MARGINGATE: Training-Free Routing for Cost-Efficient Taxonomy Construction with LLM Reasoning

Anonymous ACL submission

Abstract

Many classification systems pair a cheap embedding model with an expensive LLM; the core challenge is deciding which items need the expensive path. In evolving taxonomies, exhaustive LLM use *degrades* performance: wrong assignments become exemplars that shift centroids, causing a 16–23 point accuracy collapse. We show that *embedding margin*, the gap between the top-two similarity scores, identifies which items benefit from LLM reasoning. LLM accuracy increases monotonically with decreasing margin across three model families (open-weight and proprietary) and three benchmarks (Spearman $\rho \geq 0.87$ on 8/9 conditions), making lowest-margin-first provably optimal for any fixed budget. We build MARGINGATE, a two-stage pipeline: margin selects *which* items to route, and a confidence filter selects *whether* to accept the LLM's answer. The same signal enables category discovery: items far from all known categories are diverted to a buffer for LLM-based grouping rather than forced into wrong categories. When all categories are known, MARGINGATE matches exhaustive LLM accuracy at 30% cost; when unknown categories are present, it gains +6–9 points over exhaustive LLM ($p < 0.03$) by preventing pollution and enabling discovery. Unlike confidence-based or learned routing, the decision requires only the cheap classifier's output and no additional training.

1 Introduction

Evolving taxonomies (intent ontologies for dialogue systems, product catalogs, support ticket routing, medical coding) require classifying incoming items into a growing category set. These systems process millions of items daily, and the standard architecture pairs a cheap embedding classifier (submillisecond, pennies per 1K items) with an expensive LLM that outperforms it on hard cases (Loukas et al., 2023) but costs 10–100× more. The practical question is: *which* items deserve the expensive classifier, without training a separate router?

This question becomes critical in evolving taxonomies, where exhaustive LLM use can *degrade* system performance through ontology pollution: when an unknown-intent utterance is forced into the closest known category, it becomes an exemplar for that category, shifting its centroid and degrading nearest-neighbor lookups for all future items. Each wrong assignment makes the next one more likely, so misclassifications compound over cycles. We show empirically that this pollution causes a 16–23 point accuracy collapse (from 81–87% to 64–71%) compared to a setting where all categories are known, a failure mode invisible to standard single-pass evaluation. This motivates a training-free routing principle that composes cheap and expensive classifiers into a system outperforming either alone.

Existing approaches to cost-efficient routing each leave a gap in this setting. LLM self-reported confidence is poorly calibrated (79–98% "high confidence" predictions across our benchmarks), consistent with broader findings (Kadavath et al., 2022; Xiong et al., 2023), making it unusable as a routing signal. Learned routers (Ong et al., 2024; Chen et al., 2023) achieve strong results on static benchmarks but require training data and a fixed label space; in an evolving taxonomy, the router goes stale as new categories appear each cycle. Active-learning uncertainty (Settles, 2009) selects items for human annotation, not automated classification, a different objective that optimizes for label diversity rather than classification accuracy.

We propose *embedding margin*, the gap between the top-two similarity scores from the cheap classifier, as a training-free routing signal. LLM accuracy increases monotonically as margin decreases across 9 model×dataset conditions (Spearman $\rho \geq 0.87$ on 8/9); under this monotonicity, routing the lowest-margin items is provably optimal for any fixed budget. The signal requires no

model internals, no gradient access, no fine-tuning; it is a byproduct of the embedding classifier already in use. We build MARGINGATE, combining margin routing with a confidence filter that decides *whether* to trust the LLM's answer on routed items. The same geometric signal extends to novelty detection: items far from all known categories are diverted to a buffer for LLM-based discovery rather than forced into wrong categories.

We evaluate MARGINGATE in two configurations. Under full observability (depth-only), it matches exhaustive LLM accuracy at 30% cost ($p = 0.11$–0.32, 5 seeds), significantly outperforming top-1 threshold (+1.4–3.5pt) and entropy routing (+3.3–4.6pt) alternatives. When unknown intents are present (breadth+depth), the same LLM that achieves 81–87% under full observability collapses to 64–71% due to ontology pollution; MARGINGATE gains +6–9 points over exhaustive LLM ($p < 0.03$, 3 datasets) by preventing pollution and enabling structured discovery.

In summary, this paper makes three contributions:

1. Embedding margin as a universal routing signal for LLM deployment, validated on 9 model×dataset conditions, with a formal optimality guarantee under empirically validated monotonicity.
2. A two-stage routing pipeline (margin routing + confidence filtering) that matches exhaustive LLM at 30% cost under full observability, outperforming top-1 and entropy alternatives by 1.4–4.6 points.
3. Evidence that exhaustive LLM use *degrades* performance by 16–23 points in evolving taxonomies due to ontology pollution, and a unified pipeline that prevents this while gaining +6–9 points via margin-gated novelty detection and LLM-based discovery.

2 Related Work

LLM Routing and Cascades. FrugalGPT (Chen et al., 2023), RouteLLM (Ong et al., 2024), and CARGO (Barrak et al., 2025) train routers on quality labels or confidence signals; Dekoninck et al. (2024) unify these under cost-constrained optimization. All require labeled routing examples or model-internal signals; MARGINGATE requires neither, since the routing signal comes from similarities the embedding classifier already computes.

Training-Free Cascades and Deferral. Self-consistency cascades (Yue et al., 2024; Aggarwal et al., 2024) and semantic agreement (Soiffer et al., 2025) use agreement among stochastic decodes as a black-box routing signal; C3PO (Valkanas et al., 2025) adds conformal guarantees on cascade quality without labels. All require multiple inference passes to *generate* the signal; MARGINGATE requires zero additional inference, since margin is a byproduct of the embedding classifier already in use.

