I Know What I Don't Know: Improving Model Cascades Through Confidence Tuning

Stephan Rabanser[1,2*], Nathalie Rauschmayr[3],
Achin Kulshrestha[3], Petra Poklukar[3], Wittawat Jitkrittum[3],
Sean Augenstein[3], Congchao Wang[3], Federico Tombari[3]

[1]University of Toronto [2]Vector Institute [3]Google

February 3, 2025

Abstract

Large-scale machine learning models deliver strong performance across a wide range of tasks but come with significant computational and resource constraints. To mitigate these challenges, local smaller models are often deployed alongside larger models, relying on routing and deferral mechanisms to offload complex tasks. However, existing approaches inadequately balance the capabilities of these models, often resulting in unnecessary deferrals or sub-optimal resource usage. In this work we introduce a novel loss function called *Gatekeeper* for calibrating smaller models in cascade setups. Our approach fine-tunes the smaller model to confidently handle tasks it can perform correctly while deferring complex tasks to the larger model. Moreover, it incorporates a mechanism for managing the trade-off between model performance and deferral accuracy, and is broadly applicable across various tasks and domains without any architectural changes. We evaluated our method on encoder-only, decoder-only, and encoder-decoder architectures. Experiments across image classification, language modeling, and vision-language tasks show that our approach substantially improves deferral performance.

1 Introduction

In recent years, large-scale machine learning models such as Gemini (GeminiTeam et al., 2023), GPT-4 (Achiam et al., 2023) or Claude (Anthropic, 2024) have gained significant traction due to their remarkable ability to address a wide array of tasks. These tasks range from natural language understanding and generation, including machine translation, summarization, and conversational agents, to computer vision applications like image recognition, object detection, and image captioning. The versatility and high performance of these expansive models make them invaluable tools across diverse domains, including healthcare (Nazi & Peng, 2024), finance (Li et al., 2023), education (Wang et al., 2024b), and entertainment (Gallotta et al., 2024).

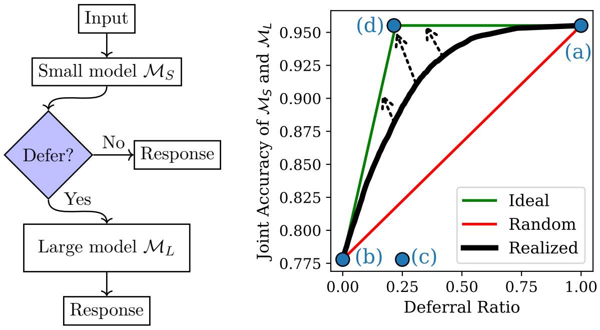

Figure 1: **Overview of the cascading setup (left) and performance trade-off (right)**. *Left*: Cascading determines which inputs should be predicted by a small model $\mathcal{M}_S$ or routed to a large model $\mathcal{M}_L$. *Right*: Performance is measured as a trade-off between joint accuracy across $\mathcal{M}_S$ and $\mathcal{M}_L$ and deferral ratio. Ideal deferral strategies optimize this trade-off and push the realized deferral curve closer to the ideal deferral depicted in (d). (a) depicts full deferral; (b) depicts no deferral; and (c) depicts excessive deferral of requests that could have been correctly handled by $\mathcal{M}_S$.

Deploying and operating such large models presents significant challenges in terms of latency, memory, compute and storage (Pope et al., 2023). Optimizing inference costs is an active research area

[*] Work done while a Student Researcher at Google. Correspondence to stephan@cs.toronto.edu.

which includes both techniques for reducing the size of the existing large model such as model compression (Hoefler et al., 2021), model pruning (Ma et al., 2023; Cheng et al., 2024) and distillation (Yang et al., 2024), and those aiming to leverage a sequence of models such as speculative decoding (Leviathan et al., 2023) and model cascades (Dohan et al., 2022). However, due to scaling laws showing that the performance of a Large Language Model (LLM) increases with its size (Kaplan et al., 2020), the latter category of methods leveraging a sequence of models is currently a more promising direction to lower inference costs without sacrificing the capabilities of large models.

Both speculative decoding and model cascading rely on the existence of a large performant model $\mathcal{M}_L$ and a small model $\mathcal{M}_S$ that is cheap, fast, and less accurate. Speculative decoding leverages $\mathcal{M}_S$ for generating a set of draft tokens that are then validated by $\mathcal{M}_L$ in parallel, a technique successfully deployed in industry applications (Leviathan, 2024). In contrast, model cascades leverage a deferral rule for selecting the most suitable model to process a given request (see Figure 1 left). While the success of the speculative decoding necessitates a highly performant $\mathcal{M}_S$ to generate quality draft tokens, model cascades allow the deployment of a less capable $\mathcal{M}_S$ by invoking $\mathcal{M}_L$ only for inference requests outside the small model's scope. In this work, we contribute to the advancement of the model cascades.

Model cascades achieve efficient deferral by optimizing for two objectives: compute budget and joint accuracy. We illustrate their trade-off on the example shown in Figure 1 (right). Assume we have x inference requests and a small model $\mathcal{M}_S$ that only requires *20%* of the compute budget of the large model $\mathcal{M}_L$. There are three worst case scenarios: (a) the small model $\mathcal{M}_S$ defers all requests to $\mathcal{M}_L$ and the system achieves the best joint accuracy (equal to the accuracy of $\mathcal{M}_L$) but the worst compute budget (*1.2x*) since all requests are run on both models; (b) $\mathcal{M}_S$ never sends a request to $\mathcal{M}_L$, resulting in the smallest compute budget (*0.2x*) but also the lowest joint accuracy (equal to the accuracy of $\mathcal{M}_S$); (c) $\mathcal{M}_S$ only sends requests that it could have answered correctly to $\mathcal{M}_L$, requiring an increased compute budget compared to (b) but still resulting in the lowest joint accuracy (equal to the accuracy of $\mathcal{M}_S$). On the other hand, an ideal case is the scenario (d) where the small model $\mathcal{M}_S$ only sends requests for which it would be incorrect, requiring a compute budget between *0.2-1x* but resulting in the optimal joint accuracy given that budget. We call the approximation of the ideal case the *deferral performance*.

In this paper we address the following research question:

How can we optimize model cascades to maximize deferral performance?

In other words, we focus on designing effective model cascades by making the small model more aware of what it does not know. We achieve this by introducing a *general-purpose* loss function, called *Gatekeeper*, that calibrates the small model's confidence in its predictions. By fine-tuning $\mathcal{M}_S$ to output high confidence for correct predictions and low confidence for incorrect ones, we enhance the reliability of its uncertainty estimates and facilitate learning of common tasks, thereby directly improving the deferral performance. Crucially, *Gatekeeper* includes an inherent mechanism for managing the trade-off between model performance and deferral accuracy that can be applied to an arbitrary architecture, making our work directly applicable to Vision-Language Models (VLMs).

We demonstrate the efficacy of the *Gatekeeper* loss across various model architectures, including encoder-only vision models for image classification, decoder-only LMs for closed- and open-form text generation, and encoder-decoder setups for VL tasks for open set classification and captioning. Our main results show that models trained with *Gatekeeper* outperform an untuned baseline by a factor of 0.72x/2x on CIFAR-100/TinyImagenet and 7x/10x on ARC-e/c, respectively, in terms of deferral performance. This advancement paves the way for more scalable and efficient deployment strategies, leveraging the strengths of both local and large-scale models to deliver high-quality results in real-time applications.

2 Related Work

Our proposed method improves model cascades through uncertainty-aware finetuning. Next, we describe related work for both research areas.

Model Cascades: A cascade consists of a series of models and a deferral rule which determines the appropriate model given an input request. The concept of model cascades has first been proposed by Viola & Jones (2001), where it is used to accelerate object detection models. Cascades have been extensively studied for classification-based computer vision (Wang et al., 2017; Trapeznikov & Saligrama, 2013; Bolukbasi et al., 2017; Jitkrittum et al., 2023) and in models for natural language processing (Dohan et al., 2022; Mamou et al., 2022; Varshney & Baral, 2022).

Cascades are particularly promising in the context of generative models such as LLMs and VLMs since they can significantly reduce inference costs. In contrast to speculative decoding (Leviathan et al., 2023), they aim to invoke the large model only for difficult examples. However, the two approached can also be combined. While Chen et al. (2024) combine the deferral logic with speculative decoding to generate initial tokens using larger models and later tokens using a smaller model, the majority of research on model cascades has focused on using pre-trained LLMs with a post-hoc deferral logic (Narasimhan et al., 2022; Jitkrittum et al., 2023; Yue et al., 2024). Kolawole et al. (2024) use agreement across multiple models to make deferral decisions, while Gupta et al. (2024) present a method to learn a deferral rule based on quantiles of per-token log probabilities.

Model cascades can be further improved through training and fine-tuning. Wang et al. (2024a) train the small model only on easier examples by masking tokens for which large and small model are incorrect. Enomoro & Eda (2021) extend the training objective of image classification models with confidence calibration. In contrast to previous research, our work extends cascades to VLMs and improves overall inference performance by making smaller models less confident when they are incorrect.

