Improving Cascade Routing for Structured Attribute Generation with Heterogeneous Confidence

Fatemeh Mansoori [1] **Andrea Scarinci** [1] **Aditya Aggarwal** [1] **Suleiman A. Khan** [1] **Ashwin Chandramouli** [1]

Abstract

Multi-model inference systems—whether based on routing, cascading, or unified strategies—often rely on confidence signals to decide when a small language model (SLM) output should be accepted or deferred. While such signals are commonly used in classification and short-form generation, their reliability in structured generation settings remains poorly understood.

In this work, we study log-probability confidence in structured attribute value generation, where a model must produce either a schema-compliant VALUE or an ABSTAIN outcome. We show that confidence is *prediction-type-conditioned*: in our setting, average token log-probability is a stronger error-detection signal for VALUE outputs than for ABSTAIN outputs. As a result, global confidence thresholding yields imbalanced trade-offs, improving VALUE precision at the cost of recall while providing weaker control over abstention behavior.

We therefore cast cascade routing as *type-aware selective deferral*, in which acceptance decisions depend on both the confidence score and the predicted output type, with VALUE thresholds specialized by attribute family. Experiments on a large-scale product attribute generation task show that a fine-tuned SLM combined with selective deferral improves quality–cost trade-offs relative to pooled thresholding. The strongest operating point routes low-confidence VALUE predictions while keeping ABSTAIN predictions from the first-stage model, highlighting the importance of modeling heterogeneous reliability in structured-generation cascades.

[1]Amazon, USA. Correspondence to: Fatemeh Mansoori <fmsamira@amazon.com>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

Large language models (LLMs) are increasingly used for structured attribute value generation in large-scale product catalogs. These systems face a fundamental tension between quality and cost: larger models can provide complementary predictions on difficult or ambiguous cases, but are too expensive to apply uniformly across millions of product–attribute instances, while smaller models are much cheaper but still make errors. This tension motivates cascade architectures that use a smaller model by default and invoke a larger fallback model only on selected cases (Chen et al., 2023). In our setting the fallback is not assumed to be uniformly superior; the cascade's value comes from selective invocation under the right output regime and confidence conditions.

A cascade is only as good as its routing policy: escalating too aggressively loses the cost advantage, while accepting too many low-quality predictions degrades output quality. A natural approach is to use token-level confidence signals, such as average log probability, to decide when a smaller model's output can be trusted.

Confidence-based decision rules have been widely studied in selective prediction and reject-option learning (El-Yaniv & Wiener, 2010; Franc et al., 2023), where a model abstains when uncertainty is high. These approaches often treat confidence as a single scalar proxy for correctness across inputs.

In structured generation, however, this assumption does not fully hold. A model may produce either a schema value or an abstention outcome, and the same scalar score need not carry the same meaning across these cases. We find that average token log-probability separates correct from incorrect predictions more clearly when the model produces a candidate value than when it produces an abstention outcome. This observation is consistent with prior work showing that sequence likelihood and model probability can be imperfectly aligned with correctness in generation and question-answering settings (Kumar & Sarawagi, 2019; Jiang et al., 2021).

As a result, a single global confidence threshold is insufficiently expressive for this setting. It can remove

low-confidence VALUE predictions and thereby improve VALUE precision, but it does not affect VALUE and ABSTAIN predictions in the same way. In particular, because ABSTAIN is itself a valid model output rather than only a rejection decision, changing the threshold also changes the system's abstention behavior. This makes global thresholding difficult to tune when the desired operating point depends on both value quality and abstention quality.

We therefore frame routing as *type-aware selective deferral*, in which the system accepts or defers a lower-tier prediction based on both its confidence score and its predicted output type. This framing is related to learning-to-defer (Madras et al., 2018; Mozannar et al., 2023), while highlighting a structured-generation setting in which reliability depends on the nature of the predicted output.

Our contributions are: (i) we identify *prediction-type-conditioned reliability*—the score–correctness relationship differs between VALUE and ABSTAIN predictions; (ii) we show that global thresholding couples value quality, abstention behavior, and routing cost, treating the two output types differently; and (iii) we propose *type-aware selective deferral* with optional attribute-family-specific thresholds, and evaluate its quality–cost trade-offs against pooled thresholding and standalone baselines.

2. Related Work

Routing and cascading for LLMs. Multi-model inference balances quality and cost by assigning inputs to models of different capacity. Routing methods make a pre-generation model choice from query-level or benchmark-derived signals (Shnitzer et al., 2023; Sikeridis et al., 2024), while cascading methods decide post-generation whether to accept an output or invoke another model (Shankar et al., 2026; Chen et al., 2025; Nie et al., 2024); recent work combines both through learned quality estimators or fallback policies (Dekoninck et al., 2025; Moslem & Kelleher, 2026). We operate within the cascading paradigm, but focus on structured generation where the model may emit either a schema value or an abstention.

Log probabilities and calibration. Token-level log probabilities are a common, readily available confidence signal for post-generation cascade decisions, aggregated as per-token probability statistics (Shankar et al., 2026) or logit features such as perplexity (Dekoninck et al., 2025). Calibration studies show that aggregate confidence can hide differences across subgroups or output regimes (Guo et al., 2017; Nixon et al., 2019), and that sequence likelihood is often misaligned with semantic correctness in machine translation (Kumar & Sarawagi, 2019) and question answering (Jiang et al., 2021). Sequence-level uncertainty for autoregressive models (Malinin & Gales, 2021) has since been followed by many alternatives—self-evaluation and P(true) (Kadavath et al., 2022), contextualized likelihood (Ren et al., 2023; Lin et al., 2024), probability-margin scores (Farr et al., 2024), sampling-based consistency (Manakul et al., 2023), and ensembles thereof (Bouchard & Chauhan, 2025; Bouchard et al., 2026)—with recent benchmarks comparing them (Vashurin et al., 2025). We revisit average log-probability and show its usefulness depends on predicted output type.

