. The overall model is defined by composition as $g = h \circ w$, inducing the function class $\mathcal{G} = \{g \mid g(x) = h \circ w(x),\ w \in \mathcal{W},\ h \in \mathcal{H}\}$.

Model training typically minimizes the *joint 0–1 loss*, which counts the number of incorrect predictions across both span endpoints. This loss is defined as $\ell_{01}^{\text{joint}} : \mathcal{Y} \times \mathcal{Y} \to \{0, 1, 2\}$, where

$$\ell_{01}^{\text{joint}}(g(x), y) = \mathbf{1}\left[g^{\text{start}}(x) \neq y^{\text{start}}\right] + \mathbf{1}\left[g^{\text{end}}(x) \neq y^{\text{end}}\right].$$

Although non-differentiable, this loss provides an intuitive and interpretable measure of token-matching performance in EQA.

Bayes and $\mathcal{G}^i$-consistency. Let $i \in \{\text{start}, \text{end}\}$. The learning objective is to find a predictor $g^i \in \mathcal{G}^i$ that minimizes the expected 0–1 error,

$$\mathcal{E}_{\ell_{01}}(g^i) \;=\; \mathbb{E}_{(X,Y^i)\sim\mathcal{D}^i}\left[\ell_{01}\big(g^i(X), Y^i\big)\right].$$

The Bayes-optimal error is defined as

$$\mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) = \inf_{g^i \in \mathcal{G}^i} \mathcal{E}_{\ell_{01}}(g^i).$$

However, direct minimization is intractable due to the discontinuity and non-convexity of the multiclass 0–1 loss ℓ_{01} (Zhang, 2002; Steinwart, 2007; Awasthi et al., 2022).

To overcome this, we adopt a family of convex surrogate losses $\Phi_{01}^{\nu} : \mathcal{G}^i \times \mathcal{X} \times \mathcal{Y}^i \to \mathbb{R}_+$, parameterized by $\nu \geq 0$, which upper bound ℓ_{01}. This family subsumes common losses such as log-softmax (for $\nu = 1$) (Mohri et al., 2012) and the mean absolute error (for $\nu = 2$) (Ghosh et al., 2017), and is defined as:

$$\Phi_{01}^{\nu}(g^i, x, y^i) = \begin{cases} \dfrac{1}{1-\nu}\left(\Psi(g^i, x, y^i)^{1-\nu} - 1\right) & \text{if } \nu \neq 1, \\ \log \Psi(g^i, x, y^i) & \text{if } \nu = 1, \end{cases} \tag{1}$$

where $\Psi(g^i, x, y^i) = \sum_{y' \in \mathcal{Y}^i} \exp\big(g^i(x, y') - g^i(x, y^i)\big)$. The expected surrogate risk is given by $\mathcal{E}_{\Phi_{01}^{\nu}}(g^i) = \mathbb{E}_{(X,Y^i)\sim\mathcal{D}^i}[\Phi_{01}^{\nu}(g^i, X, Y^i)]$, with corresponding infimum $\mathcal{E}^*_{\Phi_{01}^{\nu}}(\mathcal{G}^i) = \inf_{g^i \in \mathcal{G}^i} \mathcal{E}_{\Phi_{01}^{\nu}}(g^i)$.

A central property of surrogate losses is *Bayes consistency*, which ensures that minimizing surrogate risk also minimizes true risk. Specifically, Φ_{01}^{ν} is Bayes-consistent with respect to ℓ_{01} if, for any sequence $\{g_k^i\}_{k\in\mathbb{N}} \subset \mathcal{G}^i$,

$$\begin{aligned} \mathcal{E}_{\Phi_{01}^{\nu}}(g_k^i) - \mathcal{E}^*_{\Phi_{01}^{\nu}}(\mathcal{G}^i_{\text{all}}) \xrightarrow{k\to\infty} 0 \quad \Longrightarrow \\ \mathcal{E}_{\ell_{01}}(g_k^i) - \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i_{\text{all}}) \xrightarrow{k\to\infty} 0. \end{aligned} \tag{2}$$

While this implication holds when $\mathcal{G}^i = \mathcal{G}^i_{\text{all}}$, it need not hold under hypothesis class restrictions such as $\mathcal{G}^i_{\text{lin}}$ or $\mathcal{G}^i_{\text{ReLU}}$ (Long and Servedio, 2013; Awasthi et al., 2022). To address this, Awasthi et al. (2022) introduce $\mathcal{G}^i$*-consistency* bounds, which quantify surrogate-to-true error transfer via a monotonic function $\Gamma : \mathbb{R}_+ \to \mathbb{R}_+$:

$$\begin{aligned} \mathcal{E}_{\Phi_{01}^{\nu}}(g^i) - \mathcal{E}^*_{\Phi_{01}^{\nu}}(\mathcal{G}^i) + \mathcal{U}_{\Phi_{01}^{\nu}}(\mathcal{G}^i) \geq \\ \Gamma\left(\mathcal{E}_{\ell_{01}}(g^i) - \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) + \mathcal{U}_{\ell_{01}}(\mathcal{G}^i)\right), \end{aligned} \tag{3}$$

where the minimizability gap $\mathcal{U}_{\ell_{01}}(\mathcal{G}^i) = \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) - \mathbb{E}_{X\sim\mathcal{D}^i_X}\Big[\inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{Y^i\sim\mathcal{D}^i(\cdot|X)}\big[\ell_{01}\big(g^i(X), Y^i\big)\big]\Big]$ captures the irreducible bias induced by function class $\mathcal{G}^i$. Notably, this gap vanishes when $\mathcal{G}^i = \mathcal{G}^i_{\text{all}}$ (Steinwart, 2007; Awasthi et al., 2022), in which case inequality (3) recovers the standard Bayes consistency property in (2).

4 OPTIMAL ALLOCATION FOR EQA SYSTEMS

In this section, we formalize the problem of allocating queries $x \in \mathcal{X}$ across a set of agents comprising a primary model g and a collection of J expert models. Our objective is to learn an allocation policy that assigns each query to the agent most likely to produce a correct prediction while controlling overall computational cost.

Importantly, we show that the proposed formulation admits a Bayes-optimal solution: under mild assumptions, there exists an allocation strategy that asymptotically minimizes expected error. This result provides the theoretical foundation for our learning-to-defer framework and guarantees that the learned deferral policy approaches optimal performance in the limit.

4.1 Formulating the Allocation Problem

Setting. We consider a query allocation setup involving a primary model $g \in \mathcal{G}$ and a collection of J

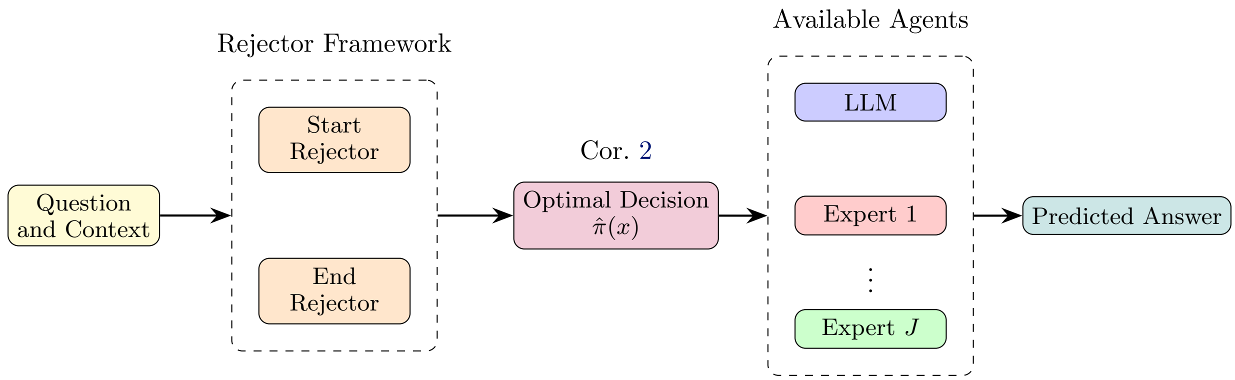


Figure 1: Overview of our approach. Rejectors estimate start and end uncertainty, inducing an optimal allocation policy $\hat{\pi}(x)$ (Corollary 2) that routes the query to an appropriate agent for answer prediction (Algorithm 2).

pre-trained expert models, collectively referred to as *agents*. The set of available agents is indexed by $\mathcal{A} = \{0\} \cup [J]$, where agent 0 corresponds to the main model and $[J] = \{1, \ldots, J\}$ indexes the experts, yielding $J+1$ agents in total. All agents are assumed to be fixed and trained offline; the objective is to learn a deferral policy that dynamically allocates each query $x \in \mathcal{X}$ to one of the agents at inference time (Mao et al., 2023a, 2024b,a, 2025; Montreuil et al., 2025b,a, 2026). Each expert M_j, when queried on an input x, produces a span prediction in the form of start and end token indices, denoted $m_j^{\text{start}}(x) \in \mathcal{Y}^{\text{start}}$ and $m_j^{\text{end}}(x) \in \mathcal{Y}^{\text{end}}$, respectively. These agents may correspond to human annotators, pretrained neural models, or other predictive systems. We denote the full set of expert predictions as $m(x) = (m_1(x), \ldots, m_J(x)) \in \mathcal{M}$, where each $m_j(x) = (m_j^{\text{start}}(x), m_j^{\text{end}}(x))$ represents the answer span returned by expert j.