Uncertainty-Aware Models: Extensive research has been conducted in the field of uncertainty quantification in deep learning and we refer to Abdar et al. (2021) for a detailed survey. While many methods have been proposed for classification-based models, measuring uncertainty for generative models is still an active area of research. Based on the assumed level of access to model internals, existing methods can be summarized into three main categories:

Black box methods operate solely via the model's query interface by injecting tailored instructions into prompts. These modify the prompt $\mathbf{x}$ by appending instructions $\mathbf{x}'$ for the model to respond less confidently: $\mathbf{x} \leftarrow \mathbf{x}|\mathbf{x}'$. Related methods are confidence quantification (Shrivastava et al., 2023), rejection and remote model awareness (Kadavath et al., 2022), and self-critiquing (Gou et al., 2023). Xiong et al. (2024) show that LLMs can express their confidence through prompting and sampling strategies and their experiments indicate that these models tend to be overconfident.

Gray box approaches employ confidence-based strategies centered on post-processing the model's logits. Many uncertainty techniques such as ensembling (Lakshminarayanan et al., 2017) and Bayesian methods (Blundell et al., 2015)) are not scalable. Related techniques are max confidence (Hendrycks & Gimpel, 2016), predictive entropy, and confidence reduction prompting. Malinin & Gales (2021) uses token-entropy as a measure of uncertainty in autoregressive models and Kuhn et al. (2023) leverages linguistic invariances via semantic entropy.

White box methods utilize uncertainty-aware fine-tuning in order to produce more accurately calibrated models. Chuang et al. (2024) introduces Self-REF, a framework which leverages confidence tokens during fine-tuning to improve performance in downstream routing. Krishnan et al. (2024) proposes an uncertainty-aware causal language modeling loss function, which captures the trade-off between predictive accuracy and uncertainty calibration. In contrast to previous work, our method aims to calibrate the model in a way such that correctly generated tokens are assigned low predictive uncertainty and incorrectly generated tokens are assigned high predictive uncertainty. We consider the uncertainty-aware model in the context of cascade inference system, where it helps to improve overall performance. Furthermore, we present in-depth ablation studies to quantify the trade-offs of the proposed loss function.

3 Method

3.1 Overview & Setup

Our framework consists of a large, highly capable model $\mathcal{M}_L$ and a smaller, resource-efficient model $\mathcal{M}_S$. We assume that $S \in \mathbb{N}$ and $L \in \mathbb{N}$ represent the parameter count of each model with $S \ll L$. Both models can either function as classifiers (i.e., $\mathcal{M} : \mathbb{R}^D \to [C]$ with C denoting the classes), or (multi-modal) sequence models (i.e., $\mathcal{M} : \mathbb{R}^D \to [V]^T$ where V is the vocabulary and T is the sequence length). We include experiments on all of these model classes in Section 4. Furthermore, we do not require a shared model family to be deployed on both $\mathcal{M}_S$ and $\mathcal{M}_L$; for example, $\mathcal{M}_S$ could be a custom convolutional neural network optimized for efficient inference and $\mathcal{M}_L$ a vision transformer (Dosovitskiy, 2020). The primary objective is to design a deferral mechanism that enables $\mathcal{M}_S$ to decide when to return its predictions without the assistance of $\mathcal{M}_L$ and when to instead defer to it.

Deferral decisions are made using signals derived from the small model $\mathcal{M}_S$ as this approach is typically more cost-effective than employing a separate routing mechanism (Teerapittayanon et al., 2016). Approaches that involve querying the large model $\mathcal{M}_L$ to assist in making deferral decisions at test time are excluded from our setup. Such methods — common in domains like LLMs — are counterproductive to our goal since querying $\mathcal{M}_L$ inherently constitutes a deferral. Examples of these inapplicable methods include collaborative LLM frameworks (Mielke et al., 2022) and techniques that rely on semantic entropy for uncertainty estimation (Kuhn et al., 2023). As part of our setup, we assume that $\mathcal{M}_S$ is strictly less capable than $\mathcal{M}_L$— a realistic scenario in practice supported by scaling laws (Kaplan et al., 2020). Under this assumption, mistakes made by $\mathcal{M}_L$ are also made by $\mathcal{M}_S$; however, $\mathcal{M}_S$ may make additional errors that $\mathcal{M}_L$ would avoid. This reflects the general observation that larger models tend to outperform smaller models across a wide range of tasks.

As discussed in Section 2, the choice of deferral strategy often depends on the level of access available to $\mathcal{M}_S$. We assume white box access with full access to $\mathcal{M}_S$'s internals. As such, deferral mechanisms can be directly integrated into the model's architecture and parameters. This involves fine-tuning $\mathcal{M}_S$ to predict deferral decisions or to incorporate rejection mechanisms within its predictive process. Our work falls into this category as it proposes a new loss function to fine-tune $\mathcal{M}_S$.

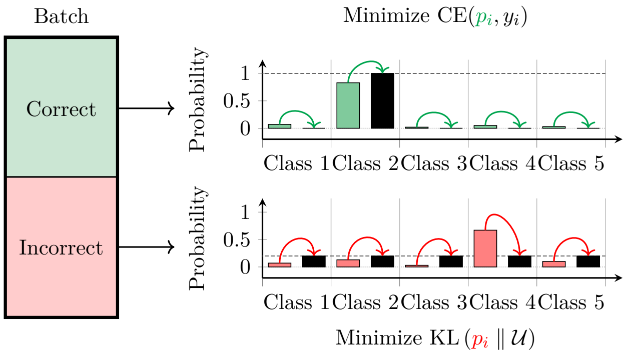

Figure 2: **Overview of *Gatekeeper***: We want correctly predicted samples maintain their current prediction by ensuring that cross entropy is decreased (top, green). At the same time, we want incorrectly predicted samples to yield a uniform confidence across all classes, leading to a low overall confidence score (bottom, red).

Our goal is to train a small model that can effectively distinguish between correct and incorrect predictions. While many past works have considered the question of whether it is possible to find proxy measures for correctness, the central question we ask is:

Can we *optimize* the small model $\mathcal{M}_S$ to separate correct from incorrect predictions?

We show that this is indeed achievable through a carefully designed fine-tuning stage that does not require any architectural modifications. This ensures that the ability to separate correct from incorrect decisions is integrated seamlessly into $\mathcal{M}_S$'s existing structure.

3.2 Confidence-Tuning for Deferral

Stage 1: Standard Training. We begin with a $\mathcal{M}_S$ that has already been trained on the tasks it is intended to perform upon deployment. However, due to its limited capacity, $\mathcal{M}_S$ cannot achieve the performance levels of $\mathcal{M}_L$. Importantly, we make no assumptions about the training process of $\mathcal{M}_S$ —whether it was trained from scratch without supervision from an external model or through a distillation approach.

Stage 2: Correctness-Aware Finetuning with *Gatekeeper*. Next, we introduce a correctness-aware loss, dubbed *Gatekeeper*, to finetune $\mathcal{M}_S$ for improved confidence calibration. Specifically, the model is trained to make correct predictions with high confidence while reducing the confidence of incorrect predictions (see Figure 2). This loss can either rely on true labels or utilize the outputs of $\mathcal{M}_L$ with soft probabilities as targets.

For a standard classification model, the calibration loss is defined as the following hybrid loss

$$\mathcal{L} = \alpha\mathcal{L}_{\text{corr}} + (1-\alpha)\mathcal{L}_{\text{incorr}} \tag{1}$$

$$\mathcal{L}_{\text{corr}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{y_i = \hat{y}_i\}\text{CE}(p_i(\mathbf{x}_i), y_i) \tag{2}$$

$$\mathcal{L}_{\text{incorr}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{y_i \neq \hat{y}_i\}\text{KL}\left(p_i(\mathbf{x}_i) \parallel \mathcal{U}\right) \tag{3}$$

where y_i and $\hat{y}_i$ are the true and predicted label, respectively, p_i is the predicted probability distribution over classes, $\mathcal{U}$ represents the uniform distribution over all classes, N denotes the number samples in the current batch, $\alpha \in (0,1)$ is a tunable hyperparameter controlling the emphasis between correct and incorrect predictions, and the cross-entropy function and KL divergence are defined as $\text{CE}(p,y) = -\sum_c y_c \log p_c$ and $\text{KL}(p \parallel q) = \sum_c p_c \log(\frac{p_c}{q_c})$, respectively. We note that a similar loss has previously been proposed in Outlier Exposure (OE) (Hendrycks et al., 2018) for out-of-distribution (OOD) sample detection. Here, the goal is to make sure that OOD examples are assigned low confidence scores by tuning the confidence on a auxiliary outlier dataset. However, to the best of our knowledge, this idea has not previously been used to improve deferral performance of a smaller model in a cascading chain.

We emphasize that the trade-off parameter α plays a critical role as part of this optimization setup as it directly influences model utility and deferral performance. A lower value of α emphasizes reducing confidence in incorrect predictions by pushing them closer to the uniform distribution, making the model more cautious in regions where it may make mistakes. Conversely, a higher value of α encourages the model to increase its confidence on correct predictions, sharpening its decision boundaries and enhancing accuracy where it is already performing well. Thus, α serves as a crucial hyperparameter that balances the trade-off between improving calibration by mitigating overconfidence in errors and reinforcing confidence in accurate classifications. By appropriately tuning α, practitioners can control the model's behavior to achieve a desired balance between reliability in uncertain regions and decisiveness in confident predictions, tailored to the specific requirements of their application.

