GATEKEEPER: Improving Model Cascades Through Confidence Tuning

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

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 evaluate 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

Large-scale machine learning models such as Gemini (GeminiTeam et al., 2023), GPT-4 (Achiam et al., 2023), and Claude (Anthropic, 2024) have demonstrated remarkable capabilities across diverse tasks, including language understanding, generation, and computer vision. Their strong generalization has enabled deployment in domains such as healthcare (Nazi & Peng, 2024), finance (Li et al., 2023), education (Wang et al., 2024b), and entertainment (Gallotta et al., 2024). However, the substantial computational, memory, and latency costs associated with these models pose significant scalability issues (Pope et al., 2023).

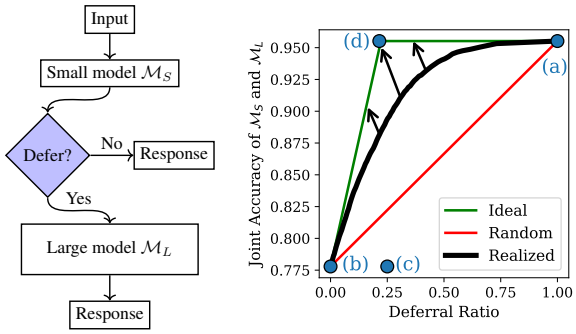

Figure 1. **Overview of the cascading setup (left) and performance trade-off (right)**. *Left*: Cascading determines which inputs should be predicted by a small model $\mathcal{M}_S$ or routed to a large model $\mathcal{M}_L$. *Right*: Performance is measured as a trade-off between joint accuracy across $\mathcal{M}_S$ 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 handled by $\mathcal{M}_S$.

To address these challenges, two main strategies have emerged. The first compresses the large model itself—via pruning (Ma et al., 2023), distillation (Yang et al., 2024), or sparsity techniques (Hoefler et al., 2021). The second—arguably more promising given empirical scaling laws (Kaplan et al., 2020)—maintains the large model $\mathcal{M}_L$ and reduces cost by selectively offloading easy inputs to a smaller, cheaper model $\mathcal{M}_S$. This idea underlies speculative decoding (Leviathan et al., 2023) and model cascades (Dohan et al., 2022; Gupta et al., 2024; Chen et al., 2024a), which pair a fast $\mathcal{M}_S$ with a more powerful $\mathcal{M}_L$.

In speculative decoding, $\mathcal{M}_S$ drafts candidate outputs that $\mathcal{M}_L$ verifies in parallel. In contrast, model cascades rely on a deferral rule to decide which model should handle a given request (Figure 1, left). Cascades are attractive because

[1]University of Toronto and Vector Institute [2]Google. Correspondence to: Stephan Rabanser <stephan@cs.toronto.edu>. Work done while a Student Researcher at Google.

Presented at the TTODLer-FM workshop at the International Conference on Machine Learning (ICML), Vancouver, Canada. 2025.

they permit the use of a less capable $\mathcal{M}_S$—provided it can accurately identify when it is likely to make mistakes. The success of such systems hinges on balancing compute cost and joint accuracy. As shown in Figure 1 (right), cascades can fail if $\mathcal{M}_S$ defers too often, too little, or defers only on correct predictions. The optimal scenario occurs when $\mathcal{M}_S$ defers *only* when it is wrong, yielding the best accuracy–efficiency trade-off. We refer to how well a cascade approximates this ideal as its *deferral performance*.

Our Contribution. We address the question: *How can we optimize model cascades to maximize deferral performance?* That is, how can we train $\mathcal{M}_S$ not only to be competent on easy examples, but also to know when to defer? We propose GATEKEEPER, a simple and general-purpose loss function that fine-tunes $\mathcal{M}_S$ to output high confidence when correct and low confidence when incorrect. This calibration improves uncertainty estimates, enabling more accurate routing and better cascade-level performance. Our approach directly shapes $\mathcal{M}_S$'s confidence through uncertainty-aware fine-tuning. GATEKEEPER also includes a tunable parameter that controls the trade-off between predictive and deferral performance, and is applicable to a wide range of architectures without architectural modifications.

We demonstrate the effectiveness of GATEKEEPER on encoder-only vision models, decoder-only language models, and encoder-decoder vision-language models. Across tasks such as image classification, closed-form text generation, and image captioning, GATEKEEPER significantly improves deferral performance. For instance, it achieves up to $2\times$ improvement on TinyImageNet and $10\times$ on ARC-e/c. These results show that GATEKEEPER enables more reliable, cost-efficient cascaded inference—paving the way for scalable deployment of machine learning systems across domains.

2. Related Work

Model Cascades: Model cascades consist of a deferral rule and a sequence of models, routing inputs to the appropriate model based on difficulty. Originally introduced to accelerate object detection (Viola & Jones, 2001), cascades have since been applied in classification (Wang et al., 2017; Trapeznikov & Saligrama, 2013; Bolukbasi et al., 2017; Jitkrittum et al., 2023) and NLP (Dohan et al., 2022; Mamou et al., 2022; Varshney & Baral, 2022).

Cascades are especially useful for LLMs and VLMs, reducing inference cost by deferring only hard examples to larger models. Unlike speculative decoding (Leviathan et al., 2023), which accelerates generation, cascades focus on selective model invocation, though both can be combined (Narasimhan et al., 2025; Chen et al., 2024b). Most prior work applies post-hoc deferral logic to pre-trained models (Narasimhan et al., 2022; Yue et al., 2024; Kolawole et al., 2024; Gupta et al., 2024). Recent approaches improve deferral through training: Wang et al. (2024a) restrict $\mathcal{M}_S$ training to easier tokens; Enomoro & Eda (2021) introduce calibration-aware training. Our method extends these ideas to VLMs and generative models by encouraging $\mathcal{M}_S$ to be uncertain when wrong.

Uncertainty-Aware Models: Uncertainty estimation is well-studied for classifiers (Abdar et al., 2021), but remains challenging for generative models. Methods differ by access to model internals:

1. *Black-box* methods modify prompts to elicit more cautious responses (Shrivastava et al., 2023; Kadavath et al., 2022; Gou et al., 2023; Xiong et al., 2024).

