ROUTING BY GEOMETRY: WEAK-TO-STRONG LLM INFERENCE WITH MAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Weak-to-strong routing reduces inference cost by escalating selected queries to a stronger language model, but effective routing requires identifying which queries are likely to benefit from escalation. We introduce Model-Adaptive Geometric Escalation (MAGE), a white-box router that probes local predictive sensitivity by applying small Gaussian perturbations to an intermediate weak-model hidden state using only forward passes. The resulting perturbation responses provide a geometric signal motivated by Fisher information complementary to output uncertainty. At its core, MAGE combines perturbation-response statistics with margin uncertainty to form a label-free default ranking for escalation. Calibration labels and predictions from both models refine this ranking to reward successful rescues and penalize harmful escalations, while reliability shrinkage limits adjustments when calibration support is weak. Across eight multiple-choice benchmarks, MAGE achieves a higher observed macro-averaged area under the accuracy–budget curve than all five adapted routing baselines on both Qwen and Llama endpoint pairs. These results highlight local predictive sensitivity as a complementary signal for balancing prediction quality and inference cost in weak-to-strong model routing.

1 INTRODUCTION

Language-model endpoints differ substantially in predictive capability and inference cost, while queries differ in how much model capability they require. Model routing exploits this heterogeneity by selecting a model for each query to balance prediction quality and cost (Chen et al., 2024; Ong et al., 2025). In a representative two-endpoint setting, a router operating under a limited strong-model call budget decides whether to retain the weak model's prediction or escalate the query to a stronger model. The key routing challenge is to identify where the strong model is likely to help, not merely where the weak model may be incorrect. Escalation may correct a weak-model error, leave it unresolved, or overturn an otherwise correct prediction. Therefore, effective routing should prioritize queries with high potential escalation benefit.

Existing routers differ in what they observe and when they act. Pre-generation methods estimate model suitability via query features (Ding et al., 2024; Feng et al., 2025) or preference supervision Ong et al. (2025). UniRoute additionally represents models through error profiles on representative prompts, supporting changing model pools (Jitkrittum et al., 2026). By contrast, post-generation cascades inspect the weak model's response before deciding whether to escalate (Chen et al., 2024). Therefore, we ask whether the weak model's response to hidden-state perturbations can complement these signals in identifying beneficial escalations.

Prior work suggests that language-model hidden states encode information related to truthfulness and correctness that is not fully reflected in output confidence (Burns et al., 2023; Azaria & Mitchell, 2023; Marks & Tegmark, 2024). We also observed that queries with similar output uncertainty may respond differently to small hidden-state perturbations, so it may provide a potential routing signal beyond the unperturbed output. This motivates our central question: can local predictive sensitivity to hidden-state perturbations complement output-side signals for budgeted weak-to-strong routing?

To explore this question, we introduce *Model-Adaptive Geometric Escalation* (MAGE), a white-box weak-to-strong router that complements output confidence with local predictive sensitivity. We hypothesize that queries with similar confidence can respond differently to small hidden-state perturbations and that these differences provide additional information about the benefit of escalation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

MAGE measures these differences using forward-only Gaussian probes at a fixed intermediate layer. For each probe, we quantify the resulting predictive change using the Kullback–Leibler (KL) divergence from the unperturbed candidate distribution to the perturbed distribution. This KL response has a local information-geometric interpretation: it is governed by the Fisher–Rao metric pulled back to hidden-state coordinates (Amari, 2016; Martens, 2020). Under suitable regularity conditions, the expected KL response to small isotropic Gaussian perturbations is proportional, to second order, to the trace of this pullback Fisher matrix. This connection motivates estimating local predictive sensitivity from a finite set of forward passes, without explicitly constructing the matrix or computing gradients (Hutchinson, 1990; Avron & Toledo, 2011).

Furthermore, to rank queries for escalation, MAGE combines KL sensitivity motivated by Fisher information, measuring the average shift of perturbed candidate distributions from the unperturbed prediction, with cross-perturbation disagreement, measuring how much these distributions vary across probes. Auxiliary candidate-score statistics describe underlying likelihood variation, while margin uncertainty indicates how closely the leading answers compete. These features form a label-free default ordering that considers both predictive sensitivity and output ambiguity. We call this design *geometry-first* because this ordering anchors calibration-based refinement. To connect these signals to escalation benefit, we construct an adaptive score that combines a selected geometry base, output uncertainty and a semantic component. Calibration labels and both endpoints' predictions guide refinement through an asymmetric utility that rewards rescues and penalizes harmful escalations. Reliability shrinkage limits departures from the label-free default under weak or inconsistent support. The resulting score provides a single ordering and nested escalation sets across strong-model call budgets.

Across eight multiple-choice benchmarks containing 20,288 examples, MAGE exceeds entropy on Qwen in macro-averaged area under the accuracy–budget curve (AUBC; 58.68% v.s. 57.63%), indicating the routing effectiveness of combination of perturbation responses and output uncertainty. At a 10% strong-model call budget, lower harmful-escalation rates (8.86% v.s. 12.98%) and higher accuracy gains over the weak model (2.59 v.s. 1.93 points) than a margin-entropy rank control support benefit-focused routing. In a post-hoc GPQA-Diamond ablation, removing perturbation features or reliability shrinkage lowers AUBC by $6.10/3.14$ points, highlighting the importance of perturbation signals and reliability-controlled adaptation. On Qwen, estimated Transformer FLOPs increase from 31.3% at a 10% strong-model call budget to 71.3% at a 50% budget relative to direct Qwen2.5-72B likelihood scoring. Overall, these results show that MAGE achieves improved routing performance with substantial computational savings relative to direct strong-model scoring.

Our contributions are:

- We introduce forward-only perturbation signals that complement output confidence, with a local Fisher interpretation linking KL responses to predictive sensitivity.
- We develop MAGE, a geometry-first router with utility-guided refinement and reliability shrinkage, limiting adaptation when calibration support is weak.
- Across eight benchmarks and two endpoint pairs, MAGE achieves higher observed macro AUBC than five adapted baselines, with stronger gains on Qwen; ablations and cost analyses characterize component benefits and probing overhead.

2 Background and Related Work

2.1 Model Routing and Selective Escalation

Model-routing methods can be broadly distinguished by whether routing decisions are made before or after obtaining a model response. For example, RouteLLM (Ong et al., 2025) and HybridLLM (Ding et al., 2024) compare the relative response quality of smaller and larger models, while GraphRouter (Feng et al., 2025) and UniRoute (Jitkrittum et al., 2026) select models based on objectives that balance performance and cost. In parallel, model cascading (Chow, 1970) describes how to organize the sequence of model calls. The cascades considered here, operate in a post-generation routing setting, in which a cheaper model first produces a response, after which the system decides whether to invoke another endpoint. For instance, FrugalGPT (Chen et al., 2024) uses learned quality estimators, and Mixture-of-Thought Cascades (Yue et al., 2024) uses agree-

ment across reasoning paths. This setting relates to selective prediction with abstention and learning to defer to a fallible alternative (Geifman & El-Yaniv, 2017; Madras et al., 2018). This raises the question of which reliability signals can guide beneficial escalation.

