When to Defer: Scope-Aware Cascades for Structured Attribute Generation in E-commerce Product Catalogs

Fatemeh Mansoori **Andrea Scarinci** **Aditya Aggarwal**
Ashwin Chandramouli **Suleiman A. Khan**
Amazon
fmsamira@amazon.com

Abstract

Fine-tuned small language models (SLMs) are attractive defaults for large-scale structured attribute generation—producing values such as materials, colors, or dimensions from product descriptions—as they are inexpensive to serve. However, a fine-tuned SLM is not reliable everywhere as its correctness varies sharply, so deploying it still requires deciding, *per output*, whether to keep the SLM's value or escalate it to a more capable fallback. The standard policy applies a single confidence threshold, but one threshold cannot separate scores on reliable tasks from those on unreliable ones, so it reduces coverage at no gain in quality. We demonstrate that in a fixed *deployment scope*, where the target tasks recur and validation data is available, a lightweight *scope-aware* router using only SLM-side features (token confidence plus task-scope identity) selects risky predictions far better. Our experiments show that at a fixed precision target it accepts about 11 points more values than a global confidence threshold (family-level calibration helps only marginally). Ablation experiments confirm the gain comes from task-scope reliability priors rather than a better confidence transformation. The router is an 8-feature gradient-boosted tree that adds negligible inference cost, making a strong fine-tuned SLM the low-cost tier of a cascade that optimizes the quality–cost tradeoff in real production settings.

1 Introduction

Modern large-scale product catalogs are populated and maintained by language models that generate structured attribute values—a material, a color, a numeric dimension—from product descriptions, at a scale of millions of product-attribute requests (Nikolakopoulos et al., 2023; Zhang et al., 2025; Satyadharma et al., 2025; Cheng et al., 2026). To control cost, these systems are commonly built as *cost-tiered cascades* (Chen et al., 2023), where a small low-cost model handles easy traffic and only the hardest cases escalate to a more capable, more expensive model.

Two levers move the quality–cost frontier of such a cascade. The first is *fine-tuning* the small tier: this does much of the heavy lifting, raising the small model's accuracy enough to handle a large share of traffic at the lowest cost. The second, which this paper studies, is the *escalation decision*: even a strong fine-tuned small model (hereafter SLM) is more reliable on some tasks than others, so the cascade must still decide, per output, whether to keep its prediction or escalate to a more capable model. We fine-tune the small tier and focus our contribution on the routing layer that decides when to defer to the more capable model.

The standard escalation rule applies a confidence threshold: accept high-confidence outputs, escalate low-confidence ones (El-Yaniv and Wiener, 2010; Franc et al., 2023), commonly using the SLM's own token log-probability as the readily available confidence signal. This confidence is useful but incomplete, because it reflects only the *output* and not the *task* it belongs to. We exploit a property of the production setting: attribute value generation runs over a fixed *deployment scope* in which the target product-type–attribute tasks recur and are known in advance, with validation data available for them. A lightweight router can use this by combining task identity with instance-level confidence.

We study this escalation layer on a fine-tuned Qwen3-4B SLM with larger Qwen models as well as a cross-family model as escalation targets, on a large English product-attribute benchmark, and make three contributions:

1. **Scope-aware selection.** A lightweight router using the SLM's confidence together with task-scope features selects risky predictions better than both global confidence and family-calibrated confidence (isotonic calibration per attribute family), at fixed operating points (Sec. 5).

2. **Mechanism.** A feature ablation shows a learned scorer using confidence alone recovers the raw threshold, and the largest gain comes from adding task-scope features, *i.e. which recurring task an output belongs to* (Sec. 6).

3. **Action decomposition.** We find that the cascade's *escalate* action and a *suppress* alternative reach different operating points: suppression raises the precision of emitted values at no fallback cost but lowers recall, while escalation preserves more recall and improves abstention balance at fallback cost. Different fallbacks expose different recovery profiles (Sec. 7).

We treat the router as a component of a cascade, in the established family of simple learned routers (Ong et al., 2025; Ding et al., 2024), rather than as a contribution in model design: our contribution is the empirical finding above, together with a deployable recipe for using it.

2 Problem Setup

Structured attribute value generation. Each input x consists of a product context (title, description, specifications) together with target-attribute metadata, which may include a natural-language definition, allowed values for closed-set categorical attributes, and formatting or unit constraints for numeric attributes. For example, given the product title "Men's Running Shoe – Black/White, Size 10" and the target attribute `color`, the model should produce *Black/White*; for the attribute `material`, it should abstain, since the material is not stated. The model produces an output $y \in \mathcal{Y} \cup \{\text{abstain}\}$, where $\mathcal{Y}$ is the valid value space for the target attribute. Abstention is a valid task output: the correct response when no schema-compliant value can be produced, covering both inapplicable attributes and values not inferable from the product information. To avoid ambiguity we write *VALUE* for non-abstention predictions and *ABSTAIN* for abstention predictions, and call an output *correct* when it agrees with the ground-truth label under the evaluation protocol (Sec. 4). The task spans five attribute families: boolean, closed-set categorical, numeric with units, numeric without units, and free text, which differ in schema constraints and difficulty.