2. *Gray-box* approaches analyze model outputs using entropy or logit post-processing (Hendrycks & Gimpel, 2016; Malinin & Gales, 2021; Kuhn et al., 2023). While effective, techniques like ensembling (Lakshminarayanan et al., 2017) and Bayesian inference (Blundell et al., 2015) are often impractical at scale.

3. *White-box* methods incorporate uncertainty into training objectives (Chuang et al., 2024; Krishnan et al., 2024). Rawat et al. (2021) pre-partition data based on $\mathcal{M}_L$'s confidence to train $\mathcal{M}_S$. In contrast, we dynamically adjust training based on $\mathcal{M}_S$'s uncertainty, ensuring it is confident when correct and uncertain when wrong. This improves performance in cascade setups.

3. The GATEKEEPER Loss

3.1. Overview & Setup

We consider a model cascade consisting of a large, accurate model $\mathcal{M}_L$ and a smaller, resource-efficient model $\mathcal{M}_S$, with parameter counts $L \gg S$. Both models may be classifiers ($\mathcal{M} : \mathbb{R}^D \to [C]$) or sequence models ($\mathcal{M} : \mathbb{R}^D \to [V]^T$), and need not share the same architecture family. For instance, $\mathcal{M}_S$ may be a lightweight CNN and $\mathcal{M}_L$ a vision transformer (Dosovitskiy, 2020). Our goal is to fine-tune $\mathcal{M}_S$ such that it knows when to trust its predictions and when to defer to $\mathcal{M}_L$. Deferral signals are derived solely from $\mathcal{M}_S$. This avoids the additional compute cost of methods that query $\mathcal{M}_L$ at inference time (Mielke et al., 2022; Kuhn et al., 2023), which defeats the purpose of deferral. Our setup assumes $\mathcal{M}_S$ is strictly less capable than $\mathcal{M}_L$ —consistent with scaling laws (Kaplan et al., 2020)—so errors made by $\mathcal{M}_L$ are likely also made by $\mathcal{M}_S$, but not vice versa.

We assume white-box access to $\mathcal{M}_S$, enabling us to integrate the deferral mechanism into its training objective. This allows $\mathcal{M}_S$ to learn during fine-tuning how to distinguish between reliable and unreliable predictions. Rather than

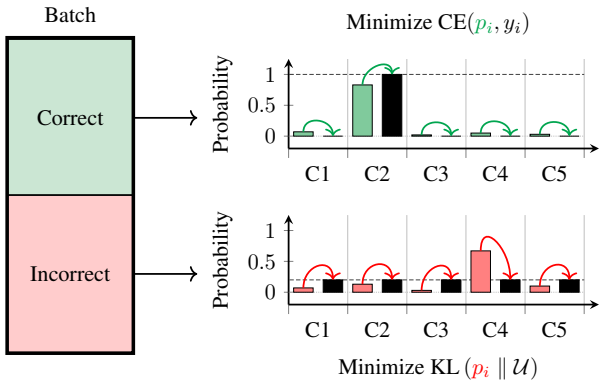

Figure 2. **GATEKEEPER Overview:** Correct predictions are made more confident (lower cross-entropy), while incorrect predictions are trained to produce uniform output distributions (higher entropy). This enables better deferral decisions at test time.

relying on post-hoc uncertainty thresholds or hand-crafted heuristics, we ask: **Can we directly optimize $\mathcal{M}_S$ to separate correct from incorrect predictions via fine-tuning?** We show this is achievable using a new loss that requires no architectural changes and integrates into standard pipelines.

3.2. Confidence-Tuning for Deferral

Stage 1: Initial Training. We begin with a $\mathcal{M}_S$ that has been trained on the target task. We make no assumptions about the training pipeline—$\mathcal{M}_S$ may be trained from scratch or via distillation.

Stage 2: Correctness-Aware Fine-Tuning. We then fine-tune $\mathcal{M}_S$ using a novel hybrid loss function, GATEKEEPER, which promotes high confidence on correct predictions and low confidence on incorrect ones (see Figure 2). This objective is based on the intuition that a reliable deferral mechanism requires well-calibrated uncertainty estimates.

The loss takes the form:

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

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

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

where p_i is the predicted distribution for input $\mathbf{x}_i$, y_i is the true label, $\hat{y}_i$ is the predicted label, $\mathcal{U}$ is the uniform distribution over all classes, and N is the batch size. The cross-entropy term reduces the entropy of correct predictions, while the KL term increases entropy of incorrect ones.

The scalar $\alpha \in (0, 1)$ balances the two terms. Smaller values emphasize penalizing overconfident errors (increasing deferral conservativeness), while larger values sharpen correct predictions and improve predictive accuracy. Thus, α directly controls the trade-off between deferral reliability and standalone model utility.

This idea is inspired by the OE loss for out-of-distribution detection (Hendrycks et al., 2018), which encourages uniform predictions on outlier data. However, to our knowledge, this formulation has not been used for improving deferral in model cascades.

Extension to Token-Based Models. For sequence models (e.g., LMs or VLMs), we compute the loss token-wise:

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

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

where T is the sequence length, and $p_{i,t}$ is the token distribution at position t. This ensures fine-grained calibration across the sequence.

Stage 3: Confidence-Based Deferral. After fine-tuning with GATEKEEPER, we use calibrated confidence scores from $\mathcal{M}_S$ to make deferral decisions. Following the selective prediction framework (El-Yaniv & Wiener, 2010), we define a gating function $g(\mathbf{x})$ and threshold τ for routing:

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

For classification models, we use max-softmax confidence:

$$g_{\text{CL}}(\mathbf{x}) = \max_c p(y = c \mid \mathbf{x}) \tag{7}$$

For sequence models, we use negative predictive entropy averaged across tokens:

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

In both cases, higher scores indicate higher confidence. By thresholding this signal, we control which inputs $\mathcal{M}_S$ handles autonomously and which are escalated to $\mathcal{M}_L$—enabling an efficient, robust cascaded inference system.

4. Experiments

In this section, we detail the experiments used to evaluate the effectiveness of GATEKEEPER across three distinct model architectures: encoder-only classification models, decoder-only language models, and encoder-decoder vision–language models. Each setup involves a cascade where a smaller model defers uncertain inputs to a larger model.