Selective Classification and Abstention. Abstain rules date to Chow (1957); Hendrycks and Gimpel (2017) propose MaxProb (our top-1 ablation) and Jitkrittum et al. (2023) categorize margin as a "differential reject" signal. We show that the top-2 *gap* consistently outperforms top-1 confidence in high-cardinality spaces (significant on 7/9 conditions).

Intent Classification and Active Learning. Arora et al. (2024) use MC-dropout uncertainty routing but require fine-tuning and a static taxonomy; IntentGPT (Lian et al., 2024) is training-free but routes every item to the LLM; Song et al. (2023) handle dynamic intents but require labeled discovery examples. Margin sampling is the most studied active-learning acquisition function (Settles, 2009; Scheffer et al., 2001; Margatina et al., 2023); our contribution is repurposing it for *automated* LLM routing (maximizing accuracy, not label diversity).

Intent Discovery. Deep Aligned Clustering (Zhang et al., 2021) and CDAC+ (Lin et al., 2020) discover intents via constrained clustering over learned representations, requiring training. MARGINGATE's novelty gate reuses the frozen embedding space with no separate model, and discovered categories immediately integrate into the routing taxonomy.

Out-of-Distribution and Novelty Detection. Energy scores (Liu et al., 2020) provide a principled alternative to MaxProb for OOD detection in neural classifiers, and conformal methods (Angelopoulos and Bates, 2021) offer distribution-free coverage guarantees. Our novelty gate uses a simpler signal (absolute top-1 similarity vs. an adaptive percentile threshold) that requires no model internals or held-out calibration set; the threshold is self-calibrating from the current taxonomy's own similarity distribution.

3 Method

3.1 Problem Formulation

Let $\mathcal{T} = \{(c_i, E_i)\}_{i=1}^{K}$ be a taxonomy of K categories, where each category c_i has an exemplar set E_i. Given a pre-trained encoder $\phi(\cdot)$, the per-category score for item x is $s_i(x) = \max_{e \in E_i} \mathrm{sim}(\phi(x), \phi(e))$. The *embedding classifier* assigns $f(x) = \arg\max_{c_i} s_i(x)$. Let $s_{(1)}(x) \geq s_{(2)}(x)$ be the two highest scores. The *embedding margin* is:

$$m(x) = s_{(1)}(x) - s_{(2)}(x) \tag{1}$$

A large margin means the embedding classifier is decisive; a small margin means the top categories are nearly tied. We also define the *absolute top-1 score* $s_{(1)}(x)$, which measures proximity to the nearest known category regardless of the runner-up (used for novelty detection in §3.2).

Items arrive in streaming batches B_t. An expensive classifier g (the LLM) operates in two modes: $g_{\text{cls}}(x)$ classifies a single item into an existing category and returns a confidence level; $g_{\text{disc}}(\mathcal{D})$ takes a batch of unclassified items and proposes new category groupings (full prompts in Appendix H). The practitioner specifies a budget k: the number of items per batch that may be routed to g. Our contribution is not how to set k, but *which items* should receive g for any fixed k.

Routing objective. A routing policy selects a subset $S \subseteq B$ with $|S| = k$ to send to g; the remainder use f. The composite accuracy is:

$$\mathrm{Acc}(S) = \frac{1}{|B|}\left[\sum_{x \in S} \mathbb{1}[g(x)=y_x] + \sum_{x \notin S} \mathbb{1}[f(x)=y_x]\right] \tag{2}$$

where y_x is the true label. We seek $S^* = \arg\max_{|S|=k} \mathrm{Acc}(S)$.

Property 1 (Margin–accuracy monotonicity)**.** *Let $a_g(m)$ and $a_f(m)$ denote the expected accuracy of classifiers g and f on items with margin m. The* LLM advantage *$\delta(m) = a_g(m) - a_f(m)$ is non-increasing in m.*

Proposition 1. *Under Property 1, routing the k items with lowest margin maximizes $\mathrm{Acc}(S)$ (proof in Appendix A).*

3.2 The MarginGate Pipeline

MarginGate operates in two modes unified by the same geometric signals: *depth* (refining known categories) and *breadth* (discovering new ones). Each cycle (Algorithm 1; Figure 1) partitions a batch into three streams based on two thresholds derived from the embedding space: items far from all categories enter a novelty buffer; items near a decision boundary go to the LLM with a confidence filter; the remainder use the embedding classifier directly. Accepted assignments and newly discovered categories feed back into the taxonomy, shifting thresholds for the next cycle.

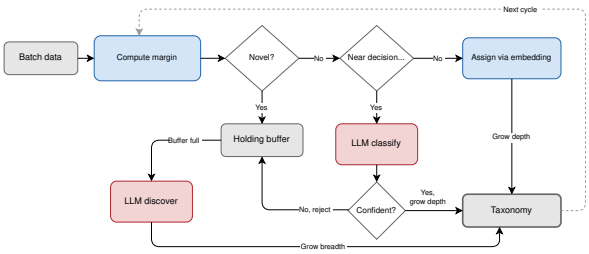

Figure 1: MarginGate pipeline: each batch is triaged by two embedding-derived signals into three streams. Items far from all categories are buffered for LLM discovery (breadth); items near a decision boundary go to LLM classification with a confidence filter (depth); the remainder are assigned directly by the embedding. Color encodes cost: blue = free embedding operations, red = LLM calls.