Learned routing, selective prediction, and deferral. Prior work trains routers or policies to choose among models or defer uncertain cases to another expert, including benchmark-trained LLM routers (Shnitzer et al., 2023), RL-based model selection (Sikeridis et al., 2024), cloud–edge routing (Yu et al., 2025), imitation-learning cascade control (Nie et al., 2024), selective prediction and reject-option learning (El-Yaniv & Wiener, 2010; Franc et al., 2023), and learning-to-defer (Madras et al., 2018; Mozannar et al., 2023). Abstention has also been studied as a reliability mechanism in LLMs (Wen et al., 2025; Kirichenko et al., 2025). Our setting differs in that ABSTAIN is itself a task output rather than only a routing action, so deferral rules should condition on whether the first-stage model predicted VALUE or ABSTAIN.

3. Problem Setup

3.1. Structured Attribute Value Generation

We consider the task of structured attribute value generation. Each input x consists of a product context, such as title, description, and specifications, together with target attribute metadata. The model generates an output conditioned on both the product information and the attribute metadata.

The attribute metadata may include a natural language definition, allowed values for closed-set categorical attributes, and formatting or unit constraints for numeric attributes. The model must produce either a schema-compliant value or an abstention outcome, $y \in \mathcal{Y} \cup \{\text{abstain}\}$, where $\mathcal{Y}$ denotes the valid value space for the target attribute. Importantly, abstention is a valid task output, not merely a routing action: the correct response may be to abstain when no schema-compliant value should be produced from the available product context.

For evaluation and routing, we treat abstention as a single output regime covering both inapplicable attributes and values not inferable from the product information, without distinguishing finer-grained abstention reasons.

The task spans multiple attribute families, including boolean attributes, closed-set categorical attributes, numeric attributes with units, numeric attributes without units, and free-text attributes. We use a to denote the attribute fam-

ily of the target attribute. These families differ in schema constraints and difficulty, and we later use them to define family-specific routing policies.

To avoid ambiguity, we use **VALUE** to denote non-abstention predictions and **ABSTAIN** to denote abstention predictions. We use **correct** to denote agreement with the ground-truth label under the evaluation protocol.

3.2. Cascade Architecture

We study a two-tier cascade consisting of a small language model (SLM) and a larger fallback model. The SLM serves as the primary low-cost tier, while the fallback model is invoked only when the routing policy does not accept the SLM output. Let y denote the output generated by the SLM.

For each input, the routing policy decides whether to accept y as the final system output or defer to the fallback model. The central problem is therefore to design a routing policy that improves output quality while limiting the fraction of inputs sent to the fallback model.

4. Non-Uniform Confidence Across Output Regimes

We define the confidence of the small-model output using the average log probability of its generated tokens. Given an input x, let $y = (y_1, \ldots, y_T)$ denote the output generated by the SLM. We compute

$$s(y; x) = \frac{1}{T} \sum_{t=1}^{T} \log p(y_t \mid y_{<t}, x), \tag{1}$$

where T is the number of generated tokens, and higher $s(y; x)$ indicates a more confident output. We compute this average after removing fixed formatting tokens (e.g. markdown fences and schema wrappers) so that it reflects only the predicted VALUE or ABSTAIN content, applying the same cleaning rule to all models and policies.

Let $d(y) \in \{\text{VALUE}, \text{ABSTAIN}\}$ denote the predicted output type of y, and a the attribute family (Section 3). We analyze how the relationship between confidence and correctness varies when conditioned on $d(y)$ and a.

Our central observation is that the score–correctness relationship depends on the predicted output type. In particular, for VALUE predictions, higher confidence is reliably associated with correctness, whereas for ABSTAIN predictions the same score is a weaker and less consistent indicator.

A single global acceptance rule, accept if $s(y; x) > \tau$ with $\tau \in \mathbb{R}$, is therefore poorly matched to the task, motivating routing policies that condition on the output regime rather than treating all generated outputs as equally calibrated under one scalar threshold.

5. Type-Aware Selective Deferral

We cast cascade routing as a selective deferral problem in which acceptance depends on both the confidence score and the predicted output type. Rather than using a single global threshold, we allow thresholds to vary by predicted output type and attribute family. Because the deferral decision is made after generation, the router observes the generated output y, its confidence score $s(y; x)$, and the attribute family a. Let $\tau_{d,a} \in \mathbb{R}$ denote the threshold, on the average log-probability scale, for predictions of output type d and attribute family a. A type-aware policy is:

$$r(a, y) = \begin{cases} \text{accept}, & s(y; x) > \tau_{d(y),a}, \\ \text{defer}, & s(y; x) \leq \tau_{d(y),a}. \end{cases} \tag{2}$$

This rule defines the most general threshold family we consider. We evaluate four special cases: a single global threshold over all outputs, separate VALUE/ABSTAIN thresholds, a single VALUE threshold that keeps ABSTAIN predictions, and family-specific VALUE thresholds that keep ABSTAIN predictions. The latter two keep ABSTAIN predictions because their score separation is weaker and the fine-tuned SLM already produces high-precision ABSTAIN outputs. The inference procedure is summarized in Appendix A.7.

We focus on threshold policies because they are transparent, deployment-friendly, and require no learned router or calibrated reward model. Thresholds are selected on a validation split disjoint from the test set and then evaluated once on the held-out test set. Rather than exhaustively optimizing many thresholds on validation, we define a small set of deployment-motivated operating regimes based on fallback budget and routing semantics. The global policy uses a single validation-selected score threshold, while the family-specific policy assigns separate thresholds by attribute family using validation score quantiles under a target routing budget. This keeps the policies interpretable and avoids overfitting many threshold degrees of freedom to the validation split. Appendix A.6 varies the target routing budget and shows that the resulting VALUE precision is stable across nearby operating points.

6. Experimental Setup

6.1. Models

We evaluate a two-tier cascade consisting of a small language model (SLM) and a larger fallback model. The SLM is based on Qwen3-4B and is evaluated in two forms: a base SLM and a fine-tuned SLM.