We further generalize this loss to token-based models (e.g., LMs and VLMs), formulated as

$$\mathcal{L}_{\text{corr}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \mathbb{1}\{y_{i,t} = \hat{y}_{i,t}\}\text{CE}(p_{i,t}(\mathbf{x}_i), y_{i,t}) \tag{4}$$

$$\mathcal{L}_{\text{incorr}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \mathbb{1}\{y_{i,t} \neq \hat{y}_{i,t}\}\text{KL}\left(p_{i,t}(\mathbf{x}_i) \parallel \mathcal{U}\right) \tag{5}$$

where $y_{i,t}$ and $\hat{y}_{i,t}$ denote the true and predicted tokens at position t for sample i, $p_{i,t}$ is the predicted token distribution at position t for sample i, and T is the sequence length for the token-based model. The token-level loss ensures that correct token predictions are made confidently while incorrect tokens are assigned smaller confidences.

Stage 3: Confidence Computation & Thresholding. After fine-tuning $\mathcal{M}_S$ with *Gatekeeper*, we apply standard confidence- and entropy-based techniques for model uncertainty to obtain a deferral signal. We use the selective prediction framework to determine whether a query point $\mathbf{x} \in \mathbb{R}^D$ should be accepted by $\mathcal{M}_S$ or routed to $\mathcal{M}_L$. Selective prediction alters the model inference stage by introducing a deferral state through a *gating mechanism* (El-Yaniv & Wiener, 2010). At its core, this mechanism relies on a deferral function $g : \mathbb{R}^D \to \mathbb{R}$ which determines if $\mathcal{M}_S$ should output a prediction for a sample $\mathbf{x}$ or defer to $\mathcal{M}_L$. Given a targeted acceptance threshold τ, the resulting predictive model can be summarized as:

$$(\mathcal{M}_S, \mathcal{M}_L, g)(\mathbf{x}) = \begin{cases} \mathcal{M}_S(\mathbf{x}) & g(\mathbf{x}) \geq \tau \\ \mathcal{M}_L(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{6}$$

Classification Models (Max Softmax). Let $\mathcal{M}_S$ produce a categorical distribution $\{p(y = c \mid \mathbf{x})\}_{c=1}^{C}$ over C classes. Then we define the gating function as

$$g_{\text{CL}}(\mathbf{x}) \;=\; \max_{1 \leq c \leq C} \; p(y = c \,|\, \mathbf{x}). \tag{7}$$

Token-based Models (Negative Predictive Entropy). Let $\mathcal{M}_S$ produce a sequence of categorical distributions $\{p(y_t = c \mid \mathbf{x})\}_{c=1}^{C}$ for each token index $t \in T$. Then we define the gating function as

$$g_{\text{NENT}}(\mathbf{x}) = \; \frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C} p\big(y_t = c \,\big|\, \mathbf{x}\big) \, \log p\big(y_t = c \,\big|\, \mathbf{x}\big), \tag{8}$$

where $y_t \in [C]$ is the predicted token at time step t, $p(y_t = c \mid \mathbf{x})$ is the (conditional) probability of token k at step t, and T is the total number of token positions for the sequence. Across both model classes, higher values of either g_{CL} or g_{NENT} indicate higher confidence in the predicted class or sequence generation, respectively.

4 Experiments

In this section we detail the experiments used to evaluate our deferral strategies across three distinct model architectures: encoder-only classification models, decoder-only LMs, and encoder-decoder VLMs. Each setup leverages a deferral setup from smaller to larger models, assessing the efficacy of deferring hard queries to more capable models.

4.1 Encoder-only Setup (Classification Models)

We comprehensively assess the performance of our deferral strategies across different model architectures and task types, starting with image classification. We train both a large model and a small model on the following datasets: CIFAR-10/100 (Krizhevsky et al., 2009), Food-101 (Bossard et al., 2014), and TinyImageNet200 (Le & Yang, 2015). For both CIFAR datasets we use a ResNet-18 (He et al., 2016) as $\mathcal{M}_L$ and a custom CNN as $\mathcal{M}_S$. For Food-101 and TinyImageNet200 we instead use a ResNet-50 (He et al., 2016) as $\mathcal{M}_L$ and a Mobilenet V3 Small (Howard et al., 2019) as $\mathcal{M}_S$, where the latter is trained using knowledge distillation from the big model.

Evaluation Metrics. We measure the performance of our confidence tuning method and the resulting deferral function $g(\cdot)$ using the following performance metrics (see example in Figure 3 for a visual overview):

1. The **Distributional Overlap of Confidences of Correct and Incorrect Predictions** s_o is defined as the integral of the minimum of the probability density functions (PDFs) of confidence scores for correctly classified samples, $\hat{p}_{\text{corr}}(c)$, and incorrectly classified samples, $\hat{p}_{\text{incorr}}(c)$ (see Figure 3a). Formally, given the confidence sets $\mathcal{C}_{\text{corr}}$ and $\mathcal{C}_{\text{incorr}}$, the overlap s_o is computed as

$$s_o = \int_0^1 \min\left\{\hat{p}_{\text{corr}}(c),\ \hat{p}_{\text{incorr}}(c)\right\} \, \mathrm{d}c, \qquad (9)$$

   where the PDFs are estimated using Kernel Density Estimation (KDE). If $s_o = 1$, then $\mathcal{M}_S$ cannot distinguish the confidence distribution of correct and incorrect predictions; if $s_o = 0$, then $\mathcal{M}_S$ can perfectly separate correct and incorrect predictions. Note that a related way of capturing the distributional separability is given by the Area Under the Receiver Operating Characteristic Curve (AUROC) which we discuss in Appendix B.3.

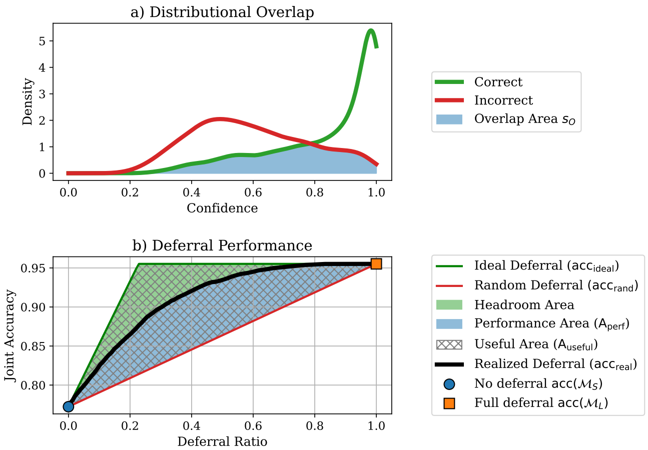

Figure 3: **Performance metrics overview**: **(a)** Distributional Overlap s_o: the densities of confidence scores for correctly (green) and incorrectly classified (red) samples, with the overlap area shaded in blue. Smaller values are better (↓). **(b)** Deferral Performance s_d: how joint accuracy between $\mathcal{M}_S$ and $\mathcal{M}_L$ varies with deferral ratio, showing random (red), ideal (green), and realized (black) deferral strategies. The blue region shows the realized performance gain, the hatched portion represents the range of useful deferral functions, and the green region indicates the potential headroom over the realized deferral. Larger values are better (↑).

2. **Deferral Performance** s_d: To formally quantify how well $\mathcal{M}_S$ defers difficult inputs to $\mathcal{M}_L$, we examine the joint performance across all possible deferral ratios $r \in [0, 1]$, where r denotes the fraction of inputs sent to $\mathcal{M}_L$ based on a particular threshold τ (recall Equation (6)). Figure 3 b) illustrates how, as r increases from 0 to 1, the overall

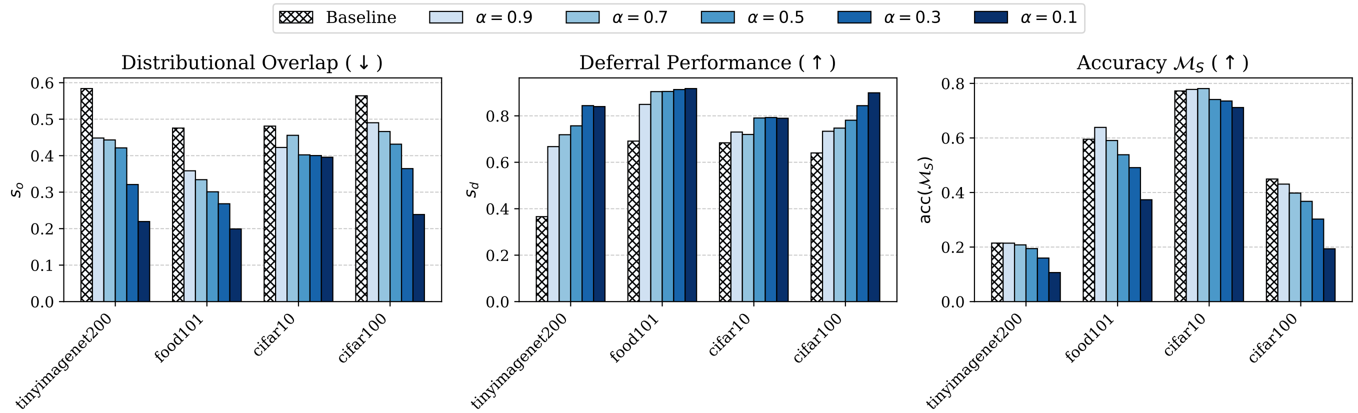

Figure 4: **Performance on image classification tasks**. We observe that lower levels of α lead to decreased distributional overlap between correct/incorrect predictions (left), increased deferral performance (center) and generally decreased performance over the full data distribution (right). These results support our conclusion that the small model $\mathcal{M}_S$ learns to refocus on easier subsets of the distribution while understanding more reliably when it should abstain and defer to the large model $\mathcal{M}_L$.