Cascade architecture. We study a two-tier cascade (Figure 1) consisting of a fine-tuned SLM and a larger fallback model. For each VALUE prediction the router decides whether to *keep* it (emit y as-is) or *escalate* it to the fallback (emit the fallback's output); the fallback is invoked only on escalated predictions. SLM ABSTAIN predictions are passed through unchanged (Sec. 4 shows that routing them is net-neutral), so the router operates on VALUE predictions only. Escalation is the cascade's default action on a low-confidence VALUE; in Sec. 7 we additionally study *suppression*, which emits an abstention with no fallback call, as an alternative action on the same selected predictions.

Deployment scope. We study a recurring production setting in which the target set of product-type–attribute *tasks* is *known in advance* and labeled validation data is available. We use *scope* in a concrete, operational sense: the recurring product-type–attribute task a prediction belongs to—not a measure of distribution shift or input novelty. Attribute value generation is an ongoing process over a stable, slowly-changing set of attributes, so the same tasks that appear at training time recur at serving time on new products. Accordingly, our train, validation, and test splits cover the same task universe but disjoint product instances. The router's task-scope features are therefore *task-scope reliability priors*: where confidence asks how sure the model is about a particular output, task identity captures how reliable it has been on that kind of task. We make no claim of cold-start generalization to tasks unseen during fine-tuning.

Objective and metrics. The central problem is to design a routing policy that improves output quality while limiting the fraction of inputs sent to the fallback. We report VALUE and ABSTAIN precision, recall, and F1 as the primary metrics, since the policy actions (suppress, escalate) affect value emission and abstention behavior differently. We define *coverage* as the fraction of VALUE predictions the router keeps (emits without escalation or suppression). As the deployment-facing control knob we expose a *target VALUE precision*: a selection operating point calibrated on validation data, from which coverage and routing rate follow.

3 Method

Confidence score. We use the average token log-probability of the SLM output, $s(y) = \frac{1}{T}\sum_t \log p(y_t \mid y_{<t}, x)$, computed over all generated tokens. Higher s indicates greater confidence.

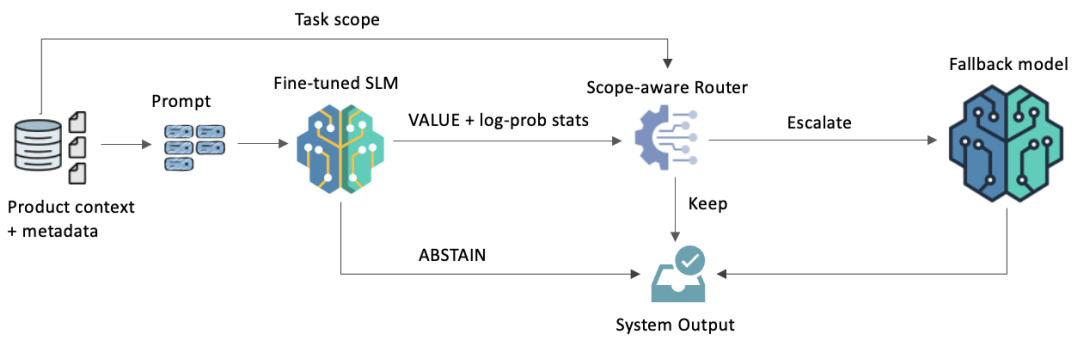

Figure 1: Scope-aware cascade routing. ABSTAIN predictions bypass the router; the router keeps or escalates each VALUE prediction.

Confidence baselines. The non-learned baselines rank VALUE predictions by a reliability score and accept the top-coverage fraction. The *global* baseline uses the average log-probability directly. The *family-calibrated* baseline fits an isotonic map from confidence to $P(\text{correct})$ per attribute family on the calibration split. Because a monotone calibration preserves within-family ranking, this baseline changes the ranking only by making scores comparable *across* families (Sec. 5).

Learned router. The router estimates $P(y \text{ correct} \mid s(y), z)$ for a VALUE prediction from SLM-side features only: raw confidence (average, minimum, and cumulative token log-probability) and *task-identity* features z: product type, attribute, attribute family, and their compositions (product-type$\times$attribute, and that $\times$locale, where locale is the catalog's marketplace). We use a gradient-boosted tree (LightGBM) with native handling of the high-cardinality categoricals; hyperparameters and feature importances are in App. C. Because it conditions on the SLM output alone, the choice of fallback is a separate empirical question, taken up in Sec. 7.

4 Experimental Setup

Data. All results are reported on a held-out English evaluation set of approximately 408K examples spanning 4,580 product-type–attribute pairs, 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 (boolean, closed-set categorical, numeric with units, numeric without units, free text). To determine correctness we normalize outputs and labels and apply a three-stage matching cascade: exact match, fuzzy (edit-distance) match, and an LLM equivalence check on the non-exact residual. We detail it in App. B. An abstention is scored correct when the ground-truth label is also an abstention.