The fine-tuned SLM is trained using LoRA with rank 64 on approximately 9M supervised examples covering both VALUE and ABSTAIN outputs. The training data spans over 30,000 product type–attribute–country combinations.

Training is performed for 1.5 epochs with a learning rate of 3×10^{-5}.

We use Qwen3-235B-A22B as the fallback model in cascade experiments, providing a larger same-family alternative prediction source for cases where the routing policy judges the fine-tuned SLM output unreliable.

Fine-tuning changes both the prediction quality and the confidence structure of the SLM. Compared with the base SLM, the fine-tuned SLM produces more accurate VALUE predictions and exhibits more useful confidence separation, making it a better candidate for confidence-based cascade routing.

6.2. Evaluation Protocol

We compare three standalone models: Base SLM, Fine-tuned SLM, and Qwen3-235B-A22B. We also evaluate cascade variants that combine the fine-tuned SLM with Qwen3-235B-A22B as a fallback. All standalone and cascade evaluations use the same product context and target-attribute metadata for a given example; model-specific formatting is limited to the required chat/template wrappers.

All results are reported on a held-out English evaluation set of approximately 408K examples spanning 4,580 product-type–attribute pairs. The evaluation set is derived from a large e-commerce product catalog. Ground-truth labels are human-annotated and audited, including the abstention labels, which mark cases where an annotator judged that no schema-compliant value applies. The data spans five attribute families, with distributional details matching the full catalog pool from which the train, validation, and test splits are drawn. To determine correctness, we first normalize model outputs and ground-truth labels and apply exact string matching. For VALUE predictions that do not exactly match the label, we apply a model-based semantic-equivalence check to account for cases where the generated value and reference label differ only in surface form. The checker is applied only to non-exact VALUE matches and is used uniformly for all models and cascade policies; details are provided in Appendix A.1. ABSTAIN predictions are evaluated against ground-truth abstention labels, grouping cases in which no schema-compliant value should be produced into a single ABSTAIN category.

We report VALUE precision and recall as the primary metrics, where VALUE denotes non-abstention predictions. We also report ABSTAIN precision and recall under the grouped abstention category described above. In the cascade setting, we additionally report routing rate, defined as the percentage of examples sent to the fallback model.

For the cascade experiments, thresholds are selected on a validation split disjoint from the held-out test set, and the held-out test set is used only for final reporting. The selected family-specific policy uses separate thresholds for VALUE predictions by attribute family and keeps ABSTAIN predictions from the fine-tuned SLM, at a target per-family routing budget of approximately 20% (T3 in Appendix A.6). We also compare against policies that threshold all outputs, including ABSTAIN predictions. The goal is not to optimize a single scalar metric, but to compare routing policies under reasonable precision–recall–cost trade-offs; Appendix A.6 reports nearby lower-budget settings.

Table 1. Effect of fine-tuning on VALUE and ABSTAIN prediction performance.

Model	VALUE P	VALUE R	ABSTAIN P	ABSTAIN R
Base SLM	0.6878	0.7487	0.5106	0.2993
Fine-tuned SLM	0.7913	0.9019	0.8828	0.3063

7. Results

7.1. Fine-Tuning Shifts the SLM Operating Point

Table 1 shows that fine-tuning substantially improves the SLM on VALUE predictions. VALUE precision increases from 0.6878 to 0.7913, and VALUE recall increases from 0.7487 to 0.9019. This indicates that fine-tuning improves both the correctness and the coverage of generated attribute values.

Fine-tuning also changes the abstention regime. ABSTAIN precision increases sharply from 0.5106 to 0.8828, meaning that abstention predictions made by the fine-tuned SLM are much more likely to be correct. ABSTAIN recall remains similar, increasing from 0.2993 to 0.3063. Thus, fine-tuning produces a strong first-stage model with high VALUE quality and high-precision abstentions, while leaving room for cascade routing to recover additional true abstention cases.

7.2. Prediction-Type-Conditioned Reliability

Figure 1 compares confidence score distributions for correct and incorrect predictions from the fine-tuned SLM. Confidence is more informative for VALUE predictions: correct VALUE outputs concentrate near high scores, while incorrect VALUE outputs extend further into lower-confidence regions. This makes average log-probability useful for identifying low-confidence VALUE outputs that should be routed. For ABSTAIN predictions, confidence is correctly ordered—abstention correctness falls from roughly 96% near a score of 0 to about 66% below -0.2—but correct and incorrect abstentions overlap heavily in the high-confidence region, with approximately 75% of correct and 47% of incorrect abstentions both falling in $(-0.1, 0]$. Because abstention precision is already high (Table 1), this overlap leaves little room for threshold-based filtering of ABSTAIN predictions, so the same confidence score provides a weaker basis for