(joint) accuracy acc(r) increases from the accuracy of $\mathcal{M}_S$ (blue circle, no deferral) to the accuracy of $\mathcal{M}_L$ (orange square, full deferral). Useful deferral models are constrained to operate between random deferral (acc$_{\text{rand}}$, red line) and ideal deferral (acc$_{\text{ideal}}$, green line). The ideal deferral acc$_{\text{ideal}}$ corresponds to the oracle solution that perfectly defers examples misclassified by $\mathcal{M}_S$ and we discuss its exact functional form in Appendix A.2. We also define the realized deferral curve, acc$_{\text{real}}$, as the joint accuracy obtained under the learned deferral strategy $g(\cdot)$ employed by $\mathcal{M}_S$ and $\mathcal{M}_L$. The deferral performance metric s_d is then given as:

$$s_d = \frac{A_{\text{perf}}}{A_{\text{useful}}} = \frac{\int_0^1 (\text{acc}_{\text{real}}(r) - \text{acc}_{\text{rand}}(r))\ \text{d}r}{\int_0^1 (\text{acc}_{\text{ideal}}(r) - \text{acc}_{\text{rand}}(r))\ \text{d}r}. \quad (10)$$

This ratio quantifies the fraction of the potential improvement over random deferral that has been realized by the achieved deferral strategy. Note that $s_d = 1$ indicates perfect deferral, matching the ideal strategy, while an $s_d = 0$ implies no improvement over random deferral.

3. **Accuracy of the small model** acc($\mathcal{M}_S$): Finally, since *Gatekeeper* emphasizes patterns for distinguishing correct/incorrect examples, the model is no longer encouraged to minimize the classification loss over the full population. As a result, improving on the correct/incorrect separation task can lead to drops in utility. Hence, practically useful deferral methods need to balance both deferral performance and the accuracy of $\mathcal{M}_S$.

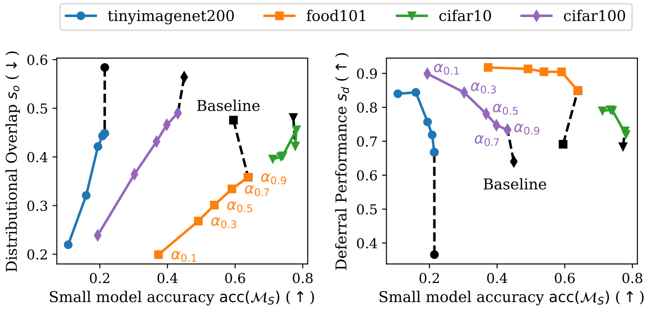

Figure 5: **Performance trade-off between small model accuracy acc($\mathcal{M}_S$) and deferral performance** s_d. The baseline model obtained without fine-tuning using *Gatekeeper* is often the most accurate model over the full data distribution. With the introduction of *Gatekeeper* we can improve distinguishability of correct/incorrect predictions (left) as well as deferral (right) at the expense of model utility. Successful cascading solutions in practice need to balance both model accuracy and deferral performance.

Results. We document our main results in Figure 4. We report performance results on both a baseline model, i.e., an instance of $\mathcal{M}_S$ that was not

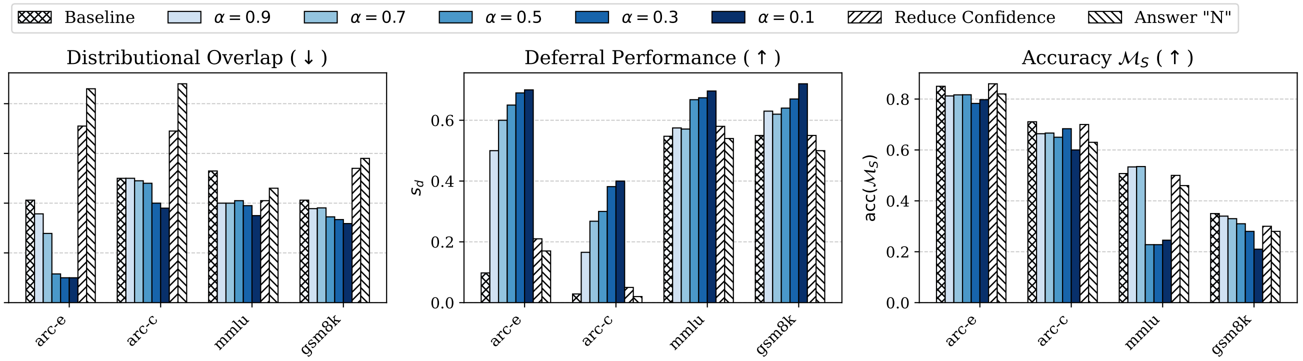

Figure 6: **Performance on language modeling tasks**. Similar as Figure 4. In addition to a non-tuned baseline, we also add an uncertainty prompting baseline as well as an Answer "N" option. Both these approaches fail to meaningfully improve deferral.

trained with *Gatekeeper*, and a set of small models trained with *Gatekeeper* at various αs. Across all models we compute the deferral performance as well as the model's correct/incorrect separation ability (center, left). We see that the strongest performance is achieved at low αs as the model strongly emphasizes pushing the outputs of incorrect examples closer to a uniform distribution. However, this strong performance comes at a cost: the accuracy of the small model consistently degrades for small αs (right). This highlights the fact that the model effectively "unlearns" to perform well on some part of the data distribution and re-focuses its classification ability of easier data points. On the other hand, for large values of α we observe consistency in model accuracy or even slight improvements as the model now emphasizes most of its training on maintaining good performance on already well-predicted points.

This result highlights a critical trade-off which is directly controlled by α: *how strongly do we want to degrade model performance over the full data distribution in order to obtain a better deferral model?* We note that this compromise between raw model utility and deferral performance is not surprising and similar trade-offs exist in fairness (Dutta et al., 2020; Yaghini et al., 2023) and privacy (Abadi et al., 2016; Rabanser et al., 2023). We study this trade-off explicitly in Figure 5 showing (i) a clear negative correlation between deferral performance and the small model's accuracy; and (ii) a clear positive correlation between the overlap of correct/incorrect confidences and the accuracy of $\mathcal{M}_S$.

4.2 Decoder-only Setup (Language Models)

In the decoder-only setup, we explore the application of LLMs. Our primary models of interest are the scalable LMs from the Gemma model class (GemmaTeam et al., 2024). We choose Gemma2B as $\mathcal{M}_S$ and Gemma7B as $\mathcal{M}_L$ with 2 billion and 7 billion parameters, respectively. Similar to the encoder-only setup, we employ smaller LMs as the initial classifiers to manage simpler next-token prediction tasks. The deferral strategy involves routing only those token sequences that exhibit high uncertainty — as determined by high predictive entropy — to the more powerful model $\mathcal{M}_L$.

Our experiments start by taking the instruction-tuned checkpoints of Gemma2B and Gemma7b and fine-tuning both models on the training split of a respective dataset to ensure that the model (i) performs well on the task; and (ii) is familiar with the desired response format. Note that this fine-tuning stage is performed using standard perplexity minimization. Then, we finetune $\mathcal{M}_S$ with *Gatekeeper* using the same training split to decrease confidence on incorrect next-token predictions. Finally, we evaluate the model yielded by *Gatekeeper* on a validation/test split. The datasets we consider are ARC-e/c (Clark et al., 2018), MMLU (Hendrycks et al., 2020), and GSM8K (Cobbe et al., 2021). We use the same evaluation metrics as previously used in Section 4.1.

Results. We document our main results in Figure 6 where we compare the baseline model's deferral and correct/incorrect separation ability against our fine-tuned model at different αs. We generally observe a similar trend as in the image classification results: higher αs maintain raw prediction performance closer to the baseline model but do not significantly improve correct/incorrect separation. At the same time, low αs improve deferral more substantially at the cost of accuracy on the full data distribution. In addition to the baseline model (i.e., a model that was not fine-tuned with *Gatekeeper* but from which we still compute the predictive entropy as a deferral signal), we also include results for two other uncertainty prompting baselines (details in Appendix B.2): (i) *Reduce Confidence*: where we append additional instructions to the input prompt to encourage the model to reduce confidence when it is uncertain; and (ii) *Answer "N"*: where we instruct the model to answer with "N" if it is uncertain about the answer. Consistent with recent findings in Kadavath et al. (2022), we find that these approaches do not reliably improve separation of correct-incorrect predictions or offer advantages as deferral models.