Models. The SLM is Qwen3-4B fine-tuned for attribute generation with LoRA (rank 64, 3 epochs, $\sim$11M training examples). Fallbacks are Qwen3-235B-A22B, Qwen3-32B, and the cross-family Nova Pro, all given the same text input as the SLM. Standalone model performance is reported in App. A.

The confidence signal is informative for the fine-tuned SLM. Fine-tuning also sharpens it: the VALUE error-detection AUROC of average log-probability rises from 0.63 for the base SLM to 0.77 for the fine-tuned SLM. We therefore study the more demanding setting in which confidence is already a useful error signal and ask whether routing can improve beyond it.

Splits. Fine-tuning, router training, and evaluation use three instance-disjoint pools over the same product-type–attribute scope: an $\sim$11M-example fine-tuning set and separate validation and test pools ($\sim$2.5–3M each); no product instance appears in more than one pool. The router is trained on SLM VALUE predictions from a $\sim$400K sample of the validation pool; cutoffs for all policies (router and thresholds) are chosen on a 20% held-out calibration split of that sample; the $\sim$408K test set used for reporting is sampled from the test pool. Reported confidence intervals are 1000-sample bootstraps over the test set.

Why route VALUE only. SLM ABSTAINs are high-precision and few; a diagnostic sweep found

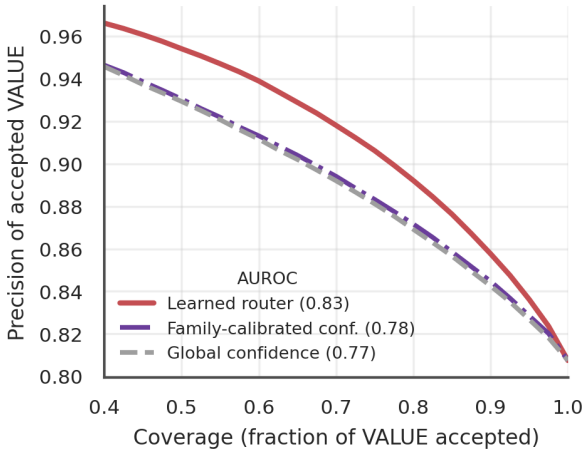

Figure 2: Selection quality on the fine-tuned SLM: precision of accepted VALUE predictions vs. coverage. The learned router dominates global and family-calibrated confidence at all plotted coverages (AUROCs in legend).

Table 1: Coverage and *achieved* test VALUE precision (router vs. global threshold) at matched dev-calibrated targets.

	Global threshold		Learned router	
Dev target	Cov.	Test P	Cov.	Test P
.88	82.7%	0.862	88.5%	0.863
.90	74.6%	0.882	83.0%	0.883
.92	64.6%	0.903	76.0%	0.903
.95	47.4%	0.933	63.2%	0.933

routing them net-neutral ($\sim$2% of routed cases, negligible effect on system metrics), so we keep them and route VALUE predictions only.

5 Scope-Aware Selection Beats Confidence Thresholding

We first ask a selection question, independent of the action taken: *which* VALUE predictions should not be trusted? A scorer ranks predictions by reliability and accepts the top-coverage fraction; a better scorer attains higher precision at any coverage. We compare the three SLM-side scorers of Sec. 3: global confidence, family-calibrated confidence, and the learned router, all calibrated on the same validation split.

Figure 2 gives the selection result. The learned router dominates all confidence-based scorers across the plotted coverage range, with error-detection AUROC 0.832 [95% CI 0.831–0.834] versus 0.770 [0.768–0.772] for global confidence (bootstrap over the test set; Sec. 4). This is a pure *selection* gain that identifies which predictions to trust, before any action is chosen: at the .92 operating point the router and the global threshold reach the *same* achieved test VALUE precision (0.903 vs. 0.903), yet the router accepts 11.4 pp more VALUE predictions [95% CI 11.2–11.5], with roughly one-third less deferral at the same quality. Table 1 shows this holds at every operating point: achieved test precision matches within 0.1 pp throughout, so the coverage advantage is not bought by emitting lower-precision values—at equal test precision the router simply accepts more. Crucially, calibrating confidence within attribute families does not close the gap: family-calibrated confidence reaches only AUROC 0.78, barely above the global threshold's 0.77. Why does task identity help? The next section localizes the gain.

Table 2: Feature ablation on the fine-tuned SLM (test error-detection AUROC). *Log-prob stats* are average, minimum, and cumulative token log-probability; *task scope* denotes product-type–attribute identity and attribute family.

Scorer / features	AUROC
Avg log-prob, threshold (no learning)	0.770
Learned: avg log-prob only	0.769
Learned: log-prob stats	0.797
Learned: log-prob stats + task scope	**0.832**
Linear (logistic): log-prob stats + scope	0.786

6 Why It Works: Task-Scope Reliability

Table 2 localizes the gain. A learned scorer given *only* confidence recovers the raw-log-probability threshold (0.769 vs. 0.770), and further log-probability statistics help only slightly (0.797); the largest gain comes when *task-scope* features are added, raising AUROC to 0.832. The router's advantage is thus not a better reading of confidence but a reliability *prior* keyed to which recurring task an output belongs to.