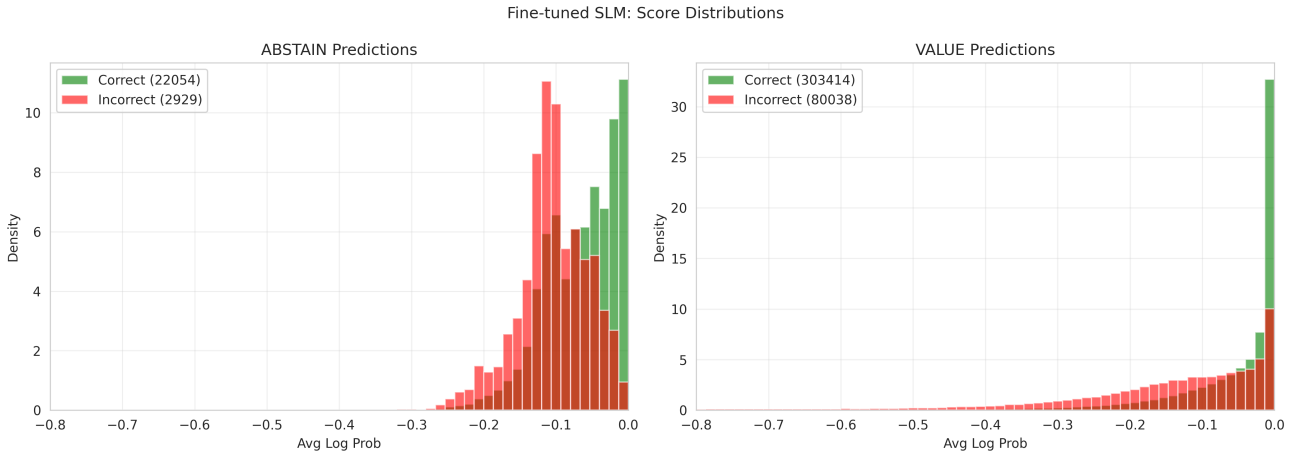

Figure 1. Score distributions of average log-probability for ABSTAIN and VALUE predictions from the fine-tuned SLM. Both panels share a common score axis and use density-normalized counts (per-panel y-axis). For VALUE predictions, incorrect outputs spread into a long low-confidence tail that is separable from the high-confidence correct mass; for ABSTAIN predictions, correct and incorrect outputs overlap more heavily near high confidence. The error-detection metrics in Table 3 quantify this asymmetry.

deferral than for VALUE and should not necessarily share the same threshold.

To quantify this separation, we compute error-detection AUROC and AUPRC by predicted output type, using $-s(y; x)$ as the score for incorrectness; full results are reported in Appendix A.3. VALUE predictions achieve AUROC/AUPRC of 0.7434/0.4362 against a prevalence baseline of 0.2087, while ABSTAIN predictions achieve 0.7154/0.2240 against a baseline of 0.1172. Both regimes show above-baseline error detection, but VALUE predictions have stronger absolute error-detection performance. This means that low-confidence VALUE outputs are more directly useful for selective deferral, while ABSTAIN predictions require separate treatment rather than sharing a single pooled threshold with VALUE predictions.

A second source of heterogeneity appears across attribute families. Restricting the same VALUE error-detection analysis to each family, average log-probability is a substantially stronger error signal for numeric attributes (AUROC 0.86 with units and 0.86 without units) than for text and categorical attributes (0.72 and 0.74), with boolean attributes in between (0.78). Numeric families also show the largest gains in AUPRC over the per-family error-prevalence baseline. Per-family error-detection metrics and the full by-family distributions are reported in Appendix A.5. These differences indicate that a single VALUE threshold does not carry the same meaning across families, and motivate routing policies that condition not only on score, but also on predicted output type and attribute family.

7.3. System-Level Effect of Thresholding

Applying a confidence threshold directly to the fine-tuned SLM illustrates why pooled thresholding is blunt. In this diagnostic, when a VALUE prediction falls below the threshold it is *rejected*: the system suppresses the generated value rather than accepting it. We use REJECTED for this confidence-based system action to distinguish it from a genuine ABSTAIN judgment, in which the model itself decides no value should be produced. As the threshold becomes stricter, more VALUE predictions are rejected, increasing VALUE precision but reducing VALUE recall and coverage. This operation also changes the system-level abstention pool: measured ABSTAIN recall rises, but ABSTAIN precision falls because some correct VALUE predictions are now rejected. This diagnostic, shown in Appendix A.4, motivates routing policies that treat VALUE and ABSTAIN predictions differently.

7.4. Routing Shifts the Quality–Cost Trade-off

Table 2 compares standalone baselines with four cascade routing policies. The standalone models represent different operating points rather than a strict quality ranking: the fine-tuned SLM achieves higher VALUE precision and recall, while Qwen3-235B-A22B is more conservative, with higher ABSTAIN recall but lower VALUE recall. This makes the fallback useful for selected cases rather than as an unconditional replacement.

Relative cost is normalized to the fine-tuned SLM and estimated under a fixed 1000-input/20-output-token request size using representative Bedrock serving prices. These estimates compare operating points under common assumptions; realized costs vary with serving configuration,

Table 2. Cascade routing results at selected operating points. Routing rate is the percentage of examples sent from the fine-tuned SLM to Qwen3-235B-A22B. Relative cost is normalized to the fine-tuned SLM.

Policy	VALUE P	VALUE R	ABSTAIN P	ABSTAIN R	Route %	Rel. Cost
Qwen3-235B-A22B only	0.7553	0.6772	0.3880	0.5755	100.0	19.8×
Fine-tuned SLM only	0.7913	0.9019	0.8828	0.3063	0.0	1.0×
Global threshold, all outputs	0.8123	0.8480	0.5977	0.4752	28.4	6.3×
Type-aware threshold, all outputs	0.8164	0.8466	0.6012	0.4973	26.5	6.0×
Global VALUE threshold, keep ABSTAIN	0.8165	0.8465	0.6011	0.4977	26.4	6.0×
Family-specific VALUE threshold, keep ABSTAIN	**0.8146**	**0.8623**	**0.6454**	**0.4687**	**19.3**	**4.6×**

throughput, token counts, region, discounts, and pricing date.

All cascade policies improve VALUE precision over the standalone fine-tuned SLM while routing only a subset of examples. The type-aware policy assigns separate validation-selected thresholds to VALUE and ABSTAIN outputs; the selected ABSTAIN threshold routes under 1% of abstentions, so it nearly coincides with the global VALUE policy (both at $\approx 26\%$). This empirically confirms that the fine-tuned SLM's high-precision abstentions should be trusted rather than routed. The strongest cost-efficient operating point applies family-specific thresholds to VALUE predictions while keeping ABSTAIN predictions from the fine-tuned SLM. Thus, the VALUE/ABSTAIN distinction matters not because both regimes should be routed aggressively, but because the routing policy should know when not to route. This policy achieves VALUE precision of 0.8146 and VALUE recall of 0.8623 while routing only 19.3% of examples, for an estimated relative cost of $4.6\times$—nearly matching the precision of the global VALUE threshold (26.4% routing) at substantially lower cost.