Algorithm 1 MarginGate — one cycle

Require: Batch B_t, taxonomy $\mathcal{T}$, budget k, novelty threshold τ_{nov}, cluster threshold n_{nov}
Ensure: Updated taxonomy $\mathcal{T}'$, buffer $\mathcal{D}$

Score: $m(x) \leftarrow s_{(1)}(x) - s_{(2)}(x)$ for all $x \in B_t$
$\mathcal{N} \leftarrow \{x \in B_t : s_{(1)}(x) < \tau_{\text{nov}}\}$ ▷ novelty gate
$R \leftarrow$ bottom-k by $m(x)$ from $B_t \setminus \mathcal{N}$ ▷ margin route
for $x \in R$ **do**
  $(\hat{y}, \mathit{conf}) \leftarrow g_{\text{cls}}(x)$ ▷ LLM classify
  if $\mathit{conf} = \texttt{high}$ **then**
    Add $(x, \hat{y})$ to $\mathcal{T}$ ▷ accept
  else
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{x\}$ ▷ reject to buffer
  end if
end for
for $x \in B_t \setminus \mathcal{N} \setminus R$ **do**
  Add $(x, f(x))$ to $\mathcal{T}$ ▷ embedding assign
end for
$\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{N}$ ▷ novelty candidates
if $|\mathcal{D}| \geq n_{\text{nov}}$ **then**
  $C_{\text{new}} \leftarrow g_{\text{disc}}(\mathcal{D})$ ▷ discover
  $\mathcal{T}' \leftarrow \mathcal{T} \cup C_{\text{new}}$ ▷ grow taxonomy
end if

Two-stage design. Margin routing identifies where g adds value, but low-margin items are exactly where g is most likely to err. The confidence filter addresses this: it prevents wrong LLM predictions from corrupting the taxonomy. Neither stage works alone: margin without the filter lets errors propagate; the filter without margin wastes budget on easy items.

Novelty gate. An item far from all known categories ($s_{(1)}$ below τ_{nov}) is unlikely to belong to any existing category. Forcing classification would inject a wrong exemplar, shifting the centroid and degrading future nearest-neighbor lookups. The novelty gate prevents this *ontology pollution* by diverting such items to the buffer. We define τ_{nov} as the 5th percentile of *ontology self-similarities*: for a sample of n=50 current exemplars, we compute each exemplar's maximum cosine similarity to the nearest exemplar of a *different* category (excluding self). This threshold adapts automatically as the taxonomy grows (the distribution shifts with new categories and exemplars), requiring no manual tuning across domains or embedding models.

Lifecycle applicability. The routing signal does not depend on a specific exemplar count or taxonomy maturity. As the taxonomy grows and exemplar sets expand, margin remains a valid routing criterion because it reflects relative category separation, not absolute embedding quality. We validate this empirically across exemplar counts in Appendix F.

3.3 Three-Tier Extension

An optional lightweight cross-encoder re-ranker screens routed items before the LLM, preserving the cost hierarchy between embedding lookup and LLM call: embedding → cross-encoder → LLM. The cross-encoder re-ranks the top-k candidates for margin-routed items; if the re-ranked margin exceeds a threshold, the item is resolved without an LLM call. The same margin-based routing decides what enters the cross-encoder tier (Appendix D).

4 Experimental Setup

4.1 Datasets

We evaluate on three intent classification benchmarks, all embedded with Cohere Embed v3 (1024-dim): Banking77 (Casanueva et al., 2020) (77 intents, 13K utterances) covers fine-grained banking intents; HWU64 (Liu et al., 2019) (64 intents, 11K utterances) spans home assistant commands across diverse categories; CLINC150 (Larson et al., 2019) (150 intents, 22.5K utterances) covers 10 diverse domains.

4.2 LLM Classifiers

We evaluate three LLM classifiers spanning different model families and capability tiers: Claude Haiku 4.5 (cheapest, serving as a conservative test), Claude Sonnet 4.6 (mid-tier), and Mistral Large 3 (Mistral AI, 2025) (open-weight, accessed via AWS Bedrock).

4.3 Evaluation Framework

Routing signal validation (§5.1). We first isolate whether margin is a reliable predictor of LLM value, independent of the full pipeline. Each dataset uses 78–80% of intents as known with 3 exemplars per intent, and $n = 200$ held-out items are scored. We perform quintile analysis of margin versus accuracy for both embedding and LLM classifiers, and compute Spearman ρ on quintile-binned accuracy across all 9 model×dataset conditions.

Depth-only simulation (§5.2). All intents are known from the start, so the pipeline only refines classification without discovery. Each intent begins with 5 exemplars. The simulation runs 10 cycles of 100 items each with a budget of $k = 30$ LLM calls per cycle (30%), repeated over 5 random seeds. This controlled setting enables comprehensive ablation of pipeline components (Table 1).

Breadth+depth simulation (§5.3). The full pipeline operates under realistic conditions with an incomplete taxonomy that must simultaneously classify known intents and discover unknown ones. We hold out 20–22% of intents as unknown at the start, with 5 exemplars per known intent. The simulation runs 10 cycles of 100 items each with a budget of $k = 30$ LLM calls per cycle (30%), repeated over 5 random seeds. A 300-item test set that includes unknown-intent items evaluates the final taxonomy.

Baselines. We compare against three reference points that bound system performance.

(a) *Oracle* uses ground-truth labels and represents the accuracy ceiling achievable with perfect classification.

(b) *Embedding-only* (breadth+depth only) assigns all items by nearest-neighbor embedding lookup with no LLM calls, establishing the cost floor.

(c) *All-LLM* routes every item to the LLM regardless of margin, testing whether exhaustive LLM use suffices without selective routing. In the breadth+depth setting, the All-LLM prompt includes the "NONE" option (identical to MARGINGATE's classification prompt); items classified as NONE are skipped (not added to the taxonomy), but no discovery

Table 1: Depth-only evaluation: all intents known, pipeline refines classification only. Accuracy is mean ± std over 5 seeds (10 cycles each). All methods except All-LLM use 300 LLM calls (30% budget).