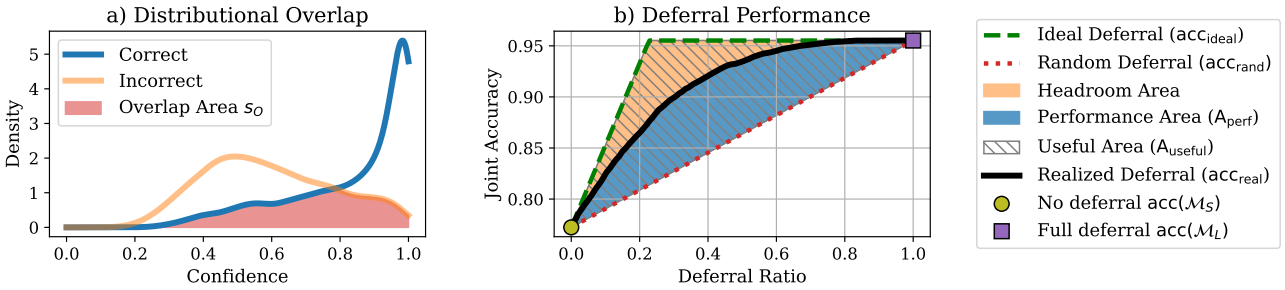

Figure 3. **Performance metrics overview**: **(a)** Distributional Overlap s_o: the densities of confidence scores for correctly (green) and incorrectly classified (red) samples, with the overlap area shaded in blue. Smaller values are better (↓). **(b)** Deferral Performance s_d: how joint accuracy 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 (↑).

4.1. Encoder-only Setup (Classification Models)

We begin our evaluation with image classification using encoder-only models. We train a small model $\mathcal{M}_S$ and a larger model $\mathcal{M}_L$ on CIFAR-10/100 (Krizhevsky et al., 2009), Food-101 (Bossard et al., 2014), and TinyImageNet200 (Le & Yang, 2015). For CIFAR, $\mathcal{M}_L$ is a ResNet-18 and $\mathcal{M}_S$ is a custom CNN. For Food-101 and TinyImageNet, $\mathcal{M}_L$ is a ResNet-50 and $\mathcal{M}_S$ is a MobileNet V3 Small (Howard et al., 2019) trained by distilling $\mathcal{M}_L$.

Evaluation Metrics. We evaluate the effectiveness of GATEKEEPER and the resulting deferral function $g(\cdot)$ using three metrics that reflect different aspects of performance (see Figure 3 for an illustrative example):

1. **Distributional Overlap of Confidence Scores** s_o: Measures how well $\mathcal{M}_S$ separates correct from incorrect predictions based on output confidence. We estimate KDEs over confidence scores for correctly ($\hat{p}_{\text{corr}}$) and incorrectly ($\hat{p}_{\text{incorr}}$) classified samples and define:

$$s_o = \int_0^1 \min\left\{\hat{p}_{\text{corr}}(c),\ \hat{p}_{\text{incorr}}(c)\right\}\, dc. \tag{9}$$

   Lower s_o indicates better separability ($s_o = 0$ for perfect, $s_o = 1$ for complete overlap). This metric is related to AUROC but compares full probability mass rather than prediction ranking.

2. **Deferral Performance** s_d: Quantifies how effectively $\mathcal{M}_S$ defers uncertain inputs to $\mathcal{M}_L$. We compute the joint accuracy $\text{acc}_{\text{real}}(r)$ over varying deferral rates r, and compare it against two baselines: random deferral $\text{acc}_{\text{rand}}(r)$ and ideal deferral $\text{acc}_{\text{ideal}}(r)$. The score is the normalized area between real and random deferral:

$$s_d = \frac{\int_0^1 \left(\text{acc}_{\text{real}}(r) - \text{acc}_{\text{rand}}(r)\right)\, dr}{\int_0^1 \left(\text{acc}_{\text{ideal}}(r) - \text{acc}_{\text{rand}}(r)\right)\, dr}. \tag{10}$$

   A value of $s_d = 1$ indicates optimal deferral; $s_d = 0$ implies no improvement over random routing. See Appendix B.3 for details.

3. **Small Model Accuracy** $\text{acc}(\mathcal{M}_S)$: Since GATEKEEPER prioritizes uncertainty calibration over classification accuracy, $\mathcal{M}_S$ may sacrifice performance on the full data distribution to better recognize its errors. We report $\text{acc}(\mathcal{M}_S)$ to quantify this trade-off and assess utility loss when optimizing for deferral.

Results. Figure 4 shows results for models trained with varying α values, compared against a baseline model and the cascading method of Narasimhan et al. (2022). At low α, GATEKEEPER effectively separates correct and incorrect confidence scores (lower s_o), improving deferral performance s_d. However, this comes with reduced small model accuracy, as $\mathcal{M}_S$ increasingly focuses on easy examples and assigns lower confidence to hard ones. As α increases, accuracy improves or stabilizes, but gains in deferral diminish. Notably, the baseline from Narasimhan et al. (2022) maintains model accuracy but requires auxiliary mechanisms to predict expert correctness—such as additional heads or networks—which can increase deployment complexity. In contrast, GATEKEEPER is architecture-agnostic and operates purely via confidence calibration.

The accuracy-deferral trade-off is further explored in Figure 5, where we observe: (i) a negative correlation between deferral performance s_d and small model accuracy, and (ii) a strong relationship between confidence overlap s_o and both deferral and accuracy. The effect mirrors well-known trade-offs in fairness (Dutta et al., 2020; Yaghini et al., 2023) and privacy (Abadi et al., 2016; Rabanser et al., 2023). Crucially, GATEKEEPER exposes this trade-off explicitly via a single tunable parameter α, allowing practitioners to tailor cascaded systems to their computational or reliability needs.