2.2 Reliability Signals for Model Routing

Output-side quantities such as likelihood, margin, and entropy provide signals of predictive reliability (Guo et al., 2017; Desai & Durrett, 2020), while verbalized confidence (Kadavath et al.) expresses the model's own assessment of its answers (Tian et al., 2023; Xiong et al., 2024). Sampling-based approaches estimate uncertainty across generated outputs. Semantic entropy (Kuhn et al., 2023) and related black-box estimators (Lin et al., 2024) measure semantic dispersion to assess answer reliability and detect hallucinations (Farquhar et al., 2024), while SelfCheckGPT uses cross-sample consistency to identify unreliable generations (Manakul et al., 2023). However, these output-side measures of confidence and predictive instability alone do not directly indicate whether invoking a stronger model will improve the final prediction.

Contrast-consistent search (Burns et al., 2023) and hidden-state probes (Azaria & Mitchell, 2023; Marks & Tegmark, 2024) have also identified accessible truth-related structure. Perturbation-based confidence estimation provides another related direction. (Khanmohammadi et al., 2025) propose CCPS, which applies targeted adversarial perturbations to final hidden states and uses the resulting response features to predict answer correctness. MAGE uses forward-only Gaussian probes at an intermediate layer to obtain a Fisher-motivated sensitivity statistic and cross-trajectory variation, complementing output uncertainty in budget-constrained escalation.

2.3 Local Sensitivity and Information Geometry

Adversarial training searches for loss-increasing perturbations (Goodfellow et al., 2015), while virtual adversarial training regularizes local changes in predictive distributions through KL divergence (Miyato et al., 2019). MAGE uses Gaussian perturbations at a fixed intermediate layer to probe a frozen weak model through batched forward passes, measuring KL divergence from unperturbed to perturbed candidate distributions.

Under suitable regularity conditions, the pullback Fisher–Rao metric governs the local quadratic approximation of this KL response in hidden-state coordinates (Amari, 2016; Martens, 2020). For small isotropic perturbations, the expected response is proportional, to second order, to the pullback trace. The normalized mean of sampled KL responses thus provides a sensitivity proxy with a stochastic trace-estimation interpretation (Hutchinson, 1990; Avron & Toledo, 2011). MAGE combines this statistic with cross-trajectory variation to rank queries for escalation.

3 Methodology

Overview. MAGE probes the weak model with hidden-state perturbations, combining resulting predictive responses and margin uncertainty into a label-free geometry-first score R. Calibration-based estimates of escalation utility yield an adaptive score A, while a reliability weight α controls shrinkage toward R to produce the final ranking for budgeted escalation(Figure 1). Complete details and proofs are provided in Appendix A.

3.1 Budgeted Weak-to-Strong Routing

Let f_{W} and f_{S} denote the weak and strong endpoints, respectively. Input x has $C = C(x) \geq 2$ ordered candidate answer strings. Under endpoint $E \in \{\mathrm{W}, \mathrm{S}\}$, candidate j has tokens $(c_{E,j,1}, \ldots, c_{E,j,T_{E,j}})$. We compute sequence log-likelihoods without length normalization:

$$s_{E,j}(x) = \sum_{t=1}^{T_{E,j}} \log p_E(c_{E,j,t} \mid x, c_{E,j,<t}), \qquad q_{E,j}(x) = \frac{\exp s_{E,j}(x)}{\sum_{m=1}^{C} \exp s_{E,m}(x)}, \tag{1}$$
$$\widehat{y}_E(x) = \arg \max_{j \in \{1,\ldots,C\}} q_{E,j}(x).$$

Here p_E denotes the conditional token distribution, $c_{E,j,<t}$ the preceding candidate tokens, and $\mathbf{q}_E(x)$ the distribution over C candidate answers. Both endpoints receive identical prompts and

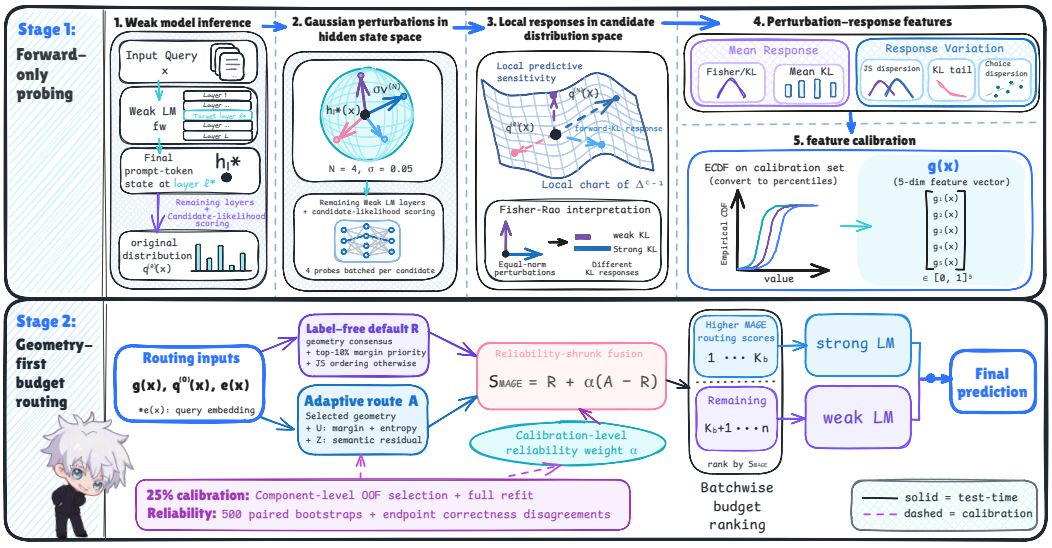


Figure 1: **Overview of MAGE.** Forward-only hidden-state perturbations characterize local predictive responses. Perturbation-response statistics and margin uncertainty define a geometry-first routing score. Calibration-based escalation utility refines this ordering, while reliability shrinkage limits adaptation when calibration support is weak. The highest-scoring queries are escalated under the strong-model call budget.

ordered candidates, although tokenizations may differ. Summed log-likelihoods can favor shorter candidates with comparable per-token probabilities. Because the strong endpoint is fallible, escalation can either rescue a weak-model error or overturn a correct weak-model prediction. This places weak-to-strong routing in the broader setting of selective prediction (Chow, 1970) and learning to defer to a fallible expert (Madras et al., 2018; Mozannar & Sontag, 2020).