4.3 Encoder-Decoder Setup (Vision-Language Models)

Finally, we examine models that incorporate both visual and textual processing capabilities, making it ideal for tasks that require a comprehensive understanding of image content in conjunction with language generation. We consider the PaliGemma (Steiner et al., 2024) model family which are encoder-decoder models designed to perform VL tasks such as image captioning, visual question answering, and image classification with descriptive outputs. In this setup, the encoder component processes the input images to extract rich feature representations, while the decoder generates corresponding textual classifications or descriptions. We use PaliGemma1B as $\mathcal{M}_S$ and PaliGemma7B as $\mathcal{M}_L$. The deferral strategy involves deploying a smaller VLM to handle the majority of classification tasks, reserving the more resource intensive 7B model for instances where $\mathcal{M}_S$'s predictive entropy falls below a predefined threshold.

Similarly to our experiments on LMs in Section 4.2, we employ two stages of fine-tuning. First, we take the instruction-tuned checkpoints of PaliGemma1B and PaliGemma7B and then fine-tune both models on the training split of a given dataset. Next, we fine-tune only $\mathcal{M}_S$ using *Gatekeeper* before evaluating the model on a validation/test split of the dataset. The datasets we consider are two classification datasets (VQAv2 (Goyal et al., 2017), AI2D (Hiippala et al., 2021)) and two captioning datasets (Cococap (Lin et al., 2014), Screen2Words (Wang et al., 2021)). This allows us to evaluate *Gatekeeper* in both closed-form vision-language classification setups as well as open-form text generation.

Factuality Scoring. For classification tasks we apply our analysis in the same way as in Section 4.2. However, for captioning datasets we need to evaluate the quality of a caption generated by PaliGemma. To do that, we compute a factuality score which judges whether the generated caption is semantically coherent with respect to a reference caption using the Gemini LLM (GeminiTeam et al., 2023). Specifically, the Gemini LLM is prompted with an instruction of the form: *"Are these captions semantically equivalent?"*, followed by both the candidate caption and the reference caption. The model then responds with either *"Yes"* or *"No"*. Finally, we compute the log-likelihood of each response and normalize it to a probability, reflecting the LLM's confidence in the captions being factually aligned. We detail this process in Appendix B.4 and denote the factuality score for input point $\mathbf{x}_i$ with candidate caption $\hat{\mathbf{y}}_i$ and ground truth caption $\mathbf{y}_i$ as $s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$.

Measuring Correlation Between Factuality and Negative Predictive Entropy. Since the result of evaluating $s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$ is no longer binary, our evaluation metrics which previously relied on accuracy cannot be used directly to evaluate deferral performance and the correct/incorrect entropy distribution separation. We address this issue by replacing the distributional overlap computation with the Pearson correlation $\rho(g_{\text{NENT}}(\mathbf{x}_i), s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i))$ between the negative predictive entropy of a caption $g_{\text{NENT}}(\mathbf{x}_i)$ and its associated factuality score $s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i))$. We also adapt our deferral performance metric from Equation (10) to rely on factuality measures instead of accuracy.

Results. We document our results in Figure 7 where we compare the baseline model's deferral ability against our fine-tuned models at different αs. For the classification results (Figure 7 left), we observe the same trends as outlined in our classification and language modeling experiments. For the captioning results (Figure 7(b) right) we observe that *Gate-*

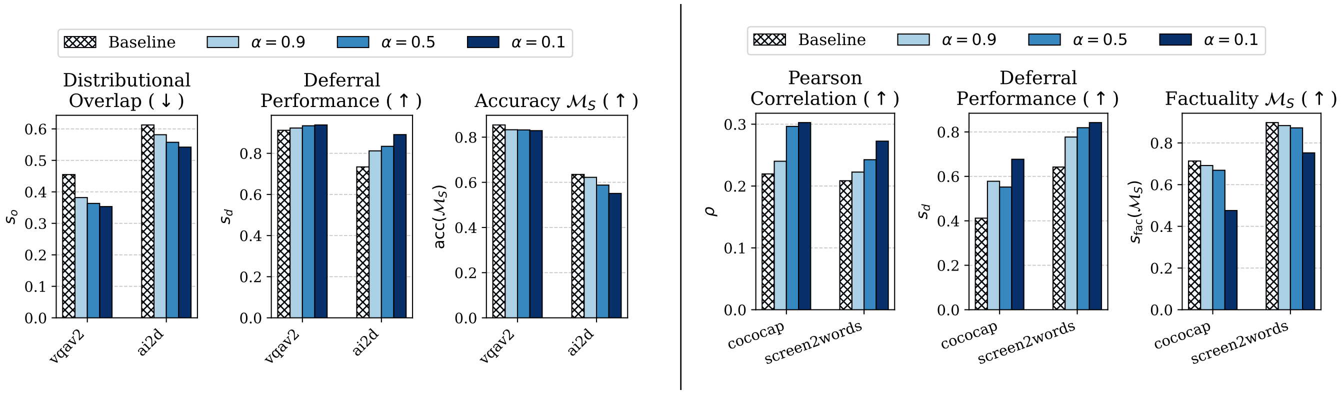

Figure 7: **Performance on VLM classification (left) and captioning tasks (right)**. Consistent with results in Figures 4 and 6, we see that smaller αs lead to improved deferral performance in both classification and generation tasks.

keeper measurably increases the correlation between factuality and negative predictive entropy, yielding better deferral from $\mathcal{M}_S$ to $\mathcal{M}_L$ with decreasing α. This demonstrates that our method does not just work on classification problems but also generalizes to sequence generation tasks. Note that while we also benchmarked the prompting baselines from Section 4.2 for these experiments, the PaliGemma model did not return any responses for these modified prompts (likely due to PaliGemma's rigid pretraining and prompting instructions (Beyer et al., 2024)).

5 Conclusion

In this work we present a novel loss function called *Gatekeeper* for improving confidence calibration in a cascade between a small local and a larger remote model. Our loss is architecture and task agnostic, making it flexibly applicable across a wide range of applications. Our results demonstrate that our approach improves over standard confidence-based deferral rules and effectively leads the small model to unlearn how to handle complex queries in favor of easier ones.

Limitations While our approach demonstrates promising results, there are a few notable constraints. First, we assume that only the smaller model can be tuned, while in some application the larger model might also adjustable for deferral. Second, our experiments primarily measure improvement over a single untuned baseline, potentially overlooking broader comparative insights. Third, we did not extensively evaluate across different model families in LLM and VLM settings (although we did so in classification experiments with ResNet vs. MobileNet). Finally, using a generative model (e.g., Gemini) to judge VLM captioning introduces the risk of erroneous assessments since LLMs are also imperfect oracles.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Models overview - anthropic. `https://docs.anthropic.com/en/docs/build-with-claude/citations`, 2024. [Online; accessed 24-January-2025].

Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Bolukbasi, T., Wang, J., Dekel, O., and Saligrama, V. Adaptive neural networks for fast test-time prediction. 2017.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Chen, Z., Yang, X., Lin, J., Sun, C., Chang, K., and Huang, J. Cascade speculative drafting for even faster LLM inference. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=lZY9u0ijP7.

Cheng, H., Zhang, M., and Shi, J. Q. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024. doi: 10.1109/TPAMI.2024.3447085.

Chuang, Y.-N., Zhou, H., Sarma, P. K., Gopalan, P., Boccio, J., Bolouki, S., and Hu, X. Learning to route with confidence tokens, 2024. URL https://arxiv.org/abs/2410.13284.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lopes, R. G., Wu, Y., Michalewski, H., Saurous, R. A., Sohl-Dickstein, J., et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Dutta, S., Wei, D., Yueksel, H., Chen, P.-Y., Liu, S., and Varshney, K. Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing. In *International conference on machine learning*, pp. 2803–2813. PMLR, 2020.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. URL http://jmlr.org/papers/v11/el-yaniv10a.html.

Enomoro, S. and Eda, T. Learning to cascade: Confidence calibration for improving the accuracy and computational cost of cascade inference systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7331–7339, May 2021. doi: 10.1609/aaai.v35i8.16900. URL https://ojs.aaai.org/index.php/AAAI/article/view/16900.

Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., and Yannakakis, G. N. Large language models and games: A survey and roadmap. *arXiv preprint arXiv:2402.18659*, 2024.

GeminiTeam, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

GemmaTeam, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Gupta, N., Narasimhan, H., Jitkrittum, W., Rawat, A. S., Menon, A. K., and Kumar, S. Language model cascades: Token-level uncertainty and beyond. 2024. URL `https://openreview.net/forum?id=KgaBScZ4VI`.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hiippala, T., Alikhani, M., Haverinen, J., Kalliokoski, T., Logacheva, E., Orekhova, S., Tuomainen, A., Stone, M., and Bateman, J. A. Ai2d-rst: A multimodal corpus of 1000 primary school science diagrams. *Language Resources and Evaluation*, 55: 661–688, 2021.