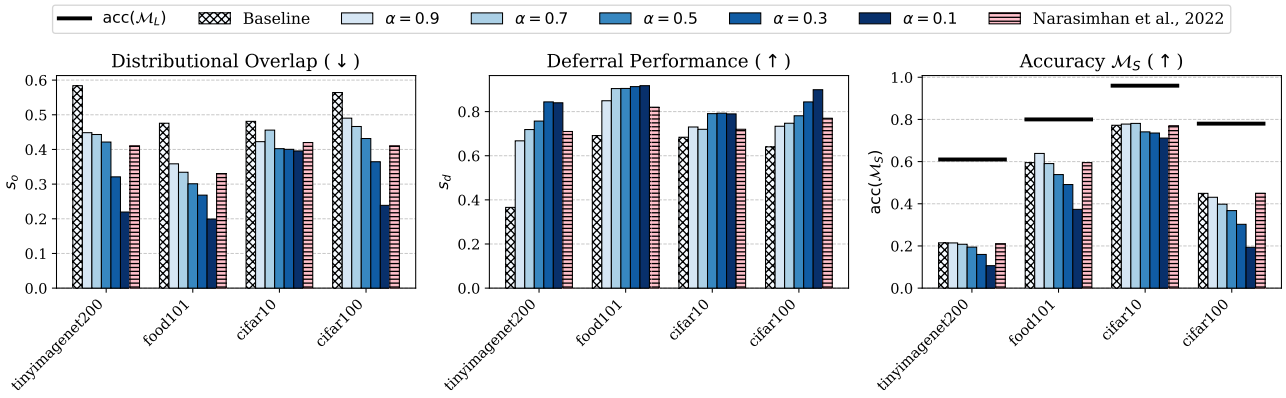

Figure 4. **Performance on image classification tasks**. We observe that lower levels of α lead to decreased distributional overlap between correct/incorrect predictions (left), increased deferral performance (center) and generally decreased performance over the full data distribution (right). These 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 defer to the large model $\mathcal{M}_L$.

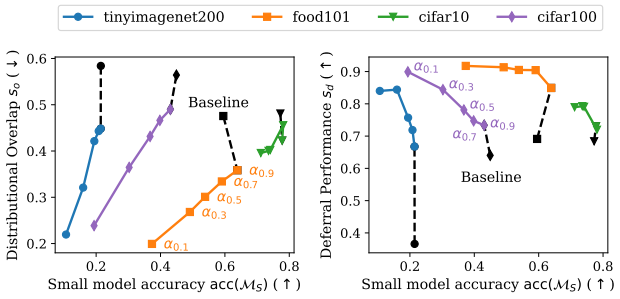

Figure 5. **Performance trade-off between small model accuracy acc($\mathcal{M}_S$) and deferral performance** s_d. The baseline model obtained without fine-tuning using GATEKEEPER is often the most accurate model over the full data distribution. With the introduction of GATEKEEPER we can improve distinguishability 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.

4.2. Decoder-only Setup (Language Models)

We next evaluate GATEKEEPER on decoder-only language models. Our setup uses Gemma2B as $\mathcal{M}_S$ and Gemma7B as $\mathcal{M}_L$ (GemmaTeam et al., 2024), forming a scalable cascade for next-token prediction. As in the classification case, the deferral decision is based solely on the output of $\mathcal{M}_S$, using predictive entropy to route uncertain tokens to $\mathcal{M}_L$.

Both models are first instruction-tuned on the training split of each dataset to ensure strong base performance and adherence to task format. We then fine-tune $\mathcal{M}_S$ with GATEKEEPER to reduce its confidence on incorrect token predictions. Evaluation is performed on the validation split using the same metrics as in Section 4.1. We use three representative benchmarks: ARC-e/c (Clark et al., 2018), MMLU (Hendrycks et al., 2020), and GSM8K (Cobbe et al., 2021). These span symbolic reasoning, factual recall, and multi-step arithmetic generation, respectively.

Results. Figure 6 shows that GATEKEEPER leads to improved deferral performance and better separation of correct/incorrect predictions, particularly at lower α. As with classification, higher α values preserve $\mathcal{M}_S$ accuracy but yield weaker separation. In addition to the baseline (Gemma2B with entropy-based deferral), we compare against: (i) the token-level cascading technique from Gupta et al. (2024); and (ii) two uncertainty prompting baselines (see Appendix C.2): *Reduce Confidence* and *Answer "N"*. These methods follow Kadavath et al. (2022) and aim to elicit uncertainty via prompting. Overall, we find that GATEKEEPER consistently outperforms both prompting and token-level deferral baselines in terms of deferral quality, without modifying the model architecture or inference pipeline.

4.3. Encoder-Decoder Setup (Vision-Language Models)

We conclude by evaluating GATEKEEPER on encoder-decoder vision-language models. Specifically, we use the PaliGemma family (Steiner et al., 2024), which supports tasks like image captioning, VQA, and descriptive classification. $\mathcal{M}_S$ is PaliGemma1B, and $\mathcal{M}_L$ is PaliGemma7B. Our cascade runs $\mathcal{M}_S$ on all inputs and defers to $\mathcal{M}_L$ when the predictive entropy of $\mathcal{M}_S$ is low. Following Section 4.2, we first fine-tune both models on the task using standard supervised instruction tuning. We then apply GATEKEEPER to $\mathcal{M}_S$ alone and evaluate the resulting cascade on 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)), covering both closed-form and generative VL tasks.

Factuality Scoring. For classification tasks, we evaluate accuracy and confidence separation as before. For captioning,

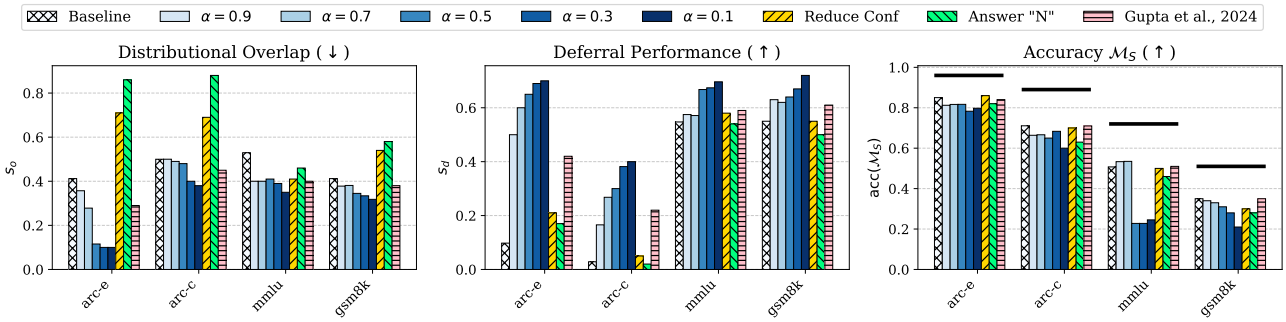

Figure 6. **Performance on language modeling tasks**. Similar as Figure 4. In addition to a non-tuned baseline, we also add an uncertainty prompting baseline, an Answer "N" option, as well as the post-hoc confidence calibration method from Gupta et al. (2024). We observe that GATEKEEPER outperforms other methods at lower levels of α.

