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

Anonymous ACL submission

Abstract

Large Language Models excel in generative tasks but exhibit inefficiencies in structured text selection, particularly in extractive question answering. This challenge is magnified in resource-constrained environments, where deploying multiple specialized models for different tasks is impractical. We propose a *Learning-to-Defer* framework that allocates queries to specialized experts, ensuring high-confidence predictions while optimizing computational efficiency. Our approach integrates a principled allocation strategy with theoretical guarantees on optimal deferral that balances performance and cost. Empirical evaluations on SQuADv1, SQuADv2, and TriviaQA demonstrate that our method enhances answer reliability while significantly reducing computational overhead, making it well-suited for scalable and efficient EQA deployment.

1 Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing, achieving state-of-the-art performance across a wide range of tasks, including machine translation, summarization, and question answering (Touvron et al., 2023; Jiang et al., 2023; OpenAI et al., 2024). Their strong generalization capabilities stem from extensive pretraining on diverse corpora, allowing them to generate fluent and contextually relevant responses. However, while LLMs perform well on open-ended and generative tasks, they often struggle in structured scenarios that demand precise, extractive reasoning. A notable example is extractive question answering (EQA), where models must retrieve exact spans from a given context rather than generate free-form responses (Chen et al., 2017; Alqifari, 2019; Lan et al., 2020). In this setting, LLMs frequently exhibit hallucinations—producing plausible yet incorrect spans that deviate from the source text (Sadat et al., 2023).

This challenge is further exacerbated in **resource-constrained environments**, such as mobile devices, IoT systems, and on-device assistants (Merenda et al., 2020), where computational efficiency is paramount. Deploying large LLMs in such settings is impractical due to their substantial memory and energy requirements. A natural solution is to use **smaller LLMs**, which offer a more efficient alternative. However, while these models perform well in general settings, they often fail on high-precision tasks such as EQA, where exact information retrieval is required. This trade-off between efficiency and accuracy creates a fundamental bottleneck: increasing model size improves task performance but is infeasible for small-device deployment, while reducing model size preserves efficiency but degrades reliability on critical tasks. Addressing this limitation requires a selective approach—one that retains the efficiency of small LLMs while ensuring robust performance on specialized tasks.

To this end, we propose a Learning-to-Defer framework (Madras et al., 2018; Mozannar and Sontag, 2021; Verma et al., 2022; Mao et al., 2023a), which adaptively delegates queries to specialized offline models, optimizing both accuracy and efficiency. Rather than relying solely on a small LLM—which may produce unreliable answers for complex queries—our method dynamically defers difficult cases to more capable, task-specific models. This strategy enables **efficient, on-device processing** for the majority of queries while leveraging specialized models only when necessary. **We establish theoretical guarantees on the optimality of our learned allocation strategy**, ensuring that queries are directed to the agent with the highest confidence. Empirically, we evaluate our approach on multiple EQA benchmarks, including SQuADv1 (Rajpurkar et al., 2016), SQuADv2 (Rajpurkar et al., 2018), and TriviaQA (Joshi et al., 2017), demonstrating that our method significantly

improves reliability while maintaining low computational overhead.

2 Related Work

Model Cascades. Model cascades (Viola and Jones, 2001; Jitkrittum et al., 2024; Saberian and Vasconcelos, 2014) sequentially pass a given query to the next model in the cascade based on a scoring criterion and a predefined threshold. The development of these criteria depends on the trade-off between cost and performance. While existing approaches have explored applying cascades to LLMs, they are not specifically designed for EQA (Kolawole et al., 2024; Yue et al., 2023). Additionally, many existing methods employ criteria that may not easily support cascades composed of specialist EQA models and free-form LLMs, which produces different output structures (Varshney and Baral, 2022). Moreover, increasing the number of LLMs in the cascade inevitably leads to higher inference latency, and the optimal model is not always selected immediately. A refinement of traditional cascades is *Agreement-Based Cascading* (Narasimhan et al., 2024), which leverages ensembles at each level of the cascade to determine whether a query should be forwarded to the next stage. This approach improves robustness but still inherits the limitations of sequential inference.

Query Routing. Query routing (Ding et al., 2024; Ong et al., 2024; Kag et al., 2023; Ding et al., 2022) aims to balance cost and quality by leveraging the strengths of a more expensive yet stronger LLM alongside a cheaper but weaker one. Typically, a lightweight routing model is trained to assign queries directly to either the small or large model based on predicted task difficulty and the desired quality level. These routing strategies are particularly useful for optimizing performance in edge-based LLM deployments. However, most existing approaches are constrained to binary routing decisions, where only two models are available for handling a task.

Our Contribution. We introduce a novel setting that integrates both the *multi-model nature of cascades* and the *direct allocation strategy of routers*. Our goal is to determine an optimal direct allocation of EQA queries across multiple models with varying point-wise performance and cost trade-offs. The distinctions between existing approaches and our proposed method are illustrated in Figure 5.

To the best of our knowledge, our approach is the first to address this specific problem setting. By enabling more fine-grained cost optimization and routing queries to multiple specialized models, potentially including non-LLM models, our method enhances adaptability and allows for more effective handling of diverse tasks.

3 Preliminaries

Extractive QA Setting. We consider an *extractive question-answering* (EQA) system, which takes as input a question q and a corresponding context c and extracts an answer a directly from c. Formally, the input is denoted as $x = (q, c) \in \mathcal{X}$, where each instance x has an associated ground truth label $y \in \mathcal{Y}$. The label consists of the *start* and *end* token indices, expressed as $y = (y^{\mathrm{s}}, y^{\mathrm{e}})$, such that the extracted answer is always a contiguous substring of c. Consequently, the label space is defined as $\mathcal{Y} = \mathcal{Y}^{\mathrm{s}} \times \mathcal{Y}^{\mathrm{e}}$. For brevity, we define a data point as $z = (x, y) \in \mathcal{Z}$. When referencing individual label components, we use $z^{\mathrm{s}} = (x, y^{\mathrm{s}})$ and $z^{\mathrm{e}} = (x, y^{\mathrm{e}})$, corresponding to the start and end token labels, respectively. Following previous studies (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2020), we assume that the *start* and *end* token labels are conditionally independent given $x \in \mathcal{X}$. Furthermore, we assume that the data points are independently and identically distributed (i.i.d.) under an underlying distribution $\mathcal{D}$ over $\mathcal{Z}$ (Mohri et al., 2012).

To model the answer extraction process, we define a *backbone* $w \in \mathcal{W}$ as a feature extractor that maps an input x to a latent representation $t = w(x)$, such that $w : \mathcal{X} \to \mathcal{T}$. The extracted representation t is then processed by a *classifier* $h \in \mathcal{H}$, which consists of two components, $h = (h^{\mathrm{s}}, h^{\mathrm{e}})$, each responsible for predicting the start and end positions of the answer span. Specifically, for $i \in \{\mathrm{s}, \mathrm{e}\}$, each classifier head h^i is a scoring function $h^i : \mathcal{T} \times \mathcal{Y}^i \to \mathbb{R}$, producing predictions according to $h^i(t) = \arg\max_{y \in \mathcal{Y}^i} h^i(t, y)$. The full prediction model, denoted as $g \in \mathcal{G}$, is defined as the composition of the feature extractor and classifier, i.e., $g(x) = h \circ w(x)$, where $g^{\mathrm{s}}(x)$ and $g^{\mathrm{e}}(x)$ are the *start* and *end* predictions, respectively. The function space is then given by $\mathcal{G} = \{g \mid g(x) = h \circ w(x),\ w \in \mathcal{W},\ h \in \mathcal{H}\}$.