8. Discussion

Our results show that confidence-guided routing is useful for structured attribute generation, but average token log-probability should not be treated as a single global signal. It is most useful for VALUE predictions, where low-confidence outputs are enriched for errors. ABSTAIN predictions show weaker separation, so applying the same threshold to both regimes can route more examples than necessary while losing VALUE recall.

The strongest operating point routes low-confidence VALUE predictions while keeping ABSTAIN predictions from the fine-tuned SLM. Its low ABSTAIN recall should be read as a first-stage operating point, not a failure of fine-tuning: abstentions are high precision, while many ambiguous VALUE predictions remain available for routing. Thus, routing complements fine-tuning rather than replacing it with a uniformly more abstention-heavy model.

The fallback model should be viewed as complementary rather than as an oracle. Qwen3-235B-A22B improves ABSTAIN recall but has lower VALUE recall than the fine-tuned SLM, so invoking it uniformly would discard many correct VALUE predictions; this also limits the VALUE-precision gains available from routing more examples. In production, later cascade stages could incorporate richer evidence, such as product images or retrieved brand-site content for attributes that are not inferable from text alone.

The thresholding results should be read as operating-point comparisons rather than fully optimized routing policies; learned routers or stronger uncertainty estimators may improve performance at the cost of additional training and calibration. The exact thresholds and gains are task- and model-specific, but the broader lesson is that confidence should be interpreted at the granularity of routing decisions.

9. Conclusion

We studied confidence-guided cascade routing for structured attribute generation. Average token log-probability is useful but heterogeneous: its relationship to correctness differs by predicted output type and attribute family, making a single global threshold limited. Type-aware selective deferral improves the quality–cost trade-off by routing low-confidence VALUE predictions while preserving high-precision ABSTAIN predictions from the first-stage model. Future work should test this behavior across additional languages, model pairs, structured-generation tasks, and finer-grained abstention taxonomies.

Impact Statement

This work aims to improve the reliability and efficiency of large-scale structured generation systems used to populate product knowledge. Efficient cascade routing can reduce inference cost and ease deployment at scale. Incorrect attributes can affect downstream search, recommendation, and customer-facing product information, so such systems should include monitoring, label-quality checks, and safeguards for high-impact attributes. The proposed method reduces some errors but does not eliminate systematic biases or mistakes in the models, data, or evaluation pipeline.

References

Bouchard, D. and Chauhan, M. S. Uncertainty quantification for language models: A suite of black-box, white-box, LLM judge, and ensemble scorers. *arXiv preprint arXiv:2504.19254*, 2025.

Bouchard, D., Chauhan, M. S., Bajaj, V., and Skarbrevik, D. Fine-grained uncertainty quantification for long-form language model outputs: A comparative study. *arXiv preprint arXiv:2602.17431*, 2026.

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., Lu, X., Li, J., Chen, P., Li, Z., Sun, K., Luo, Y., Mao, Q., Li, M., Xiao, L., Yang, D., Huang, X., Ban, Y., Sun, H., and Yu, P. S. Harnessing multiple large language models: A survey on LLM ensemble. *arXiv preprint arXiv:2502.18036*, 2025.

Dekoninck, J., Baader, M., and Vechev, M. A unified approach to routing and cascading for LLMs. In *International Conference on Machine Learning*, 2025.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Farr, D., Cruickshank, I., Manzonelli, N., Clark, N., Starbird, K., and West, J. LLM confidence evaluation measures in zero-shot CSS classification. *arXiv preprint arXiv:2410.13047*, 2024.

Franc, V., Prusa, D., and Voracek, V. Optimal strategies for reject option classifiers. *Journal of Machine Learning Research*, 24:1–49, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 2021.

Kadavath, S., Conerly, T., Askell, A., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kirichenko, P., Ibrahim, M., Chaudhuri, K., and Bell, S. J. AbstentionBench: Reasoning LLMs fail on unanswerable questions. In *Advances in Neural Information Processing Systems*, 2025.

Kumar, A. and Sarawagi, S. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*, 2019.

Lin, Z., Trivedi, S., and Sun, J. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10351–10368, 2024.

Madras, D., Pitassi, T., and Zemel, R. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, 2018.

Malinin, A. and Gales, M. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations (ICLR)*, 2021.

Manakul, P., Liusie, A., and Gales, M. J. F. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Moslem, Y. and Kelleher, J. D. Dynamic model routing and cascading for efficient LLM inference: A survey. *arXiv preprint arXiv:2603.04445*, 2026.

Mozannar, H., Lang, H., Wei, D., Sattigeri, P., Das, S., and Sontag, D. Who should predict? exact algorithms for learning to defer to humans. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Nie, L., Ding, Z., Hu, E., Jermaine, C., and Chaudhuri, S. Online cascade learning for efficient inference over streams. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Nixon, J., Dusenberry, M., Jerfel, G., Nguyen, T., Liu, J., Zhang, L., and Tran, D. Measuring calibration in deep learning. In *CVPR Workshops*, 2019.

Ren, J. et al. Self-evaluation improves selective generation in large language models. *arXiv preprint arXiv:2312.09300*, 2023.

Shankar, S., Zeighami, S., and Parameswaran, A. G. Task cascades for efficient unstructured data processing. *Proceedings of the ACM on Management of Data*, 4(1):1–21, 2026. doi: 10.1145/3786702.

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.

Sikeridis, D., Ramdass, D., and Pareek, P. PickLLM: Context-aware RL-assisted large language model routing. *arXiv preprint arXiv:2412.12170*, 2024.

Vashurin, R., Fadeeva, E., Vazhentsev, A., Rvanova, L., Tsvigun, A., Vasilev, D., Xing, R., Sadallah, A. B., Grishchenkov, K., Petrakov, S., Panchenko, A., Baldwin, T., Nakov, P., Panov, M., and Shelmanov, A. Benchmarking uncertainty quantification methods for large language models with LM-Polygraph. *Transactions of the Association for Computational Linguistics (TACL)*, 13:220–248, 2025. doi: 10.1162/tacl_a_00737.