True Deferral Loss. To enable cost-sensitive allocation of queries among multiple *agents*, we define a *rejector* $r \in \mathcal{R}$, which maps each input $x \in \mathcal{X}$ to an agent index in $\mathcal{A}$. The rejector is decomposed into two components—$r^{\text{start}} \in \mathcal{R}^{\text{start}}$ and $r^{\text{end}} \in \mathcal{R}^{\text{end}}$—corresponding to independent deferral decisions for the start and end span predictions. Each $r^i \in \mathcal{R}^i$, for $i \in \{\text{start}, \text{end}\}$, is a scoring function $r^i : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ that selects the agent with maximal score:

$$r^i(x) = \arg\max_{j \in \mathcal{A}} r^i(x, j).$$

To learn such rejectors, we adopt the *True Deferral Loss (TDL)* from Mao et al. (2023a), adapted here to structured prediction in EQA.

Definition 1 (True Deferral Loss)**.** *Given an input $x \in \mathcal{X}$ and a rejector $r \in \mathcal{R}$, the true deferral loss is defined as*

$$\ell_{def}(r(x), y) = \sum_{i \in \{start, end\}} \sum_{j=0}^{J} c_j(x, y^i) \mathbf{1}\{r^i(x) = j\},$$

where c_j denotes the cost of assigning input x to agent j. For the main model g, this cost is defined as

$$c_0\big(x, y^i\big) = \mathbf{1}\big\{g^i(x) \neq y^i\big\},$$

which penalizes incorrect predictions. For expert $j > 0$, the cost incorporates both prediction error and invocation penalty:

$$c_j\big(x, y^i\big) = \alpha_j \mathbf{1}\{m_j^i(x) \neq y^i\} + \beta_j,$$

where $\alpha_j \geq 0$ scales the error penalty and $\beta_j \geq 0$ models the cost of consultation. Notably, setting $\alpha_j = 0$ reduces expert j to an oracle: always correct but not free to query (Chow, 2003; Cortes et al., 2016).

The deferral decision thus hinges on minimizing expected deferral cost while maintaining high accuracy. When $r^i(x) = 0$, the query is assigned to the main model; otherwise, if $r^i(x) = j > 0$, it is deferred to expert j, whose prediction $m_j^i(x)$ incurs a penalty based on both error and query cost. A principled deferral strategy must therefore balance predictive reliability with the expense of expert consultation.

4.2 Optimality of the Allocation

An ideal deferral strategy allocates each query $x \in \mathcal{X}$ to the agent most likely to predict correctly, thereby maximizing reliability and decision confidence (Madras et al., 2018). To formalize this intuition, we analyze the Bayes-optimal risk under the *true deferral loss* and characterize the *Bayes-rejector*—the rejection function that minimizes expected deferral cost.

Let $\mathcal{D}^i(\cdot \mid X = x)$ denote the conditional distribution of the label Y^i given an input x, where $(X, Y^i) \sim \mathcal{D}^i$. For any $x \in \mathcal{X}$ we define the per-task, per-agent *conditional risks*

$$\eta_j^i(x) = \mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[c_j(x, Y^i)\big] \tag{4}$$

which leads to $\eta_0^i(x) = \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[g^i(x) \neq Y^i\big]$ and for expert $\eta_j^i(x) = \alpha_j \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[m_j^i(x) \neq Y^i\big] + \beta_j$, where $\alpha_j \geq 0$ scales the expert's prediction error and $\beta_j \geq 0$ is its fixed query cost.

Lemma 1 (Bayes-Rejector)**.** *Given an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$, the Bayes-optimal rejector that minimizes the conditional true deferral loss is*

$$r^{B,i}(x) = \begin{cases} 0, & \textit{if } \inf_{g^i \in \mathcal{G}^i} \eta_0^i(x) \leq \min_{j \in [J]} \eta_j^i(x), \\ j^*, & \textit{otherwise,} \end{cases}$$

where $j^* = \arg\min_{j \in [J]} \eta_j^i(x)$.

A proof is provided in the appendix. Lemma 1 formalizes the decision rule induced by the true deferral loss: defer only when an expert exhibits strictly lower expected risk than the main model. This ensures that each query is routed to the lowest-risk agent under the conditional distribution induced by $\mathcal{D}$, yielding asymptotically optimal deferral performance.

Computational Challenge. While Lemma 1 prescribes the optimal deferral strategy, learning the Bayes-rejector is computationally intractable in practice. The problem is known to be NP-hard (Zhang and Agarwal, 2020; Steinwart, 2007; Bartlett et al., 2006; Mohri et al., 2012), due to the discontinuity and non-convexity of the *true deferral loss*, which complicates optimization. This difficulty is characteristic of many structured prediction tasks where exact minimization of non-differentiable loss functions is infeasible (Cortes et al., 2016; Mao et al., 2023b). In the next subsection, we propose a surrogate formulation that approximates the Bayes-rejector while preserving its theoretical guarantees.

4.3 Accurate Approximation of the True Deferral Loss

To approximate the *true deferral loss* while preserving the optimality of the decision rule in Lemma 1, we leverage tools from consistency theory as formalized in Section 3. A standard approach in statistical learning is to introduce a *surrogate loss*, that is, a differentiable proxy for a target loss. In our setting, the goal is to construct a surrogate for the *true deferral loss* that is both Bayes-consistent and $(\mathcal{G}, \mathcal{R})$-consistent, ensuring that minimizing the surrogate yields a rejector $r^{*,i}$ that converges to the Bayes-optimal rejector from Lemma 1. In particular, we aim to guarantee that, as the surrogate loss is minimized, the learned deferral strategy asymptotically approaches the optimal allocation rule.

Formulating the Surrogate Deferral Loss. To construct a tractable alternative to the non-differentiable true deferral loss (TDL), we adopt the cross-entropy multiclass surrogate family $\Phi_{01}^{\nu} : \mathcal{R}^i \times \mathcal{X} \times \mathcal{A} \to \mathbb{R}_+$, which upper bounds the multiclass 0–1 loss and enjoys favorable optimization properties. Following Mao et al. (2023a), we adapt this surrogate to our structured EQA setting to define the *Surrogate Deferral Loss (SDL).*

Lemma 2 (Surrogate Deferral Loss)**.** *Given an input $x \in \mathcal{X}$ and a labeled instance (x, y), the surrogate deferral loss is defined as*

$$\Phi_{\text{def}}^{\nu}(r, x, y) = \sum_{i \in \{start, end\}} \sum_{j=0}^{J} \tau_j(x, y^i) \Phi_{01}^{\nu}\big(r^i, x, j\big),$$

where $\tau_j(x, y^i) = \sum_{q \neq j} c_q(x, y^i)$ quantifies the nonnegative relative cost gap between agent j and the competing agents.

We give the proof of Lemma 2 in the appendix. The term $\tau_j(x, y^i)$ reflects a soft preference for agent j by measuring how much worse the alternatives are in terms of cost. Intuitively, minimizing the SDL encourages the rejector r^i to assign queries to agents with lower relative cost. The surrogate loss Φ_{01}^{ν} is typically instantiated as a log-softmax and serves as a smooth approximation to the discontinuous 0–1 decision boundary.

This surrogate formulation preserves the agent-comparison structure of the TDL while introducing differentiability, enabling end-to-end training via stochastic gradient descent. As such, it supports efficient integration with standard deep learning frameworks (Bartlett et al., 2006), allowing scalable learning of deferral strategies. We give the detailed training procedure in Algorithm 1.

In the next subsection, we analyze the theoretical guarantees of the SDL. Specifically, we establish that under suitable conditions, minimizing Φ_{def}^{ν} yields a rejector r^i that approximates the Bayes-optimal deferral rule. This *Bayes consistency* ensures that our surrogate not only facilitates optimization but also preserves statistical optimality in the asymptotic regime.

Algorithm 1 Training the rejector r

Input: Dataset $\{(x_k, y_k^{\text{start}}, y_k^{\text{end}})\}_{k=1}^K$, multi-task model $g \in \mathcal{G}$, experts $m \in \mathcal{M}$, rejectors $r = (r^{\text{start}}, r^{\text{end}})$, number of epochs EPOCH, batch size BATCH, learning rate λ, surrogate parameter ν.
Initialization: Initialize parameters $\theta = (\theta^{\text{start}}, \theta^{\text{end}})$.
for $i = 1$ to EPOCH **do**
  Shuffle dataset $\{(x_k, y_k^{\text{start}}, y_k^{\text{end}})\}_{k=1}^K$.
  for each $\mathcal{B} \subset \{(x_k, y_k^{\text{start}}, y_k^{\text{end}})\}_{k=1}^K$ of size BATCH **do**
    Extract input-output pairs $z = (x, y^{\text{start}}, y^{\text{end}}) \in \mathcal{B}$.
    Query model $g(x)$ and experts $m(x)$.
    Evaluate costs $c_j(x, y^{\text{start}})$ and $c_j(x, y^{\text{end}})$
    Compute the regularized empirical risk minimization:
      $\widehat{\mathcal{E}}_{\Phi_{\text{def}}}(r; \theta) = \frac{1}{\text{BATCH}} \sum_{z \in \mathcal{B}} \Big[\Phi_{\text{def}}^\nu(r, x, y)\Big]$.
    Update parameters θ:
      $\theta \leftarrow \theta - \lambda \nabla_\theta \widehat{\mathcal{E}}_{\Phi_{\text{def}}}(r; \theta)$.
  end for
end for
Return: trained rejector model $\hat{r}$.