Typically, EQA systems are trained using a *true multiclass* 0-1 loss, which measures the number of mispredictions made by the model across the

start and *end* token positions. Formally, this loss function is defined as $\ell_{01}^{\mathrm{s,e}} : \mathcal{G} \times \mathcal{Z} \to \{0, 1, 2\}$, and takes the form:

$$\ell_{01}^{\mathrm{s,e}}(g, z) = \sum_{i \in \{\mathrm{s,e}\}} \ell_{01}(g^i, z^i). \tag{1}$$

This loss penalizes the model by counting the number of incorrect start or end token predictions, providing a discontinuous but interpretable measure of model performance.

Consistency in Classification: Let $i \in \{\mathrm{s}, \mathrm{e}\}$. The primary goal is to learn a classifier $g^i \in \mathcal{G}^i$ that minimizes the true error $\mathcal{E}_{\ell_{01}}(g^i)$, defined as $\mathcal{E}_{\ell_{01}}(g^i) = \mathbb{E}_{z^i}[\ell_{01}(g^i, z^i)]$. The Bayes-optimal error is given by $\mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) = \inf_{g^i \in \mathcal{G}^i} \mathcal{E}_{\ell_{01}}(g^i)$. However, directly minimizing $\mathcal{E}_{\ell_{01}}(g^i)$ is challenging due to the non-differentiability of the *true multiclass* 0-1 loss ℓ_{01} (Zhang, 2002; Steinwart, 2007; Awasthi et al., 2022).

To address this, the cross-entropy *multiclass surrogate* family, denoted by $\Phi^\nu_{01} : \mathcal{G}^i \times \mathcal{X} \times \mathcal{Y}^i \to \mathbb{R}^+$, provides a convex upper bound to ℓ_{01}. This family is parameterized by $\nu \geq 0$ and includes widely used surrogate losses such as MAE for $\nu = 2$ (Ghosh et al., 2017) and log-softmax (Mohri et al., 2012) for $\nu = 1$, defined as:

$$\Phi^\nu_{01} = \begin{cases} \frac{1}{1-\nu}\left(\Psi(g^i, z^i)^{1-\nu} - 1\right) & \nu \neq 1, \\ \log\left(\Psi(g^i, z^i)\right) & \nu = 1, \end{cases} \tag{2}$$

with $\Psi(g^i, x, y^i) = \sum_{y' \in \mathcal{Y}^i} e^{g^i(x,y') - g^i(x,y^i)}$. The corresponding surrogate error is $\mathcal{E}_{\Phi^\nu_{01}}(g^i) = \mathbb{E}_{z^i}[\Phi^\nu_{01}(g^i, z^i)]$, with its optimal value given by $\mathcal{E}^*_{\Phi^\nu_{01}}(\mathcal{G}^i) = \inf_{g^i \in \mathcal{G}^i} \mathcal{E}_{\Phi^\nu_{01}}(g^i)$.

A crucial property of a surrogate loss is *Bayes-consistency*, ensuring that minimizing the surrogate error also minimizes the true error (Zhang, 2002; Steinwart, 2007; Bartlett et al., 2006; Tewari and Bartlett, 2007). Formally, Φ^ν_{01} is Bayes-consistent with respect to ℓ_{01} if, for any sequence $\{g^i_k\}_{k \in \mathbb{N}} \subset \mathcal{G}^i$, the following holds for the *true* and *surrogate excess risk*:

$$\begin{aligned} &\mathcal{E}_{\Phi^\nu_{01}}(g^i_k) - \mathcal{E}^*_{\Phi^\nu_{01}}(\mathcal{G}^i) \xrightarrow{k\to\infty} 0 \\ &\implies \mathcal{E}_{\ell_{01}}(g^i_k) - \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) \xrightarrow{k\to\infty} 0. \end{aligned} \tag{3}$$

This assumption holds under $\mathcal{G}^i = \mathcal{G}^i_{\text{all}}$, but not necessarily for restricted hypothesis classes like $\mathcal{G}^i_{\text{lin}}$ or $\mathcal{G}^i_{\text{ReLU}}$ (Long and Servedio, 2013; Awasthi et al., 2022). To mitigate this limitation, Awasthi et al. (2022) proposed $\mathcal{G}^i$-consistency bounds, which depend on a non-decreasing function $\Gamma : \mathbb{R}^+ \to \mathbb{R}^+$ and take the form:

$$\begin{aligned} &\mathcal{E}_{\Phi^\nu_{01}}(g^i) - \mathcal{E}^*_{\Phi^\nu_{01}}(\mathcal{G}^i) + \mathcal{U}_{\Phi^\nu_{01}}(\mathcal{G}^i) \geq \\ &\Gamma\Big(\mathcal{E}_{\ell_{01}}(g^i) - \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) + \mathcal{U}_{\ell_{01}}(\mathcal{G}^i)\Big), \end{aligned} \tag{4}$$

where the minimizability gap $\mathcal{U}_{\ell_{01}}(\mathcal{G}^i)$ quantifies the difference between the best-in-class generalization error and the expected pointwise minimum error

$$\mathcal{U}_{\ell_{01}}(\mathcal{G}^i) = \mathcal{E}^B_{\ell_{01}}(\mathcal{G}^i) - \mathbb{E}_x\Big[\inf_{g^i \in \mathcal{G}^i} \mathbb{E}_{y^i|x}[\ell_{01}(g^i, z^i)]\Big].$$

Notably, this minimizability gap vanishes when $\mathcal{G}^i = \mathcal{G}^i_{\text{all}}$ (Steinwart, 2007; Awasthi et al., 2022). In the asymptotic limit, inequality (4) ensures the recovery of Bayes-consistency, aligning with (3).

4 Optimal Allocation for EQA systems

In this section, we formalize the problem of allocating queries $x \in \mathcal{X}$ among multiple agents, including the main model g and J expert models. Crucially, we demonstrate that our formulation facilitates the learning of an optimal allocation strategy, thereby ensuring asymptotic optimality in performance.

4.1 Formulating the Allocation Problem

Setting: We consider a main model $g \in \mathcal{G}$ and J distinct experts, each available on demand. We collectively refer to the main model g and the J experts as *agents*. The agent space is defined as $\mathcal{A} = \{0\} \cup [J]$, where the cardinality is $|\mathcal{A}| = J+1$, representing the total number of agents in the system. We assume that all agents have been pretrained offline, and our focus is on the allocation of queries among them (Mao et al., 2023a, 2024; Montreuil et al., 2024, 2025). When an expert M_j is queried for an input x, it generates two outputs: a *start* span $m^{\mathrm{s}}_j(x) \in \mathcal{Y}^{\mathrm{s}}$ and an *end* span $m^{\mathrm{e}}_j(x) \in \mathcal{Y}^{\mathrm{e}}$. These experts may be human annotators, AI models, or other decision-making systems capable of predicting both spans. We aggregate the predictions of all experts into the variable $m(x) = (m_1(x), \ldots, m_J(x)) \in \mathcal{M}$.

True Deferral Loss: To learn allocations among multiple *agents*, we define a rejector $r \in \mathcal{R}$ that determines the *agent* to which a given query $x \in \mathcal{X}$ should be assigned. We decompose this rejector into two components: the *start* rejector $r^s \in \mathcal{R}^s$

and the *end* rejector $r^e \in \mathcal{R}^e$. Each rejector $r^i \in \mathcal{R}^i$ for $i \in \{s, e\}$ is defined as $r^i : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and assigns the query according to the rule $r^i(x) = \arg\min_{j \in \mathcal{A}} r^i(x, j)$. To learn these rejectors $r \in \mathcal{R}$, we introduce the *true deferral loss*, adapted from (Mao et al., 2023a) for standard classification tasks.