To fit the router and assess its escalation decisions on held-out queries, we use the following calibration and evaluation protocol. For each benchmark and seed, a seeded permutation partitions examples into disjoint calibration and evaluation sets, $\mathcal{I}_{\text{cal}}$ and $\mathcal{I}_{\text{eval}}$, allocating 25% to calibration with rounding and safeguards for nonempty splits (see Appendix A.1). Calibration labels guide fitting, selection, and reliability estimation; all compared routers are assessed on the same evaluation examples, whose labels are used only for assessment. Router parameters are fitted separately for each calibration split, while hyperparameter settings remain fixed across benchmarks and endpoint pairs. This protocol evaluates adaptation within each benchmark distribution, with calibration-size sensitivity examined in Appendix B.7. MAGE operates in a batchwise routing setting: its ranking may use unlabeled within-pool feature ranks, but never evaluation labels. A single ordering of each unlabeled evaluation pool is evaluated at strong-model call fractions $\mathcal{B} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

3.2 Forward-Only Local Predictive Geometry

Hidden-state perturbations. We use isotropic Gaussian perturbations to probe local predictive responses without favoring a particular hidden-state direction. We perturb a fixed intermediate layer selected by relative depth ρ. For L zero-indexed Transformer blocks, we set $\ell^\star = \text{round}(\rho(L-1))$ with default $\rho = 0.78$. Let $\mathbf{h}_{\ell^\star}(x) \in \mathbb{R}^d$ be the final prompt-token state after this block. We apply $N = 4$ independent perturbations:

$$\mathbf{v}^{(k)} \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \qquad \widetilde{\mathbf{h}}^{(k)}_{\ell^\star}(x) = \mathbf{h}_{\ell^\star}(x) + \sigma \mathbf{v}^{(k)}, \qquad k = 1, \ldots, N, \tag{2}$$

where $\mathbf{I}_d$ is the identity matrix and $\sigma = 0.05$ is the per-coordinate standard deviation. For each query, the same perturbation vector $\mathbf{v}^{(k)}$ is reused across all candidate-scoring passes. Batched forward passes produce scores $s_j^{(k)}(x)$ and distributions $\mathbf{q}^{(k)}(x)$, without requiring gradient and Hessian computation. Implementation details are provided in Appendix A.2.

Geometric interpretation. For a fixed query, let $\mathbf{h} \mapsto \mathbf{q}_{\mathbf{h}}$ denote the mapping from the monitored hidden state to the candidate distribution. The pullback Fisher–Rao metric $\boldsymbol{\mathcal{F}}_{\mathbf{h}}$ measures hidden-state displacements by their predictive effect, distinguishing directions that produce different probability changes despite equal displacement norms (Amari, 2016; Martens, 2020). Under the regularity conditions in Appendix A.3, small perturbations satisfy

$$D_{\mathrm{KL}}(\mathbf{q}_{\mathbf{h}} \,\|\, \mathbf{q}_{\mathbf{h}+\Delta\mathbf{h}}) \approx \frac{1}{2}\Delta\mathbf{h}^{\top}\boldsymbol{\mathcal{F}}_{\mathbf{h}}\Delta\mathbf{h}, \qquad \mathbb{E}_{\mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}[D_{\mathrm{KL}}(\mathbf{q}_{\mathbf{h}} \,\|\, \mathbf{q}_{\mathbf{h}+\sigma\mathbf{v}})] \approx \frac{\sigma^2}{2}\,\mathrm{Tr}(\boldsymbol{\mathcal{F}}_{\mathbf{h}})\,. \tag{3}$$

The second relation motivates a forward-pass estimate of the trace by averaging KL responses and scaling by $2/\sigma^2$, without explicitly constructing the Fisher matrix. Writing $\mathbf{q}^{(0)} = \mathbf{q}_{\mathrm{W}}$, we define

$$\kappa_k(x) = D_{\mathrm{KL}}\Big(\mathbf{q}^{(0)}(x) \,\Big\|\, \mathbf{q}^{(k)}(x)\Big)\,, \qquad F_{\mathrm{KL}}(x) = \frac{2}{N\sigma^2}\sum_{k=1}^{N}\kappa_k(x). \tag{4}$$

Thus, F_{KL} provides a finite-probe proxy for average local predictive sensitivity in the monitored coordinates. Calibration-based refinement connects these responses to escalation utility.

Perturbation-response representation. We combine mean sensitivity with cross-probe disagreement, upper-tail response, and auxiliary score variation to distinguish response patterns with similar average KL. Let $\overline{\mathbf{q}} = N^{-1}\sum_k \mathbf{q}^{(k)}$ and $\overline{\kappa} = N^{-1}\sum_k \kappa_k$. Define

$$\begin{gathered} D_{\mathrm{JS}} = \frac{1}{N}\sum_{k=1}^{N} D_{\mathrm{KL}}\Big(\mathbf{q}^{(k)} \,\Big\|\, \overline{\mathbf{q}}\Big)\,, \qquad D_{\mathrm{choice}} = \frac{1}{C}\sum_{j=1}^{C}\mathrm{Var}_{k=1:N}\Big[s_j^{(k)}\Big]\,, \\ \kappa_{\mathrm{tail}} = \max\!\left(0, \frac{\max_k \kappa_k - \mathrm{median}_k\,\kappa_k}{\max(\overline{\kappa}, \varepsilon_{\mathrm{num}})}\right). \end{gathered} \tag{5}$$

where query dependence is omitted, $\varepsilon_{\mathrm{num}} = 10^{-12}$, and variance uses divisor N. We form

$$\begin{aligned} \mathbf{g}_{\mathrm{raw}}(x) &= [F_{\mathrm{KL}}(x), D_{\mathrm{JS}}(x), \kappa_{\mathrm{tail}}(x), \overline{\kappa}(x), D_{\mathrm{choice}}(x)]^{\top}\,, \\ g_r(x) &= \frac{1}{|\mathcal{I}_{\mathrm{cal}}|}\sum_{i\in\mathcal{I}_{\mathrm{cal}}} \mathbf{1}[u_r(x_i) \le u_r(x)]\,, \qquad r = 1,\dots,5, \end{aligned} \tag{6}$$

Here, F_{KL} captures Fisher-motivated local sensitivity, D_{JS} measures disagreement among perturbed distributions, κ_{tail} captures unusually large KL responses relative to the mean, $\overline{\kappa}$ summarizes mean KL response, and D_{choice} measures variation in candidate log-scores. These statistics distinguish queries with similar mean sensitivity but different disagreement or tail behavior. We set $u_r = \log(1 + g_{\mathrm{raw},r})$. The second line applies the calibration empirical cumulative distribution function (ECDF), mapping each coordinate to a calibration percentile without correctness labels and yielding $\mathbf{g}(x) \in [0,1]^5$. We treat average predictive displacement as the core sensitivity signal and use the remaining statistics to characterize variation around that response. At fixed σ, $F_{\mathrm{KL}} = 2\overline{\kappa}/\sigma^2$, so their ECDF ranks coincide in exact arithmetic. Retaining both in the default response aggregate therefore gives mean KL twice the weight of each individual variation statistic (Appendix A.5).