	Strategy	Banking77	HWU64	CLINC150	LLM calls
Baselines	Oracle	92.8 ± 1.8	85.7 ± 0.9	90.7 ± 1.2	0
	All-LLM	86.4 ± 2.8	81.3 ± 1.1	87.1 ± 1.6	1000
Ablations	No confidence filter	84.1 ± 2.9	77.9 ± 1.7	86.5 ± 1.7	300
	Top-1 threshold	84.1 ± 2.7	74.8 ± 2.8	84.5 ± 2.3	300
	Entropy routing	82.2 ± 3.8	73.7 ± 2.6	82.8 ± 2.8	300
Ours	**MARGINGATE**	**85.5 ± 3.1**	**78.3 ± 2.1**	**86.7 ± 1.8**	300
p (MARGINGATE vs All-LLM)		0.32	0.059	0.11	—

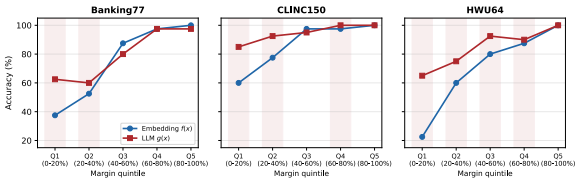

Figure 2: Accuracy by embedding margin quintile (n=40 per bin, Haiku 4.5). The LLM dominates at low margin (Q1–Q2) while the embedding classifier matches or exceeds it at high margin (Q4–Q5), motivating margin-based routing. Shading marks quintiles where the LLM outperforms the embedding.

mechanism groups them into new categories. This isolates the benefit of margin-gated routing and structured discovery from the mere presence of a NONE option.

Ablations. We isolate each component of MARGINGATE by replacing or removing it while holding the LLM budget constant at 30%.

(a) *No confidence filter* (depth-only) accepts all LLM predictions without quality gating, isolating the value of the two-stage design.
(b) *Top-1 threshold* (depth-only) routes by lowest absolute similarity $s_{(1)}$ instead of margin, testing whether the top-2 gap adds value over a MaxProb baseline.
(c) *Entropy routing* (depth-only) routes by highest entropy of the full similarity distribution, an information-theoretic alternative to margin.
(d) *K-means discovery* (breadth+depth) uses the same margin routing but replaces LLM-based discovery with k-means clustering, isolating the contribution of semantic grouping.

Statistics. We report bootstrap 95% confidence intervals computed from 10K resamples. All strategy comparisons use paired t-tests across 5 seeds ($\alpha = 0.05$).

5 Results

5.1 Routing Signal Validation

We validate that embedding margin is a reliable routing signal (Figure 2). Both embedding and LLM accuracy increase monotonically with margin, and the LLM's advantage concentrates in the low-margin zone: in the bottom quintile (Q1), the LLM outperforms the embedding by 25pt on Banking77, 25pt on CLINC150, and 42.5pt on HWU64. The two classifiers converge by Q5 (≥97.5% on all three datasets), confirming that high-margin items need no LLM verification. The crossover—where the embedding begins to match or exceed the LLM—occurs near Q3 (40–60% margin) on all three datasets, providing a natural operating point for the routing threshold.

The monotonic relationship holds across all three LLM families (Spearman $\rho \geq 0.87$ on 8/9 model×dataset conditions, Appendix B) without learning: the signal comes free from the embedding classifier. Margin also outperforms alternative routing signals—top-1 threshold (+2.5pt, $p < 10^{-11}$) and entropy (+4.2pt, $p < 10^{-19}$)—across all 9 conditions (Appendix B).

5.2 Depth-Only Evaluation

When all intents are known from the start, MARGINGATE matches All-LLM accuracy at 30% of the cost (Table 1). The differences are not statistically significant ($p = 0.11$–0.32), confirming that margin routing loses nothing by skipping 70% of LLM calls.

Ablations. We isolate each component by replacing or removing it (Table 1). Removing the confidence filter loses 0.2–1.4 points, confirming the two-stage design's value. Replacing margin with top-1 threshold (MaxProb routing) loses 1.4–3.5 points, confirming that the top-2 gap outperforms

Table 2: Breadth+depth evaluation: 20–22% of intents unknown at start, pipeline must classify known intents and discover new ones. Accuracy is mean ± std over 5 seeds (10 cycles each). †Includes 9 additional LLM calls for batch-clustering at discovery events. ‡Same routing as MARGINGATE but with k-means clustering instead of LLM discovery.

	Strategy	Banking77	HWU64	CLINC150	LLM calls
Baselines	Oracle (all intents known)	84.7 ± 2.1	81.6 ± 1.3	89.2 ± 1.9	0
	Embedding-only (cycle 0)	62.1 ± 3.1	59.8 ± 1.9	69.4 ± 0.3	0
	All-LLM	63.7 ± 2.5	64.4 ± 1.3	70.8 ± 1.5	1000
Ablation	K-means discovery‡	66.9 ± 0.5	66.4 ± 1.1	77.0 ± 2.3	300
Ours	**MARGINGATE**	**72.2 ± 1.2**	**70.0 ± 2.3**	**80.2 ± 1.1**	309†
p (MARGINGATE vs All-LLM)		0.003	0.022	<0.001	—

absolute confidence. Entropy routing loses 3.3–4.6 points, the weakest alternative — distributional uncertainty dilutes the decision-boundary signal. Since all ablations are comprehensively evaluated in this controlled setting, the breadth+depth evaluation (§5.3) focuses on the pipeline-level comparison without repeating them.

5.3 Breadth+Depth Evaluation

In the depth-only setting, exhaustive LLM use matched selective routing (Table 1). When 20–22% of intents are unknown, the picture reverses: All-LLM now *degrades* performance by forcing unknown-intent items into wrong categories, and each wrong exemplar shifts centroids, compounding errors over cycles (Table 2). Over 10 cycles, All-LLM gains only +1.4 to +4.6 points over the starting accuracy despite 1000 LLM calls; MARGINGATE gains +10 points with 300.

MARGINGATE vs. baselines. MARGINGATE significantly outperforms All-LLM on all 3 datasets (paired t-test: Banking77 $p = 0.003$, HWU64 $p = 0.022$, CLINC150 $p < 0.001$) and k-means discovery on all 3 (Banking77 $p = 0.002$, HWU64 $p = 0.020$, CLINC150 $p = 0.018$). MARGINGATE closes 45–55% of the gap between the starting accuracy and the oracle ceiling, compared to only 7–21% for All-LLM. The advantage appears immediately (by cycle 1) and is consistent across LLM families: in the routing signal validation (§5.1), the cross-model accuracy range is ≤ 1.2 points per dataset, confirming that the routing principle drives the gain rather than the specific LLM.