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*

$$\ell_{def} = \sum_{i \in \{s,e\}} \sum_{j=0}^{J} c_j(g^i(x), m_j^i(x), z^i) 1_{\{r^i(x)=j\}},$$

where the cost function c_j quantifies the penalty associated with agent misclassification. Specifically, the cost incurred when relying on the main model g is defined as $c_0(g^i(x), z^i) = 1_{\{g^i(x) \neq y^i\}}$. Similarly, the cost of consulting expert j is given by $c_{j>0}(m_j^i(x), z^i) = \alpha_j c_0(m_j^i(x), z^i) + \beta_j$, where $\alpha_j \geq 0$ and $\beta_j \geq 0$ accounts for the additional expense of querying expert j. Notably, setting $\alpha_j = 0$ corresponds to evaluating the main model g against an oracle (a perfectly correct expert) while still incurring an additional querying cost (Chow, 1970; Cortes et al., 2016).

The rejector function $r^i \in \mathcal{R}$ determines the allocation of queries. If $r^i(x) = 0$, the query is assigned to the main model g, which produces the prediction $g^i(x)$. Otherwise, if $r^i(x) = j$ for $j > 0$, the query is deferred to expert j, yielding the prediction $m_j^i(x)$.

An important question remains: how is the deferral decision made? The rejector $r^i(x)$ must balance predictive accuracy and the cost of expert consultation. An effective deferral strategy should minimize overall prediction errors while limiting unnecessary expert queries.

4.2 Optimality of the Allocation

Ideally, the query $x \in \mathcal{X}$ should be allocated to the agent with the highest confidence in its prediction (Madras et al., 2018), thereby improving the reliability and trustworthiness of the system. To formalize this decision-making process, we analyze the optimal risk associated with our *true deferral loss* and characterize the *Bayes-rejector*, which defines the optimal rejection strategy in our framework.

Given the conditional probability distribution $\mathcal{D}(\cdot|X = x)$, we denote the main model's confidence as $\eta_0^i(x) = \mathcal{D}(g^i(x) \neq y^i|X = x)$, and the confidence of expert j as $\eta_j^i(x) = \alpha_j \mathcal{D}(m_j^i(x) \neq y^i|X = x) + \beta_j$. We introduce the following Lemma 1:

Lemma 1 (Bayes-Rejector)**.** *Given an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$, the optimal rejection rule that minimizes the risk associated with the true deferral loss is given by:*

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

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

We provide a proof of this relationship in Appendix C. Lemma 1 suggests that minimizing the *true deferral loss* defined in Definition 1 leads to an optimal decision rule that compares agents' confidence levels. If the main model exhibits higher confidence than the most reliable expert, i.e., if $\eta_0^i(x) \leq \min_{j \in [J]} \eta_j^i(x)$, the query $x \in \mathcal{X}$ is assigned to the main model g. Conversely, if there exists an expert j with the lowest expected risk, the query is deferred to this expert. This deferral mechanism ensures that queries are allocated to the most confident agent in the system, thereby enhancing the reliability and trustworthiness of the allocation process.

Current issue: Learning the Bayes-rejector is a well-established NP-hard problem (Zhang and Agarwal, 2020; Steinwart, 2007; Bartlett et al., 2006; Mohri et al., 2012), primarily due to the discontinuity of the *true deferral loss*. This challenge is prevalent in machine learning, where optimizing discontinuous loss functions is notoriously difficult (Cortes et al., 2016; Mao et al., 2023b). In the following subsection, we present an approach to accurately approximate this deferral rule while preserving theoretical guarantees.

4.3 Accurate Approximation of the True Deferral Loss

To effectively approximate the *true deferral loss* while preserving the optimality of the decision rule in Lemma 1, we leverage key concepts from consistency theory, as defined in Preliminaries 3. A standard approach in statistical learning is to introduce a *surrogate loss* that serves as a differentiable proxy for a target loss—in this case, the *true deferral loss*. Our goal is to construct a *surrogate loss* that is both Bayes-consistent and $(\mathcal{G}, \mathcal{R})$-consistent, ensuring that minimizing this loss results in a learned rejector $r^{*,i}$ that closely approximates the Bayes-rejector defined in Lemma 1. This guarantees that,

as the surrogate loss is minimized, **the learned decision rule asymptotically approaches the optimal rejection strategy**.

Formulating the Surrogate Deferral Loss. To construct this surrogate loss, we introduce a new hypothesis $\overline{r}^i \in \overline{\mathcal{R}}^i$, where $\overline{r} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. The first dimension is set to a zero-scoring function, $\overline{r}^i(x, 0) = 0$, and is compared against the remaining scores, defined as $\overline{r}^i(x, j) = -r^i(x, j)$ for expert indices $j \in [J]$. We leverage the cross-entropy *multiclass surrogate* family, denoted by $\Phi^{\nu}_{01} : \overline{\mathcal{R}}^i \times \mathcal{X} \times \mathcal{A} \to \mathbb{R}^+$, for the *true multiclass loss*. Adapting the approach introduced by Mao et al. (2023a) to our setting, we define the *surrogate deferral loss* as a proxy for the *true deferral loss*. This formulation enables a structured comparison between the rejection model and the expert alternatives. We now introduce the following definition:

Definition 2 (Surrogate Deferral Loss)**.** *Given an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$, the surrogate loss for the true deferral loss is defined as:*

$$\Phi^{\nu}_{def} = \sum_{i \in \{s,e\}} \sum_{j=0}^{J} \tau_j(g^i(x), m^i_j(x), z^i) \Phi^{\nu}_{01}(\overline{r}^i, x, j),$$

where $\tau_j = 1 - c_j$ for $j \in \mathcal{A}$.

This surrogate formulation ensures that minimizing Φ^{ν}_{def} aligns with the objective of minimizing the *true deferral loss*, while maintaining desirable optimization properties such as differentiability and convexity under suitable conditions. A key advantage of this surrogate loss is that it enables gradient-based optimization, making it compatible with standard deep learning frameworks (Bartlett et al., 2006).

To further analyze the properties of Φ^{ν}_{def}, we study its Bayes consistency and its $(\mathcal{G}, \mathcal{R})$-consistency, ensuring that a minimizer of the surrogate loss recovers a rejector $r^{*,i}$ that closely approximates the Bayes-optimal rejector. In the following subsection, we derive theoretical guarantees for the surrogate loss and discuss its implications for model training.

4.4 Theoretical Guarantees of the Surrogate Deferral Loss

Proving Consistency Properties: In the previous subsection, we introduced the *surrogate deferral loss* as a proxy for approximating the *true deferral loss*. Our goal is to establish that minimizing the surrogate excess risk $\mathcal{E}_{\Phi^{\nu}_{\text{def}}}(r) - \mathcal{E}^*_{\Phi^{\nu}_{\text{def}}}(\mathcal{R}) + \mathcal{U}_{\Phi^{\nu}_{\text{def}}}(\mathcal{R})$ leads to minimizing the true excess risk $\mathcal{E}_{\ell_{\text{def}}}(r, g) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{R}, \mathcal{G}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{R}, \mathcal{G})$. Establishing this relationship implies that the learned rejector r^* will closely approximate the Bayes-optimal rejector r^B, thereby ensuring an optimal allocation of queries, as stated in Lemma 1.