4.4 Theoretical Guarantees of the Surrogate Deferral Loss

Consistency Guarantees. In the previous subsection, we introduced the *surrogate deferral loss* as a differentiable surrogate to approximate the true deferral loss *true deferral loss*. We now establish that minimizing the surrogate excess risk leads to a reduction in the true excess deferral risk. That is,

$$\mathcal{E}_{\Phi_{\text{def}}^\nu}(r) - \mathcal{E}^*_{\Phi_{\text{def}}^\nu}(\mathcal{R}) + \mathcal{U}_{\Phi_{\text{def}}^\nu}(\mathcal{R})$$

acts as a valid upper bound for

$$\mathcal{E}_{\ell_{\text{def}}}(g, r) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}),$$

thereby implying that any minimizer $r^* \in \mathcal{R}$ of the surrogate loss closely approximates the Bayes-optimal rejector r^B, as defined in Lemma 1.

Theorem 1 (($\mathcal{R}, \mathcal{G}$)-Consistency)**.** *Let $x \in \mathcal{X}$ and let $\mathcal{D}$ denote any distribution over $\mathcal{X} \times \mathcal{Y}$. Fix any $\nu \geq 0$, and suppose there exists a non-decreasing, concave function $\Gamma^\nu : \mathbb{R}^+ \to \mathbb{R}^+$ such that, for all $r \in \mathcal{R}$,*

$$\mathcal{E}_{\Phi_{01}^\nu}(r) - \mathcal{E}^*_{\Phi_{01}^\nu}(\mathcal{R}) + \mathcal{U}_{\Phi_{01}^\nu}(\mathcal{R}) \geq \Gamma^\nu \left(\mathcal{E}_{\ell_{01}}(r) - \mathcal{E}^B_{\ell_{01}}(\mathcal{R}) + \mathcal{U}_{\ell_{01}}(\mathcal{R})\right)$$

Then, for any $(g, r) \in \mathcal{G} \times \mathcal{R}$,

$$\begin{aligned}\mathcal{E}_{\ell_{def}}(g, r) - \mathcal{E}^B_{\ell_{def}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{def}}(\mathcal{G}, \mathcal{R}) \leq \\ \overline{\Gamma}^\nu \left(\mathcal{E}_{\Phi_{def}^\nu}(r) - \mathcal{E}^*_{\Phi_{def}^\nu}(\mathcal{R}) + \mathcal{U}_{\Phi_{def}^\nu}(\mathcal{R})\right) \\ + \sum_{i \in \{start, end\}} \left(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i)\right),\end{aligned}$$

where the rescaled function is defined by

$$\overline{\Gamma}^\nu(u) = \left(\sum_{i \in \{start, end\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right) \Gamma^\nu \left(\frac{u}{\sum_{i \in \{start, end\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right),$$

and where the expected cost weights are given by $\overline{\boldsymbol{\tau}}^{\boldsymbol{i}} = \{\mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}[\tau_j(x, Y^i)]\}_{j \in \mathcal{A}}$.

The proof is deferred to the Appendix. Intuitively, Theorem 1 establishes a quantitative link between surrogate and true deferral risk minimization. The first term in the bound captures the impact of optimizing the surrogate objective, while the second term quantifies the approximation error due to holding the predictor g fixed. Assuming,

$$\begin{aligned}\mathcal{E}_{\Phi_{\text{def}}^\nu}(r) - \mathcal{E}^*_{\Phi_{\text{def}}^\nu}(\mathcal{R}) + \mathcal{U}_{\Phi_{\text{def}}^\nu}(\mathcal{R}) \leq \epsilon_0, \\ \sum_{i \in \{\text{start}, \text{end}\}} \left(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i)\right) \leq \epsilon_1,\end{aligned}$$

then the true deferral excess risk satisfies

$$\begin{aligned}\mathcal{E}_{\ell_{\text{def}}}(g, r) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) \leq \\ \epsilon_1 + \left(\sum_{i \in \{\text{start}, \text{end}\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right) \Gamma^\nu \left(\frac{\epsilon_0}{\sum_{i \in \{\text{start}, \text{end}\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right).\end{aligned}$$

Importantly, in the realizable regime where both function classes are unrestricted (i.e., $\mathcal{R} = \mathcal{R}_{\text{all}}$ and $\mathcal{G} = \mathcal{G}_{\text{all}}$), we recover exact minimization of both surrogate and true risks.

Corollary 1 (Bayes-consistency)**.** *Suppose the conditions of Theorem 1 hold and $\mathcal{R} = \mathcal{R}_{all}, \mathcal{G} = \mathcal{G}_{all}$. Then the surrogate deferral loss is Bayes-consistent: for any sequence $(g_k, r_k) \in \mathcal{G}_{all} \times \mathcal{R}_{all}$ with*

$$\mathcal{E}_{\Phi_{def}^\nu}(r_k) - \mathcal{E}^*_{\Phi_{def}^\nu}(\mathcal{R}_{all}) \xrightarrow{k \to \infty} 0,$$

we have $\mathcal{E}_{\ell_{def}}(g_k, r_k) - \mathcal{E}^B_{\ell_{def}}(\mathcal{G}_{all}, \mathcal{R}_{all}) \xrightarrow{k \to \infty} 0$.

This result confirms that minimizing our surrogate training objective leads to asymptotically optimal deferral policies under general conditions.

Single expert allocation. When both the *start* and *end* span predictions must be delegated to the same agent, we define a unified deferral policy $\pi^*(x)$ that minimizes the total Bayes risk across tasks. This yields the following optimal allocation rule, proved in the appendix:

Lemma 3 (Bayes-Optimal Deferral Policy)**.** *Given $x \in \mathcal{X}$, the Bayes-optimal policy assigns the query to the agent $j \in \mathcal{A}$ minimizing the expected cumulative risk:*

$$\pi^*(x) = \arg\min_{j \in \mathcal{A}} \sum_{i \in \{start, end\}} \eta_j^i(x),$$

where $\eta_j^i(x) = \mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}[c_j(x, Y^i)]$ denotes the expected cost of assigning task i to agent j.

In practice, we approximate this policy by learning per-task rejector scores $\hat{r}^i(x, j)$ that estimate $\eta_j^i(x)$. This gives rise to the learned allocation rule:

Corollary 2 (Learned allocation policy)**.** *Assume the per-task surrogate Φ_{01}^{ν} is strictly proper (Lemma 2), so it admits a strictly decreasing link $\psi\colon [0,1] \to \mathbb{R}$. Suppose further that the excess surrogate risk of the learned score function $\hat{r}$ satisfies*

$$\mathcal{E}_{\Phi_{\text{def}}^{\nu}}(\hat{r}) \; - \; \mathcal{E}^*_{\Phi_{\text{def}}^{\nu}}(\mathcal{R}_{all}) \xrightarrow[n \to \infty]{} 0.$$

Then, as the training-sample size n grows,

$$\Pr_{X \sim \mathcal{D}}\big[\hat{\pi}(X) \neq \pi^*(X)\big] \xrightarrow{n \to \infty} 0,$$

so the learned allocation rule

$$\hat{\pi}(x) = \arg\max_{j \in \mathcal{A}} \sum_{i \in \{start, end\}} \hat{r}^{\,i}(x, j),$$

converges in probability to the Bayes-optimal deferral policy of Lemma 3.

A proof is given in the appendix. Under the consistency guarantees established in Theorem 1, this converges to the Bayes-optimal allocation $\pi^*(x)$, ensuring reliable and cost-sensitive deferral in the single-agent allocation setting. The query is allocated to an agent according to Algorithm 2.

Algorithm 2 Inference with Corollary 2

1: **Input:** Query $x = (q, c)$ with question q and context c

2: **Rejector Evaluation:** Compute scores $\hat{r}(x) = (\hat{r}^{\text{start}}(x), \hat{r}^{\text{end}}(x))$

3: **Agent Allocation:** Select agent via the learned policy: $\hat{\pi}(x) = \arg\max_{j \in \mathcal{A}} \sum_{i \in \{\text{start,end}\}} \hat{r}^i(x, j)$

4: **Prediction:**

- **If** $\hat{\pi}(x) = 0$, return prediction $g(x)$
- **Else** return expert prediction $m_{\hat{\pi}(x)}(x)$

5 EVALUATION

5.1 Motivation

We evaluate L2D for balancing predictive accuracy and computational efficiency in EQA, focusing on realistic deployment scenarios with lightweight on-device models and stronger remote experts. Our goal is not to surpass state-of-the-art accuracy, but to show that L2D effectively trades off answer quality against computational cost, reliably identifying when deferral is warranted and minimizing reliance on expensive models.

5.2 Setting

We test L2D on three standard EQA benchmarks: SQuADv1 (Rajpurkar et al., 2016), SQuADv2 (Rajpurkar et al., 2018), and TriviaQA (Joshi et al., 2017). The training and inference procedures are given in Algorithms 1 and 2.

Agents. We evaluate our L2D framework in a deployment-motivated setting with one lightweight general-purpose model and two stronger EQA experts. As the on-device model, we select LLaMA-3.2-1B (Touvron et al., 2023), which offers broad task coverage while remaining computationally viable for resource-constrained environments. For simplicity, we use the base weights without fine-tuning. The expert models, M_1 and M_2, are ALBERT-Base and ALBERT-XXL, respectively (Lan et al., 2020). These models are well suited to EQA and achieve high span-selection accuracy, but they do not offer LLaMA's general-purpose capabilities, making them less suitable as the on-device default model. To capture the disparity in inference cost between M_1 and M_2, we define a cost ratio $R = \frac{\text{GFLOPs}(M_2)}{20 \cdot \text{GFLOPs}(M_1)}$ and apply an expert penalty $\beta_2 = R\beta_1$ to discourage excessive reliance on the more computationally intensive model.