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Jitkrittum, W., Gupta, N., Menon, A. K., Narasimhan, H., Rawat, A., and Kumar, S. When does confidence-based cascade deferral suffice? In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9891–9906. Curran Associates, Inc., 2023.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kolawole, S., Dennis, D., Talwalkar, A., and Smith, V. Agreement-based cascading for efficient inference, 2024. URL `https://arxiv.org/abs/2407.02348`.

Krishnan, R., Khanna, P., and Tickoo, O. Enhancing trust in large language models with uncertainty-aware fine-tuning, 2024. URL `https://arxiv.org/abs/2412.02904`.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=VD-AYtP0dve`.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Le, Y. and Yang, X. S. Tiny imagenet visual recognition challenge. 2015.

Leviathan, Y. Looking back at speculative decoding, 2024. URL `https://research.google/blog/looking-back-at-speculative-decoding/`.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Li, Y., Wang, S., Ding, H., and Chen, H. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pp. 374–382, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Ma, X., Fang, G., and Wang, X. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=J8Ajf9WfXP`.

Malinin, A. and Gales, M. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=jN5y-zb5Q7m`.

Mamou, J., Pereg, O., Wasserblat, M., and Schwartz, R. Tangobert: Reducing inference cost by using cascaded architecture, 2022. URL `https://arxiv.org/abs/2204.06271`.

Mielke, S. J., Szlam, A., Dinan, E., and Boureau, Y.-L. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10: 857–872, 2022.

Narasimhan, H., Jitkrittum, W., Menon, A. K., Rawat, A., and Kumar, S. Post-hoc estimators for learning to defer to an expert. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 29292–29304. Curran Associates, Inc., 2022.

Nazi, Z. A. and Peng, W. Large language models in healthcare and medical domain: A review. *Informatics*, 11(3), 2024. ISSN 2227-9709. doi: 10.3390/informatics11030057. URL `https://www.mdpi.com/2227-9709/11/3/57`.

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently scaling transformer inference. In Song, D., Carbin, M., and Chen, T. (eds.), *Proceedings of Machine Learning and Systems*, volume 5, pp. 606–624. Curan, 2023.

Rabanser, S., Thudi, A., Guha Thakurta, A., Dvijotham, K., and Papernot, N. Training private models that know what they don't know. *Advances in Neural Information Processing Systems*, 36:53711–53727, 2023.

Shrivastava, V., Liang, P., and Kumar, A. Llamas know what gpts don't show: Surrogate models for confidence estimation. *arXiv preprint arXiv:2311.08877*, 2023.

Steiner, A., Pinto, A. S., Tschannen, M., Keysers, D., Wang, X., Bitton, Y., Gritsenko, A., Minderer, M., Sherbondy, A., Long, S., et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

Teerapittayanon, S., McDanel, B., and Kung, H.-T. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Trapeznikov, K. and Saligrama, V. Supervised sequential classification under budget constraints. In Carvalho, C. M. and Ravikumar, P. (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 581–589, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL `https://proceedings.mlr.press/v31/trapeznikov13a.html`.

Varshney, N. and Baral, C. Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the*

2022 Conference on Empirical Methods in Natural Language Processing, pp. 11007–11021, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.756. URL `https://aclanthology.org/2022.emnlp-main.756`.

Viola, P. and Jones, M. 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, pp. I–I, 2001. doi: 10.1109/CVPR.2001.990517.

Wang, B., Li, G., Zhou, X., Chen, Z., Grossman, T., and Li, Y. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 498–510, 2021.

Wang, C., Augenstein, S., Rush, K., Jitkrittum, W., Narasimhan, H., Rawat, A. S., Menon, A. K., and Go, A. Cascade-aware training of language models, 2024a. URL `https://arxiv.org/abs/2406.00060`.

Wang, S., Xu, T., Li, H., Zhang, C., Liang, J., Tang, J., Yu, P. S., and Wen, Q. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*, 2024b.

Wang, X., Luo, Y., Crankshaw, D., Tumanov, A., and Gonzalez, J. E. Idk cascades: Fast deep learning by learning not to overthink. In *Conference on Uncertainty in Artificial Intelligence*, 2017.

Xiong, M., Hu, Z., Lu, X., LI, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gjeQKFxFpZ`.

Yaghini, M., Liu, P., Boenisch, F., and Papernot, N. Learning to walk impartially on the pareto frontier of fairness, privacy, and utility. 2023.

Yang, C., Zhu, Y., Lu, W., Wang, Y., Chen, Q., Gao, C., Yan, B., and Chen, Y. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Trans. Intell. Syst. Technol.*, October 2024. ISSN 2157-6904. doi: 10.1145/3699518. URL `https://doi.org/10.1145/3699518`. Just Accepted.

Yue, M., Zhao, J., Zhang, M., Du, L., and Yao, Z. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=6okaSfANzh`.

A Additional Background

A.1 Model Access Levels

In Figure 8, we show a schematic overview of different model access levels discussed in Section 2.

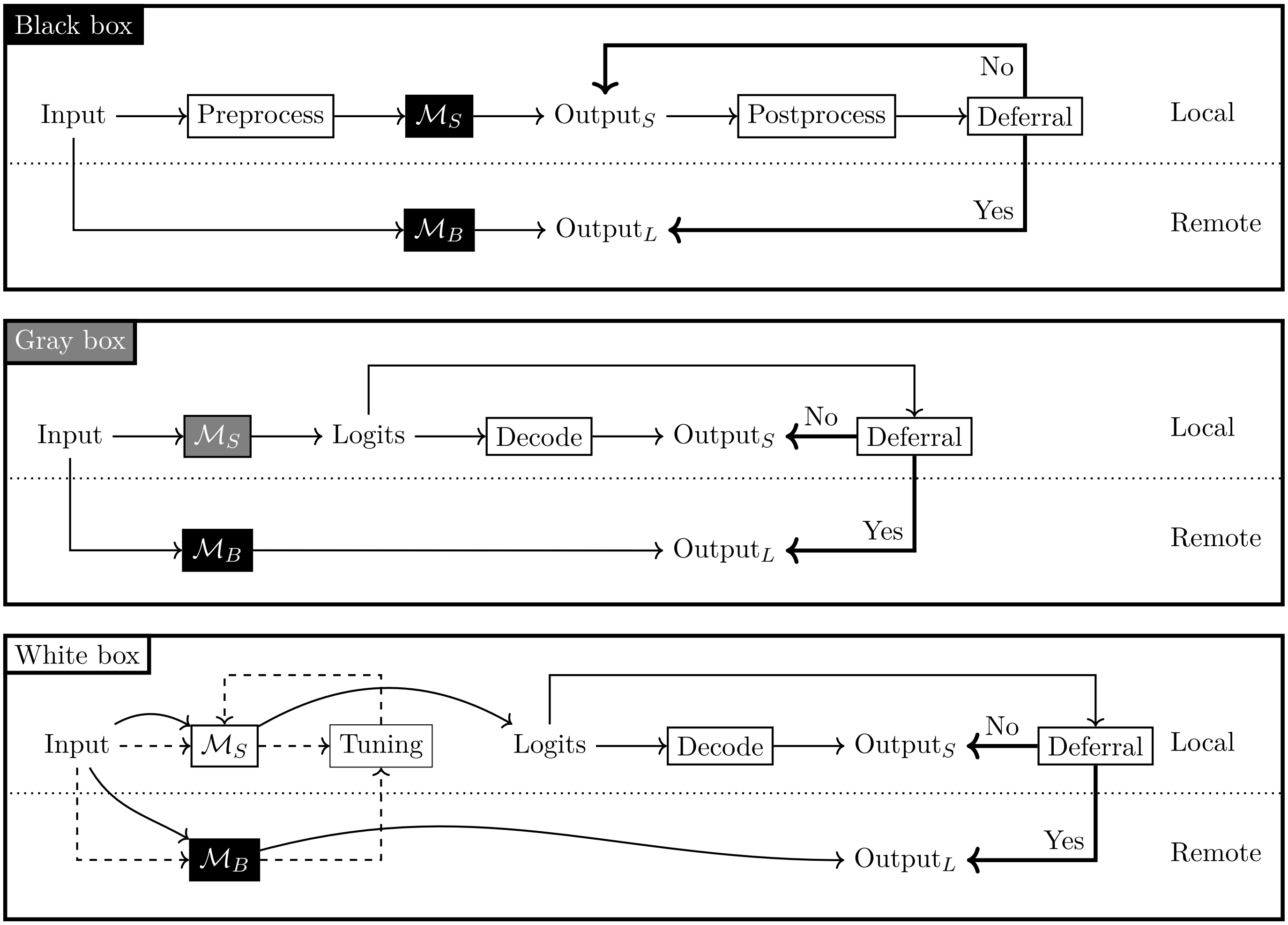

Figure 8: **An overview of different uncertainty quantification strategies depending on model access level**.

A.2 Ideal Deferral Curve

We present the functional form of the *ideal deferral* curve, denoted $\text{acc}_{\text{ideal}}(r)$, for a small (student) model $\mathcal{M}_S$ and a large (teacher) model $\mathcal{M}_L$. Recall that $r \in [0, 1]$ denotes the deferral ratio, i.e., the fraction of inputs that $\mathcal{M}_S$ "defers" to $\mathcal{M}_L$. Let $p_s = \text{acc}(\mathcal{M}_S)$, and $p_l = \text{acc}(\mathcal{M}_L)$ with $0 \leq p_s \leq p_l \leq 1$. Our goal is to describe the maximum achievable joint accuracy if exactly a fraction r of the data is deferred to the large model.