Theorem 1 ($(\mathcal{R}, \mathcal{G})$–consistency)**.** *Given 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 $\mathcal{R}$-consistency bounds hold for any distribution $\mathcal{D}$:*

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

then for any $(g, r) \in \mathcal{G} \times \mathcal{R}$, any distribution $\mathcal{D}$ and any $x \in \mathcal{X}$,

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

with the expected cost vector $\overline{\boldsymbol{\tau}}^{\boldsymbol{i}} = \{\mathbb{E}_{y^i|x}[\tau^i_j]\}_{j \in \mathcal{A}}$ and $\overline{\Gamma}^{\nu}(u) = \left(\sum_{i \in \{s,e\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right) \Gamma^{\nu}\left(\frac{u}{\sum_{i \in \{s,e\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right)$.

The proof of Theorem 1, along with additional bounds for $\nu \geq 0$, is provided in Appendix D. It is reasonable to assume that at the end of training, the *surrogate deferral excess risk* has been minimized, leading to the bound $\mathcal{E}_{\Phi^{\nu}_{\text{def}}}(r) - \mathcal{E}^*_{\Phi^{\nu}_{\text{def}}}(\mathcal{R}) + \mathcal{U}_{\Phi^{\nu}_{\text{def}}}(\mathcal{R}) \leq \epsilon_0$. Since the model g has been trained offline, it is mild to assume that the c_0-excess risk satisfies $\sum_{i \in \{\text{s,e}\}} \left(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i)\right) \leq \epsilon_1$. This result implies that the left-hand side is bounded above, yielding the inequality

$$\mathcal{E}_{\ell_{\text{def}}}(g, r) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) \leq \epsilon_1 + \left(\sum_{i \in \{\text{s,e}\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right) \overline{\Gamma}^{\nu}(\epsilon_0). \tag{5}$$

By leveraging properties of $\overline{\Gamma}^{\nu}$, we have established that minimizing the *surrogate deferral loss* effectively leads to minimizing the *true deferral loss*. Using standard arguments from statistical learning theory (Steinwart, 2007; Mao et al., 2023b; Awasthi et al., 2022), Theorem 1 further implies Bayes-consistency when considering the hypothesis spaces $\mathcal{R} = \mathcal{R}_{\text{all}}$ and $\mathcal{G} = \mathcal{G}_{\text{all}}$.

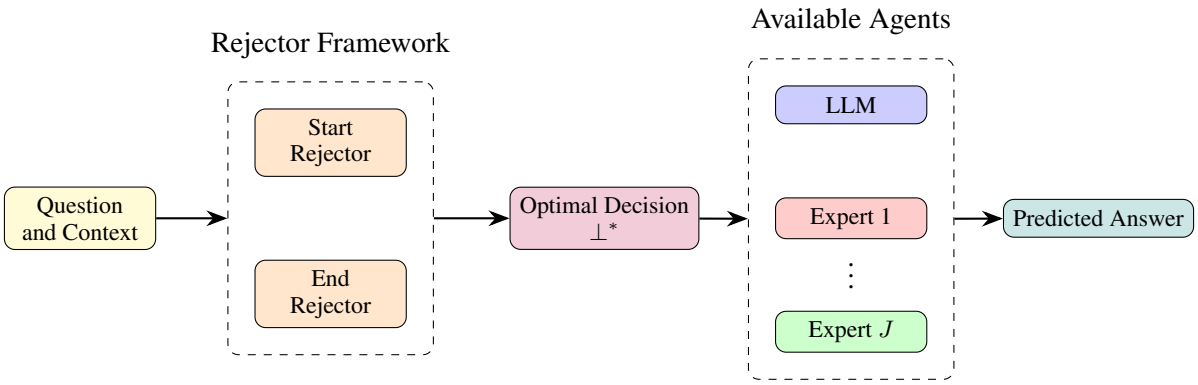

Figure 1: Inference Step of Our Approach: The input data is processed through the rejector framework, which predicts both *start* and *end* spans. Based on the optimal rule defined in Equation 2, the query is assigned to an agent that subsequently predicts the answer.

Implications: Our theoretical guarantees establish that the learned rejector r^i follows the same optimal deferral rule as defined in Lemma 1. Specifically, the learned rule is given by $r^i(x) = \arg\min_{j \in \mathcal{A}} r^i(x, j)$. This deferral rule independently allocates the *start* and *end* extraction decisions. However, in scenarios where the query $x \in \mathcal{X}$ must be deferred to a single *agent*, an optimal unified deferral rule $\perp^*(x)$ is applied:

Lemma 2 (Optimal Deferral Rule for Single Allocation)**.** *Let $x \in \mathcal{X}$ and any distribution $\mathcal{D}$. Assigning the query to a single agent leads to the following optimal decision rule:*

$$\perp^*(x) = \arg\min_{j \in \mathcal{A}} \sum_{i \in \{s,e\}} r^{*,i}(x, j).$$

At the optimum, $r^{*,i}$ follows the deferral rule prescribed in Lemma 1. Consequently, the following equivalence holds:

$$\perp^*(x) \approx \begin{cases} 0, & \text{if } \sum_{i \in \{\mathrm{s,e}\}} \eta_0^i(x) \leq \min_{j \in [J]} \sum_{i \in \{\mathrm{s,e}\}} \eta_j^i(x), \\ j, & \text{otherwise.} \end{cases}$$

This formulation ensures that allocation is directed to the most confident agent across both the *start* and *end* spans, thereby preserving optimality in the allocation process.

5 Evaluation

In this section, we evaluate our approach on three widely used question-answering benchmarks: SQuADv1 (Rajpurkar et al., 2016), SQuADv2 (Rajpurkar et al., 2018), and TriviaQA (Joshi et al., 2017). Our experiments demonstrate that while LLMs generally perform well on broad question-answering tasks, they struggle with EQA. Therefore, incorporating a system where experts are available on demand significantly improves the overall performance of the system. Settings of our experiment can be found in E.2. We ensure the reproducibility of our results by making our implementation publicly available. We provide algorithms for both training and inference in Appendix 1 and 2.

Agents: In the context of small devices, we select Llama-3.2-1B as our primary model (Touvron et al., 2023), as it offers strong performance across various tasks while remaining computationally feasible for low-resource settings. To showcase the ease of integrating our approach into existing LLMs, we use the publicly available Llama-3.2-1B base weights without additional training. We use two expert models with distinct computational capacities: M_1, ALBERT-Base and M_2, is the more computationally demanding ALBERT-XXL (Lan et al., 2020). Although the specialist models from the ALBERT family offer lower computational cost and superior performance on EQA tasks, they lack the generality of Llama-3.2-1B, making them unsuitable to be the on-device/main model g that should have task-agnostic performance.