Ontology pollution explains All-LLM's ceiling. The same LLM that achieves 81–87% under full observability (Table 1) collapses to 64–71% when unknown intents are present (Table 2)—a 16–23 point drop attributable entirely to ontology pollution. The cause is structural: All-LLM must assign every item to a known category, so unknown-intent items are forced into wrong categories with high confidence, and each wrong exemplar shifts centroids for all future items. MARGINGATE avoids this because its routing is geometric rather than generative: the novelty gate diverts items far from all categories before any classification occurs, and the confidence filter rejects 9–18% of uncertain LLM assignments that would otherwise corrupt the taxonomy (Appendix C).

Decomposition of the advantage. Subtracting All-LLM from k-means isolates the novelty gate's contribution (pollution prevention: +2.0 to +6.2 points), while subtracting k-means from MARGINGATE isolates LLM discovery (+3.2 to +5.3 points). On CLINC150, pollution prevention accounts for 66% of the total gain because its diverse domains make novelty detection reliable; on HWU64, LLM discovery contributes more (64%) because same-domain intents require semantic reasoning to separate. CLINC150 achieves near-exact discovery (averaging 30.4 discovered clusters vs. 30 held-out ground-truth intents); Banking77 and HWU64 over-fragment (∼2×) but still achieve 68–72% cluster purity vs. 54–58% for k-means (Appendix G). A failure taxonomy of 61 LLM-win cases (Appendix E) confirms that 45% require reasoning beyond surface similarity, explaining why margin correctly routes these items to the LLM.

6 Discussion

Routing dominates model selection. All three LLM families achieve nearly identical MARGINGATE accuracy (within 1.2 points per dataset), yet the difference between *which items* reach the LLM yields 2.5–4.2 points at the same cost. This

suggests that for deployed systems, investing in smarter item selection yields better returns than upgrading the LLM itself.

Longitudinal evaluation matters. Standard single-pass evaluation misses the longitudinal failure mode that dominates our results: ontology pollution is cumulative, creating a feedback loop visible only over multiple cycles. MARGINGATE gains 7–9 points in cycles 1–5 (active discovery), then 1–3 points in cycles 6–10 (consolidation), while All-LLM gains at most +0.5 to +3.7 points in the first 5 cycles and barely improves thereafter. Evolving-taxonomy benchmarks, where models interact with their own past outputs, may offer a more realistic testbed than static accuracy.

Routing value spans the ontology lifecycle. The LLM outperforms the embedding on 7/9 dataset×exemplar conditions, and the advantage persists even at n=10 exemplars per intent on HWU64 (+6.0pt) and CLINC150 (+3.0pt; Appendix F). LLM-win cases concentrate in the bottom 60% by margin at all exemplar counts (89–100%), confirming that margin-based routing remains effective from cold-start through maturity.

Why training-free rather than learned routing? Learned routers (Ong et al., 2024; Chen et al., 2023) assume a fixed label space and require labeled routing examples. In an evolving taxonomy, both assumptions break: new categories appear each cycle, and labeled routing data does not exist at cold start. MARGINGATE sidesteps this by using the embedding geometry, which updates automatically as new exemplars arrive. A static-snapshot comparison would favor learned approaches, but would not capture the longitudinal setting where MARGINGATE operates without retraining.

Confidence filter: limitations and alternatives. The accept/reject filter relies on the LLM's self-reported confidence, which we observe is poorly calibrated globally (79–98% "high confidence" rates). However, the filter operates only on the margin-routed subset (low-margin, hard items), where miscalibration is less extreme and the binary high/low signal still rejects 9–18% of responses that would otherwise corrupt the taxonomy. The filter's contribution is modest in single-cycle terms (0.2–1.4 points) but compounds over cycles by preventing error propagation. Stronger training-free alternatives exist: semantic agreement across stochastic decodes (Soiffer et al., 2025), embedding-space verification (checking whether the LLM's predicted label is geometrically plausible given the item's position), or small verifier models. We chose LLM self-confidence for simplicity and zero additional cost; these alternatives could replace the filter as a drop-in improvement without changing the routing mechanism itself.

When does MARGINGATE fail? The framework requires meaningful embedding structure: if the embedding space carries no category-relevant signal, margin is uninformative. HWU64 has the weakest embedding ceiling (85.7% oracle vs. 90.7–92.8% for the others) yet margin monotonicity still holds ($\rho \geq 0.87$) and MARGINGATE still gains +5.6 points over All-LLM. Discovery over-fragments on same-domain taxonomies ($\sim$2$\times$ on Banking77 and HWU64), a limitation of single-domain category sets rather than of the routing mechanism. A post-discovery merge step—consolidating clusters whose centroids fall within a similarity threshold, or prompting the LLM to judge whether two discovered groups represent the same intent—could reduce fragmentation without sacrificing purity. We do not implement this here because fragmentation does not affect routing accuracy (the majority-vote evaluation handles many-to-one mappings), but it would improve interpretability of the discovered taxonomy in production.

Budget sensitivity and token cost. The 30% budget is not tuned. Composite accuracy varies by less than 2 points across the 30–70th percentile range on Banking77 and CLINC150; HWU64 shows a wider range (4 points) but increases monotonically, so practitioners can set budget purely by cost constraints (Appendix C). Because all classification calls use the same prompt structure (all intents enumerated with exemplars), the 30% call-rate reduction maps directly to a 30% reduction in token consumption and latency. Token cost per call scales linearly with $|\mathcal{T}|$ ($\sim$45 tokens per intent); at $|\mathcal{T}| = 150$, each call consumes $\sim$6.9K tokens. For large taxonomies ($|\mathcal{T}| > 200$), the cross-encoder tier (§3.3) provides additional savings by resolving items without any LLM call (Appendix C).