Wen, B., Yao, J., Feng, S., Xu, C., Tsvetkov, Y., Howe, B., and Wang, L. L. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556, 2025.

Yu, S., Goudarzi, M., and Toosi, A. N. Efficient routing of inference requests across LLM instances in cloud-edge computing. *arXiv preprint arXiv:2507.15553*, 2025.

A. Appendix

A.1. Evaluation Details

All models and cascade variants are evaluated on the same cleaned held-out English set of approximately 408K examples spanning 4,580 product-type–attribute pairs. The same filtering, label-cleaning, output-normalization, and correctness procedure is applied to every standalone model and cascade policy before computing metrics.

For VALUE predictions, we first compare the normalized prediction against the normalized ground-truth label. Normalization removes superficial formatting differences such as casing, leading or trailing whitespace, and minor punctuation artifacts. If the normalized strings do not match, we apply a model-based semantic-equivalence check to account for cases where the generated value and reference label differ only in surface form, including synonyms, reordered phrases, unit-compatible surface forms, and equivalent numeric formats. Using a fixed judge model and prompt, the judge is given the target attribute, the normalized prediction, and the normalized reference label, and determines whether the two values are equivalent for that attribute. A VALUE prediction is counted as correct if either the normalized exact match or the semantic-equivalence check succeeds. Predictions that fail both checks are counted as incorrect. The same judge model and prompt are used for all standalone and cascade systems, so any judge noise applies uniformly across comparisons; we did not separately validate the judge's agreement with human equivalence judgments, which may add some noise to absolute VALUE-correctness levels.

For ABSTAIN predictions, we compare against the grouped ground-truth abstention class. As in the main paper, ABSTAIN covers cases in which no schema-compliant value should be produced from the available product context. VALUE and ABSTAIN precision and recall are then computed over these two grouped output regimes using the same correctness labels for all standalone and cascade systems. The semantic-equivalence checker is applied only to non-exact VALUE predictions and is not used for ABSTAIN predictions or for routing decisions.

A.2. Confidence Behavior Before and After Fine-Tuning

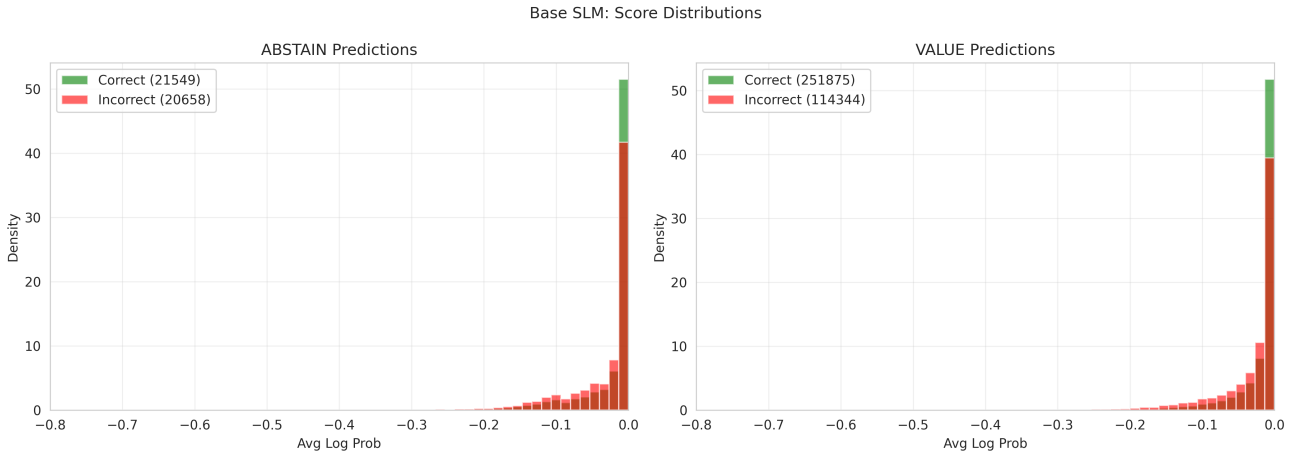

Figure 2. Score distributions for ABSTAIN and VALUE predictions from the base SLM. Compared with the fine-tuned SLM shown in the main paper, the base model exhibits weaker separation between correct and incorrect outputs. This makes its confidence scores less useful for selective routing.

Figure 2 shows the confidence distribution of the base SLM. Compared with the fine-tuned SLM in Figure 1, the base model shows weaker separation between correct and incorrect VALUE predictions. Incorrect VALUE predictions appear across a wider range of confidence scores, making them harder to isolate with a simple threshold.

After fine-tuning, correct VALUE predictions become more concentrated in the high-confidence region, while incorrect VALUE predictions are more likely to appear in the lower-confidence tail. This supports the observation that fine-tuning improves not only prediction quality, but also the usefulness of confidence as a routing signal. ABSTAIN predictions remain less cleanly separated than VALUE predictions, which motivates output-type-aware routing rather than a single global threshold.

A.3. Error-Detection Metrics

To quantify the usefulness of average log-probability as an error-detection signal, we compute AUROC and AUPRC separately by predicted output type. The positive class is an incorrect prediction and the ranking score is $-s(y; x)$, so higher scores correspond to lower confidence and greater likelihood of error. Baseline AUPRC is the error prevalence within each output type. These metrics characterize the confidence signal as an error detector; the downstream routing consequence of this asymmetry is shown in Table 2, where family-specific VALUE thresholding reduces routing rate from 26.4% to 19.3% relative to global VALUE thresholding while preserving nearly the same VALUE precision.

Table 3. Error-detection performance of average log-probability by predicted output type. The positive class is an incorrect prediction and the ranking score is $-s(y; x)$. Baseline AUPRC equals the error prevalence within each output type.