Cost: We model our agents' costs for $i \in \{\mathrm{s,e}\}$ as $c_0(g^i(x), y^i) = 1_{\{g^i(x) \neq y^i\}}$, leading to $c_{j>0}(m^i(x), y^i) = c_0(m^i(x), y^i) + \beta_j$, with $\beta_j \geq 0$. Frequent expert queries can significantly increase latency and resource consumption, making the model less suitable for real-time or resource-constrained environments. The consultation cost β_j penalizes experts to prevent excessive query-

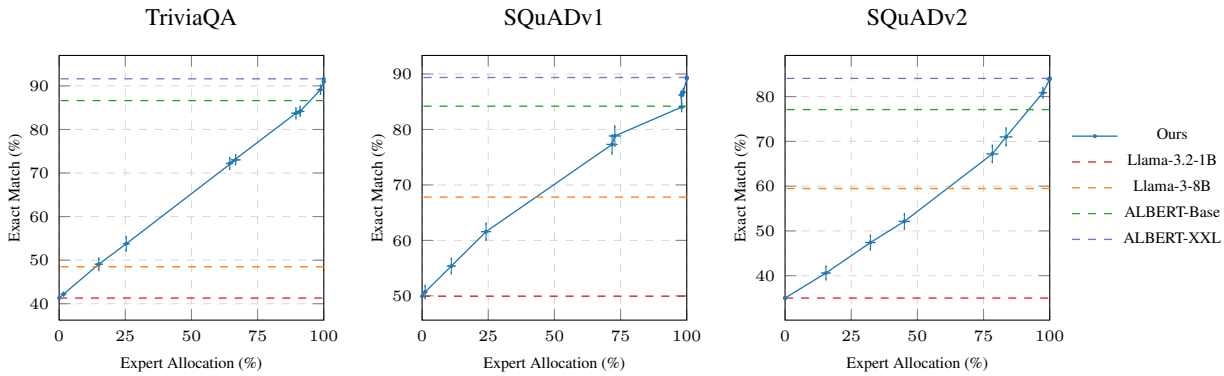

Figure 2: Comparison between the Exact Match metric and the Expert Allocation: (a) TriviaQA, (b) SQuADv1, (c) SQuADv2.

ing, reflecting the fact that querying across a set of offline experts should be done while considering the cost-performance tradeoff. Across the experts, we choose the ratio $R = \frac{\text{GFLOPs } (M_2)}{20 \text{ GFLOPs } (M_1)}$ and let $\beta_2 = R\beta_1$. This represents how the cost should scale with the relative computational complexity of the expert models.

Rejector: To efficiently allocate queries among the system's agents, we employ a highly lightweight architecture specifically designed for small-device deployment (Fig. 6). We utilize TinyBERT architecture (Devlin et al., 2018) to train our rejector. This contains only 4.39M parameters —just 0.35% of the main Llama-3.2-1B model, making it suitable for low-compute deployment.

Benchmark: We chose to benchmark our performance using vote-based ensembles (Breiman, 1996; Trad and Chehab, 2024). This closely mirrors our setting by providing supports multiple different models while providing direct allocation. However, ensembles do so by querying all models in parallel. We are interested in observing the difference in efficiency between our direct allocation and such a approach. We also benchmark against a larger model from the Llama-3 family, Llama-3-8B (Grattafiori et al., 2024), highlighting that our method, which utilizes the more compact 1B variant, not only matches but often outperforms the larger model. This is particularly significant in emphasizing that our smaller 1B model overcomes the challenges of deploying a higher performing 8B model on edge devices whilst facing no performance loss. When prompting both Llama-3.2-1B and Llama-3-8B, we employ few-shot demonstrations (Brown et al., 2020), with the specific demonstrations detailed in E.1.

Metrics: We measure performance on EQA using Exact Match (EM). We emphasis specialist models, although stronger in performance, are not suitable candidates for g. GLOPs/EM ratio is another metric which measures the computational cost we are paying for a performance gain. This is important in displaying the efficiency of our allocation strategy. We track allocation ratios, this represents the percentage of queries allocated to the experts, displaying how r is effectively taking cost into consideration when developing allocation decisions Lastly, we also take True Positive/False Positive Rate (TPR/FPR) into consideration. A TP outcome occurs when the model is incorrect and we successfully defer to an accurate expert. An FP outcome occurs when the query is allocated to an incorrect expert while the model is correct.

Results: Expert allocation refers to the percentage deferrals to experts, be it M_1 or M_2. Each expert allocation corresponds to a β_1 cost selection, the relationship between the two is displayed in 4. Additionally, we report detailed experimental results within E.2.

Performance: From Figure 2, we observe that our approach is able to outperform or match the Llama 3 family models across all datasets with appropriate expert allocation. This emphasizes the importance of having a system that allows for expert involvement. It also shows that our approach can improves performance of smaller edge-based LLM on EQA tasks whilst allowing for performance comparable to a otherwise larger LLM.

Efficiency: From Figure 3, we observe that with the exception of ALBERT-Base, our approach maintains the best computational efficiency across all datasets. We observe from 4 that our approach

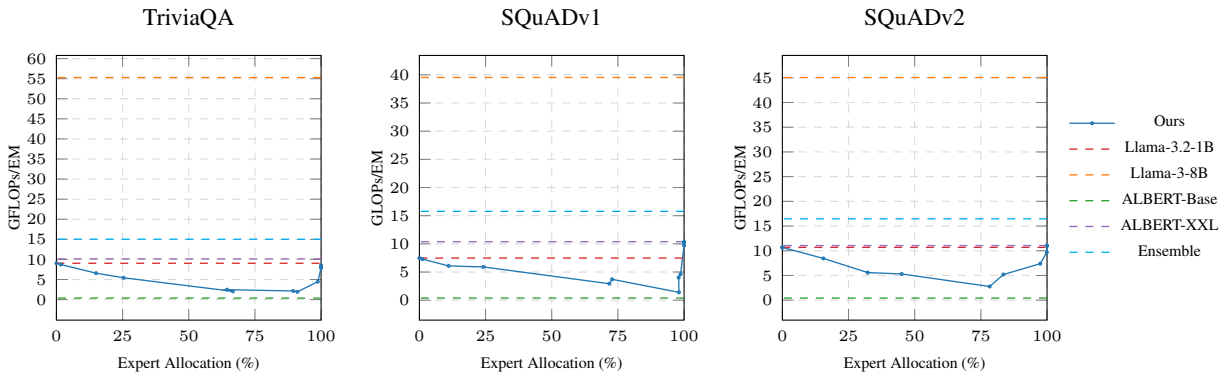

Figure 3: Combined Efficiency Comparison across benchmarks: (a) TriviaQA, (b) SQuADv1, (c) SQuADv2.

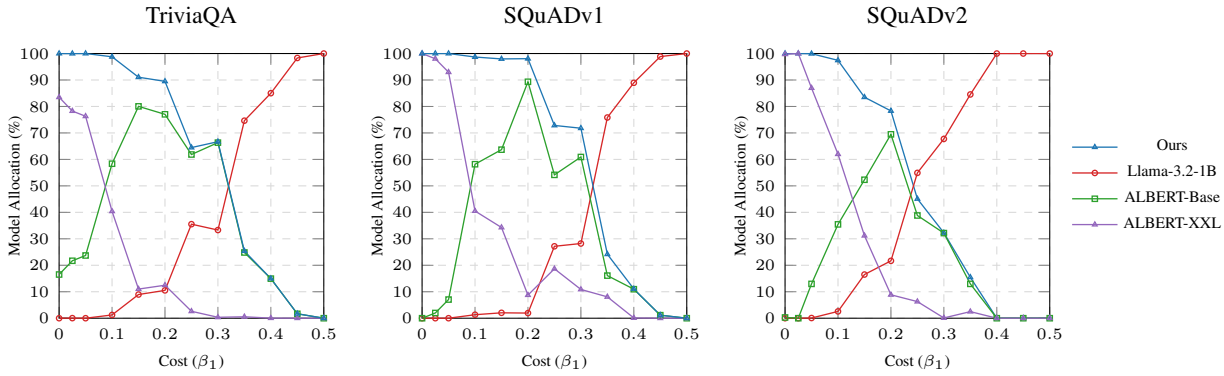

Figure 4: Combined Allocation Percentage across benchmarks: (a) TriviaQA, (b) SQuADv1, (c) SQuADv2.

is able to defer to the cheaper ALBERT-Base by increasing β_1. We note that when β_1 is set to extremes of 0.5 and 0, we are able to create scenarios to discourage allocations to g and M_2 respectively. While not significant from a performance standpoint, this observation proves that our rejector framework is able to successfully learn the cost distribution and factor this in when allocating queries, resulting in a more efficient system. Although we use GFLOPs to develop our ratio and evaluate the approach, it would be interesting for users of our approach to experiment with supplementing or developing β cost distributions based on network latency & cloud-related cost. Metrics which could better mirror real world deployment requirements.