To avoid tying the analysis to deployment-specific factors such as infrastructure or communication stack, we use GFLOPs as a simple proxy when defining this ratio. In practical deployments, users could instead calibrate the β costs using quantities such as network latency or monetary cost.

Rejector. To enable cost-aware query allocation across agents, we employ a compact model designed for small-device deployment (Figure 5). Specifically, we instantiate the rejector using the TinyBERT architecture (Devlin et al., 2019), which contains only 4.39M parameters, just 0.35% the size of the LLaMA-3.2-1B model, making it well suited to low-compute environments. The rejector is trained using the surrogate deferral loss (SDL) introduced in Lemma 2. We adopt the multiclass formulation Φ_{01}^{ν} with $\nu = 1$, corresponding to the standard log-softmax loss (Mao et al., 2023b).

Benchmark: We benchmark against vote-based ensembles (Breiman, 1996; Trad and Chehab, 2024), which most closely resemble our setting because they combine multiple models under direct allocation. However, ensembles query all models in parallel and therefore operate under a different computational regime. We also benchmark against a larger model from the Llama-3 family, Llama-3-8B (Grattafiori et al., 2024), to assess whether our framework can match or exceed the performance of a substantially larger model while retaining the deployment advantages of the 1B variant. When prompting both Llama-3.2-1B and Llama-3-8B, we use few-shot demonstrations (Brown et al., 2020), listed in the appendix.

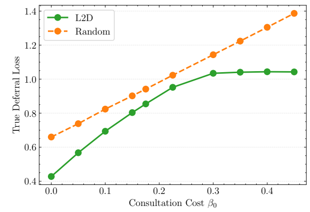
Additionally, we benchmark our approach against the family of single-expert routers introduced in (Ding et al., 2024): the probabilistic router r_{prob}, the deterministic router r_{deter}, and the relaxed router with transformation r_{trans}. We use ALBERT-Base as the model and ALBERT-XXL as the expert. To adapt this framework to EQA, we use $c_j(x, y^i)$ as the quality function in place of BART (Yuan et al., 2021).

Metrics: We measure EQA performance using Exact Match (EM). We emphasize that, although the specialist models are stronger, they are not suitable candidates for g. We also report the GFLOPs/EM ratio, which measures computational cost per unit of performance, and allocation ratios, which quantify the fraction of queries deferred to experts and therefore reflect how r accounts for cost. Finally, we report true-positive and false-positive rates (TPR/FPR). A true positive occurs when the main model is incorrect and the query is correctly deferred to an accurate expert. A false positive occurs when the query is deferred to an incorrect expert even though the main model is correct.

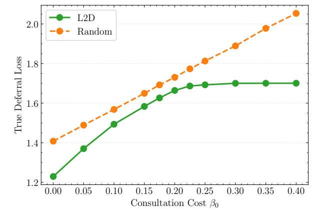
5.3 Results

Sanity Check Against Random Allocation. We begin with a comparison against random allocation as a sanity check. Figure 2 compares the true deferral loss of L2D with that of a random allocation policy across all benchmark datasets. This is the most direct way to verify that the rejector is learning a non-trivial allocation rule rather than merely redistributing queries across agents.

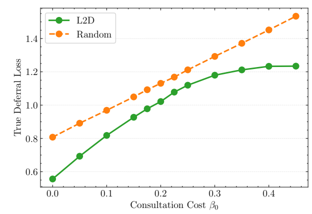
For all values of β_0, L2D achieves substantially lower TDL, demonstrating that it defers selectively rather than arbitrarily. Because TDL penalizes both unnecessary expert usage and incorrect on-device predictions, the consistent gap over random allocation indicates that the learned policy captures the intended cost-quality trade-off.



(a) SQuADv1



(b) SQuADv2



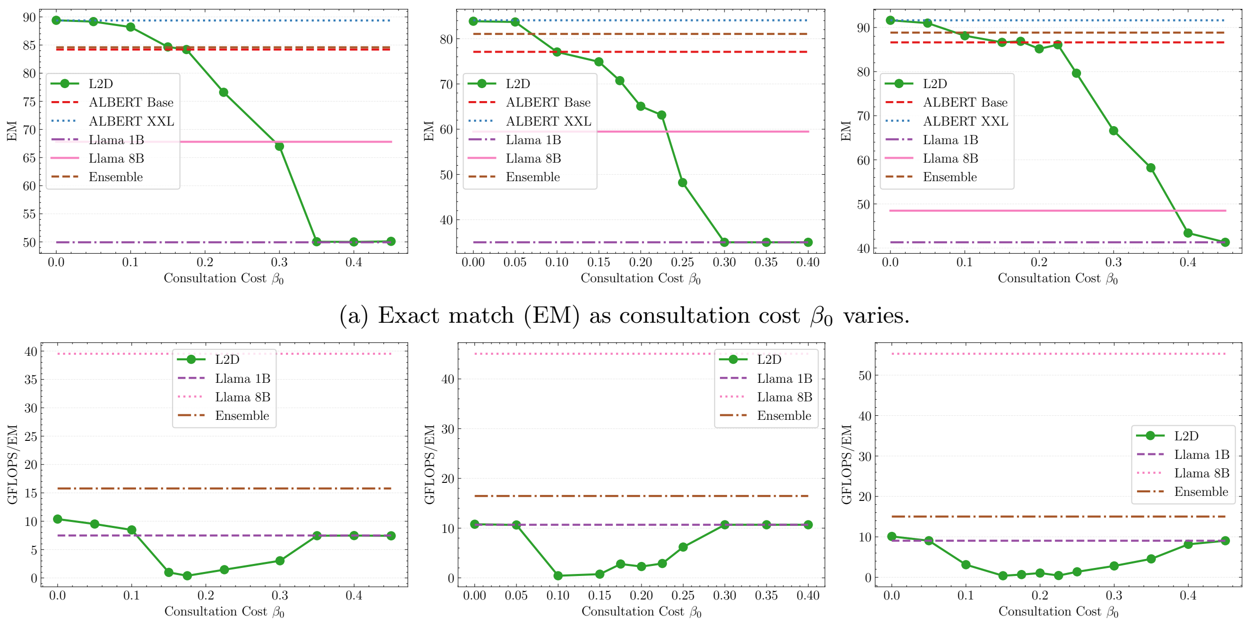
(c) TriviaQA

Figure 2: **TDL against random allocation as consultation cost β_0 varies.**

Cost-Aware Allocation Behavior. We next examine how the learned policy responds to the consultation cost. Across all three datasets, increasing the consultation penalty systematically reduces reliance on expensive experts and shifts traffic toward cheaper agents. In particular, increasing β_1 moves allocation toward the cheaper ALBERT-Base expert, while extreme values of β_1 can suppress allocation to either g or M_2. This behavior indicates that the rejector is not merely optimizing accuracy, but is explicitly responding to the specified cost structure. Additional plots of normalized cost and expert query rate are provided in the appendix.

Accuracy-Efficiency Trade-Offs. The next question is whether these cost-aware routing decisions preserve answer quality. Figure 3 reports the evolution of exact match and GFLOPs/EM as the consultation cost varies. These plots make the performance-efficiency trade-off explicit and clarify where the learned policy remains competitive while reducing computation.

As β_0 increases, the policy is encouraged to use cheaper agents more aggressively, which leads to a gradual reduction in EM, as shown in the top row of Figure 3. This behavior is expected: larger consultation penalties make aggressive expert usage less attractive. The key point is that the resulting loss in exact match is controlled rather than abrupt, which



(a) Exact match (EM) as consultation cost β_0 varies.

(b) GFLOPs/EM as consultation cost β_0 varies.

Figure 3: **Accuracy-efficiency trade-offs across datasets.** Columns correspond to SQuADv1, SQuADv2, and TriviaQA from left to right. The top row reports exact match, and the bottom row reports computational cost per unit of performance.

indicates that the rejector is making meaningful trade-offs instead of collapsing to a single fixed agent.

The bottom row of Figure 3 shows that, across all benchmark datasets, our approach uses the fewest computational resources per unit of EM. This highlights the cost-performance efficiency of the proposed method. We also obtain at least a 4x efficiency improvement relative to naively running the larger, more expensive Llama 8B on the edge device while maintaining comparable performance. This underscores the practical value of selective expert involvement.

Single-Expert Router Comparison. Finally, we compare our framework with single-expert routing baselines adapted from Ding et al. (2024). This comparison isolates the effect of the deferral objective from the multi-expert setting and tests whether our loss remains beneficial in the simpler two-agent case.

Table 1: Single-Expert Setting, Exact Match on SQuADv1, L2D against Query Routers (Ding et al., 2024)

Cost β_0	Our	r_{prob}	r_{deter}	r_{trans}
0.0	84.2	84.2	84.2	84.2

Table 1 shows that, in the zero-cost single-expert setting, our method matches the performance of existing query routers. Table 2 then shows that, once expert consultation incurs nonzero cost, our method outperforms the router baselines. This suggests that the proposed objective is better aligned with discontinuous cost structures of the form $c_j(x, y^i)$ and therefore remains effective even when the routing problem becomes explicitly cost-sensitive.