3.3 Reliability-Gated Geometry-First Routing

Label-free default. The geometry-first score combines two calibration-ranked bases. The first summarizes perturbation responses as $G_H(x) = \mathrm{Rank}_{\mathrm{cal}}[H](x)$, where $H(x) = \frac{1}{5}\sum_{j=1}^{5} g_j(x)$. The second, $G_{2\mathrm{st}}(x)$, is the calibration-ranked score of a two-stage ordering that promotes the top 10% of queries by margin uncertainty and ranks the remaining queries by D_{JS}. Margin uncertainty is $1 - (q_{(1)}^{(0)} - q_{(2)}^{(0)})$, where $q_{(1)}^{(0)}$ and $q_{(2)}^{(0)}$ are the two largest candidate probabilities. We define

$$G_{\mathrm{rob}}(x) = \frac{G_H(x) + G_{2\mathrm{st}}(x)}{2}, \qquad R(x) = \mathrm{Rank}_{\mathrm{cal}}[G_{\mathrm{rob}}](x) \in [0,1]. \tag{7}$$

Here $\mathrm{Rank}_{\mathrm{cal}}$ maps a score to its empirical percentile among calibration scores. Both bases use scores without correctness labels, and the two-stage ordering uses unlabeled within-pool ranks. Thus, R provides a label-free default for calibration-based refinement; implementation details appear in Appendix A.6.

From responses to escalation utility. Let $a_E(x)$ denote endpoint correctness. We define the per-example escalation utility $\eta(x)$ as +1 for a rescue, 0 when the two endpoints have the same correctness outcome, and $-\gamma$ for a harmful escalation. For a routing representation $\phi(x)$, the conditional expected utility is

$$\eta_\gamma(\phi) = \Pr(a_{\mathrm{W}} = 0, a_{\mathrm{S}} = 1 \mid \phi) - \gamma \Pr(a_{\mathrm{W}} = 1, a_{\mathrm{S}} = 0 \mid \phi). \tag{8}$$

At a fixed number of escalations to the strong endpoint, ordering queries by their conditional expected utilities maximizes expected total escalation utility. This provides the decision target for learning from perturbation responses, without assuming a monotone relationship between perturbation sensitivity and weak-model error. We fix $\gamma = 1.25$ to penalize harmful escalation 25% more heavily than successful rescue; $\gamma = 1$ corresponds to net accuracy gain.

Calibration-based refinement. The adaptive score is defined as $A(x) = \mathrm{Rank}_{\mathrm{cal}}[G^+_{\mathrm{sel}}(x) + w_U U(x) + w_Z Z(x)]$, where $\mathrm{Rank}_{\mathrm{cal}}$ is the scalar calibration ECDF. The response-based term G^+_{sel} is selected from the learned and label-free base scores after an order-preserving upper-tail spacing transform. The transform uses fixed strength $\lambda_{\mathrm{anchor}} = 0.5$ to control the added score amplitude and temperature $\tau_{\mathrm{anchor}} = 0.08$ to control the transition width. The score U averages calibration-ranked margin uncertainty and candidate entropy normalized by $\log C$; Z is a calibration-ranked semantic residual score learned from fixed category-and-question embeddings to predict residual escalation utility, defined as the observed utility η minus the geometry learner's OOF prediction. We select $w_U, w_Z \in \{0, 0.25, 0.5, 0.75, 1\}$, a coarse nonnegative grid that includes disabling either component.

We train a bootstrap ensemble of ridge regressors with bounded pairwise-ranking corrections on calibration data $(\mathbf{g}(x_i), \eta_i)$. The ensemble-averaged utility predictions are mapped to calibration percentiles to obtain the learned geometry base (Appendix A.7.2). Semantic out-of-fold (OOF) predictions use label-independent outer folds. Within each outer training set, semantic residual targets are constructed from geometry OOF predictions obtained using only the outer-training data. This ensures that labels from the held-out outer fold are excluded from both residual construction and semantic fitting.

We select the base and additive weights by maximizing complexity-penalized, budget-averaged calibration utility, using OOF predictions for learned components and direct scores for label-free components. We denote the resulting calibration-stage adaptive score by A^{sel}. After selection, learned components are refit on all calibration data to obtain the evaluation-time adaptive score $A(x)$. Component-level cross-fitting does not independently validate the subsequent ranking, selection, or reliability estimation, which reuse calibration data (see Appendix A.7).

Reliability shrinkage. Let n_{event} count calibration examples with differing endpoint correctness. With calibration scores fixed, p_+ is the fraction of $M = 500$ paired bootstrap resamples in which the calibration-stage adaptive score A^{sel} achieves higher budget-averaged utility than the default score R. Both orderings are evaluated on the same resampled occurrences. We set

$$\alpha = \frac{n_{\mathrm{event}}}{n_{\mathrm{event}} + 200} \,\mathrm{clip}\left(\frac{p_+ - 0.80}{0.95 - 0.80}, 0, 1\right), \quad S_{\mathrm{MAGE}}(x) = (1 - \alpha)R(x) + \alpha A(x). \tag{9}$$

Here $\mathrm{clip}(u, 0, 1) = \min\big(1, \max(0, u)\big)$. The event factor reaches $1/2$ at 200 disagreements, shrinking adaptation when informative calibration events are scarce. The support factor disables adaptation at $p_+ \le 0.80$ and saturates at $p_+ \ge 0.95$. These reliability controls are fixed regularization settings, and p_+ is a calibration support measure rather than a test p-value. The shared $\alpha \in [0, 1)$ recovers R at zero and bounds score adaptation by $|S_{\mathrm{MAGE}} - R| \le \alpha$ (Appendix A.8).

3.4 Budgeted Inference

For $n_{\mathrm{eval}} = |\mathcal{I}_{\mathrm{eval}}|$, set $K_b = \mathrm{round}(b n_{\mathrm{eval}})$. Sort evaluation indices once by decreasing $S_{\mathrm{MAGE}}(x_i)$ with stable ties, and let $\mathcal{R}_b$ contain the first K_b indices. Then

$$z_b(x_i) = \mathbf{1}[i \in \mathcal{R}_b], \qquad \widehat{y}_b(x_i) = \begin{cases} \widehat{y}_{\mathrm{W}}(x_i), & z_b(x_i) = 0, \\ \widehat{y}_{\mathrm{S}}(x_i), & z_b(x_i) = 1. \end{cases} \tag{10}$$

The escalation sets contain exactly K_b queries and are nested across budgets because they share one ordering (Appendix A.9).