6 Conclusion

In this work, we introduced a novel Learning-to-Defer framework for extractive question answering that dynamically allocates queries to the most suitable agent, optimizing both accuracy and computational efficiency. By leveraging theoretical guarantees, our method effectively balances performance and cost, making it well-suited for deployment in resource-constrained environments. Empirical evaluations on standard EQA benchmarks, including SQuADv1, SQuADv2, and TriviaQA, demonstrated that our approach enhances reliability while reducing computational overhead, outperforming larger LLMs and ensemble methods in both effectiveness and efficiency.

Limitations

Despite the demonstrated effectiveness of our Learning-to-Defer framework for optimal query allocation in EQA, several limitations remain.

Task Generalization and Theoretical Guarantees: Our framework is specifically designed for extractive question answering, where answers correspond to contiguous spans in a given context. The structured nature of EQA enables well-defined loss functions and confidence-based allocation criteria, allowing us to establish theoretical guarantees on optimal query allocation. However, these guarantees do not directly extend to more complex NLP tasks such as generative QA, multi-hop reasoning, or open-domain retrieval, where outputs are not constrained to predefined spans. The lack of structured outputs in these tasks introduces additional

challenges in defining optimal deferral strategies and ensuring theoretical consistency. Future work should explore whether similar optimality guarantees can be formulated for these broader settings, potentially requiring new loss functions and deferral mechanisms.

Deferral Cost Estimation: The deferral mechanism relies on predefined cost parameters β_j to regulate expert consultation. However, there is currently no explicit and effective method to dynamically track how query allocation varies across agents in response to changes in these cost parameters.

References

Reem Alqifari. 2019. Question answering systems approaches and challenges. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 69–75, Varna, Bulgaria. INCOMA Ltd.

Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2022. Multi-class h-consistency bounds. *Advances in Neural Information Processing Systems*, 35:782–795.

Peter Bartlett, Michael Jordan, and Jon McAuliffe. 2006. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

C. Chow. 1970. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Learning with rejection. In *Algorithmic Learning Theory*, pages 67–82, Cham. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Aritra Ghosh, Himanshu Kumar, and P. Shanti Sastry. 2017. Robust loss functions under label noise for deep neural networks. *ArXiv*, abs/1712.09482.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick

Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. 2024. When does confidence-based cascade deferral suffice? *Preprint*, arXiv:2307.02764.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Anil Kag, Igor Fedorov, Aditya Gangrade, Paul Whatmough, and Venkatesh Saligrama. 2023. Efficient edge inference by selective query. In *The Eleventh International Conference on Learning Representations*.

Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. 2024. Agreement-based cascading for efficient inference. *Preprint*, arXiv:2407.02348.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *Preprint*, arXiv:1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Phil Long and Rocco Servedio. 2013. Consistency versus realizable h-consistency for multiclass classification. In *Proceedings of the 30th International Conference on Machine Learning*, number 3 in Proceedings of Machine Learning Research, pages 801–809, Atlanta, Georgia, USA. PMLR.

David Madras, Toniann Pitassi, and Richard Zemel. 2018. Predict responsibly: Improving fairness and accuracy by learning to defer. *Preprint*, arXiv:1711.06664.

Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. 2023a. Two-stage learning to defer with multiple experts. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023b. Cross-entropy loss functions: Theoretical analysis and applications. *Preprint*, arXiv:2304.07288.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2024. Regression with multi-expert deferral. *Preprint*, arXiv:2403.19494.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. The MIT Press.

Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2025. Adversarial robustness in two-stage learning-to-defer: Algorithms and guarantees. *Preprint*, arXiv:2502.01027.

Yannis Montreuil, Shu Heng Yeo, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2024. Two-stage learning-to-defer for multi-task learning. *Preprint*, arXiv:2410.15729.

Hussein Mozannar and David Sontag. 2021. Consistent estimators for learning to defer to an expert. *Preprint*, arXiv:2006.01862.

Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Faster cascades via speculative decoding. *Preprint*, arXiv:2405.19261.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *Preprint*, arXiv:2406.18665.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun

Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh R Menon, Md Rizwan Parvez, and Zhe Feng. 2023. Delucionqa: Detecting hallucinations in domain-specific question answering. *Preprint*, arXiv:2312.05200.

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

Ambuj Tewari and Peter L. Bartlett. 2007. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(36):1007–1025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Fouad Trad and Ali Chehab. 2024. To ensemble or not: Assessing majority voting strategies for phishing detection with large language models. *Preprint*, arXiv:2412.00166.

Neeraj Varshney and Chitta Baral. 2022. Model cascading: Towards jointly improving efficiency and accuracy of nlp systems. *Preprint*, arXiv:2210.05528.

Rajeev Verma, Daniel Barrejon, and Eric Nalisnick. 2022. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics*.

P. Viola and M. Jones. 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.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2023. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*.

Mingyuan Zhang and Shivani Agarwal. 2020. Bayes consistency vs. h-consistency: The interplay between surrogate loss functions and the scoring function class. In *Advances in Neural Information Processing Systems*, volume 33, pages 16927–16936. Curran Associates, Inc.

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

A Current Approaches

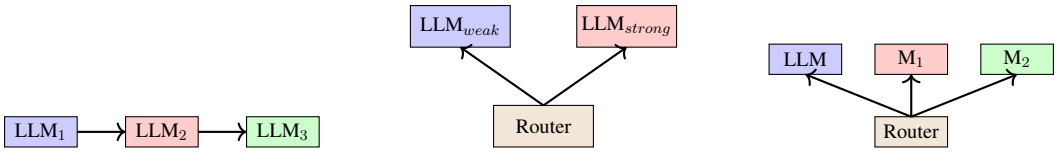

Figure 5: From left to right: Model Cascades, Query Routing, Learning-To-Defer (Ours), we support the multi-model nature of Model Cascades while allowing for direct inferences in Query Routing approaches.

B Approach Details

B.1 Training Algorithm

Algorithm 1 Training

Input: Dataset $\{(x_k, y_k^{\mathrm{s}}, y_k^{\mathrm{e}})\}_{k=1}^K$, multi-task model $g \in \mathcal{G}$, experts $m \in \mathcal{M}$, rejectors $r = (r^{\mathrm{s}}, r^{\mathrm{e}})$, number of epochs EPOCH, batch size BATCH, learning rate λ, surrogate parameter ν.
Initialization: Initialize rejectors parameters $\theta = (\theta^{\mathrm{s}}, \theta^{\mathrm{e}})$.
for $i = 1$ to EPOCH **do**
  Shuffle dataset $\{(x_k, y_k^{\mathrm{s}}, y_k^{\mathrm{e}})\}_{k=1}^K$.
  for each mini-batch $\mathcal{B} \subset \{(x_k, y_k^{\mathrm{s}}, y_k^{\mathrm{e}})\}_{k=1}^K$ of size BATCH **do**
    Extract input-output pairs $z = (x, y^{\mathrm{s}}, y^{\mathrm{e}}) \in \mathcal{B}$.
    Query model $g(x)$ and experts $m(x)$. {Agents have been trained offline and fixed}
    Evaluate costs $c_0(g(x), z)$ and $c_{j>0}(m(x), z)$. {Compute costs}
    Compute the regularized empirical risk minimization:
      $\widehat{\mathcal{E}}_{\Phi_{\mathrm{def}}}(r; \theta) = \frac{1}{\mathrm{BATCH}} \sum_{z \in \mathcal{B}} \left[\Phi_{\mathrm{def}}^{\nu}(r, g, m, z) \right]$.
    Update parameters θ:
      $\theta \leftarrow \theta - \lambda \nabla_\theta \widehat{\mathcal{E}}_{\Phi_{\mathrm{def}}}(r; \theta)$. {Gradient update}
  end for
end for
Return: trained rejector model r^*.