Table 2: Single-Expert Setting, Exact Match / Cost on SQuADv1, L2D against Query Routers (Ding et al., 2024)

Cost β_0	Our	r_{prob}	r_{deter}	r_{trans}
0.25	12.16	3.36	3.37	3.36
0.5	3.04	1.68	1.68	1.68

6 CONCLUSION

We introduced a Learning-to-Defer framework for EQA that dynamically allocates queries to the most suitable agent while balancing accuracy and computational cost. The framework is supported by theoretical guarantees and is intended for resource-constrained deployment settings. Empirical evaluations on SQuADv1, SQuADv2, and TriviaQA show improved reliability-efficiency trade-offs relative to larger LLMs, ensembles, and routing baselines.

ACKNOWLEDGMENTS

This research was supported by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No. AISG2-PhD-2023-01-041-J) and by A*STAR, and forms part of the DesCartes programme, which is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

Alqifari, R. (2019). Question answering systems approaches and challenges. In Kovatchev, V., Temnikova, I., Šandrih, B., and Nikolova, I., editors, *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 69–75, Varna, Bulgaria. INCOMA Ltd.

Awasthi, P., Mao, A., Mohri, M., and Zhong, Y. (2022). Multi-class h-consistency bounds. *Advances in neural information processing systems*, 35:782–795.

Bartlett, P., Jordan, M., and McAuliffe, J. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156.

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Chen, S., Jiang, W., Lin, B., Kwok, J., and Zhang, Y. (2024). Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328.

Chen, Z., Li, J., Chen, P., Li, Z., Sun, K., Luo, Y., Mao, Q., Yang, D., Sun, H., and Yu, P. S. (2025). Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Chow, C. (2003). On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46.

Cortes, C., DeSalvo, G., and Mohri, M. (2016). Learning with rejection. In Ortner, R., Simon, H. U., and Zilles, S., editors, *Algorithmic Learning Theory*, pages 67–82, Cham. Springer International Publishing.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ding, D., Amer-Yahia, S., and Lakshmanan, L. (2022). On efficient approximate queries over machine learning models. *Proc. VLDB Endow.*, 16(4):918–931.

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L. V. S., and Awadallah, A. H. (2024). Hybrid LLM: cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Ding, D., Mallick, A., Zhang, S., Wang, C., Madrigal, D., del Carmen Hipolito Garcia, M., Xia, M., Lakshmanan, L. V. S., Wu, Q., and Rühle, V. (2025). Best-route: Adaptive llm routing with test-time optimal compute. *ArXiv*, abs/2506.22716.

Egashira, K., Vero, M., Staab, R., He, J., and Vechev, M. T. (2024). Exploiting LLM quantization. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C., editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Garcia, A., Clavel, C., Essid, S., and d'Alche Buc, F. (2018). Structured output learning with abstention: Application to accurate opinion prediction. In Dy, J. and Krause, A., editors, *Proceedings of the*

35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 1695–1703. PMLR.

Ghosh, A., Kumar, H., and Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. In Singh, S. P. and Markovitch, S., editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1919–1925. AAAI Press.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b.

Jitkrittum, W., Gupta, N., Menon, A. K., Narasimhan, H., Rawat, A. S., and Kumar, S. (2023). When does confidence-based cascade deferral suffice? In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Kag, A., Fedorov, I., Gangrade, A., Whatmough, P. N., and Saligrama, V. (2023). Efficient edge inference by selective query. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Kolawole, S., Dennis, D., Talwalkar, A., and Smith, V. (2024). Agreement-based cascading for efficient inference.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. (2024). Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Long, P. M. and Servedio, R. A. (2013). Consistency versus realizable h-consistency for multiclass classification. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 801–809. JMLR.org.

Lu, K., Yuan, H., Lin, R., Lin, J., Yuan, Z., Zhou, C., and Zhou, J. (2024). Routing to the expert: Efficient reward-guided ensemble of large language models. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico. Association for Computational Linguistics.

Madras, D., Pitassi, T., and Zemel, R. S. (2018). Predict responsibly: Improving fairness and accuracy by learning to defer. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing*

Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 6150–6160.

Mao, A., Mohri, C., Mohri, M., and Zhong, Y. (2023a). Two-stage learning to defer with multiple experts. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Mao, A., Mohri, M., and Zhong, Y. (2023b). Cross-entropy loss functions: Theoretical analysis and applications. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 23803–23828. PMLR.

Mao, A., Mohri, M., and Zhong, Y. (2024a). Realizable h-consistent and bayes-consistent loss functions for learning to defer. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C., editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Mao, A., Mohri, M., and Zhong, Y. (2024b). Regression with multi-expert deferral. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Mao, A., Mohri, M., and Zhong, Y. (2025). Mastering multiple-expert routing: Realizable h-consistency and strong guarantees for learning to defer. *arXiv preprint arXiv:2506.20650*.

Merenda, M., Porcaro, C., and Iero, D. (2020). Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. The MIT Press.

Montreuil, Y., Carlier, A., Ng, L. X., and Ooi, W. T. (2025a). Adversarial robustness in two-stage learning-to-defer: Algorithms and guarantees. In *Forty-second International Conference on Machine Learning*.

Montreuil, Y., Carlier, A., Ng, L. X., and Ooi, W. T. (2026). Why ask one when you can ask \$k\$? learning-to-defer to the top-\$k\$ experts. In *The Fourteenth International Conference on Learning Representations*.

Montreuil, Y., Heng, Y. S., Carlier, A., Ng, L. X., and Ooi, W. T. (2025b). A two-stage learning-to-defer approach for multi-task learning. In *Forty-second International Conference on Machine Learning*.

Mozannar, H. and Sontag, D. A. (2020). Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7076–7087. PMLR.

Narasimhan, H., Jitkrittum, W., Rawat, A. S., Kim, S., Gupta, N., Menon, A. K., and Kumar, S. (2025). Faster cascades via speculative decoding. In *The Thirteenth International Conference on Learning Representations*.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. (2025). RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2023). Gpt-4 technical report.

Qu, G., Chen, Q., Wei, W., Lin, Z., Chen, X., and Huang, K. (2025). Mobile edge intelligence for large language models: A contemporary survey. *IEEE Communications Surveys & Tutorials*, 27(6):3820–3860.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Saberian, M. and Vasconcelos, N. (2014). Boosting algorithms for detector cascade learning. *Journal of Machine Learning Research*, 15(74):2569–2605.

Sadat, M., Zhou, Z., Lange, L., Araki, J., Gundroo, A., Wang, B., Menon, R., Parvez, M., and Feng, Z. (2023). DelucionQA: Detecting hallucinations in domain-specific question answering. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 822–835, Singapore. Association for Computational Linguistics.

Steinwart, I. (2007). How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287.

Stojkovic, J., Zhang, C., Goiri, Í., Torrellas, J., and Choukse, E. (2025). Dynamollm: Designing llm inference clusters for performance and energy efficiency. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1348–1362. IEEE.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., and Zhou, D. (2020). MobileBERT: a compact task-agnostic BERT for resource-limited devices. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

Trad, F. and Chehab, A. (2024). To ensemble or not: Assessing majority voting strategies for phishing detection with large language models.

Varshney, N. and Baral, C. (2022). Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11007–11021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Verma, R., Barrejón, D., and Nalisnick, E. T. (2023). Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In Ruiz, F. J. R., Dy, J. G., and van de Meent, J., editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 11415–11434. PMLR.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I.

Yuan, W., Neubig, G., and Liu, P. (2021). Bartscore: Evaluating generated text as text generation. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27263–27277.

Yue, M., Zhao, J., Zhang, M., Du, L., and Yao, Z. (2023). Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *ArXiv preprint*, abs/2310.03094.

Zhang, M. and Agarwal, S. (2020). Bayes consistency vs. h-consistency: The interplay between surrogate loss functions and the scoring function class. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Zhang, T. (2002). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A APPENDIX

A.1 Current Approaches

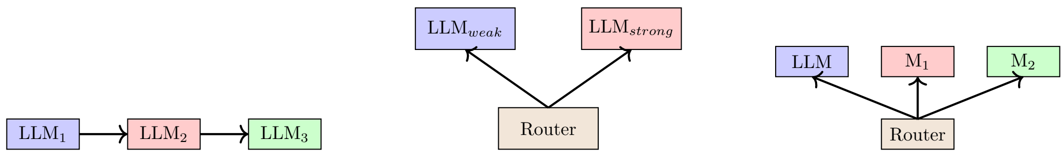


Figure 4: From left to right: model cascades, query routing, and Learning-to-Defer (ours). Our framework retains the multi-model nature of cascades while allowing direct allocation, as in query routing.

A.2 Proof Lemma 1

Lemma 1 (Bayes-Rejector). *Given an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$, the Bayes-optimal rejector that minimizes the conditional true deferral loss is*

$$r^{B,i}(x) = \begin{cases} 0, & \text{if } \inf_{g^i \in \mathcal{G}^i} \eta_0^i(x) \leq \min_{j \in [J]} \eta_j^i(x), \\ j^*, & \text{otherwise}, \end{cases}$$

where $j^* = \arg\min_{j \in [J]} \eta_j^i(x)$.

Proof. For $i \in \{\text{start}, \text{end}\}$ and any input $x \in \mathcal{X}$, define

$$\eta_0^i(x) \;=\; \Pr_{(X,Y^i)\sim\mathcal{D}^i}\big[g^i(x) \neq Y^i \mid X = x\big],$$

$$\eta_j^i(x) \;=\; \alpha_j \Pr_{(X,Y^i)\sim\mathcal{D}^i}\big[m_j^i(x) \neq Y^i \mid X = x\big] \;+\; \beta_j, \qquad j = 1, \ldots, J,$$

where $\alpha_j \geq 0$ and $\beta_j \geq 0$ are the per-expert error penalty and fixed consultation cost, respectively.