MAGE requires white-box access to the weak model and four additional perturbation replicas per candidate-scoring pass. Inference cost includes weak-model scoring, perturbations, semantic encoding, router scoring, and routed strong-model calls. Calibration additionally requires labeled examples scored by both endpoints, calibration response extraction, and router fitting. This setup cost is incurred per calibration fit and can be amortized over subsequent queries; strong-model call rate describes only the routed inference component.

4 RESULTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate weak-to-strong routing on eight multiple-choice benchmarks: MMLU-Pro (Wang et al., 2024), ARC-Challenge (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024), CommonsenseQA (Talmor et al., 2019), MuSR (Sprague et al., 2024), MedQA (Jin et al., 2021), MathQA (Amini et al., 2019), and LogiQA (Liu et al., 2020). The selected splits comprise 20,288 examples spanning general and expert knowledge, science, commonsense, medicine, mathematics, and logic.

Models and calibration. We evaluate Qwen2.5-3B-Instruct→Qwen2.5-72B-Instruct as the primary endpoint pair and repeat the evaluation with Llama-3.2-3B-Instruct→Llama-3.1-70B-Instruct using the same default hyperparameters (Table 3). Within each pair, methods share prompts, candidate order, cached endpoint predictions, and data splits. For each benchmark and seed (42–46), approximately 25% of examples form the calibration set, with the remainder reserved for evaluation. MAGE is fitted separately on each calibration split using nested semantic OOF (Appendix A.7). Calibration labels are used for fitting, selection, and reliability estimation, whereas evaluation labels are used only for assessment and oracle diagnostics.

Routing budgets. For each $b \in \mathcal{B} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, each routing method escalates its top $K_b = \mathrm{round}(b|\mathcal{I}_{\mathrm{eval}}|)$ scored queries. This batchwise protocol compares routing orderings at matched strong-model call counts. Computation cost is assessed separately.

Baseline adaptations. We compare five adapted routers: RouteLLM-BERT (Ong et al., 2025), FrugalGPT (Chen et al., 2024), HybridLLM (Ding et al., 2024), GraphRouter (Feng et al., 2025), and UniRoute (Jitkrittum et al., 2026). Random, Always-Weak, Always-Strong, and a label-dependent Oracle serve as references. For example i, Random has expected correctness $(1-b)a_{\mathrm{W},i} + ba_{\mathrm{S},i}$ at nominal budget b, where $a_{E,i}$ denotes correctness under endpoint E. Detailed baseline adaptations are described in Appendix B.3.

Metrics and statistical analysis. We report accuracy averaged equally over benchmarks and routing seeds. We use the normalized trapezoidal area under the five-point accuracy–budget curve as our primary metric:

$$\mathrm{AUBC}(m) = \sum_{r=1}^{5} w_r \,\mathrm{Acc}_m(b_r), \qquad (w_1, \ldots, w_5) = (0.125, 0.25, 0.25, 0.25, 0.125), \tag{11}$$

where $b_r = 0.1r$ and $\mathrm{Acc}_m(b_r)$ denotes macro accuracy. Accuracy and AUBC are expressed as percentages, with differences reported in percentage points. For macro-AUBC inference, we use an example-clustered bootstrap to obtain 95% confidence intervals and Holm-adjusted tests across the planned comparisons. Budget-specific and benchmark-level comparisons are descriptive.

4.2 ROUTING EFFECTIVENESS

Table 1 shows that MAGE achieves the highest macro accuracy among the five adapted routers at every tested budget on both endpoint pairs. On Qwen, its gains over the strongest adapted router at each budget range from 1.14 to 2.47 pp, yielding a macro AUBC of 58.68%. On Llama, MAGE reaches a macro AUBC of 51.91%, compared with 51.31% for UniRoute and 50.81% for GraphRouter.

Table 1: **Routing results across model families.** Macro accuracy (%) is averaged equally over eight benchmarks and five routing seeds. AUBC summarizes performance across the five call budgets. Boldface indicates the highest value within each model-pair block.

Model pair	Method	10%	20%	30%	40%	50%	AUBC
Qwen2.5-3B → Qwen2.5-72B	RouteLLM-BERT (Ong et al., 2025)	53.89	54.77	55.85	56.89	58.36	55.91
	FrugalGPT (Chen et al., 2024)	53.68	54.87	56.13	57.31	58.69	56.12
	HybridLLM (Ding et al., 2024)	53.56	54.93	56.20	57.55	58.82	56.22
	GraphRouter (Feng et al., 2025)	53.74	55.13	56.30	57.70	59.23	56.40
	UniRoute (Jitkrittum et al., 2026)	53.74	55.15	56.44	57.83	59.22	56.47
	Random	53.75	55.06	56.37	57.67	58.98	56.37
	MAGE (ours)	**55.03**	**57.23**	**58.89**	**60.30**	**61.55**	**58.68**
Llama-3.2-3B → Llama-3.1-70B	RouteLLM-BERT (Ong et al., 2025)	46.19	47.63	49.61	51.45	53.46	49.63
	FrugalGPT (Chen et al., 2024)	46.31	48.12	50.23	52.15	54.12	50.18
	HybridLLM (Ding et al., 2024)	46.64	48.59	50.50	52.41	54.24	50.49
	GraphRouter (Feng et al., 2025)	46.73	48.80	50.93	52.78	54.73	50.81
	UniRoute (Jitkrittum et al., 2026)	46.99	49.23	51.52	53.41	55.19	51.31
	Random	46.53	48.44	50.35	52.26	54.18	50.35
	MAGE (ours)	**47.36**	**49.89**	**51.78**	**54.05**	**56.46**	**51.91**

Table 2: **Per-benchmark AUBC across model families.** Entries report AUBC (%). Values are averaged over five routing seeds. Boldface indicates the highest value within each model-pair block.