B.2 Inference Algorithm

Algorithm 2 Inference

Query: Input $x \in \mathcal{X}$ for $x = (q, c)$ with a question q and a context c.
Evaluation: Rejectors $r^*(x) = (r^{\mathrm{s},*}(x), r^{\mathrm{e},*}(x))$
Allocation: Allocate the query using the optimal rule $\perp^*(x) = \arg\min_{j \in \mathcal{A}} \sum_{i \in \{\mathrm{s},\mathrm{e}\}} r^{*,i}(x, j)$.
Output: Prediction from the main model $g(x)$ if $(\perp^*(x) = 0)$ or $m_{\perp^*(x)}(x)$ otherwise.

C Proof Lemma 1

Lemma 1 (Bayes-Rejector)**.** *Given an input $x \in \mathcal{X}$ and any distribution $\mathcal{D}$, the optimal rejection rule that minimizes the risk associated with the true deferral loss is given by:*

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

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

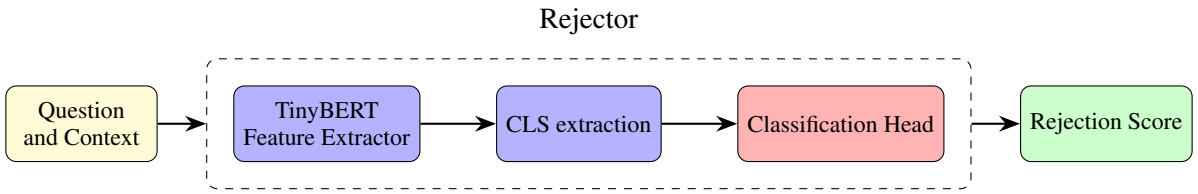

Figure 6: Rejector Architecture: The input data is processed through a TinyBERT embedding (Devlin et al., 2018), which serves as a feature extractor. The extracted CLS token is then used by the classification head to predict the allocation.

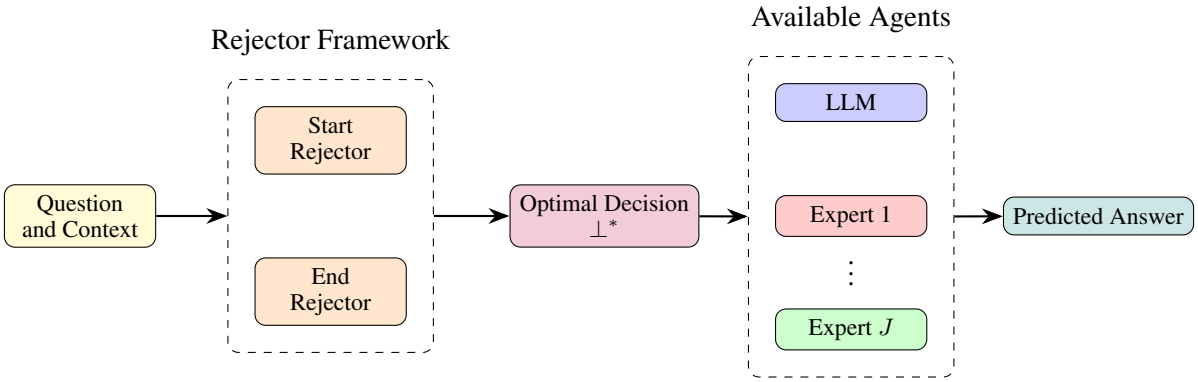

Figure 7: Inference Step of Our Approach: The input data is processed through the rejector framework, which predicts both *start* and *end* spans. Based on the optimal rule defined in Equation 2, the query is assigned to an agent that subsequently predicts the answer.

Proof. Leveraging the *true deferral loss* for a single $i \in \{\mathrm{s}, \mathrm{e}\}$, we can formalize the conditional risk associated with the *true deferral loss*:

$$\begin{aligned}
\mathcal{C}_{\ell_{\text{def}}}(g^i, r^i, m^i, z^i) &= \mathbb{E}_{y^i|x}\Big[c_j(g^i(x), m^i_j(x), z^i)1_{\{r^i(x)=j\}}\Big] \\
&= \mathcal{D}(h^i(x) \neq y^i|X=x)\mathbb{I}_{r^i(x)=0} + \sum_{j=1}^{J} \mathbb{E}_{y^i|x}[c_j(m^i_j(x), z^i)]\mathbb{I}_{r^i(x)=j} \\
&= \mathcal{D}(h^i(x) \neq y_i|X=x)\mathbb{I}_{r^i(x)=0} + \sum_{j=1}^{J} \Big(\alpha_j \mathcal{D}(m^i_j(x) \neq y^i|X=x) + \beta_j\Big)\mathbb{I}_{r^i(x)=j}
\end{aligned}$$

Now, let's study this quantity at its optimum.

$$\begin{aligned}
\inf_{r^i \in \mathcal{R}^i, g^i \in \mathcal{G}^i} \mathcal{C}_{\ell_{\text{def}}}(g^i, r^i, m^i, z^i) &= \inf_{r^i \in \mathcal{R}^i, g^i \in \mathcal{G}^i} \left(\mathbb{E}_{y^i|x}\Big[c_j(r^i(x), m^i_j(x), z^i)1_{\{g^i(x)=j\}}\Big]\right) \\
&= \min\left\{ \inf_{g^i \in \mathcal{G}^i} \mathcal{D}(g^i(x) \neq y^i|X=x), \min_{j\in[J]} \alpha_j \mathcal{D}(m^i_j(x) \neq y^i|X=x) + \beta_j \right\}
\end{aligned}$$

It is then easy to observe, that the Bayes-rejector follows this form:

$$r^{B,i}(x) = \begin{cases} 0, & \text{if } \inf_{g^i \in \mathcal{G}^i} \eta^i_0(x) \leq \min_{j\in[J]} \eta^i_j(x), \\ \arg\min_{j\in[J]} \eta^i_j(x), & \text{otherwise.} \end{cases}$$

with $\eta^i_{j>0}(x) = \mathcal{D}(m^i_j(x) \neq y^i|X=x) + \beta_j$ and $\eta^i_0(x) = \mathcal{D}(g^i(x) \neq y^i|X=x)$. □

D Proof Theorem 1

Theorem 1 ($(\mathcal{R}, \mathcal{G})$–consistency)**.** *Given 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 $\mathcal{R}$-consistency bounds hold for*

any distribution $\mathcal{D}$:

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

then for any $(g, r) \in \mathcal{G} \times \mathcal{R}$, any distribution $\mathcal{D}$ and any $x \in \mathcal{X}$,

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

with the expected cost vector $\overline{\boldsymbol{\tau}}^{\boldsymbol{i}} = \{\mathbb{E}_{y^i|x}[\tau^i_j]\}_{j\in\mathcal{A}}$ and $\overline{\Gamma}^\nu(u) = \left(\sum_{i\in\{s,e\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1\right)\Gamma^\nu\left(\frac{u}{\sum_{i\in\{s,e\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1}\right)$.