We analyze the conditional risk associated with the *true deferral loss* for a fixed task $i \in \{\text{start}, \text{end}\}$. For a given input $x \in \mathcal{X}$, this risk is defined as the expected cost incurred when using a rejector $r^i \in \mathcal{R}^i$ to select an agent:

$$\begin{aligned} \mathcal{C}^i_{\ell_{\text{def}}}(g^i, r^i, x) &= \mathbb{E}_{Y^i \sim \mathcal{D}^i(\,\cdot\,|X=x)}\Big[\sum_{j=0}^{J} c_j\big(x, Y^i\big)\, \mathbf{1}\{r^i(x) = j\}\Big] \\ &= \eta_0^i(x)\, \mathbf{1}\{r^i(x) = 0\} \;+\; \sum_{j=1}^{J} \eta_j^i(x)\, \mathbf{1}\{r^i(x) = j\}. \end{aligned}$$

To determine the optimal rejection strategy, we minimize this conditional risk over all possible choices of $g^i \in \mathcal{G}^i$ and $r^i \in \mathcal{R}^i$:

$$\inf_{r^i \in \mathcal{R}^i,\, g^i \in \mathcal{G}^i} \mathcal{C}^i_{\ell_{\text{def}}}(g^i, r^i, x) = \min\left\{ \inf_{g^i \in \mathcal{G}^i} \eta_0^i(x), \min_{j \in [J]} \eta_j^i(x) \right\}.$$

This implies the structure of the Bayes-optimal rejector $r^{B,i}$, which assigns the query to the agent with minimal conditional risk:

$$r^{B,i}(x) = \begin{cases} 0, & \text{if } \inf\limits_{g^i \in \mathcal{G}^i} \eta_0^i(x) \leq \min\limits_{j \in [J]} \eta_j^i(x), \\ \arg\min_{j \in [J]} \eta_j^i(x), & \text{otherwise}. \end{cases}$$

Thus, the Bayes-rejector minimizes the expected deferral loss by comparing the cost-adjusted risks of the main model and the experts, selecting the most reliable agent accordingly. □

A.3 Proof Lemma 2

Lemma 2 (Surrogate Deferral Loss)**.** *Given an input $x \in \mathcal{X}$ and a labeled instance (x, y), the surrogate deferral loss is defined as*

$$\Phi^{\nu}_{\text{def}}(r, x, y) = \sum_{i \in \{start, end\}} \sum_{j=0}^{J} \tau_j(x, y^i) \Phi^{\nu}_{01}\big(r^i, x, j\big),$$

where $\tau_j(x, y^i) = \sum_{q \neq j} c_q(x, y^i)$ quantifies the nonnegative relative cost gap between agent j and the competing agents.

Proof. We consider a unified agent space $\mathcal{A} = \{0, 1, \ldots, J\}$, where index 0 denotes the main model and indices $j \in [J]$ denote experts. For each query–label pair (x, y^i), define the agent-specific cost as $c_j(x, y^i) \geq 0$ for all $j \in \mathcal{A}$. Let the total consultation cost be:

$$C^i_{\text{tot}}(x, y^i) = \sum_{j \in \mathcal{A}} c_j(x, y^i).$$

Define the deferral complement cost (excluding agent j) as

$$\tau^i_j(x, y^i) = \sum_{\substack{q \in \mathcal{A} \\ q \neq j}} c_q(x, y^i) = C^i_{\text{tot}}(x, y^i) - c_j(x, y^i).$$

Now, for a selector $r^i : \mathcal{X} \to \mathcal{A}$, the deferral loss component is

$$\ell^i_{\text{def}}(r^i(x), y^i) = c_{r^i(x)}(x, y^i).$$

Because $\mathcal{A} \setminus \{r^i(x)\}$ is the complement of the selected agent, we can equivalently write:

$$\begin{aligned} \ell^i_{\text{def}}(r^i(x), y^i) &= C^i_{\text{tot}}(x, y^i) - \sum_{j \neq r^i(x)} c_j(x, y^i) \\ &= \sum_{j \in \mathcal{A}} \tau^i_j(x, y^i) \mathbf{1}\{r^i(x) \neq j\} - (J-1) C^i_{\text{tot}}(x, y^i). \end{aligned} \tag{5}$$

Now aggregate the deferral loss over both span components $i \in \{\text{start}, \text{end}\}$ to obtain the total deferral loss:

$$\sum_{i \in \{\text{start},\text{end}\}} \ell^i_{\text{def}}(r^i(x), y^i) = \sum_{i \in \{\text{start},\text{end}\}} \left[\sum_{j \in \mathcal{A}} \tau^i_j(x, y^i) \mathbf{1}\{r^i(x) \neq j\} - (J-1) C^i_{\text{tot}}(x, y^i) \right].$$

Let $\Phi^u_{01}(r^i, x, j)$ be a surrogate loss that upper bounds the 0-1 indicator:

$$\mathbf{1}\{r^i(x) \neq j\} \leq \Phi^u_{01}(r^i, x, j), \quad \forall j \in \mathcal{A}.$$

This holds for the cross-entropy surrogate and other upper-bounding families (Mao et al., 2023b). Since each $\tau^i_j(x, y^i) \geq 0$, the total loss satisfies:

$$\sum_{i \in \{\text{start},\text{end}\}} \ell^i_{\text{def}}(r^i(x), y^i) \leq \sum_{i \in \{\text{start},\text{end}\}} \left[\sum_{j \in \mathcal{A}} \tau^i_j(x, y^i) \Phi^u_{01}(r^i, x, j) - (J-1) C^i_{\text{tot}}(x, y^i) \right].$$

Furthermore, as the term $(J-1) C^i_{\text{tot}}(x, y^i)$ does not depend on r, we can formalize the following surrogate loss

$$\Phi(r, x, y) = \sum_{i \in \{\text{start},\text{end}\}} \sum_{j \in \mathcal{A}} \tau^i_j(x, y^i) \Phi^u_{01}(r^i, x, j) \tag{6}$$

□

A.4 Proof Theorem 1

Theorem 1 (($\mathcal{R}, \mathcal{G}$)-Consistency)**.** *Let $x \in \mathcal{X}$ and let $\mathcal{D}$ denote any distribution over $\mathcal{X} \times \mathcal{Y}$. Fix any $\nu \geq 0$, and suppose there exists a non-decreasing, concave function $\Gamma^\nu : \mathbb{R}^+ \to \mathbb{R}^+$ such that, for all $r \in \mathcal{R}$,*

$$\begin{aligned}\mathcal{E}_{\Phi_{01}^\nu}(r) - \mathcal{E}^*_{\Phi_{01}^\nu}(\mathcal{R}) + \mathcal{U}_{\Phi_{01}^\nu}(\mathcal{R}) \geq \\ \Gamma^\nu \left(\mathcal{E}_{\ell_{01}}(r) - \mathcal{E}^B_{\ell_{01}}(\mathcal{R}) + \mathcal{U}_{\ell_{01}}(\mathcal{R})\right)\end{aligned}$$

Then, for any $(g, r) \in \mathcal{G} \times \mathcal{R}$,

$$\begin{aligned}\mathcal{E}_{\ell_{def}}(g, r) - \mathcal{E}^B_{\ell_{def}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{def}}(\mathcal{G}, \mathcal{R}) \leq \\ \overline{\Gamma}^\nu \left(\mathcal{E}_{\Phi^\nu_{def}}(r) - \mathcal{E}^*_{\Phi^\nu_{def}}(\mathcal{R}) + \mathcal{U}_{\Phi^\nu_{def}}(\mathcal{R})\right) \\ + \sum_{i \in \{start, end\}} \left(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i)\right),\end{aligned}$$

where the rescaled function is defined by

$$\overline{\Gamma}^\nu(u) = \left(\sum_{i \in \{start, end\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right) \Gamma^\nu \left(\frac{u}{\sum_{i \in \{start, end\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right),$$

and where the expected cost weights are given by $\overline{\boldsymbol{\tau}}^{\boldsymbol{i}} = \{\mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}[\tau_j(x, Y^i)]\}_{j \in \mathcal{A}}$.

Proof. The proof of Theorem 1 uses the following lemma, which states the relevant consistency property for a general distribution.