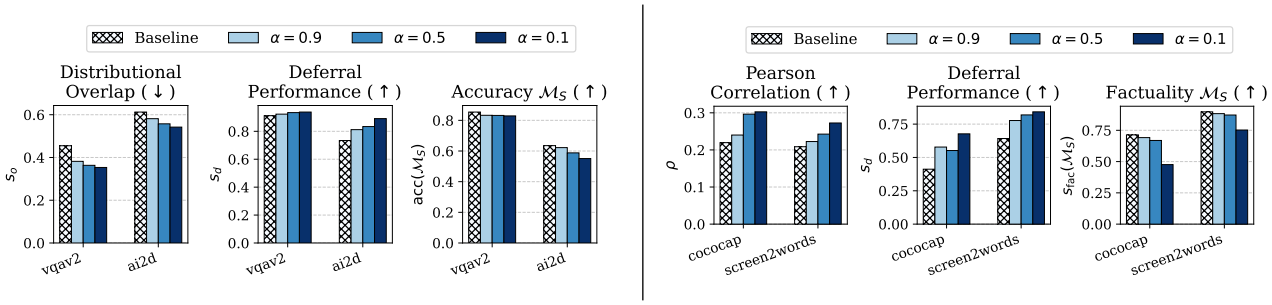

Figure 7. **Performance on VLM classification (left) and captioning tasks (right)**. Consistent with results in Figures 4 and 6, we see that smaller αs lead to improved deferral performance.

however, we assess factual alignment between generated and reference captions using the Gemini LLM (GeminiTeam et al., 2023). The model is prompted with: *"Are these captions semantically equivalent?"* and outputs "Yes" or "No." We compute the normalized log-likelihood of each response to obtain a factuality score $s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$, reflecting the model's confidence in semantic agreement. Full details are provided in Appendix C.4.

Uncertainty/Factuality Correlation. Since factuality scores are continuous, we adapt our metrics accordingly. Instead of binary accuracy-based separation, we compute the Pearson correlation $\rho\big(g_{\text{NENT}}(\mathbf{x}_i), s_{\text{Fac}}(\hat{\mathbf{y}}_i, \mathbf{y}_i)\big)$ between negative predictive entropy and factuality. We also generalize our deferral performance metric s_d to operate on factuality instead of accuracy.

Results. Figure 7 shows our main results. For classification tasks (left), trends match those observed in Sections 4.1 and 4.2: lower α improves deferral but reduces raw accuracy. For captioning tasks (right), GATEKEEPER increases the correlation between confidence and factuality, demonstrating effective deferral in generative settings. Prompting baselines from Section 4.2 could not be evaluated: PaliGemma failed to return valid outputs under prompt modifications, likely due to rigid pretraining (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. Despite achieving strong performance across tasks and architectures, several limitations remain: **(i)** We assume that only $\mathcal{M}_S$ is fine-tuned. Although this simplifies deployment and avoids retraining $\mathcal{M}_L$, it may overlook gains achievable through joint adaptation. **(ii)** In language modeling, GATEKEEPER may be overly strict: different token sequences can express the same meaning, and penalizing deviations based on exact tokens may suppress valid linguistic variation. Ideally, deferral decisions should reflect semantic correctness rather than surface-level mismatches. **(iii)** While we evaluate across multiple model families for classification, our experiments in the LLM and VLM settings focus on a single architecture per task, limiting insights into generalization. **(iv)** Our use of a generative model (Gemini) to score factuality introduces potential noise, as LLMs may produce inconsistent or inaccurate judgments.

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].

Azaria, A. and Mitchell, T. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*, 2023.

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, B., Zhu, M., Dolan-Gavitt, B., Shafique, M., and Garg, S. Model cascading for code: Reducing inference costs with model cascading for LLM based code generation, 2024a. URL https://arxiv.org/abs/2405.15842.

Chen, L., Zaharia, M., and Zou, J. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

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*, 2024b. URL https://openreview.net/forum?id=lZY9u0ijP7.

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.

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Ruhle, V., Lakshmanan, L. V., and Awadallah, A. H. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024.

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.

Geng, J., Cai, F., Wang, Y., Koeppl, H., Nakov, P., and Gurevych, I. A survey of confidence estimation and calibration in large language models. *arXiv preprint arXiv:2311.08298*, 2023.

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.

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., 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, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

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.

Liu, L., Pan, Y., Li, X., and Chen, G. Uncertainty estimation and quantification for llms: A simple supervised approach. *arXiv preprint arXiv:2404.15993*, 2024.

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.

Mahaut, M., Aina, L., Czarnowska, P., Hardalov, M., Müller, T., and Màrquez, L. Factual confidence of llms: On reliability and robustness of current estimators. *arXiv preprint arXiv:2406.13415*, 2024.

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.

Narasimhan, H., Jitkrittum, W., Rawat, A. S., Kim, S., Gupta, N., Menon, A. K., and Kumar, S. Faster cascades via speculative decoding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vo9t20wsmd.

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.

Nie, L., Ding, Z., Hu, E., Jermaine, C., and Chaudhuri, S. Online cascade learning for efficient inference over streams. *arXiv preprint arXiv:2402.04513*, 2024.

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.

Rawat, A. S., Zaheer, M., Menon, A. K., Ahmed, A., and Kumar, S. When in doubt, summon the titans: Efficient inference with large models, 2021. URL https://arxiv.org/abs/2110.10305.

Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., and Yurochkin, M. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 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.

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. Broader Impact

This work contributes to the responsible and efficient deployment of machine learning systems by improving the decision-making capabilities of smaller, local models in model cascade architectures. By introducing a loss function that calibrates model confidence with respect to correctness, our approach enhances both the performance and transparency of automated systems that must decide when to act autonomously and when to defer to a more capable model. This design can improve the accessibility and sustainability of machine learning applications by reducing reliance on large, energy-intensive models—particularly important in low-resource environments or edge computing.