Intuition and Setup Since $\mathcal{M}_S$ achieves accuracy p_s, it misclassifies a fraction $(1 - p_s)$ of the inputs. In an *ideal* scenario, we defer exactly those inputs that $\mathcal{M}_S$ is going to misclassify. Because $\mathcal{M}_L$ is more accurate $(p_l \geq p_s)$ every example misclassified by $\mathcal{M}_S$ benefits from being passed to $\mathcal{M}_L$.

- **Case 1:** $r \le (1 - p_s)$.
  We can use our entire deferral "budget" r to cover only those inputs $\mathcal{M}_S$ would get wrong. Hence, deferring a fraction r of the data (all from $\mathcal{M}_S$'s mistakes) raises the overall accuracy by substituting $\mathcal{M}_S$'s errors with $\mathcal{M}_L$'s accuracy p_l on that fraction.

- **Case 2:** $r > (1 - p_s)$.
  We have enough capacity to defer *all* of $\mathcal{M}_S$'s mistakes, so the joint accuracy saturates at p_l. Deferring *additional* examples (which $\mathcal{M}_S$ would have classified correctly) will not improve the overall accuracy beyond p_l.

Piecewise Functional Form Thus, the *ideal deferral* curve can be expressed as:

$$\mathrm{acc}_{\mathrm{ideal}}(r) \;=\; \begin{cases} p_s + \dfrac{p_l - p_s}{1 - p_s}\, r, & 0 \;\le\; r \;\le\; (1 - p_s), \\[2ex] p_l, & (1 - p_s) \;<\; r \;\le\; 1. \end{cases} \tag{11}$$

When $0 \le r \le (1 - p_s)$, the overall accuracy grows linearly from $\mathrm{acc}_{\mathrm{ideal}}(0) = p_s$ to $\mathrm{acc}_{\mathrm{ideal}}(1 - p_s) = p_l$. Past $r = (1 - p_s)$, it remains constant at p_l.

Figure 3 (b) in the main paper plots this ideal deferral curve (green line). It serves as an upper bound on how effective any real deferral strategy can be. In contrast, a purely random deferral strategy produces a linear interpolation (the red line), which is strictly below the ideal curve for most r. Consequently, the difference $\mathrm{acc}_{\mathrm{ideal}}(r) - \mathrm{acc}_{\mathrm{rand}}(r)$ represents the *maximum possible* gain one can achieve by carefully selecting which examples to defer rather than choosing them at random.

Summary We summarize the key take-aways below:

- **Ideal Deferral Routes All Mistakes:** Only the inputs misclassified by $\mathcal{M}_S$ get deferred, guaranteeing the highest possible joint accuracy at each deferral level r.

- **Piecewise Definition:** Accuracy increases linearly from p_s to p_l over the interval $r \in [0, (1 - p_s)]$, then remains at p_l.

- **Upper Bound on Realized Deferral:** No actual strategy can exceed this ideal curve, as it assumes perfect knowledge of which specific inputs $\mathcal{M}_S$ would misclassify.

B Additional Experimental Details

B.1 CNN Used in Image Classification Experiments

Below we include a representation of the SmallCNN model used as $\mathcal{M}_S$ in image classification experiments discussed in Section 4.1:

```
SmallCNN(
  (features): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU(inplace=True)
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Linear(in_features=2048, out_features=64, bias=True)
    (1): ReLU(inplace=True)
    (2): Linear(in_features=64, out_features=10, bias=True)
  )
)
```

B.2 Reduce Confidence and Answer "N" Baselines

In addition to the baseline model in Section 4.2 (i.e., a model that was not fine-tuned with our specialized $\mathcal{L}_{\text{def}}$ loss but from which we still compute predictive entropy as a deferral signal), we also examine two additional methods aimed at eliciting uncertainty from the model directly via prompt modifications. Both methods are *black box* approaches that only rely on a query interface to the model via prompt injection, and we provide their implementation details below.

Reduce Confidence. In this setting, we modify the original prompt $\mathbf{x}$ by appending an additional instruction $\mathbf{x}'$ that encourages the model to respond with lower confidence when it is uncertain: $\mathbf{x} \leftarrow \mathbf{x} \mid \mathbf{x}'$. For instance, the instruction we add is:

$$\mathbf{x}' = \texttt{``Respond with low confidence if you are uncertain.''}$$

We treat this appended text as a hint to the model to self-regulate its confidence when producing an answer. This is similar in spirit to other black box approaches such as confidence quantification, rejection awareness, remote model notice, and self-critiquing. Although Xiong et al. (2024) show that large language models can express aspects of their confidence via prompting, our experiments indicate that simply prompting the model to express lower confidence does not reliably improve the separation of correct versus incorrect predictions, nor does it offer advantages in a deferral setting. These findings are in line with those reported in Kadavath et al. (2022).

Answer "N." We also consider an alternate prompt modification, in which the appended instruction is:

$$\mathbf{x}' = \texttt{``Respond with `N' if you are uncertain.''}$$

This approach explicitly instructs the model to produce a special "N" token to indicate uncertainty or lack of confidence. The intuition is that by introducing a designated "uncertain" response, one might isolate uncertain cases for deferral. However, our results in Section 4.2 similarly show that the model's ability to follow this instruction is inconsistent and does not substantially improve performance as a deferral model. The model often remains overconfident and fails to produce "N" in cases where it is in fact incorrect.

B.3 Additional metrics

In addition to the metrics outlined in Section 4, we also consider the **Area Under the Receiver Operating Characteristic Curve** (AUROC) (s_{AUROC}). The AUROC quantifies the model's ability to discriminate between correctly and incorrectly classified data points by evaluating the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across various confidence thresholds τ. Formally, given the confidence sets $\mathcal{C}_{\text{corr}}$ and $\mathcal{C}_{\text{incorr}}$, the AUROC is defined as

$$s_{\text{AUROC}} = \int_0^1 \text{TPR}(\tau)\, \text{dFPR}(\tau), \tag{12}$$

where for each threshold $\tau \in [0, 1]$ we compute $\text{TPR}(\tau) = \frac{|\{c \in \mathcal{C}_{\text{corr}} | c \geq \tau\}|}{|\mathcal{C}_{\text{corr}}|}$ and $\text{FPR}(\tau) = \frac{|\{c \in \mathcal{C}_{\text{incorr}} | c \geq \tau\}|}{|\mathcal{C}_{\text{incorr}}|}$. Note that $s_{\text{AUROC}} = 1$ indicates perfect separability and $s_{\text{AUROC}} = 0.5$ corresponds to a random guessing baseline.

B.4 Factuality Scoring

Factuality scoring with Gemini for a reference caption r and a candidate caption c is computed as follows:

1. **Compute the log-likelihoods.** Let $\ell_{\text{Same}}(c, r)$ be the log-likelihood that the model outputs "Same" for a given candidate caption c and reference r, and let $\ell_{\text{Diff}}(c, r)$ be the log-likelihood that the model outputs "Different".

2. **Apply softmax.** To convert these log-likelihoods into probabilities, we exponentiate and normalize:

$$p(\text{Same} \mid c, r) = \frac{\exp\big(\ell_{\text{Same}}(c, r)\big)}{\exp\big(\ell_{\text{Same}}(c, r)\big) + \exp\big(\ell_{\text{Diff}}(c, r)\big)},$$

$$p(\text{Diff} \mid c, r) = \frac{\exp\big(\ell_{\text{Diff}}(c, r)\big)}{\exp\big(\ell_{\text{Same}}(c, r)\big) + \exp\big(\ell_{\text{Diff}}(c, r)\big)}.$$

3. **Interpret the probability.** The value $p(\text{Same} \mid c, r)$ is then taken as the factual alignment score, expressing how confidently the model believes the candidate caption is factually aligned with the reference.

B.5 Additional Experimental Results

In this section, we provide additional experimental results further supporting our findings reported for image classification experiments in Section 4.1. In particular, we show ROC curves in Figure 9 and distributional overlap in Figure 10, both demonstrating that *Gatekeeper* increases the separation of correct/incorrect confidence scores. Similarly, the deferral curves in Figure 11 clearly show that *Gatekeeper* successfully pushed the realized deferral (black line) closer to the ideal one (marked with dashed upper line). Lastly, we report the joint accuracy of $\mathcal{M}_S$ across varying α parameter in Figure 12. As discussed in Section 4, we observe that $\mathcal{M}_S$'s accuracy generally decreases with $\alpha \to 0$.