The mechanism is that SLM reliability varies sharply and predictably by task: across product-type$\times$attribute pairs, VALUE correctness ranges from $\sim$0.63 (10th pct) to $\sim$0.98 (90th pct), median $\sim$0.82. This variation is only partly captured by confidence, as a model can be confidently wrong on a chronically hard task, so a router that knows the task can discount confidence where it is unreliable. The benefit concentrates on free text (the majority of predictions) and closed-set categorical attributes, where confidence is weakest at separating correct from incorrect, and is small on numeric

and boolean attributes, where confidence already works.

The gain also requires using these features non-linearly: a logistic scorer over the same confidence-plus-scope features reaches only 0.786 (Table 2), short of the router's 0.832—the router learns a confidence$\times$task interaction a linear model cannot. The router's advantage over confidence thresholding also holds at nearby fine-tuning checkpoints (App. D).

Robustness to catalog refresh. Does a router calibrated once stay useful as the catalog is refreshed within the same scope? Applying the same validation-selected operating points, without recalibration, to a fresh catalog pull (159,491 slots, 3,210 product-type–attribute pairs, collected later) preserves the quality–coverage dial under a reference-free LLM judge: stricter points accept fewer VALUE predictions at higher judged correctness, monotonically (App. E); the .90 point, for instance, accepts 74.5% at 75.3% judged correctness. The learned ranking is thus not tied to the original product instances.

Robustness across languages. While the main study is on English, we find that training the identical router recipe on the English scope augmented with four European languages (German, French, Spanish, and Italian) reproduces the selection gain in every language: error-detection AUROC of 0.81–0.84 for the router versus 0.76–0.79 for raw confidence, and a $+10$ to $+12$ pp coverage advantage at matched test precision (App. F). Pooling the additional languages leaves the English result unchanged (router AUROC 0.832).

7 Suppress and Escalate Optimize Different Operating Points

The router decides *which* risky VALUE predictions to act on (Sec. 5); a separate, orthogonal choice is *what* action to take on that selected set. We compare two actions on the identical selected set: *escalate* (defer the value to a fallback model, the cascade's default) or *suppress* (withhold the value and emit an abstention, scored in the ABSTAIN metrics, with no fallback call). Table 3 shows they optimize different operating points.

Suppression is a precision-gating action. Withholding the router-selected values raises emitted-VALUE precision the most—to 0.903 at the $P{=}.92$ operating point—at no fallback cost, but it lowers VALUE recall (those inputs emit no value) and, by converting values to abstentions, trades abstention precision for abstention recall.

Escalation is a recovery action. Sending the selected values to a fallback recovers VALUE recall *relative to suppression* (it can return a value rather than nothing), though recall stays below the no-routing SLM (0.856 vs. 0.907 at $P{=}.92$), since these are the hardest cases. Because the fallback is no oracle, escalation also does not raise precision as aggressively as suppression. The two same-family fallbacks expose *different recovery profiles*: at $P{=}.92$ Qwen3-235B gives higher VALUE precision (0.835 vs. 0.809) and abstention recall (0.565 vs. 0.484), while the smaller Qwen3-32B gives higher abstention precision (0.750 vs. 0.640); across targets the 235B holds VALUE precision roughly flat while the 32B's declines. Fallback choice is thus a choice over recovery profiles, not determined by model size alone. To test whether the ceiling is a same-family artifact, we also escalate to a cross-family fallback (Nova Pro; App. G, VALUE F1 0.839, ABSTAIN F1 0.586 at $P{=}.92$). Neither a different family nor greater scale substantially raises the ceiling here, consistent with these being intrinsically hard cases that no *same-information* fallback recovers, pointing to complementary evidence (e.g. product images or retrieval) rather than a different model.

These actions sit at different points on the cost axis. Suppression adds no inference cost. Escalation cost scales with the escalated fraction: the 32B and 235B fallbacks cost roughly $14\times$ and $20\times$ the SLM per call ($\approx$\$12, \$162, and \$238 per million inferences).[1] At $P{=}.92$, escalation touches 24% of VALUE predictions ($\approx$22% of all traffic), so the fallback premium applies to that fraction rather than to total volume. A deployment can thus pick the action and the fallback that match its precision, recall, and budget priorities.

Deployment and integration. The router drops into an existing cascade rather than changing it: at serving time it is a single tree-ensemble evaluation running in-process on CPU, negligible next to SLM generation, with no GPU and no extra service call. The features it needs are already at the deci-

[1] Estimated from Amazon Bedrock's publicly listed prices at 1000 input / 20 output tokens: the 4B SLM uses Custom Model Import pricing (one model unit at 80 req/s); the fallbacks use on-demand token pricing. These are operating-point comparisons under common assumptions; realized cost varies with region, throughput, token counts.

Table 3: Suppress vs. escalate on the same router-selected predictions at P=.92 (test). **Dev target** is the validation-calibrated precision target, not achieved test precision (VALUE P). Full grid: App. G.