At the same time, the ability to fine-tune smaller models to strategically abstain from uncertain predictions raises important considerations for fairness and accountability. In high-stakes applications such as healthcare or finance, improper tuning of the deferral threshold—or uncalibrated confidence estimates—could lead to the systematic denial of service or misallocation of computational resources. Care must be taken to ensure that such systems are thoroughly evaluated not only for average performance but also for differential performance across subgroups. Moreover, the use of large models as fallback decision-makers assumes their correctness, which may not always hold, especially in underrepresented domains. We therefore encourage developers and practitioners to accompany deployments of cascade-based systems with rigorous audits of fairness, reliability, and alignment with human values.

B. Additional Background

B.1. Related Work

B.1.1. LLM Routing

Ding et al. (2024) propose a hybrid LLM inference pipeline that routes each query either to a small on-device model or a larger high-quality model based on the query's predicted difficulty and a tunable quality threshold. This cost-aware router allows dynamically trading off accuracy for efficiency, enabling up to a 40% reduction in expensive model calls without degrading answer quality. Similarly, Shnitzer et al. (2023) present a method to select the best model from a pool of pre-trained LLMs for each input by learning a "router" on many benchmark tasks. Without requiring labeled examples from the new target task, their approach uses existing datasets to train input-based model selectors, which consistently outperform always using the single best LLM for all queries.

B.1.2. Model Cascade Learning

Nie et al. (2024) introduce an online cascade-learning framework where lightweight models are incrementally trained to imitate a powerful LLM's decisions on a data stream, deferring to the LLM only when necessary. They cast cascade construction as an imitation-learning problem with theoretical no-regret guarantees, achieving LLM-level accuracy while cutting inference cost by up to 90% and maintaining robustness to distribution shifts over time. Chen et al. (2023) outline strategies for reducing LLM usage cost and present *FrugalGPT*, a cascade approach that learns to route queries through combinations of smaller or larger LLMs to balance cost and performance. Their experiments show that an adaptive use of multiple models can match the accuracy of the strongest individual LLM (e.g., GPT-4) with up to 98% cost savings. It can also slightly exceed GPT-4's accuracy at equal cost, highlighting the benefit of cascades that allocate queries to the most appropriate model for each input.

B.1.3. Confidence Calibration in LLMs

Jitkrittum et al. (2023) analyze the classical strategy of confidence-based deferral in model cascades, wherein a model hands off to a stronger model if its confidence is below a threshold, to determine when this simple strategy succeeds or breaks down. They derive the optimal deferral policy in theory and show that naïve confidence thresholds perform well in general but can fail when later models are specialists (only reliable on certain inputs), when there is label noise, or under distribution shift – scenarios where more sophisticated deferral criteria yield better performance. Geng et al. (2023) provide a comprehensive survey of methods for confidence estimation and calibration in LLM outputs. They review recent techniques to quantify uncertainty in large language model predictions, discuss challenges unique to LLMs, and highlight advancements that improve alignment between a model's reported confidence and its actual accuracy across tasks. Azaria & Mitchell (2023) find evidence that an LLM's internal activations encode whether or not it is producing a truthful answer, even when the model's output is incorrect or fabricated. By training a classifier on the model's hidden state (without fine-tuning the LLM itself), they can often detect when the model is "lying" or unsure, suggesting that large models internally recognize their

mistakes or uncertainty despite outwardly confident responses. Similarly, Liu et al. (2024) propose a supervised approach to LLM uncertainty quantification that leverages labeled examples and the model's hidden representations to predict the correctness of its answers. They show that incorporating features from the model's internal layers yields significantly improved uncertainty estimates and calibration across diverse tasks, with these gains transferring robustly to new domains. Notably, their method is easy to implement and can be adapted to different levels of model access (black-box vs. white-box), making it widely applicable.

B.1.4. CONFIDENCE VERBALIZATION IN LLMS

Lin et al. (2022) demonstrate that GPT-3 can be fine-tuned to output a calibrated verbal confidence (e.g., "I'm 90% sure") along with each answer. This model's stated confidence levels align well with its true correctness likelihood and remain fairly well-calibrated even under distribution shift, marking the first instance of an LLM explicitly expressing useful uncertainty estimates in natural language. Xiong et al. (2024) thoroughly evaluate black-box methods for eliciting an LLM's self-reported confidence through prompting and answer sampling. They find that current LLMs tend to verbalize overly high confidence (mirroring human overconfidence), but that carefully designed prompts, consistency checks across multiple sampled answers, and improved aggregation strategies can mitigate this issue. Moreover, larger models generally show better calibration and an improved ability to predict their own failures, though room for further improvement remains in making their expressed uncertainty truly reliable. Mielke et al. (2022) examine whether a conversational agent's expressed certainty corresponds to its actual knowledge, showing that off-the-shelf dialogue models are poorly "linguistically calibrated." They demonstrate that a model's likelihood of giving a correct answer can be estimated via an auxiliary model and used as a control signal to adjust the agent's responses. The resulting dialogue agent exhibits far less overconfident language when it is likely to be wrong, improving transparency about uncertainty in its answers. Finally, Mahaut et al. (2024) assess the reliability of various methods to estimate an LLM's *factual confidence* – the probability that its answer is correct – under both in-domain and paraphrased inputs. Through a rigorous evaluation on QA and fact-checking tasks, they conclude that the most trustworthy confidence scores come from model-introspective approaches (e.g., a trained probe on hidden states), albeit at the cost of requiring full model access and training data. They also highlight that an LLM's confidence can be unstable under meaning-preserving input variations (paraphrases), underscoring the need for more robust and stable confidence estimation techniques for factual correctness.

B.2. Model Access Levels

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

B.3. Ideal Deferral Curve

We present the functional form of the *ideal deferral* curve, denoted $\mathrm{acc}_{\mathrm{ideal}}(r)$, 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 $p_s = \mathrm{acc}(\mathcal{M}_S)$, and $p_l = \mathrm{acc}(\mathcal{M}_L)$ 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:** $r \leq (1 - p_s)$.
  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.