Model pair	Method	MMLU-Pro	ARC-C	GPQA	CSQA	MuSR	MedQA	MathQA	LogiQA	Macro
Qwen2.5-3B → Qwen2.5-72B	RouteLLM-BERT (Ong et al., 2025)	39.32	87.35	35.83	82.73	54.60	58.41	38.43	50.60	55.91
	FrugalGPT (Chen et al., 2024)	40.01	86.98	35.88	82.17	55.32	58.80	38.74	51.09	56.12
	HybridLLM (Ding et al., 2024)	40.24	87.45	35.91	82.28	55.08	58.77	38.88	51.13	56.22
	GraphRouter (Feng et al., 2025)	41.73	87.19	35.96	82.23	54.68	58.57	39.57	51.29	56.40
	UniRoute (Jitkrittum et al., 2026)	42.58	86.67	35.05	82.45	56.01	57.86	39.43	51.74	56.47
	Random	41.04	86.92	35.72	82.41	55.15	58.81	39.12	51.76	56.37
	MAGE (ours)	**43.26**	**91.21**	**39.97**	**84.91**	**57.06**	**61.08**	**39.98**	**51.98**	**58.68**
Llama-3.2-3B → Llama-3.1-70B	RouteLLM-BERT (Ong et al., 2025)	33.16	79.58	29.73	71.95	47.48	62.95	32.39	39.78	49.63
	FrugalGPT (Chen et al., 2024)	33.28	79.14	29.26	71.73	50.74	62.39	33.03	41.85	50.18
	HybridLLM (Ding et al., 2024)	33.18	78.67	30.57	71.62	53.37	62.42	32.85	41.21	50.49
	GraphRouter (Feng et al., 2025)	36.39	78.47	28.83	72.40	51.94	62.57	**33.85**	42.03	50.81
	UniRoute (Jitkrittum et al., 2026)	**37.29**	78.79	30.15	71.89	**53.49**	62.06	33.27	**43.53**	51.31
	Random	35.23	78.20	29.97	71.60	50.01	62.32	33.10	42.39	50.35
	MAGE (ours)	37.06	**81.72**	**32.64**	**73.07**	50.67	**64.88**	32.33	42.89	**51.91**

Primary AUBC inference. On Qwen, MAGE achieves positive macro-AUBC gains over all five adapted baselines, with the 95% bootstrap interval for each comparison excluding zero. Against UniRoute, the strongest adapted baseline, the gain is 2.20 pp with a 95% interval of $[1.54, 2.85]$ pp. On Llama, the gain over UniRoute is 0.60 pp, with a 95% interval of $[-0.12, 1.31]$ pp, which includes zero.

Benchmark-level heterogeneity. Table 2 shows that MAGE has the highest observed AUBC among the five adapted routers on every Qwen benchmark. Gains over the strongest baseline are largest on GPQA ($+4.00$ pp) and ARC-Challenge ($+3.75$ pp), and smallest on LogiQA ($+0.24$ pp) and MathQA ($+0.41$ pp). On Llama, MAGE attains the highest AUBC on ARC-Challenge, GPQA, CommonsenseQA, and MedQA. UniRoute is highest on MMLU-Pro, MuSR, and LogiQA, while GraphRouter is highest on MathQA. Macro averaging weights all benchmarks equally.

4.3 Understanding the Routing Signal

Beneficial escalation. A post-hoc, descriptive analysis on Qwen compares MAGE with the uncertainty control U, which averages calibration ranks of margin uncertainty and normalized entropy (Appendix Table 9). At the 10% call budget, macro accuracy gains over the weak endpoint are 2.59 pp for MAGE and 1.93 pp for U, with macro harm rates among escalated queries of 8.86% and 12.98%, respectively. Harm denotes replacing a correct weak-model prediction with an incorrect strong-model prediction. As a complementary uncertainty comparison, MAGE exceeds the candidate-entropy control by 1.054 pp in macro AUBC on Qwen, while the two are nearly identical on Llama ($+0.008$ pp; Appendix B.5).

Component analysis. Figure 2 presents a GPQA-Diamond case study. Removing perturbation-derived features reduces AUBC by 6.098 pp, while using the unshrunk adaptive score A ($\alpha = 1$) reduces it by 3.142 pp. The label-free default R nearly matches full MAGE, with an AUBC difference of -0.034 pp. The default R nearly matches full MAGE, while removing reliability shrinkage substantially reduces AUBC; the remaining fusion changes have comparatively small observed effects in this case study. With 148 evaluation examples per seed, these component comparisons are descriptive. Detailed component and control comparisons are reported in Appendix B.6.

Sensitivity and Stability. Across ten perturbation repeats on the three-benchmark Qwen diagnostic subset (routing seed fixed at 42), four trajectories achieve the highest mean AUBC among the tested trajectory counts. Increasing the trajectory count to eight improves agreement without improving AUBC. With four trajectories, mean routed-set Jaccard agreement at the 10% budget is 0.270, showing that individual routing decisions vary across repeats even when aggregate AUBC remains relatively stable. Under the fixed-evaluation calibration-size protocol, reducing the calibration fraction from 25% to 5% changes macro AUBC by -0.08 pp. Across the additional routing-hyperparameter sweeps, the absolute change in macro AUBC remains below 0.024 pp. Full results are reported in Appendix Tables 6, 11, 7, and 8, and Figure 4.

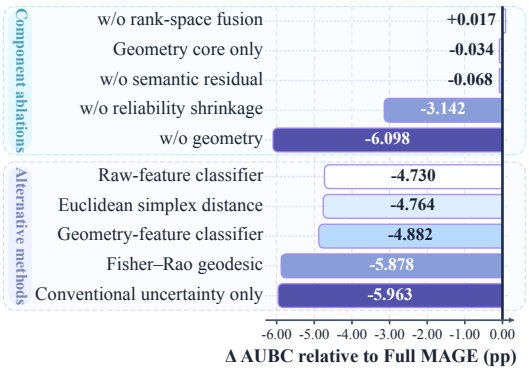


Figure 2: **Component ablations on GPQA-Diamond.** Bars show AUBC differences in percentage points relative to Full MAGE under the nested semantic OOF construction.

4.4 End-to-End Efficiency

MAGE requires white-box weak-model access and four additional perturbation trajectories per candidate-scoring pass. Appendix Table 12 reports nested-OOF accuracies and profile-based Qwen costs, covering weak scoring, perturbations, semantic encoding, amortized router fitting and scoring, and routed strong inference. Estimated wall time ranges from 0.449 to 0.898 seconds per query; these estimates exclude the additional computation required for nested cross-fitting during calibration, as well as model loading and warm-up. Estimated Transformer FLOPs are 31.3%–71.3% of direct Qwen2.5-72B likelihood scoring, excluding non-Transformer router arithmetic.

Appendix Figures 6 and 7 show component costs and accuracy–compute curves under common weak-first accounting, with baseline-router overhead set to zero. The FLOP savings are measured relative to direct strong-model scoring, while the accuracy gains over the five routing adapters are established at matched strong-model call budgets. Native query-only routers may avoid weak inference on escalated queries.

5 Conclusion

Our results suggest that local predictive sensitivity measured through hidden-state perturbations can complement output uncertainty in weak-to-strong routing. Across eight multiple-choice benchmarks and two endpoint pairs, MAGE achieves higher observed macro AUBC than five adapted routing baselines, with stronger gains on Qwen. The GPQA-Diamond case study further indicates a strong geometry-first default and the value of reliability shrinkage in controlling calibration-based adaptation.