Dev target	Policy	acted-on %	VALUE P	VALUE R	VALUE F1	ABSTAIN P	ABSTAIN R	ABSTAIN F1
—	FT SLM (no routing)	0.0	0.807	0.907	0.854	0.875	0.369	0.519
.92	Suppress	24.0	0.903	0.771	0.832	0.435	0.732	0.546
	Escalate→235B	24.0	0.835	0.856	0.845	0.640	0.565	0.600
	Escalate→32B	24.0	0.809	0.871	0.839	0.750	0.484	0.588

sion point—confidence statistics from the SLM's generation output, and task identity from request metadata—so integration adds one predicate to the cascade's escalation trigger. The threshold is the deployment knob: the quality–cost point can be moved without retraining either model, though it should be re-selected on fresh validation data as the catalog drifts.

8 Related Work

Routing and cascading for LLMs. Multi-model inference balances quality and cost by assigning inputs to models of different capacity: *routing* makes a pre-generation model choice from query-level signals (Shnitzer et al., 2023; Ong et al., 2025; Ding et al., 2024), while *cascading* decides post-generation whether to accept an output or invoke another model (Chen et al., 2023; Shankar et al., 2026; Nie et al., 2024; Mansoori et al., 2026), with recent work combining both (Dekoninck et al., 2025). We operate within the cascading paradigm, but study structured generation where the output may be a schema value or an abstention, and ask which *features* of the first-stage output best predict when to defer.

Confidence signals and calibration. Token-level log probabilities and derived confidence scores are a common, readily available signal for post-generation cascade decisions, including confidence-thresholded model and task cascades (Shankar et al., 2026); learned quality estimators may instead use logit-derived features such as perplexity and logit quantiles (Dekoninck et al., 2025). Calibration studies show that aggregate confidence is often miscalibrated (Guo et al., 2017; Nixon et al., 2019) and that sequence likelihood is misaligned with semantic correctness (Kumar and Sarawagi, 2019; Jiang et al., 2021), motivating many alternatives to average log-probability—self-evaluation, contextualized likelihood, sampling-based consistency, and ensembles, among others (Malinin and Gales, 2021; Kadavath et al., 2022; Manakul et al., 2023; Vashurin et al., 2025). Our finding is related to subgroup calibration and multicalibration (Hébert-Johnson et al., 2018; Detommaso et al., 2024), which argue that reliability should be calibrated over meaningful subpopulations rather than only globally; our setting differs in using such task-scope structure for the post-generation keep/escalate decision in an LLM cascade, and we find (Sec. 5) that calibrating confidence within attribute families helps only marginally—the gain comes from using task identity as a feature rather than from recalibrating confidence within subgroups. Thus, rather than proposing a new confidence score, we show that within a fixed deployment scope the *task scope* of a prediction supplies deferral signal that confidence alone does not, and that a learned router can exploit the two jointly.

Selective prediction and learning-to-defer. Confidence-based accept/defer decisions are formalized in selective prediction and reject-option learning (El-Yaniv and Wiener, 2010; Franc et al., 2023) and in learning-to-defer (Madras et al., 2018; Mozannar et al., 2023); abstention has separately been studied as an LLM reliability mechanism (Wen et al., 2025; Kirichenko et al., 2025). Our setting differs in two ways: abstention is itself a task output, so the policy must distinguish a genuine abstention from a confidence-based suppression; and the deferral decision is single-shot and supervised by an observed correctness label, which we model with a simple discriminative router rather than a sequential or decision-theoretic controller (Sikeridis et al., 2024; Aggarwal et al., 2024).

9 Conclusion

We show a practical recipe for fine-tuned-SLM cascades: fine-tune, calibrate a scope-aware router on validation data, then choose the action and fallback that fit the precision, recall, and cost target.

We demonstrate that a lightweight scope-aware router selects risky VALUE predictions better than confidence thresholding, with the gain coming

from task-scope reliability priors rather than a better confidence transform. We further show that suppressing versus escalating the selected set, and the choice of fallback, expose distinct operating points a deployment can pick to optimize the quality–cost tradeoff.

Limitations

We study the routing decision in isolation and offline: we report selection quality, operating-point trade-offs, and relative serving cost, but end-to-end serving throughput and a live or A/B deployment are out of scope. The study is likewise scoped to a single task (attribute-value generation) and a single SLM family (Qwen), though we demonstrate multiple fallback models including an out-of-family one. We also do not benchmark a single mid-sized distilled model as an end-to-end alternative to the cascade; such a model would fit naturally as the cheap tier, with the same deferral question applying on top, and comparing the two is left to future work. The headline analysis is on English/US-marketplace data; however, App. F shows the selection result extends to four additional European languages (German, French, Spanish, Italian). By design the router exploits *within-scope* task reliability and we make no claim that it generalizes to product-type–attribute pairs unseen during fine-tuning, i.e. the cold-start case. A new deployment scope requires rerunning the fine-tune/calibrate/deploy procedure on data for that scope. Finally, cost ratios depend on token-size and serving assumptions but remain valid operating-point comparisons.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2024. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Zhu Cheng, Zhenming Wang, Yu Tang, Dan Liu, Bryan Zhang, Athanasios N. Nikolakopoulos, Pranav Souri Itabada, Jing Zhang, Chih-Chi Chou, Peng Gao, Fatemeh Mansoori, Bharat Bojja, Sarath Chander, Sameer Thombare, Umit Batur, and Tarik Arici. 2026. CatalogAgent: A supervisor-mediated self-learning system enabling context engineering for GenAI models. *arXiv preprint arXiv:2607.14396*.