Proof. Proving Theorem 1 requires the following lemma 3, introducing the consistency property for a general distribution.

Lemma 3 ($\mathcal{R}^i$-consistency bound)**.** *Given 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 $\mathcal{D}$:*

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

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

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

Let denote a cost for $j \in \mathcal{A} = \{0, \ldots, J\}$:

$$\overline{c}^{i,*}_j = \begin{cases}\inf_{g^i\in\mathcal{G}} \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] & \text{if } j = 0 \\ \mathbb{E}_{y^i|x}[c_j(m^i_j(x), z^i)] & \text{otherwise}\end{cases}$$

Let's recall a previous established results proven in C.

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

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

$$\begin{aligned}\Delta\mathcal{C}^i_{\ell_{\text{def}}} &= \mathcal{C}^i_{\ell_{\text{def}}} - \min_{j\in\mathcal{A}}\overline{c}^{i,*}_j \\ &= \mathcal{C}^i_{\ell_{\text{def}}} - \min_{j\in\mathcal{A}}\overline{c}^i_j + \Big(\min_{j\in\mathcal{A}}\overline{c}^i_j - \min_{j\in\mathcal{A}}\overline{c}^{i,*}_j\Big)\end{aligned} \tag{6}$$

We now define the first term $A = \mathcal{C}^i_{\ell_{\text{def}}} - \min_{j\in\mathcal{A}}\overline{c}^i_j$ and the second term $B = \min_{j\in\mathcal{A}}\overline{c}^i_j - \min_{j\in\mathcal{A}}\overline{c}^{i,*}_j$, such that $\Delta\mathcal{C}^i_{\ell_{\text{def}}} = A + B$. It is important to notice that:

$$\min_{j\in\mathcal{A}}\overline{c}^i_j = \inf_{r^i\in\mathcal{R}}\sum_{j\in\mathcal{A}}\overline{c}^i_j 1_{\{r^i(x)=j\}} = \inf_{r^i\in\mathcal{R}}\sum_{j\in\mathcal{A}}\overline{\tau}^i_j 1_{\{\overline{r}^i(x)\neq j\}} \tag{7}$$

It follows by definition of the conditional risk:

$$A = \sum_{j\in\mathcal{A}}\overline{\tau}^i_j 1_{\{\overline{r}^i(x)\neq j\}} - \inf_{r^i\in\mathcal{R}}\sum_{j\in\mathcal{A}}\overline{\tau}^i_j 1_{\{\overline{r}^i(x)\neq j\}} \tag{8}$$

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

$$\boldsymbol{p}^i = \frac{\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}}{\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \in \Delta^{|\mathcal{A}|}, \tag{9}$$

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

$$\begin{aligned}
A &= \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Big(\sum_{j\in\mathcal{A}} p^i_j 1_{\{\overline{r}^i(x)\neq j\}} - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} p^i_j 1_{\{\overline{r}^i(x)\neq j\}} \Big) \\
&\leq \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Gamma^\nu \Big(\sum_{j\in\mathcal{A}} p^i_j \Phi^\nu_{01}(\overline{r}^i, x, j) - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} p^i_j \Phi^\nu_{01}(\overline{r}^i, x, j) \Big) \quad \text{(using Lemma 3)} \\
&= \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Gamma^\nu \Big(\frac{1}{\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \Big[\sum_{j\in\mathcal{A}} \overline{\tau}^i_j \Phi^\nu_{01}(\overline{r}^i, x, j) - \inf_{r^i\in\mathcal{R}} \sum_{j\in\mathcal{A}} \overline{\tau}^i_j \Phi^\nu_{01}(\overline{r}^i, x, j) \Big] \Big) \\
&= \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(\overline{r}^i)}{\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \Big)
\end{aligned} \tag{10}$$

Now, we have the following relationship:

$$B = \min_{j\in\mathcal{A}} \overline{c}^i_j - \min_{j\in\mathcal{A}} \overline{c}^{i,*}_j \leq \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] \tag{11}$$

Injecting B, it follows:

$$\Delta\mathcal{C}^i_{\ell_{\text{def}}}(r^i, g^i) \leq \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(\overline{r}^i)}{\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \Big) + \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] \tag{12}$$

Applying the summation:

$$\Delta\mathcal{C}_{\ell_{\text{def}}}(r, g) \leq \sum_{i\in\{\text{s,e}\}} \Big[\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Gamma^\nu \Big(\frac{\Delta\mathcal{C}^i_{\text{def}}(\overline{r}^i)}{\|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \Big) + \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{y^i|x}[c_0(g^i(x), z^i)] \Big] \tag{13}$$

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

$$\Delta\mathcal{C}_{\ell_{\text{def}}}(r, g) \leq \Big(\sum_{i\in\{\text{s,e}\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1 \Big) \Gamma^\nu \Bigg(\frac{\Delta\mathcal{C}_{\text{def}}(\overline{r})}{\sum_{i\in\{\text{s,e}\}} \|\overline{\boldsymbol{\tau}}^{\boldsymbol{i}}\|_1} \Bigg) + \sum_{i\in\{\text{s,e}\}} \Big[\mathbb{E}_{y^i|x}\Big[c_0\big(g^i(x), z^i\big)\Big] - \inf_{g^i\in\mathcal{G}^i} \mathbb{E}_{y^i|x}\Big[c_0\big(g^i(x), z^i\big)\Big] \Big] \tag{14}$$

Then, applying the expectation $\mathbb{E}_x[\cdot]$ to recover the excess risk $\mathbb{E}_x[\Delta\mathcal{C}_\ell] := \mathcal{E}_\ell - \mathcal{E}^B_\ell + \mathcal{U}_\ell$, we show the desired results:

$$\begin{aligned}
\mathcal{E}_{\ell_{\text{def}}}(g, r) - \mathcal{E}^B_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) + \mathcal{U}_{\ell_{\text{def}}}(\mathcal{G}, \mathcal{R}) \leq \overline{\Gamma}^\nu \Big(\mathcal{E}_{\Phi^\nu_{\text{def}}}(r) - \mathcal{E}^*_{\Phi^\nu_{\text{def}}}(\mathcal{R}) + \mathcal{U}_{\Phi^\nu_{\text{def}}}(\mathcal{R}) \Big) \\
+ \sum_{i\in\{\text{s,e}\}} \Big(\mathcal{E}_{c_0}(g^i) - \mathcal{E}^B_{c_0}(\mathcal{G}^i) + \mathcal{U}_{c_0}(\mathcal{G}^i) \Big),
\end{aligned} \tag{15}$$

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

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

□

E Experiments

E.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 where no answer is found within the provided context. In these cases, we aim for the model to return no output, which we represent using 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: "?"

E.2 Agent Training and Performance Details

We train our models using a single NVIDIA H-100 GPU. Additionally, we take the results across 4 independent experimental runs. We train both ALBERT-Base and ALBERT-XXL offline, we will publicly release the weights for these agents. We do not train Llama-3.2-1B or Llama-3-8B from scratch. Instead, we utilize the publicly available weights from *meta-llama* on HuggingFace out of the box. For each dataset, we use the following hyperparameters on an NVIDIA H100 GPU:

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

Table 1: Hyperparameters for SQuADv1, SQuADv2, and TriviaQA.

We report the following performance metrics for our agents on the test set being a subsample of the validation set (3000 inputs):

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

	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

Table 3: 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 model is significantly more lightweight, with only 4.39M parameters and 0.15 GFLOPs, making it well-suited for deployment in resource-constrained environments.