Broader applicability. The framework applies wherever (i) a cheap similarity classifier exists, (ii) an expensive reasoning classifier adds value on a subset, and (iii) the category set evolves. Product categorization, medical coding, support ticket routing, and document taxonomy construction all

share this structure.

7 Conclusion

We introduced MARGINGATE, a training-free routing framework that uses embedding margin to selectively invoke LLM classification. Margin's monotonic relationship with LLM accuracy makes bottom-k routing provably optimal for any fixed budget (Proposition 1). Under full observability, MARGINGATE matches exhaustive LLM accuracy at 30% cost, outperforming top-1 and entropy routing alternatives by 1.4–4.6 points. When unknown intents are present, exhaustive LLM use collapses by 16–23 points due to ontology pollution; MARGINGATE gains +6–9 points over All-LLM by preventing pollution and enabling structured category discovery via the same geometric signal.

Because the routing signal is a byproduct of the embedding classifier, MARGINGATE requires no labeled routing data and generalizes across LLM providers. Routing the right items to the right model outperforms exhaustive expensive classification by wider margins than any model upgrade we tested, suggesting that item selection matters more than model selection for deployed systems. The finding that exhaustive LLM use *degrades* performance in evolving taxonomies challenges the assumption that more LLM calls are always better; system-level quality depends on knowing when *not* to use the expensive resource. While we validate on intent taxonomies, the underlying principle applies to any evolving-taxonomy setting where cheap and expensive classifiers coexist.

Limitations

We use simulated taxonomy growth on static benchmarks rather than a deployed production system with real streaming data. Each dataset uses a single embedding model (Cohere Embed v3); we do not ablate across embedding architectures, though the monotonicity property is validated across three LLM families. The confidence filter relies on LLM self-reported confidence rather than a stronger training-free signal (e.g., stochastic-decode agreement or embedding-space verification); we chose simplicity over optimality here and expect stronger filters to improve the 0.2–1.4 point contribution. Discovery over-fragments on same-domain taxonomies (Banking77 and HWU64 fragment $\sim 2\times$) because all intents share a narrow semantic domain; merge heuristics (e.g., centroid-similarity consolidation) could reduce fragmentation. The failure taxonomy was validated via inter-model agreement ($\kappa = 0.51$) rather than human annotation. The test set is static; we do not evaluate under distribution shift between training and deployment.

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. *Advances in Neural Information Processing Systems*.

Anastasios N. Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of LLMs. In *Proceedings of EMNLP (Industry Track)*.

Amine Barrak, Halima Bouzidi, and Foutse Khomh. 2025. CARGO: Confidence-aware routing for cost-efficient LLM cascades. *arXiv preprint arXiv:2502.03261*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerber, Milica Gasic, and Nikola Mrkšić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI*.

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*.

C. K. Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. 2024. A unified approach to routing and cascading for LLMs. *arXiv preprint arXiv:2410.10347*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of ICLR*.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. 2023. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of EMNLP-IJCNLP*.

Juan Lian, Fanding Liu, Jingji Shen, and Ming Liu. 2024. IntentGPT: Few-shot intent discovery with large language models. *arXiv preprint arXiv:2404.07907*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of AAAI*.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of IWSDS*.

Lefteris Loukas, Ilias Stogiannidis, Odysseas Diamantopoulos, Prodromos Malakasiotis, and Stavros Vassos. 2023. Making LLMs worth every penny: Resource-limited text classification in banking. In *Proceedings of ICAIF*.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of EMNLP*.

Mistral AI. 2025. Mistral large 3. Accessed via AWS Bedrock as `mistral.mistral-large-3-675b-instruct`.

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

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden Markov models for information extraction. In *Proceedings of IDA*.

Burr Settles. 2009. Active learning literature survey. *Computer Sciences Technical Report 1648, University of Wisconsin–Madison*.

Duncan Soiffer, Steven Kolawole, and Virginia Smith. 2025. Semantic agreement enables efficient open-ended LLM cascades. In *Proceedings of EMNLP: Industry Track*.

Hao Song, Haiyang Xu, Wei Zhang, Fei Wu, and Zhijian Yan. 2023. A class-guided approach for intent discovery in open-domain dialogue systems. In *Proceedings of ACL*.

Antonios Valkanas, Soumyasundar Pal, Pavel Rumiantsev, Yingxue Zhang, and Mark Coates. 2025. C3PO: Optimized large language model cascades with probabilistic cost constraints for reasoning. In *Advances in Neural Information Processing Systems*, volume 38.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. *arXiv preprint arXiv:2306.13063*.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2024. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *Proceedings of ICLR*.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. In *Proceedings of AAAI*.

A Proof of Proposition 1

Proof. For any x, routing to g instead of f changes composite accuracy by $\delta(m(x))/|B|$. Maximizing Eq. 2 is equivalent to choosing k items that maximize $\sum_{x \in S} \delta(m(x))$. Since δ is non-increasing in m (Property 1), this sum is maximized by selecting the k items with smallest $m(x)$—the rearrangement inequality applied to a monotone sequence. □

B Routing Signal Validation

Table 3 reports per-condition pipeline accuracy for margin, top-1, and entropy routing (5 seeds each, 300 LLM calls total). Margin is directionally better on 9/9 conditions, statistically significant on 7/9 vs. top-1 and 9/9 vs. entropy. The two non-significant conditions (Banking77 with Sonnet and Mistral) show the same directional advantage (+1.3 and +2.0 points) but higher variance prevents significance with 5 seeds.

Property 1 is an empirical assumption. Table 4 reports Spearman ρ between margin quintile and LLM accuracy for all 9 conditions, confirming monotonicity holds on 8/9 (all $\rho \geq 0.87$, $p < 0.054$). The single condition below 0.9 (Haiku 4.5 on Banking77, $\rho = 0.87$, $p = 0.054$) is directionally consistent but underpowered with 5 quintile bins.