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

Gianluca Detommaso, Martín Bertran, Riccardo Fogliato, and Aaron Roth. 2024. Multicalibration for confidence scoring in LLMs. In *International Conference on Machine Learning (ICML)*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*.

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

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. 2018. Multicalibration: Calibration for the (Computationally-identifiable) masses. In *International Conference on Machine Learning (ICML)*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*.

Saurav Kadavath, Tom Conerly, Amanda Askell, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. 2025. AbstentionBench: Reasoning LLMs fail on unanswerable questions. In *Advances in Neural Information Processing Systems*.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

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

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

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. 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)*.

Fatemeh Mansoori, Andrea Scarinci, Aditya Aggarwal, Suleiman Khan, and Ashwin Chandramouli. 2026. Improving cascade routing for structured attribute generation with heterogeneous confidence. In *ICML 2026 Workshop on Resource-Adaptive Foundation Model Inference (AdaptFM)*.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. 2023. Who should predict? exact algorithms for learning to defer to humans. In *International Conference on Artificial Intelligence and Statistics*.

Lunyiu Nie, Zhimin Ding, Erdong Hu, Christopher Jermaine, and Swarat Chaudhuri. 2024. 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*.

Athanasios N. Nikolakopoulos, Swati Kaul, Siva Karthik Gade, Bella Dubrov, Umit Batur, and Suleiman Ali Khan. 2023. SAGE: Structured attribute value generation for billion-scale product catalogs. *arXiv preprint arXiv:2309.05920*.

Jeremy Nixon, Michael W. Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. 2019. Measuring calibration in deep learning. In *CVPR Workshops*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*.

Soham Satyadharma, Fatemeh Sheikholeslami, Swati Kaul, Aziz Umit Batur, and Suleiman A. Khan. 2025. Auto prompting without training labels: An LLM cascade for product quality assessment in e-commerce catalogs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 937–953.

Shreya Shankar, Sepanta Zeighami, and Aditya G. Parameswaran. 2026. Task cascades for efficient unstructured data processing. *Proceedings of the ACM on Management of Data*, 4(1):1–26.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

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

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with LM-Polygraph. *Transactions of the Association for Computational Linguistics (TACL)*, 13:220–248.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. 2025. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556.

Bryan Zhang, Suleiman Khan, and Stephan Walter. 2025. Leveraging product catalog patterns for multilingual e-commerce product attribute prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*.

Table 4: Standalone models on the test set (VALUE and ABSTAIN precision/recall).

Model	VALUE P	VALUE R	ABSTAIN P	ABSTAIN R
Base SLM	0.688	0.749	0.511	0.299
Fine-tuned SLM	0.807	0.907	0.875	0.369
Qwen3-32B	0.710	0.751	0.515	0.377
Qwen3-235B	0.753	0.677	0.389	0.573
Nova Pro	0.708	0.754	0.504	0.351

A Standalone Model Performance

Table 4 reports the standalone test-set performance of the SLM and fallbacks. Fine-tuning lifts VALUE precision and recall well above the base SLM; the fallbacks, applied alone, do not dominate the fine-tuned SLM—the 235B trades higher abstention recall for lower VALUE recall, while the 32B and the cross-family Nova Pro land below the SLM on VALUE—so they are complementary on the routed hard subset (Sec. 7), not standalone replacements.

B Gold-Set Evaluation Protocol

All primary results are scored against human-annotated, audited labels. A prediction is judged correct through a three-stage matching cascade applied identically to every model and policy:

1. **Exact match.** The prediction and the gold value set are lowercased and compared; a prediction found in the gold set is correct.

2. **Fuzzy (edit-distance) match.** After removing stopwords and punctuation, we compute a token-sorted edit-distance similarity (rapidfuzz `token_sort_ratio`, on a 0–100 scale) between the prediction and the gold value. A prediction that is not an exact match but scores above 90 is credited as correct, absorbing word-order, spacing, and minor spelling differences.

3. **LLM synonym fallback.** Predictions that are not exact matches are sent to a frontier instruction-tuned LLM, prompted to decide whether the prediction is the same as or semantically equivalent to the gold value.

Stages 1–2 are deterministic and resolve 72% of the fine-tuned SLM's VALUE predictions; the remaining 28% (105,973 of 378,115; non-exact, non-abstention predictions) are adjudicated by the LLM. Because stages 1–2 are deterministic and the same judge prompt is applied identically to every model and policy, any judge error is shared across systems and largely cancels in the relative comparisons that carry our claims; it bears on absolute accuracy, not on the router-vs-baseline gap.