AI use statement

In this work, generative AI tools were used for language polishing and improving the clarity of written passages. They were not used to generate experimental results, fabricate data, or make research decisions without author verification. All AI-assisted text was carefully reviewed and revised by the

authors, and all technical claims, mathematical formulations, experimental procedures, and results were independently verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer, 2016. doi: 10.1007/978-4-431-55978-8.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2357–2367, 2019. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245/.

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8:1–8:34, 2011. doi: 10.1145/1944345.1944349.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023. doi: 10.18653/v1/2023.findings-emnlp.68.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1007/BF00058655.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005. doi: 10.1145/1102351.1102363.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018. doi: 10.1111/ectj.12097.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003. doi: 10.1002/rsa.10073.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 295–302. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.21.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, 1994. doi: 10.1201/9780429246593.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A graph-based router for LLM selections. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eU39PDsZtT.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6572.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ka82fvJ5f1.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Dario Ringer, Sam andAmodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know.

Reza Khanmohammadi, Erfan Miahi, Mehrsa Mardikoraem, Simerjot Kaur, Ivan Brugere, Charese Smiley, Kundan S Thind, and Mohammad M Ghassemi. Calibrating llm confidence by probing perturbed representation stability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10448–10514, 2025. doi: 10.18653/v1/2025.emnlp-main.530.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=DWkJCSxKU5.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3622–3628, 2020. doi: 10.24963/ijcai.2020/501.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 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*, pp. 9004–9017, 2023. doi: 10.18653/v1/2023.emnlp-main.557.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL https://jmlr.org/papers/v21/17-678.html.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. doi: 10.1109/TPAMI.2018.2858821.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 2020.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2311.12022.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.16049.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4149–4158, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023. doi: 10.18653/v1/2023.emnlp-main.330.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788, 2020.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01574.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024.

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

A Methodology Details

This appendix provides the complete specification and supporting proofs for the MAGE pipeline in Section 3, covering local predictive geometry, calibration-based adaptation, and budgeted inference. Numerical controls are fixed, while router parameters are fitted separately on each calibration split. Calibration labels guide fitting, selection, and reliability estimation; evaluation labels are used only for assessment.

Hyperparameter configuration. Table 3 summarizes the default settings, shared across benchmarks and endpoint pairs; learned component parameters, adaptive weights, and the reliability weight α are estimated separately on each calibration split. The harm penalty $\gamma = 1.25$ makes the penalty magnitude 25% larger than the unit rescue reward. The order-preserving anchor adjusts score spacing in adaptive combinations and cancels in R (Lemma 1), while the weight grid and complexity penalty control calibration-based additions.

A.1 Problem Setup and Notation

Input x has $C = C(x) \geq 2$ ordered candidate strings. Under endpoint $E \in \{\mathrm{W}, \mathrm{S}\}$, candidate j has tokens $(c_{E,j,1}, \ldots, c_{E,j,T_{E,j}})$. Define

$$\begin{aligned} s_{E,j}(x) &= \sum_{t=1}^{T_{E,j}} \log p_E(c_{E,j,t} \mid x, c_{E,j,<t}), \\ q_{E,j}(x) &= \frac{\exp s_{E,j}(x)}{\sum_{m=1}^{C} \exp s_{E,m}(x)}, \qquad \widehat{y}_E(x) = \arg\max_j q_{E,j}(x). \end{aligned} \tag{12}$$

Scores are computed without length normalization. Both endpoints receive the same prompts and ordered candidates, with model-specific tokenizations (Ding et al., 2024; Ong et al., 2025). Either endpoint can make prediction errors, motivating routing to a fallible expert (Madras et al., 2018; Mozannar & Sontag, 2020). Superscripts (0) and (k) denote unperturbed and perturbed weak-model quantities, respectively:

$$q^{(0)}(x) = [q_{\mathrm{W},1}(x), \ldots, q_{\mathrm{W},C}(x)]^\top \in \Delta^{C-1}. \tag{13}$$

Here Δ^{C-1} is the probability simplex over the C candidate answers. We use natural logarithms and round-to-nearest with ties to even; score ties are resolved by the original candidate or example order.

For each benchmark, a seeded permutation defines calibration and evaluation pools of sizes

$$n_{\mathrm{cal}} = \min(n_{\mathrm{all}} - 1, \max(1, \mathrm{round}(0.25 n_{\mathrm{all}}))), \qquad n_{\mathrm{eval}} = n_{\mathrm{all}} - n_{\mathrm{cal}}. \tag{14}$$

The first n_{cal} permuted indices form $\mathcal{I}_{\mathrm{cal}}$, and the remainder form $\mathcal{I}_{\mathrm{eval}}$; both pools retain the original index order. All experimental calibration splits contain at least three examples, supporting nested semantic cross-fitting. The budget grid is

$$\mathcal{B} = \{0.1, 0.2, 0.3, 0.4, 0.5\}, \qquad b_{\min} = 0.1. \tag{15}$$

For an ordered pool $\mathcal{P}$ and $b \in [0, 1]$, $\mathrm{Top}_b(v; \mathcal{P})$ selects the first $\mathrm{round}(b|\mathcal{P}|)$ indices after stable decreasing sorting by v.

Table 3: **Hyperparameters and implementation controls.** Default settings are shared across benchmarks and endpoint pairs. Adaptive weights are selected separately on each calibration split.