Table 4: Spearman ρ (margin quintile rank vs. LLM accuracy) across all model×dataset conditions.

	Banking77	CLINC150	HWU64
Haiku 4.5	0.87 (p=0.054)	0.97 (p=0.005)	0.90 (p=0.037)
Sonnet 4.6	1.00 (p<0.001)	0.97 (p=0.005)	0.97 (p=0.005)
Mistral 3	1.00 (p<0.001)	0.95 (p=0.014)	0.97 (p=0.005)

C Pipeline Sensitivity

Budget sensitivity. Table 5 shows composite accuracy at varying margin thresholds. Accuracy varies by less than 2 points across the 30–70th percentile range on Banking77 and CLINC150. HWU64 shows a wider range (4 points) but accuracy increases monotonically with budget, confirming that 30% is a conservative default.

Table 5: Composite accuracy (%) at varying margin thresholds ($n = 200$, Haiku 4.5). %LLM = fraction of items routed to the LLM classifier.

Pctile	%LLM	Banking77	CLINC150	HWU64
20	20%	80.0	91.5	78.5
30	30%	80.5	93.0	80.5
40	40%	81.5	94.5	81.5
50	50%	81.0	94.0	82.5
60	60%	80.0	94.0	84.0
70	70%	80.0	94.0	84.5

Confidence filter. Without the confidence filter (margin-only strategy), HWU64 loses 0.4 points and Banking77 loses 1.4 points in the depth-only setting. The filter rejects 9–18% of LLM responses depending on dataset. Its primary role is preventing error propagation over cycles rather than maximizing single-cycle accuracy.

Token cost analysis. The classification prompt enumerates all known intents with 3 exemplars each (Appendix H). Table 6 reports estimated token counts per call and total token consumption at 30% vs. 100% call rates. Because all calls use the same prompt structure regardless of routing strategy, the 30% call-rate reduction translates directly to a 30% token reduction: MARGINGATE saves proportionally in both calls and tokens.

Token cost per call scales linearly with taxonomy size ($|\mathcal{T}|$): each intent contributes ∼45 tokens (name + 3 exemplars + formatting). At $|\mathcal{T}| = 150$ (CLINC150), each call consumes ∼6.9K input tokens; at $|\mathcal{T}| = 300$ (a plausible production taxonomy), this grows to ∼13.6K tokens.

Table 6: Estimated token cost per classification call and total consumption over 10 cycles (N=1000 items). Token ratio = MARGINGATE / All-LLM.

Dataset	$\|\mathcal{T}\|$	Tokens/call	Total (30%)	Total (100%)
Banking77	77	3,570	1.07M	3.57M
HWU64	64	2,985	0.90M	2.99M
CLINC150	150	6,855	2.06M	6.86M

D Cross-Encoder Extension

Adding a cross-encoder middle tier (ms-marco-MiniLM-L-6-v2) between embedding and LLM pre-screens margin-routed items: items whose reranked margin exceeds a threshold are resolved without an LLM call (Table 7). With the same 30% LLM budget, the CE resolves ∼55% of screened items, reducing LLM calls by ∼10% at a cost of 0.1–1.4 points. Recovery across failure categories

Table 3: Pipeline accuracy (%) by routing signal across 9 model×dataset conditions (mean of 5 seeds). p-values from paired t-tests (margin vs. alternative). Bold = margin significantly better ($p < 0.05$).

Model	Dataset	Margin	Top-1	Entropy	p (vs Top-1)	p (vs Entropy)
Haiku 4.5	Banking77	85.5	84.1	82.2	**0.013**	**0.004**
Haiku 4.5	HWU64	78.3	74.8	73.7	**0.006**	**<0.001**
Haiku 4.5	CLINC150	86.7	84.5	82.8	**0.012**	**0.007**
Sonnet 4.6	Banking77	85.5	84.2	81.7	0.157	**0.001**
Sonnet 4.6	HWU64	79.2	75.3	74.2	**0.016**	**0.003**
Sonnet 4.6	CLINC150	87.3	84.9	82.7	**0.020**	**0.008**
Mistral 3	Banking77	84.9	82.9	81.1	0.129	**0.042**
Mistral 3	HWU64	78.0	74.6	73.5	**0.024**	**<0.001**
Mistral 3	CLINC150	86.5	84.1	82.6	**0.027**	**0.029**
Aggregated ($N = 45$)		—	—	—	$< 10^{-11}$	$< 10^{-19}$

is uniform (50–71%), confirming the LLM tier remains necessary for the reasoning-dependent 45%.

Table 7: Cross-encoder extension: pipeline accuracy (%) and call counts (mean of 5 seeds, 30 items routed per cycle). CE Resolved = items resolved by cross-encoder without LLM call.

	Banking77	HWU64	CLINC150
MARGINGATE	85.5	78.3	86.7
+ CE	85.5	76.9	85.5
Δ accuracy (pt)	−0.1	−1.4	−1.1
LLM calls saved	30 (10%)	30 (10%)	30 (10%)
CE calls	613	601	601
CE resolved	344 (56%)	331 (55%)	331 (55%)

E Failure Taxonomy

Table 8 categorizes 61 LLM-win cases by failure mode. Inter-model agreement reaches $\kappa = 0.51$ (moderate) at the super-category level and $\kappa = 0.35$ (fair) at the fine-grained level.

Table 8: Failure taxonomy of LLM-win cases.