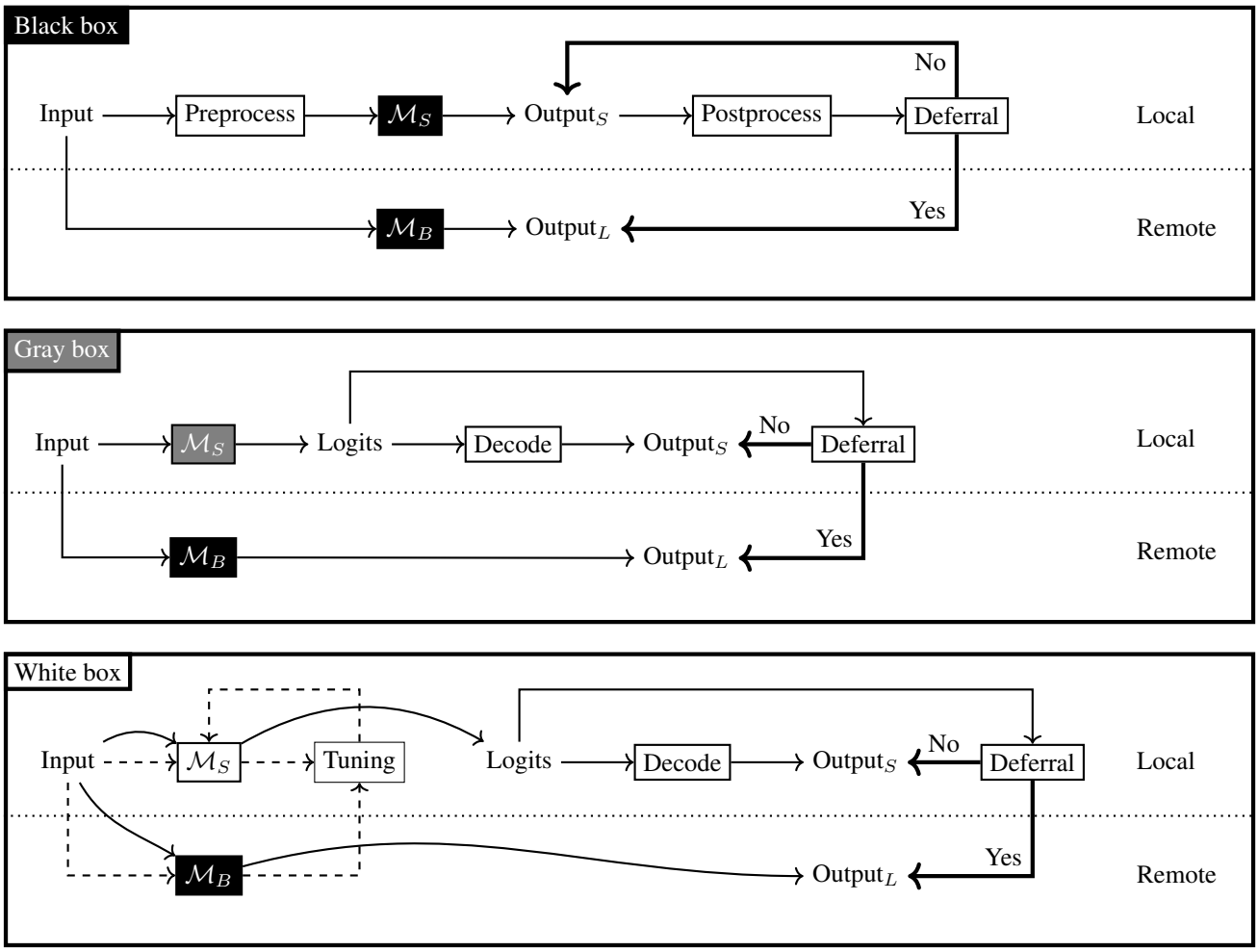

Figure 8. **An overview of different uncertainty quantification strategies depending on model access level**.

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

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

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.