Lemma 4 ($\mathcal{R}^i$-consistency bound)**.** *Fix an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$. Suppose there exists a non-decreasing, concave function $\Gamma^\nu : \mathbb{R}^+ \to \mathbb{R}^+$ for $\nu \geq 0$ such that the $\mathcal{R}^i$-consistency bounds hold for any distribution $\mathcal{D}$:*

$$\mathcal{E}_{\Phi_{01}^\nu}(r^i) - \mathcal{E}^*_{\Phi_{01}^\nu}(\mathcal{R}^i) + \mathcal{U}_{\Phi_{01}^\nu}(\mathcal{R}^i) \geq \Gamma^\nu(\mathcal{E}_{\ell_{01}}(r^i) - \mathcal{E}^B_{\ell_{01}}(\mathcal{R}^i) + \mathcal{U}_{\ell_{01}}(\mathcal{R}^i)),$$

or, equivalently, for $\boldsymbol{p^i} \in \Delta^{|\mathcal{A}|}$,

$$\sum_{j \in \mathcal{A}} p^i_j 1_{\{r^i(x) \neq j\}} - \inf_{r^i \in \mathcal{R}^i} \sum_{j \in \mathcal{A}} p^i_j 1_{\{r^i(x) \neq j\}} \leq \Gamma^\nu \Big(\sum_{j \in \mathcal{A}} p^i_j \Phi^\nu_{01}(r^i, x, j) - \inf_{r^i \in \mathcal{R}^i} \sum_{j \in \mathcal{A}} p^i_j \Phi^\nu_{01}(r^i, x, j)\Big)$$

Define, for each $j \in \mathcal{A} = \{0, \dots, J\}$, the cost

$$\overline{c}^{i,*}_j(x) = \begin{cases} \inf_{g^i \in \mathcal{G}^i} \mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[c_0(x, Y^i)\big] \\ \quad = \inf_{g^i \in \mathcal{G}^i} \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[g^i(x) \neq Y^i\big], \quad j = 0, \\ \\ \mathbb{E}_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[c_j(x, Y^i)\big] \\ = \alpha_j \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[m^i_j(x) \neq Y^i\big] + \beta_j, \quad j = 1, \dots, J. \end{cases}$$

Recall the previously established result from Lemma 1:

$$\begin{aligned}\mathcal{C}^{*,i}_{\ell_{\text{def}}}(g^i, r^i, x) &= \min \Big\{ \inf_{g^i \in \mathcal{G}^i} \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[g^i(x) \neq Y^i\big], \alpha_j \Pr_{Y^i \sim \mathcal{D}^i(\cdot|X=x)}\big[m^i_j(x) \neq Y^i\big] + \beta_j \Big\} \\ &= \min_{j \in \mathcal{A}} \overline{c}^{i,*}_j(x)\end{aligned}$$

Therefore, we introduce the calibration gap $\Delta\mathcal{C}^i_{\ell_{\text{def}}} := \mathcal{C}^i_{\ell_{\text{def}}} - \mathcal{C}^{*,i}_{\ell_{\text{def}}}$:

$$\begin{aligned}\Delta\mathcal{C}^i_{\ell_{\text{def}}}(r^i, g^i, x) &= \mathcal{C}^i_{\ell_{\text{def}}}(r^i, g^i, x) - \min_{j \in \mathcal{A}} \overline{c}^{i,*}_j(x) \\ &= \mathcal{C}^i_{\ell_{\text{def}}}(r^i, g^i, x) - \min_{j \in \mathcal{A}} \overline{c}^i_j(x) + \Big(\min_{j \in \mathcal{A}} \overline{c}^i_j(x) - \min_{j \in \mathcal{A}} \overline{c}^{i,*}_j(x)\Big)\end{aligned} \tag{7}$$

Define the first term as $A = \mathcal{C}^i_{\ell_{\text{def}}} - \min_{j\in\mathcal{A}} \overline{c}^i_j$ and the second term as $B = \min_{j\in\mathcal{A}} \overline{c}^i_j - \min_{j\in\mathcal{A}} \overline{c}^{i,*}_j$, so that $\Delta\mathcal{C}^i_{\ell_{\text{def}}} = A + B$. By Lemma 2,

$$\min_{j\in\mathcal{A}} \overline{c}^i_j(x) = \inf_{r^i\in\mathcal{R}} \mathcal{C}^i_{\ell_{def}}(r^i, g^i, x) = \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} \overline{\tau}^i_j(x) 1_{\{r^i(x)\neq j\}} \tag{8}$$

It follows by definition of the conditional risk:

$$A = \sum_{j\in\mathcal{A}} \overline{\tau}^i_j(x) 1_{\{r^i(x)\neq j\}} - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} \overline{\tau}^i_j(x) 1_{\{r^i(x)\neq j\}} \tag{9}$$

We normalize the cost vector $\boldsymbol{\overline{\tau}^i}$ using the ℓ_1-norm:

$$\boldsymbol{p}^i = \frac{\boldsymbol{\overline{\tau}^i}}{\|\boldsymbol{\overline{\tau}^i}\|_1} \in \Delta^{|\mathcal{A}|}, \tag{10}$$

where $\|\boldsymbol{\overline{\tau}^i}\|_1$ denotes the ℓ_1-norm, ensuring that $\boldsymbol{p}^i$ lies within the probability simplex $\Delta^{|\mathcal{A}|} = \Big\{\boldsymbol{p}^i \in \mathbb{R}^{|\mathcal{A}|} \mid p^i_j \geq 0, \sum_j p^i_j = 1\Big\}$. Then,

$$\begin{aligned}
A &= \|\boldsymbol{\overline{\tau}^i}\|_1 \Big(\sum_{j\in\mathcal{A}} p^i_j 1_{\{r^i(x)\neq j\}} - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} p^i_j 1_{\{r^i(x)\neq j\}} \Big) \\
&\leq \|\boldsymbol{\overline{\tau}^i}\|_1 \Gamma^\nu \Big(\sum_{j\in\mathcal{A}} p^i_j \Phi^\nu_{01}(r^i, x, j) - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} p^i_j \Phi^\nu_{01}(r^i, x, j) \Big) \quad \text{(using Lemma 4)} \\
&= \|\boldsymbol{\overline{\tau}^i}\|_1 \Gamma^\nu \Big(\frac{1}{\|\boldsymbol{\overline{\tau}^i}\|_1} \Big[\sum_{j\in\mathcal{A}} \overline{\tau}^i_j(x) \Phi^\nu_{01}(r^i, x, j) - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} \overline{\tau}^i_j(x) \Phi^\nu_{01}(r^i, x, j) \Big] \Big) \\
&= \|\boldsymbol{\overline{\tau}^i}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(r^i, x)}{\|\boldsymbol{\overline{\tau}^i}\|_1} \Big)
\end{aligned} \tag{11}$$

Now, we have the following relationship:

$$B = \min_{j\in\mathcal{A}} \overline{c}^i_j(x) - \min_{j\in\mathcal{A}} \overline{c}^{i,*}_j(x) \leq \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] \tag{12}$$

Substituting the bound on B, we obtain:

$$\Delta\mathcal{C}^i_{\ell_{\text{def}}}(r^i, g^i, x) \leq \|\boldsymbol{\overline{\tau}^i}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(r^i)}{\|\boldsymbol{\overline{\tau}^i}\|_1} \Big) + \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] \tag{13}$$

Summing over i, we obtain

$$\Delta\mathcal{C}_{\ell_{\text{def}}}(r, g, x) \leq \sum_{i\in\{\text{start,end}\}} \Big[\|\boldsymbol{\overline{\tau}^i}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(r^i)}{\|\boldsymbol{\overline{\tau}^i}\|_1} \Big) + \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] \Big] \tag{14}$$

Using the fact that the function Γ is concave and that the *start* and *end* are conditionally independent given x:

$$\begin{aligned}
\Delta\mathcal{C}_{\ell_{\text{def}}}(r, g, x) \leq & \Bigg(\sum_{i\in\{\text{start,end}\}} \|\boldsymbol{\overline{\tau}^i}\|_1 \Bigg) \Gamma^\nu \Bigg(\frac{\Delta\mathcal{C}_{\text{def}}(r)}{\sum_{i\in\{\text{start,end}\}} \|\boldsymbol{\overline{\tau}^i}\|_1} \Bigg) \\
& + \sum_{i\in\{\text{start,end}\}} \Big[\mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{Y^i\sim\mathcal{D}(\cdot|X=x)}[c_0(x, Y^i)] \Big]
\end{aligned} \tag{15}$$

Taking expectation with respect to X yields the excess-risk form $\mathbb{E}_{X\sim\mathcal{D}_X}[\Delta\mathcal{C}_\ell] := \mathcal{E}_\ell - \mathcal{E}^B_\ell + \mathcal{U}_\ell$, which gives the desired result:

$$\begin{aligned}\mathcal{E}_{\ell_{\text{def}}}(g,r) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{G},\mathcal{R}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{G},\mathcal{R}) \leq \overline{\Gamma}^{\nu}\left(\mathcal{E}_{\Phi^{\nu}_{\text{def}}}(r) - \mathcal{E}^*_{\Phi^{\nu}_{\text{def}}}(\mathcal{R}) + \mathcal{U}_{\Phi^{\nu}_{\text{def}}}(\mathcal{R})\right) \\ + \sum_{i\in\{\text{start},\text{end}\}}\left(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i)\right),\end{aligned} \tag{16}$$

with $\overline{\Gamma}^{\nu}(u) = \left(\sum_{i\in\{\text{start},\text{end}\}}\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right)\Gamma^{\nu}\left(\frac{u}{\sum_{i\in\{\text{start},\text{end}\}}\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right)$ and from Mao et al. (2023b), it follows for $\nu \geq 0$ the inverse transformation $\Gamma^{\nu}(u) = \mathcal{T}^{-1,\nu}(u)$:

$$\mathcal{T}^{\nu}(u) = \begin{cases} \frac{2^{1-\nu}}{1-\nu}\left[1 - \left(\frac{(1+u)^{\frac{2-\nu}{2}} + (1-u)^{\frac{2-\nu}{2}}}{2}\right)^{2-\nu}\right] & \nu \in [0,1) \\ \frac{1+u}{2}\log[1+u] + \frac{1-u}{2}\log[1-u] & \nu = 1 \\ \frac{1}{(\nu-1)n^{\nu-1}}\left[\left(\frac{(1+u)^{\frac{2-\nu}{2}} + (1-u)^{\frac{2-\nu}{2}}}{2}\right)^{2-\nu} - 1\right] & \nu \in (1,2) \\ \frac{1}{(\nu-1)n^{\nu-1}}u & \nu \in [2,+\infty). \end{cases}$$

□

A.5 Proof Lemma 3

Lemma 3 (Bayes-Optimal Deferral Policy)**.** *Given $x \in \mathcal{X}$, the Bayes-optimal policy assigns the query to the agent $j \in \mathcal{A}$ minimizing the expected cumulative risk:*

$$\pi^*(x) = \arg\min_{j\in\mathcal{A}} \sum_{i\in\{start,end\}} \eta^i_j(x),$$

where $\eta^i_j(x) = \mathbb{E}_{Y^i\sim\mathcal{D}^i(\cdot|X=x)}[c_j(x,Y^i)]$ denotes the expected cost of assigning task i to agent j.