Super-category	Category	n	%
Semantic	Implicit intent	23	37.7
	Pragmatic context	4	6.6
Surface	Lexical overlap	16	26.2
	Entity confusion	8	13.1
	Action ambiguity	3	4.9
Structural	Granularity mismatch	7	11.5

F Exemplar Sensitivity

Table 9 shows that the LLM outperforms the embedding classifier on 7/9 dataset×exemplar conditions. Banking77 is the exception: Cohere embeddings close the gap at $n \geq 5$ (the embedding actually surpasses the LLM due to rich representation of fine-grained banking terms). On HWU64 and CLINC150, the LLM advantage persists at $n = 10$: the gap narrows (from +14.5 to +6.0 on HWU64, from +8.0 to +3.0 on CLINC150) but never closes, confirming that routing value extends well beyond cold-start. LLM-win cases consistently concentrate in the bottom 60% by margin (89–100%), validating margin as the correct routing criterion at all exemplar counts.

Table 9: Accuracy (%) by exemplar count. Gap = $g - f$. LLM outperforms embedding on 7/9 conditions; Banking77 with strong (Cohere) embeddings is the sole exception at $n \geq 5$.

Dataset	n	f	g	Gap	LLM wins in bot-60%
Banking77	3	75.0	79.5	+4.5	100%
	5	81.5	79.5	−2.0	89%
	10	90.0	86.5	−3.5	100%
HWU64	3	70.0	84.5	+14.5	97%
	5	72.0	82.0	+10.0	96%
	10	80.5	86.5	+6.0	100%
CLINC150	3	86.5	94.5	+8.0	94%
	5	92.0	98.0	+6.0	100%
	10	93.0	96.0	+3.0	100%

G Discovery Diagnostics

Table 10 summarizes quality metrics across 5 seeds.

Label alignment procedure. Discovered clusters receive synthetic IDs during the simulation and are never shown ground-truth labels. For evaluation only, we align each discovered cluster C_j to the ground-truth intent via majority vote over the true labels of items in C_j; this mapping is strictly post-hoc and not fed back into the pipeline. Purity is the fraction of items in C_j matching its majority-vote

label; coverage is the number of distinct ground-truth intents represented across all clusters (coverage may exceed the held-out count when novelty-gated known-intent items enter clusters).

CLINC150 achieves near-perfect discovery because its 150 intents span 10 diverse semantic domains; LLM discovery covers 29.8/30 held-out intents with 55% of clusters at ≥80% purity. Banking77 and HWU64 over-fragment because all intents are within the same narrow domain. The purity advantage of LLM over k-means (+14–21 points) is consistent across datasets, confirming that LLM semantic grouping provides value beyond geometric proximity.

Table 10: Discovery quality (mean across 5 seeds). Frag. = clusters/held-out intents. High-P = clusters with purity ≥80%. Cov. = unique GT intents represented / held-out count.

		CLINC150	Banking77	HWU64
Frag.	K-means	1.01×	1.71×	2.19×
	LLM	1.01×	1.82×	2.11×
Purity	K-means	57.3%	53.8%	58.0%
	LLM	78.6%	67.8%	71.7%
High-P	K-means	28%	19%	28%
	LLM	55%	36%	45%
Cov.	K-means	24.6/30	22.6/17	20.2/14
	LLM	29.8/30	27.2/17	26.2/14

H LLM Prompts

We use three prompts throughout the paper. All calls use temperature 0 and require structured JSON output.

Classification prompt (depth and breadth+depth). Used for margin-routed items in both evaluation settings. The breadth+depth variant adds a "NONE" option for items that do not match any known intent.

```
Classify the following utterance into one of
the intents listed below.
If the utterance does NOT clearly match any
intent, respond with "NONE".

Utterance: "{utterance}"

Intent: "{intent_name}"
  Examples:
  - "{example_1}"
  - "{example_2}"
  - "{example_3}"
[... repeated for each known intent ...]

Respond with ONLY a JSON object:
{"intent": "<intent_name>",
 "confidence": "high|medium|low"}
Use "NONE" as intent if it doesn't fit any
of the above.
```

Discovery prompt (breadth+depth). Used when the novelty buffer reaches the discovery threshold. The prompt receives up to 30 known intent names and all buffered utterances.

```
Below are utterances that don't clearly match
any known intent in our system.

Known intents: "intent_1", "intent_2", ...
Previously discovered intents (assign
utterances here if they fit):
  "discovered_intent_1":
    - "example_1"
    - "example_2"

Unmatched utterances:
  1. "utterance_1"
  2. "utterance_2"
  [...]

Your task:
1. First, check if any utterances match a
   previously discovered intent listed above.
   If so, assign them using that EXACT name.
2. For remaining utterances, identify groups
   that represent the SAME new intent.
3. Give each new group a short, descriptive
   intent name.
4. Only group utterances you are confident
   belong together (minimum 2 per group).
5. Leave ungroupable utterances out.

Respond with ONLY a JSON array:
[{"intent_name": "<name>",
  "utterance_numbers": [1, 3, 5]}]

If no clear groups exist, respond with: []
```

Failure taxonomy prompt (Appendix E). Used to classify embedding failure modes on the 61 LLM-win cases. Each case is annotated independently.

```
Analyze why an embedding-based intent
classifier misclassified this utterance,
while an LLM got it right.

Utterance: "{text}"
True intent: {true_intent}
Embedding predicted: {emb_intent}
LLM predicted (correct): {llm_intent}

Classify the PRIMARY reason the embedding
failed into exactly ONE of these categories:

1. LEXICAL_OVERLAP: The utterance shares key
   words with the wrong intent
2. ACTION_AMBIGUITY: The utterance implies an
   action that maps to multiple intents
3. IMPLICIT_INTENT: The utterance doesn't
   explicitly state the intent -- requires
   inference from context
```

```
4. ENTITY_CONFUSION: The utterance mentions
   entities that belong to a different
   intent's typical context
5. GRANULARITY_MISMATCH: The true and
   predicted intents are sub-categories of
   the same broader intent
6. PRAGMATIC_CONTEXT: Understanding requires
   world knowledge or pragmatic reasoning
7. OTHER: None of the above

Respond with ONLY a JSON object:
{"category": "<CATEGORY>",
 "reasoning": "<one sentence>"}
```