Caveats of the scoring protocol. The fuzzy-match threshold is a heuristic and can occasionally credit a near-miss that changes meaning, or pass a borderline case to the LLM. The LLM synonym fallback has its own error. Labels are single-reference, so a prediction that is correct but phrased differently from the recorded gold value may be undercredited if the fallback does not recognize it. We do not separately quantify the LLM judge's agreement with human annotators on the residual it scores. We apply the protocol uniformly across systems, so relative comparisons are less sensitive to scorer choice than absolute accuracy, though scorer errors remain a limitation.

C Router Configuration

The router is a LightGBM gradient-boosted tree: 400 trees, `num_leaves` $= 31$, `min_data_in_leaf` $= 500$, learning rate 0.05, `feature_fraction` $=$ `bagging_fraction` $= 0.8$, ℓ_2 regularization 1.0. It takes eight features: three raw confidence statistics (average, minimum, and cumulative token log-probability) and five task-identity categoricals—product type, attribute, attribute family, product-type$\times$attribute, and product-type$\times$attribute$\times$locale—handled natively by LightGBM's categorical splits (no one-hot encoding); unseen categories at test time are routed by LightGBM's default branch. The router is fit on the SLM's validation-set VALUE predictions with label $\mathbb{1}[\text{correct}]$; accept thresholds are chosen on a 20% calibration split to meet each target VALUE precision (Sec. 3).

By gain importance, the dominant features are average log-probability and product-type$\times$attribute (co-dominant: 338K and 327K), followed by minimum log-probability (110K), product type (89K), product-type$\times$attribute$\times$locale (86K), attribute (78K), cumulative log-probability (54K), and attribute family (11K). That product-type$\times$attribute rivals confidence—while attribute family alone contributes little—is consistent with the ablation (Table 2): the signal is the specific recurring task, not its coarse family.

D Checkpoint Robustness

The headline results are reported at our main 3E checkpoint, which we treat as a representative point

in this saturated region rather than a loss-optimal one. Table 5 repeats the analysis at nearby fine-tuning checkpoints, each router trained on its own checkpoint's validation predictions and evaluated on its own test set with the identical recipe. Two things hold across checkpoints. First, standalone fine-tuned quality is essentially saturated: VALUE and ABSTAIN precision/recall barely move from 2E to 4E, so the checkpoint choice is not a sensitive quality lever. Second, the router improves error-detection AUROC over raw confidence by a consistent margin (router AUROC drifts slightly—0.833/0.832/0.828 at 2E/3E/4E—while its coverage advantage at the .92 target holds at $\sim$10–12 points), so the routing benefit is stable rather than specific to 3E.

E Fresh-Catalog Robustness Check

To test that the calibrated operating points remain useful as the catalog is refreshed, we draw a fresh catalog pull from the same product-type–attribute scope, collected later, comprising 159,491 evaluable slots across 3,210 product-type–attribute pairs. We apply the router and the validation-selected operating points unchanged—no recalibration, no re-fitting—and route VALUE predictions exactly as in the main experiments.

Unlike the primary evaluation (App. B), which scores against human-annotated labels, this pull has no gold labels. We therefore score each accepted VALUE prediction with a reference-free judge—the same frontier LLM as in App. B, but given the product context, the target-attribute definition, and the predicted VALUE instead of a gold label. It marks a value correct if the context states it, an equivalent of it, or reasonably implies it (e.g. from product type or material), and incorrect if the value contradicts the context or makes an unsupported specific claim. Judged correctness is the fraction of accepted VALUE predictions the judge marks correct. This is a label-free automated proxy—distinct from, and weaker than, the gold-anchored protocol used for our main results—so we use it only to check that the operating points preserve the expected quality–coverage ordering on newly sampled items, not to report absolute accuracy.

Table 6 reports the result for the FT (3E) router across four dev-calibrated operating points. The pattern is monotone and matches the labeled-test behavior: tightening the target lowers coverage and raises judged correctness. Coverage at a given target differs from the labeled test set (e.g. 74.5% vs. 83.0% at .90) because the fresh pull has a different score distribution, so the same dev-calibrated threshold accepts a different fraction; the point here is the preserved monotonic ordering, not the absolute coverage. The calibrated dial therefore transfers to fresh catalog data without re-tuning.

F Robustness Across Languages

The headline study is on English. To test whether scope-aware selection is specific to that language, we train the *identical* router recipe (same features, hyperparameters, and calibration procedure) on the English scope augmented with four European languages—German, French, Spanish, and Italian—pooled into one router. Each language corresponds to one regional marketplace (US/English, Germany/German, France/French, Spain/Spanish, Italy/Italian); the router sees this only through the *locale* component of its task-identity features (Sec. 3), not as an explicit language input. Train, validation, and test pools are instance-disjoint, as in the main study: the pooled router is trained on $\sim$1.0M validation-set VALUE predictions (English plus the four languages) with a 20% calibration split for threshold selection, and evaluated on held-out per-language test sets ranging from roughly 100K to nearly 400K VALUE predictions each. Per language we report error-detection AUROC and, with each language calibrated to its own dev .92 VALUE-precision target, the coverage and achieved test VALUE-precision of the router versus the raw-confidence threshold (Table 7). Non-English labels are scored by the same three-stage matching protocol (App. B); the LLM judge's accuracy may vary across languages, but since the router-vs-confidence comparison within each language uses the same labels for both scorers, any such variation cancels in the within-language gap we report.