Model / Type	Error Rate	AUROC	AUPRC	Baseline AUPRC
Base SLM VALUE	0.3122	0.6263	0.4171	0.3122
Base SLM ABSTAIN	0.4894	0.6112	0.5683	0.4894
Fine-tuned SLM VALUE	0.2087	0.7434	0.4362	0.2087
Fine-tuned SLM ABSTAIN	0.1172	0.7154	0.2240	0.1172

The fine-tuned SLM improves error detection in both regimes, especially for VALUE predictions, where low-confidence outputs are more useful for selective deferral.

A.4. Global Threshold Diagnostic

Figure 3 shows the diagnostic thresholding experiment discussed in the main text.

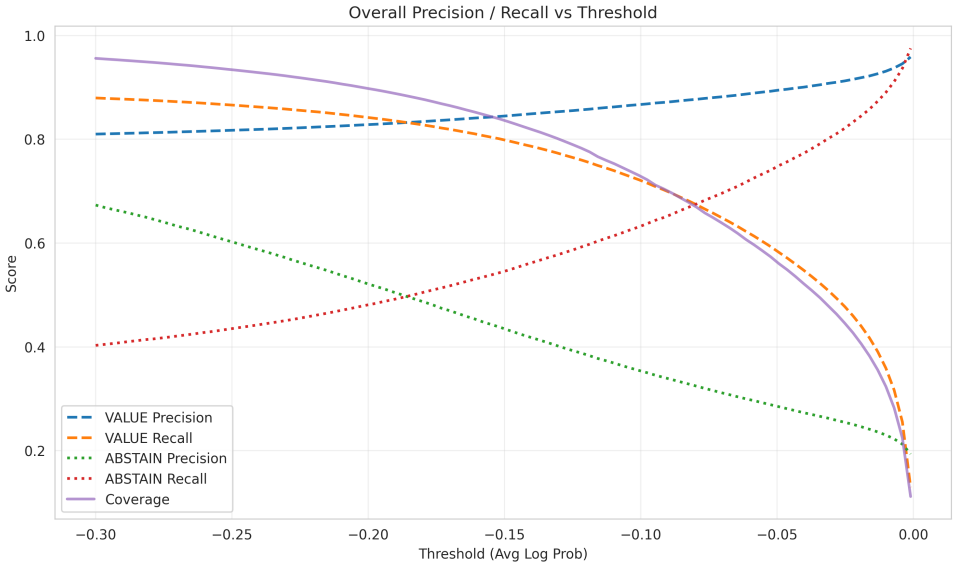

Figure 3. System-level precision and recall as a function of applying a confidence threshold directly to the fine-tuned SLM. In this diagnostic analysis, VALUE predictions below the threshold are rejected rather than routed to a fallback model. As the threshold becomes stricter, VALUE precision improves and coverage decreases, while grouped ABSTAIN metrics change because rejected VALUE predictions enter the system-level abstention pool.

A.5. Attribute-Family Heterogeneity

Figure 4 shows that confidence behavior varies across attribute families. Numeric and text attributes have wider confidence ranges, giving thresholding more room to separate reliable and unreliable predictions. Boolean and categorical attributes are more concentrated near high confidence, so aggressive thresholding can remove many correct predictions without a proportional gain in precision. This supports the attribute-family component of the routing policy: the same average log-probability threshold does not have the same meaning across all attribute families.