C. Additional Experimental Details

C.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)
  )
)
```

C.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: $\mathbf{x} \leftarrow \mathbf{x} \,\|\, \mathbf{x}'$. 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.

C.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.

C.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.

C.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 $\alpha \to 0$.

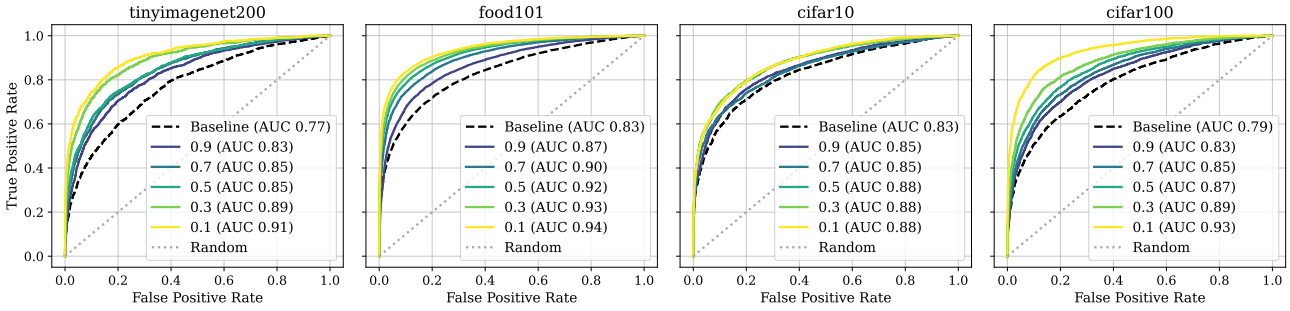

Figure 9. **ROC curves for image classification experiments**. Each figure shows the ROC curves for each of the datasets considered in Section 4.1. We observe that GATEKEEPER consistently increases separation of correct and incorrect confidence scores across varying α (colored curves) compared to the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

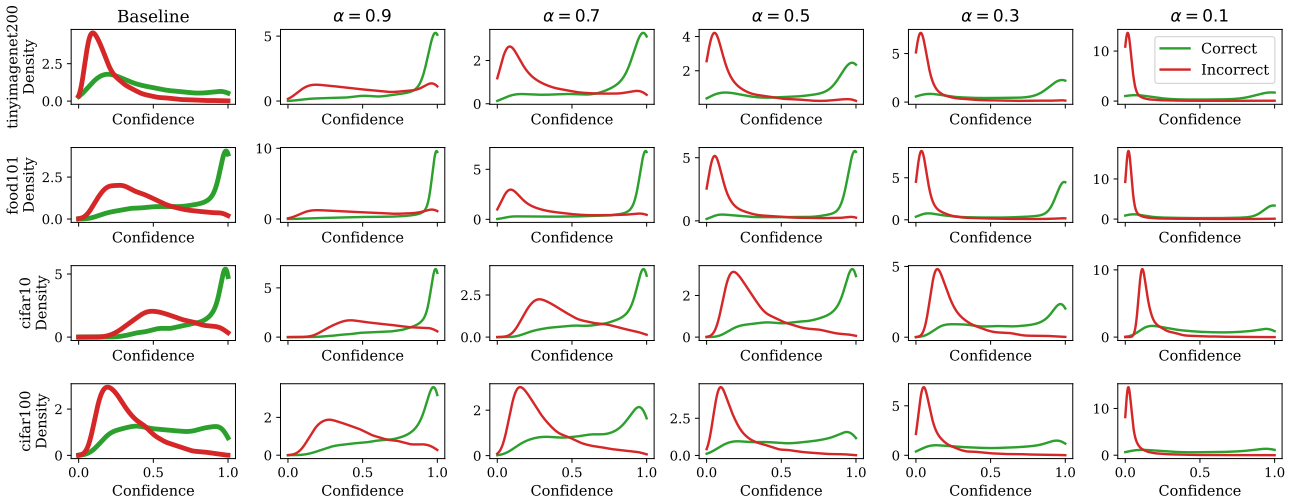

Figure 10. **Distributional overlap for image classification experiments**. Left-most column shows the results obtained using the untuned baseline, while the remaining columns correspond to the results obtained using GATEKEEPER with decreasing α values. Rows correspond to the datasets considered in Section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

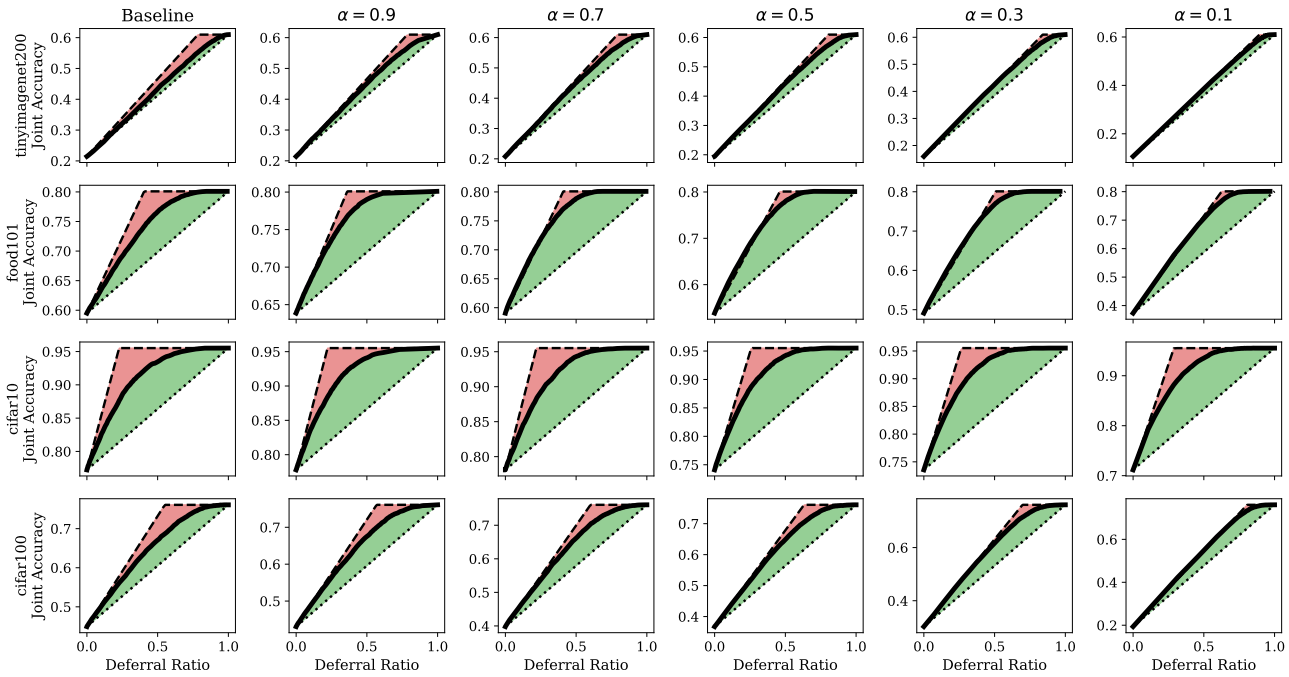

Figure 11. **Deferral curves for image classification experiments**. Left-most column shows the results obtained using the untuned baseline, while the remaining columns correspond to the results obtained using GATEKEEPER with decreasing α values. Rows correspond to the datasets considered in Section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

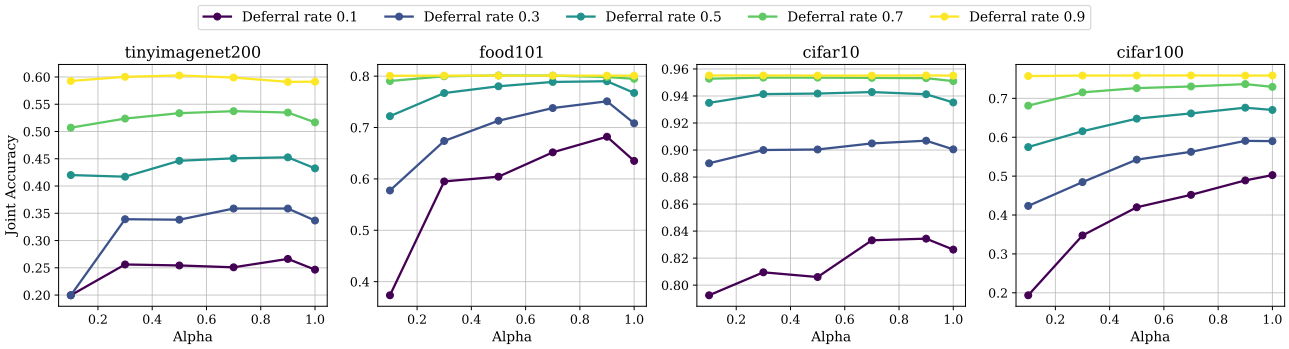

Figure 12. **Joint accuracy across different levels of α.** For varying fixed deferral ratios, we observe that the accuracy of $\mathcal{M}_S$ generally decreases as $\alpha \to 0$.