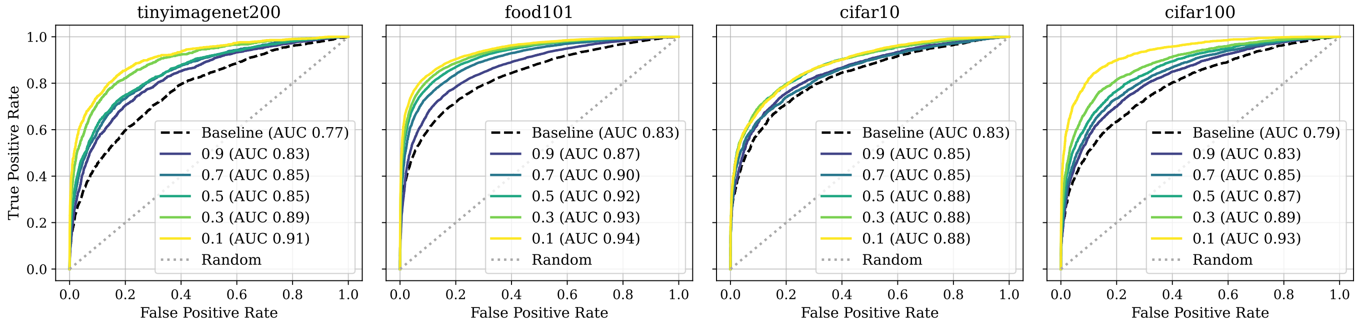

Figure 9: **ROC curves for image classification experiments**. Each figure shows the ROC curves for each of the datasets considered in Section 4.1. We observe that *Gatekeeper* consistently increases separation of correct and incorrect confidence scores across varying α (colored curves) compared to the baseline (denoted with black dashed line).

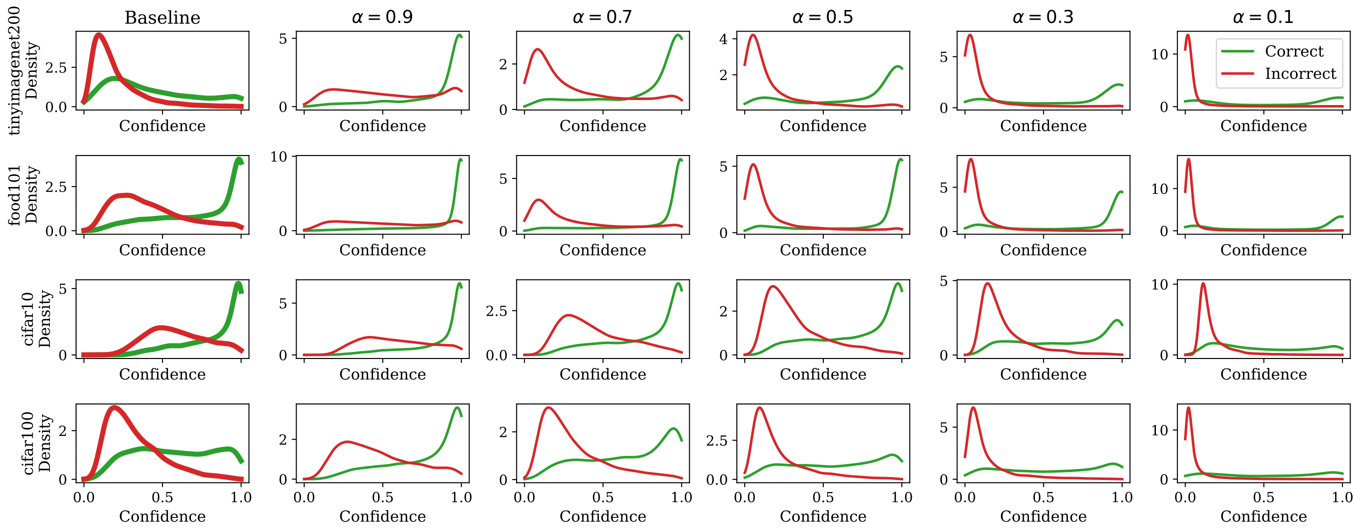

Figure 10: **Distributional overlap for image classification experiments**. Left-most column shows the results obtained using the untuned baseline, while the remaining columns correspond to the results obtained using *Gatekeeper* with decreasing α values. Rows correspond to the datasets considered in Section 4.1. We see that *Gatekeeper* increases separation of correct and incorrect confidence scores compared to the baseline.

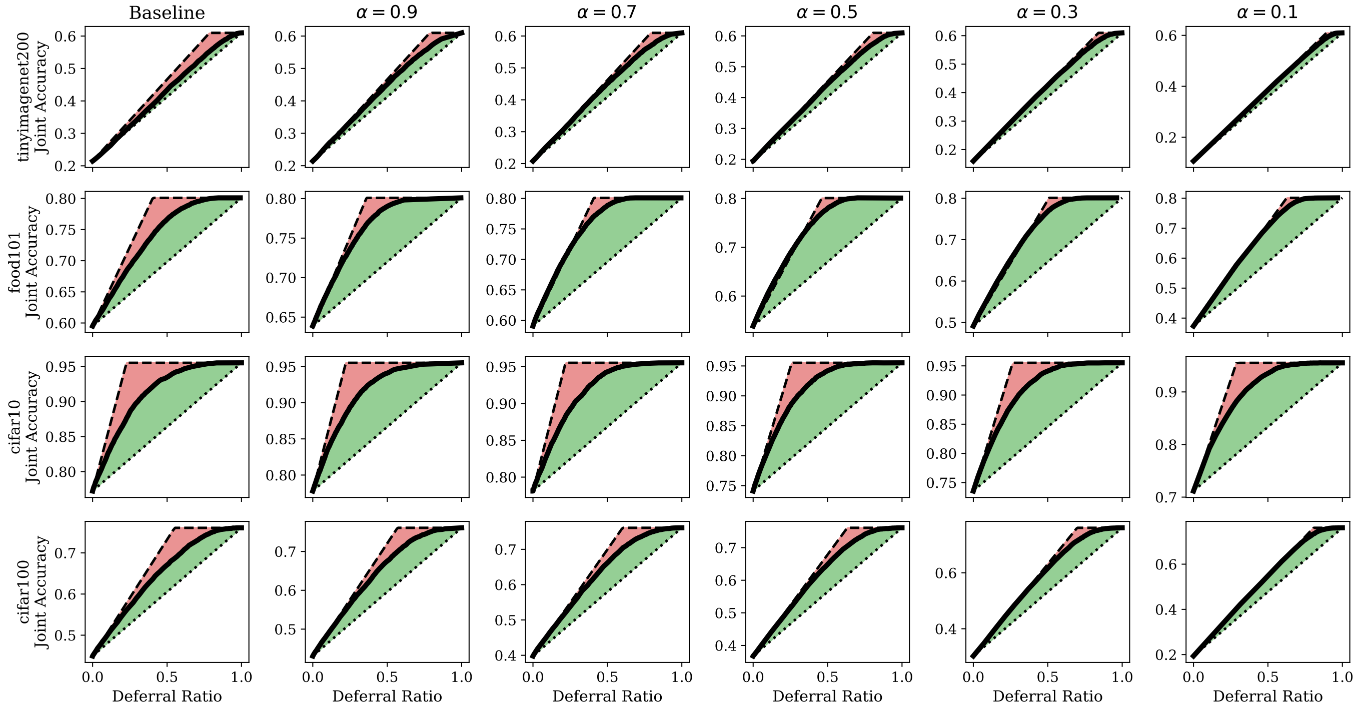

Figure 11: **Deferral curves for image classification experiments**. Left-most column shows the results obtained using the untuned baseline, while the remaining columns correspond to the results obtained using *Gatekeeper* with decreasing α values. Rows correspond to the datasets considered in Section 4.1 The results show that *Gatekeeper* brings the realized deferral (black line) closer to the ideal deferral (dashed upper line).

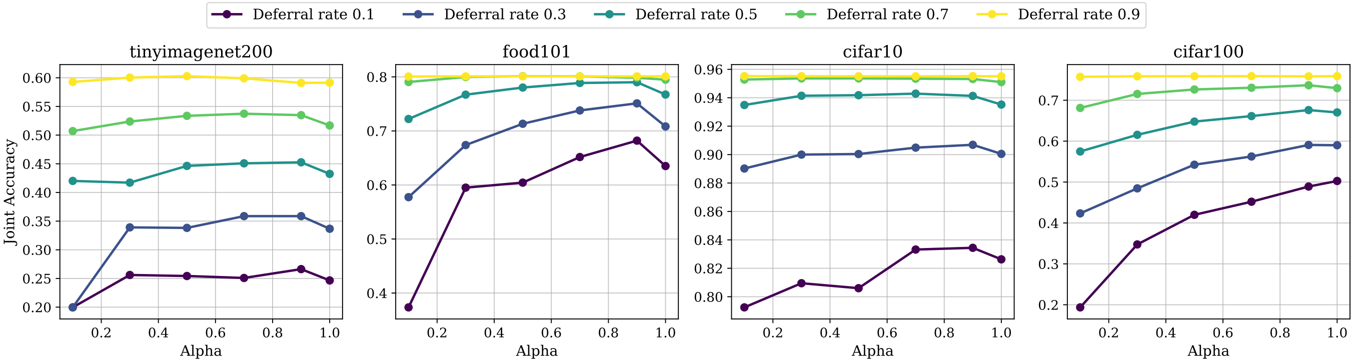

Figure 12: **Joint accuracy across different levels of** α. For varying fixed deferral ratios, we observe that the accuracy of $\mathcal{M}_S$ generally decreases as $\alpha \to 0$.