Proof. Fix $x \in \mathcal{X}$. Because start and end are conditionally independent given $X = x$, the conditional expected true-deferral cost when always deferring *both* sub-tasks to agent j is the *additive* conditional risk $\mathcal{C}_j(x) = \eta^{\text{start}}_j(x) + \eta^{\text{end}}_j(x)$. Let Π be the space of all measurable single-agent policies $\pi : \mathcal{X} \to \mathcal{A}$. For every $\pi \in \Pi$ the conditional expected loss at x is $\mathcal{C}_{\pi(x)}(x)$. Hence $\inf_{\pi\in\Pi}\mathcal{C}_{\pi(x)}(x) = \min_{j\in\mathcal{A}}\mathcal{C}_j(x)$, and the minimiser is any argument of that minimum. Defining $\pi^*(x)$ exactly as the arg min above therefore achieves the Bayes (i.e. pointwise minimum) conditional risk, which completes the proof. □

A.6 Proof Corollary 2

Corollary 2 (Learned allocation policy)**.** *Assume the per-task surrogate Φ^{ν}_{01} is strictly proper (Lemma 2), so it admits a strictly decreasing link $\psi\colon [0,1]\to\mathbb{R}$. Suppose further that the excess surrogate risk of the learned score function $\hat{r}$ satisfies*

$$\mathcal{E}_{\Phi^{\nu}_{\text{def}}}(\hat{r}) \;-\; \mathcal{E}^*_{\Phi^{\nu}_{\text{def}}}(\mathcal{R}_{all}) \xrightarrow[n\to\infty]{} 0.$$

Then, as the training-sample size n grows,

$$\Pr_{X\sim\mathcal{D}}\left[\hat{\pi}(X) \neq \pi^*(X)\right] \xrightarrow{n\to\infty} 0,$$

so the learned allocation rule

$$\hat{\pi}(x) = \arg\max_{j\in\mathcal{A}} \sum_{i\in\{start,end\}} \hat{r}^{\,i}(x,j),$$

converges in probability to the Bayes-optimal deferral policy of Lemma 3.

Proof. After training, each task-specific rejector outputs *scores* $\hat{r}^{\,i}(x, j) \in \mathbb{R},\ i \in \{\text{start}, \text{end}\},\ j \in \mathcal{A}$, and the inference-time decision for the two tasks jointly is

$$\hat{\pi}(x) \ = \ \arg\max_{j \in \mathcal{A}} \hat{S}_j(x), \qquad \hat{S}_j(x) = \sum_{i \in \{\text{start}, \text{end}\}} \hat{r}^{\,i}(x, j).$$

The $(\mathcal{G}, \mathcal{R})$–consistency (Theorem 1) implies that

$$\sup_{j,i,x} \left| \hat{r}^{\,i}(x, j) - \psi(\eta_j^i(x)) \right| \xrightarrow{p} 0 \tag{17}$$

Because ψ is strictly *decreasing*, maximising $\hat{r}^{\,i}(x, \cdot)$ is equivalent—w.p. $\to 1$—to minimising $\eta_j^i(x)$. Write $E_i(x)$ for this event.

On $E(x) = E_s(x) \cap E_e(x)$ we have, for any j_1, j_2,

$$\hat{S}_{j_1}(x) > \hat{S}_{j_2}(x) \iff R_{j_1}(x) < R_{j_2}(x),$$

hence $\hat{\pi}(x) = \pi^*(x)$ on $E(x)$.

By the union bound, $\Pr[\hat{\pi}(X) \neq \pi^*(X)] \leq \Pr[\neg E_s(X)] + \Pr[\neg E_e(X)] \to 0$. $\square$

(i) The link ψ is unique for strictly proper Φ_{01}^{ν}, hence no ambiguity arises. (ii) If surrogate optimisers are not uniquely defined, any tie-breaking rule yields the same convergence result provided it is deterministic. (iii) The corollary formally justifies the inference Algorithm 1 in the main paper: the single argmax over summed scores is asymptotically optimal.

A.7 Experiments

A.7.1 Few-Shot Demonstrations

We present the few-shot demonstrations used to prompt the Llama-3 family of models. Datasets such as SQuADv2 contain questions for which no answer appears in the provided context. In these cases, we instruct the model to return no output, represented by the symbol '¿.

1. **Demonstration 1:**

   Context:"The Eiffel Tower is located in Paris, France."
   Question: "Where is the Eiffel Tower?"
   Output: "Paris, France"

2. **Demonstration 2:**

   Context: "Albert Einstein developed the theory of relativity in the early 20th century."
   Question: "What did Albert Einstein develop?"
   Output: "the theory of relativity"

3. **Demonstration 3:**

   Context: "Marie Curie won the Nobel Prize in Physics in 1903 and in Chemistry in 1911."
   Question: "What year was Marie Curie born?"
   Output: "?"

4. **Demonstration 4:**

   Context: "The Great Wall of China was built to protect against invasions. It stretches over 13,000 miles."
   Question: "Who built the Great Wall of China?"
   Output: "?"

A.7.2 Agent Training and Performance Details

We train our models on a single NVIDIA H100 GPU and report results averaged over four independent runs. We train both ALBERT-Base and ALBERT-XXL offline and will publicly release their weights. We do not train Llama-3.2-1B or Llama-3-8B from scratch; instead, we use the publicly available *meta-llama* weights from HuggingFace without further training. For each dataset, we use the following hyperparameters:

Table 3: Hyperparameters for SQuADv1, SQuADv2, and TriviaQA.

Experts	Batch Size	Epochs	Learning Rate	Warm-up	Scheduler	Max Length	Stride
ALBERT-Base	32	2	5e-5	0.1	linear	384	128
ALBERT-XXL	32	2	5e-5	0.1	linear	384	128

We report the following performance metrics on a test set formed by subsampling 3,000 inputs from the validation set:

Table 4: Exact Match (EM) and F1 scores for each dataset.

Agents	SQuADv1	SQuADv2	TriviaQA
ALBERT-Base	84.20/90.63	77.10/79.54	86.63/90.86
ALBERT-XXL	89.37/94.91	84.07/86.57	91.63/94.21
Llama-3.2-1B	49.93/60.12	35.00/38.79	41.30/48.02
Llama-3-8B	67.80/80.22	59.47/66.47	48.47/56.66
Ensemble	84.60/90.80	81.06/84.19	88.84/91.78

Table 5: Computational efficiency of different models. We compare the number of parameters (in millions) and computational cost (in GFLOPs) for processing a sequence of length $L = 384$. The rejector is substantially lighter, with only 4.39M parameters and 0.15 GFLOPs, making it well suited to deployment in resource-constrained environments.

	Llama-3.2-1B	ALBERT-Base	ALBERT-XXL	Llama-3-8B	Rejector	Ensemble
Parameters (M)	1240	11.10	206	8030	4.39	1457.1
GFLOPs	373.66	32.68	928.08	2,680.06	0.15	1,334.42

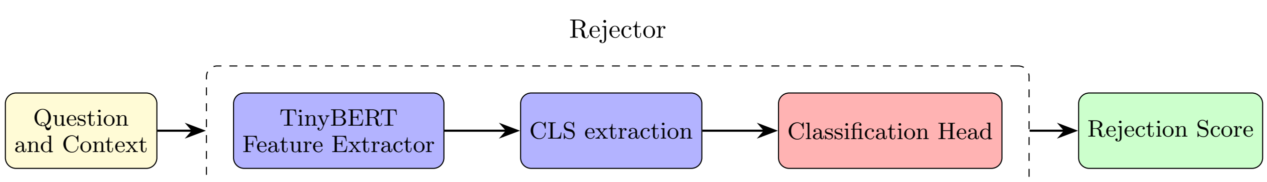


Figure 5: Rejector architecture. The input is processed by TinyBERT (Devlin et al., 2019), which serves as the feature extractor. The resulting CLS token is then used by the classification head to predict the allocation.

A.7.3 Rejector Training

Using the architecture illustrated in Figure 5, we train the rejector according to the surrogate deferral loss (SDL) defined in Lemma 2. Specifically, we adopt the standard multiclass log-softmax loss

$$\Phi_{01}^{\nu=1}(r^i, x, j) = -\log\left(\frac{e^{r^i(x,j)}}{\sum_{j' \in \mathcal{A}} e^{r^i(x,j')}}\right),$$

and optimize the rejector using the procedure detailed in Algorithm 1. For all datasets—SQuADv1, SQuADv2, and TriviaQA—we employ the following hyperparameters:

Table 6: Key training hyperparameters for L2D rejector.

lr	epochs	batch size	dropout	optimizer	warmup ratio	weight decay
1e−5	6	128	0.1	AdamW	0.1	0.001

For the query routers from (Ding et al., 2024), we use the following hyperparameters across all variants.

Table 7: Key training hyperparameters for r_{trans}, r_{prob} and r_{deter}.

lr	epochs	batch size	dropout	optimizer	warmup ratio	weight decay
1e−2	10	32	0.1	AdamW	0.1	0.001