The selection gain replicates in every language: the router improves error-detection AUROC over raw confidence by about 0.05–0.06 throughout, and at matched test precision accepts 10–12 pp more VALUE predictions than the confidence threshold. On this pooled router the English row reproduces the headline AUROC of 0.832 (Sec. 5); its coverage figures differ from Table 1 by under 1.5 pp because thresholds here are selected on the pooled calibration split. Adding the other languages therefore leaves the English scope essen-

Table 5: Checkpoint robustness (test set). Standalone fine-tuned VALUE/ABSTAIN precision and recall are flat across checkpoints; the router's error-detection AUROC and its coverage advantage over confidence at the dev-calibrated .92 operating point are stable. 3E (our main checkpoint) is a representative point in this saturated region, not a loss-optimal one.

	VALUE		ABSTAIN		Error-det. AUROC		Coverage @ dev .92	
Checkpoint	P	R	P	R	Confidence	Router	Confidence	Router
2E	0.804	0.906	0.876	0.357	0.768	0.833	63.8%	75.8%
3E	0.807	0.907	0.875	0.369	0.770	0.832	64.6%	76.0%
4E	0.808	0.908	0.874	0.373	0.771	0.828	67.6%	77.6%

Dev op. point	Coverage	Judged corr.
.95	53.4%	78.9%
.92	66.5%	76.7%
.90	74.5%	75.3%
.88	81.5%	73.9%

Table 6: Fresh-catalog robustness (FT 3E router). Validation-selected operating points applied without recalibration to a fresh catalog pull (159,491 slots, 3,210 product-type–attribute pairs; 117,867 VALUE predictions), scored by a reference-free LLM judge. Coverage = accepted VALUE / all VALUE; judged correctness = judge-correct / accepted VALUE. The quality–coverage ordering is preserved.

tially unchanged.

G Full Suppress/Escalate Operating-Point Grid

Table 8 reports the complete suppress-vs-escalate grid across all four dev-calibrated VALUE-precision targets (0.88–0.95); the main text (Table 3) shows the 0.92 operating point. All numbers are on the test set, computed with the same scorer applied to each policy's constructed final prediction. "Dev target" is the validation-calibrated selection threshold, not achieved test precision.

Table 7: Robustness across languages (test set). One router trained on English+4 European languages, evaluated per language; each language is calibrated to the dev .92 VALUE-precision target. AUROC is VALUE error-detection. At matched test precision the router accepts more VALUE predictions than the confidence threshold in every language.

	Error-det. AUROC		Coverage @ dev .92		Achieved test P	
Language	Confidence	Router	Confidence	Router	Confidence	Router
English	0.770	0.832	65.9%	76.2%	0.900	0.903
German	0.764	0.810	58.1%	68.6%	0.900	0.897
French	0.773	0.828	55.4%	67.2%	0.908	0.906
Spanish	0.778	0.836	63.4%	74.7%	0.908	0.906
Italian	0.787	0.841	62.9%	74.7%	0.916	0.913

Table 8: Full operating-point grid: suppress vs. escalate at dev VALUE-precision targets 0.88–0.95 (test set). Acted-on % is the fraction of VALUE predictions selected (suppressed or escalated). The same router selection is applied throughout.

Dev tgt	Policy	acted-on %	VALUE P	VALUE R	VALUE F1	ABSTAIN P	ABSTAIN R	ABSTAIN F1
—	FT SLM (no routing)	0.0	0.807	0.907	0.854	0.875	0.369	0.519
.88	Suppress	11.5	0.863	0.859	0.861	0.571	0.586	0.578
	Escalate→235B	11.5	0.830	0.888	0.858	0.735	0.494	0.591
	Escalate→32B	11.5	0.815	0.894	0.853	0.812	0.444	0.574
	Escalate→Nova Pro	11.5	0.815	0.894	0.853	0.812	0.444	0.574
.90	Suppress	17.0	0.883	0.823	0.852	0.499	0.656	0.567
	Escalate→235B	17.0	0.834	0.875	0.854	0.687	0.529	0.598
	Escalate→32B	17.0	0.813	0.885	0.847	0.783	0.463	0.582
	Escalate→Nova Pro	17.0	0.813	0.885	0.848	0.785	0.462	0.581
.92	Suppress	24.0	0.903	0.771	0.832	0.435	0.732	0.546
	Escalate→235B	24.0	0.835	0.856	0.845	0.640	0.565	0.600
	Escalate→32B	24.0	0.809	0.871	0.839	0.750	0.484	0.588
	Escalate→Nova Pro	24.0	0.809	0.872	0.839	0.754	0.480	0.586
.95	Suppress	36.9	0.933	0.662	0.774	0.356	0.840	0.500
	Escalate→235B	36.9	0.832	0.821	0.826	0.578	0.612	0.595
	Escalate→32B	36.9	0.799	0.847	0.822	0.704	0.509	0.591
	Escalate→Nova Pro	36.9	0.799	0.848	0.822	0.708	0.504	0.589