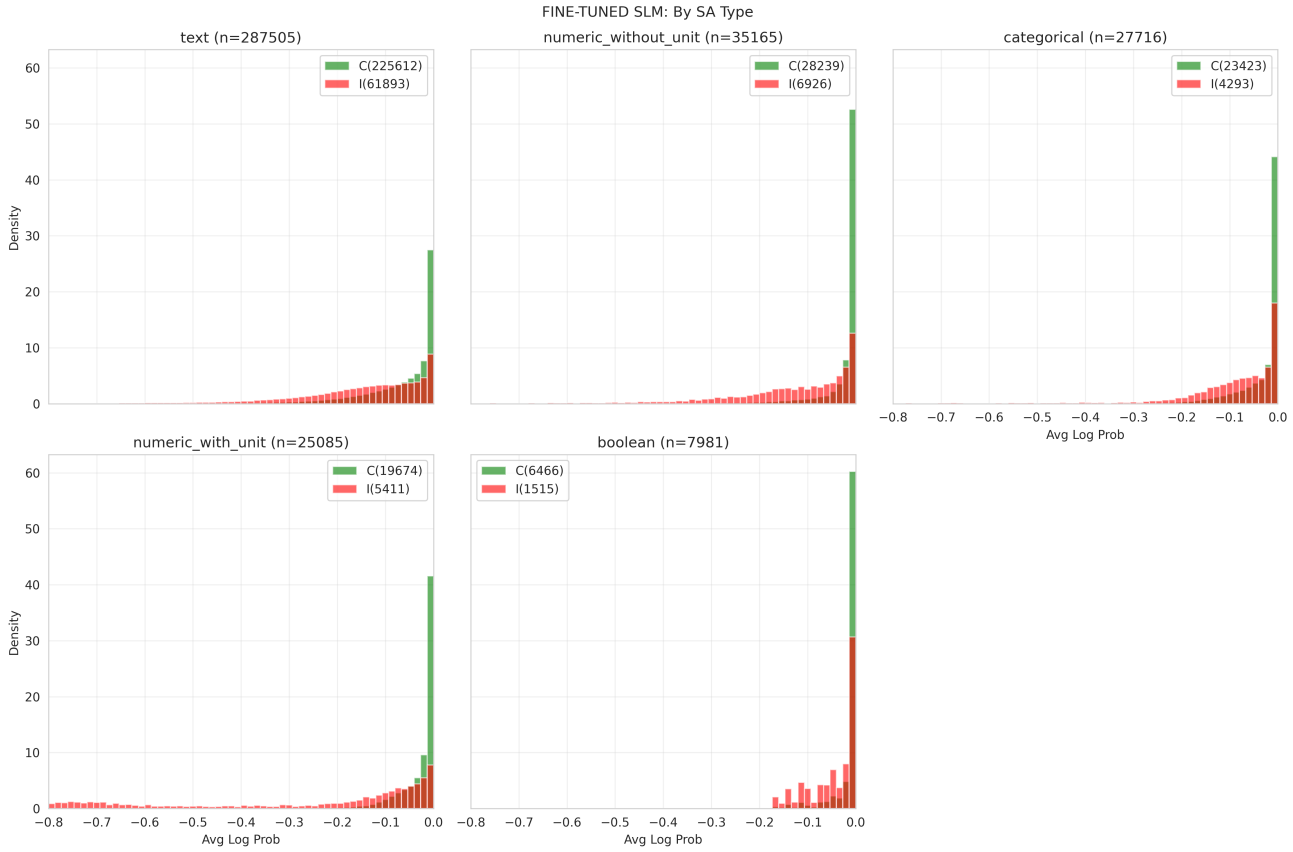

Figure 4. Confidence behavior across attribute families for the fine-tuned SLM. The usefulness of confidence varies by attribute family. Text and numeric attributes show broader confidence ranges and more visible low-confidence regions, while boolean and categorical attributes are more concentrated near high-confidence scores. This heterogeneity motivates thresholding policies that condition on attribute family.

Table 4 quantifies this by restricting the VALUE error-detection analysis (same protocol as Appendix A.3) to each attribute family of the fine-tuned SLM. Average log-probability is a markedly stronger error signal for numeric families (AUROC 0.86 both with and without units) than for text and categorical attributes (0.72 and 0.74), with boolean in between (0.78). The numeric families also show the largest AUPRC gains over their per-family error-prevalence baselines, confirming that confidence-based deferral is most reliable for numeric VALUE predictions.

This heterogeneity also explains why a single global threshold is poorly suited to the task: because the families occupy different score ranges, a common threshold routes very different fractions of each family. Table 5 illustrates this, reporting the fraction of each family's VALUE predictions that a single global threshold would route. At a common threshold of -0.05, only 14% of boolean predictions are routed but 49% of text predictions, motivating per-family VALUE thresholds that route a comparable fraction from each family.

Table 4. Per-family VALUE error-detection performance of average log-probability for the fine-tuned SLM. The positive class is an incorrect prediction and the ranking score is $-s(y; x)$. Baseline AUPRC equals the error prevalence within each family.

Attribute family	Error Rate	AUROC	AUPRC	Baseline AUPRC
Boolean	0.1898	0.7831	0.4389	0.1898
Categorical	0.1549	0.7392	0.3513	0.1549
Numeric (w/ unit)	0.2157	0.8645	0.7412	0.2157
Numeric (w/o unit)	0.1970	0.8572	0.6203	0.1970
Text	0.2153	0.7190	0.3962	0.2153

Table 5. Fraction (%) of each attribute family's VALUE predictions routed by a single common global threshold on the average-log-probability score. A common threshold routes very different fractions across families, motivating per-family thresholds.

Family	-0.03	-0.05	-0.07	-0.10	-0.15
Boolean	19.1	14.5	10.3	5.9	1.3
Categorical	35.9	27.8	21.3	13.9	6.5
Numeric (w/ unit)	41.1	32.1	25.3	18.0	12.4
Numeric (w/o unit)	28.8	22.8	19.1	14.9	9.6
Text	57.5	48.3	40.5	31.2	19.8

A.6. Threshold Sensitivity

The main results use one selected cost-efficient operating point. To check that the conclusion is not tied to a single threshold choice, we evaluate nearby configurations for the main policy: family-specific VALUE thresholding while keeping ABSTAIN predictions from the fine-tuned SLM.

Thresholds are applied only to VALUE predictions; ABSTAIN predictions are kept directly. We vary the target per-family routing budget, setting each family's VALUE threshold to the corresponding validation score quantile (Table 7). Across budgets, family-specific routing improves VALUE precision over the standalone fine-tuned SLM while routing far fewer than all examples.

Table 6. Threshold sensitivity for family-specific VALUE thresholding with SLM ABSTAIN predictions kept, at three target routing budgets. Routing rate is the percentage of examples sent to Qwen3-235B-A22B.

Config.	VALUE P	VALUE R	ABSTAIN P	ABSTAIN R	Route%
T1	0.8089	0.8808	0.7168	0.4193	10.9
T2	0.8128	0.8709	0.6749	0.4497	15.5
T3	0.8146	0.8623	0.6454	0.4687	19.3

The three configurations represent operating points along the same routing trade-off at increasing routing budgets. Tighter budgets (T1) route fewer examples and retain higher VALUE recall, while looser budgets (T3) raise VALUE precision. T3 is the operating point used in the main results because it gives the highest VALUE precision among these validation-budget settings; across all three, VALUE precision stays within a narrow band, indicating that the conclusion is not tied to a single budget.

Table 7. Attribute-family-specific VALUE thresholds used for the sensitivity configurations. These thresholds apply only to VALUE predictions; ABSTAIN predictions are kept directly for the policy reported in Table 6.

Config.	Boolean	Categorical	Num. w/ unit	Num. w/o unit	Text
T1	-0.050	-0.104	-0.484	-0.117	-0.207
T2	-0.032	-0.080	-0.211	-0.077	-0.169
T3	-0.022	-0.064	-0.133	-0.054	-0.145

A.7. Routing Algorithm

Algorithm 1 expands the post-generation acceptance rule used by the threshold policies in the main text.

Algorithm 1 Type-Aware Selective Deferral

Input: product–attribute instance x, attribute family a, thresholds $\{\tau_{d,a}\}$
Generate SLM output y and score $s(y; x)$
Determine predicted output type $d(y)$
Select threshold $\tau_{d(y),a}$
if $s(y; x) \leq \tau_{d(y),a}$ **then**
  Defer to fallback model
else
  Accept y
end if