Parameter or control	Description and default setting
Calibration and perturbation extraction	
Calibration fraction	Fraction of examples used for calibration: 25%.
$\mathcal{B}$	Strong-model call budgets: $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.
ρ	Relative intervention depth: 0.78, with $\ell^\star = \mathrm{round}(\rho(L-1))$.
N	Number of Gaussian perturbation trajectories per query: 4.
σ	Per-coordinate perturbation standard deviation: 0.05.
Default ordering and adaptive selection	
Promotion fraction	Fraction promoted by margin uncertainty in the two-stage ordering: 0.10.
Default base weights	Weights assigned to G_H and $G_{2\mathrm{st}}$: $(1/2, 1/2)$.
$\lambda_{\mathrm{anchor}}$	Amplitude of the order-preserving anchor adjustment: 0.50.
τ_{anchor}	Anchor sigmoid temperature: 0.08, centered at rank 0.90.
γ	Harm penalty: 1.25, relative to a unit rescue reward.
$\mathcal{W}$	Candidate values for each adaptive weight w_U, w_Z: $\{0, 0.25, 0.5, 0.75, 1\}$.
λ_c	Coefficient penalizing $w_U + w_Z$ during calibration selection: 0.002.
Component learning and semantic representation	
Ridge penalties	Geometry: 8; semantic: 32, under summed squared-error loss with unpenalized intercepts.
c	Pairwise inverse regularization: 0.25, with penalty $\|\mathbf{w}\|_2^2/(2c)$.
Pairwise correction amplitude	Maximum magnitude of the additive $\tanh$ correction: 0.20.
Sampled pair limit	At most 4096 retained pairs per ensemble member before sign augmentation; repeats are allowed.
Pairwise optimization	At most 80 updates from zero initialization, with update-norm tolerance 10^{-7}.
Bootstrap ensemble size	Number of members: 12 for full fits and 4 for fold fits.
Cross-fitting folds	At most 5 folds; semantic outer partitions are label-independent.
Semantic encoder	Fixed `all-MiniLM-L6-v2` encoder with unit-normalized output embeddings.
Projection dimension	$\min(d_e, 96)$; Gaussian projection entries have variance $1/96$ when $d_e > 96$.
Projection seed	Fixed random seed for semantic projection: 1729.
Reliability shrinkage	
M	Number of paired calibration bootstrap resamples: 500.
κ_{event}	Event-count scale: 200, giving weight $n_{\mathrm{event}}/(n_{\mathrm{event}} + 200)$.
(p_0, p_1)	Support thresholds: $(0.80, 0.95)$, with linear interpolation from zero to unit support weight.
Numerical controls	
$\varepsilon_{\mathrm{num}}$	Probability and relative-tail denominator floor: 10^{-12}.
Standardization threshold	Feature standard deviations below 10^{-8} are replaced by one.
Sigmoid clipping range	Sigmoid arguments are clipped to $[-30, 30]$.

A.2 Forward-Only Hidden-State Perturbations

For L zero-indexed Transformer blocks, relative depth ρ selects

$$\ell^\star = \mathrm{round}(\rho(L-1)). \tag{16}$$

Let $h_{\ell^\star}(x) \in \mathbb{R}^d$ be the final-prompt state after this block. Sample

$$v^{(k)} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, I_d), \qquad \widetilde{h}_{\ell^\star}^{(k)}(x) = h_{\ell^\star}(x) + \sigma v^{(k)}, \quad k = 1, \ldots, N. \tag{17}$$

We perturb the final prompt-token state at the intervention block, keep the other positions unchanged, and recompute downstream layers. Each query–trajectory vector is shared across candidate-scoring passes, yielding

$$q_j^{(k)}(x) = \frac{\exp s_j^{(k)}(x)}{\sum_{m=1}^{C} \exp s_m^{(k)}(x)}. \tag{18}$$

We use fixed settings $(\rho, N, \sigma) = (0.78, 4, 0.05)$, where σ is the per-coordinate standard deviation and $\mathbb{E}\|\sigma v^{(k)}\|_2^2 = d\sigma^2$. Perturbation replicas are batched within candidate-scoring passes, with model parameters held fixed and perturbation directions sampled from an isotropic Gaussian.

A.3 Local Fisher Interpretation

Fix the query and the other states at the intervention block. Assume that the predictive map $h \mapsto q_h$ has positive probabilities and is twice continuously differentiable near h. Let $J_h = \partial q_h / \partial h$ and

define

$$D_{\mathrm{KL}}(p\|q) = \sum_j p_j \log(p_j/q_j),$$
$$\mathcal{F}_h = \sum_j q_h(j)\nabla_h \log q_h(j) \nabla_h \log q_h(j)^\top = J_h^\top \operatorname{diag}(q_h)^{-1} J_h. \tag{19}$$

The matrix $\mathcal{F}_h$ is the Fisher–Rao pullback through the predictive map (Amari, 2016; Martens, 2020). It is positive semidefinite by construction, and the normalization constraint $\mathbf{1}^\top J_h = 0$ gives $\operatorname{rank}(\mathcal{F}_h) \le \min(d, C-1)$. Its null space consists of hidden-state directions that leave the candidate distribution unchanged to first order.

Proposition 1 (local KL expansion). Under these assumptions,

$$D_{\mathrm{KL}}(q_h\|q_{h+\Delta}) = \tfrac{1}{2}\Delta^\top \mathcal{F}_h \Delta + o(\|\Delta\|_2^2), \qquad \Delta \to 0. \tag{20}$$

Proof. For $L(\Delta) = D_{\mathrm{KL}}(q_h\|q_{h+\Delta})$, normalization gives $L(0) = 0$ and $\nabla L(0) = -\sum_j \nabla q_h(j) = 0$. Differentiating $\sum_j q_h(j)\nabla \log q_h(j) = 0$ gives

$$\nabla^2 L(0) = -\sum_j q_h(j)\nabla^2 \log q_h(j) = \mathcal{F}_h.$$

Second-order Taylor expansion proves the claim. □

Define

$$\kappa_k(x) = D_{\mathrm{KL}}(q^{(0)}(x)\|q^{(k)}(x)), \qquad F_{\mathrm{KL}}(x) = \frac{2}{N\sigma^2}\sum_{k=1}^{N}\kappa_k(x). \tag{21}$$

For $V \sim \mathcal{N}(0, I_d)$, set

$$\varepsilon_\sigma(V) = \frac{2}{\sigma^2} D_{\mathrm{KL}}(q_h\|q_{h+\sigma V}) - V^\top \mathcal{F}_h V.$$

Assume additionally that $\epsilon_\sigma(V)$ is almost surely defined and converges to zero in L^2 as $\sigma \to 0$. This Gaussian-tail condition does not follow from local smoothness alone.

Proposition 2 (Gaussian trace moments). For fixed $N \ge 1$,

$$\mathbb{E}_v[F_{\mathrm{KL}}] = \operatorname{Tr}(\mathcal{F}_h) + o(1), \qquad \operatorname{Var}_v(F_{\mathrm{KL}}) = \frac{2}{N}\operatorname{Tr}(\mathcal{F}_h^2) + o(1), \quad \sigma \to 0. \tag{22}$$

Proof. For symmetric B and $V \sim \mathcal{N}(0, I_d)$, diagonalization and independence of Gaussian coordinates give

$$\mathbb{E}[V^\top B V] = \operatorname{Tr}(B), \qquad \operatorname{Var}(V^\top B V) = 2\operatorname{Tr}(B^2).$$

Apply these identities to $2\kappa_k/\sigma^2 = (v^{(k)})^\top \mathcal{F}_h v^{(k)} + \varepsilon_\sigma(v^{(k)})$. The assumed L^2 convergence makes the remainder's mean and variance vanish; Cauchy–Schwarz also gives vanishing covariance with the quadratic term. Averaging N independent probes divides the variance by N (Hutchinson, 1990; Avron & Toledo, 2011), proving the stated moments. □

Consequently, $\mathbb{E}[D_{\mathrm{KL}}(q_h\|q_{h+\sigma V})] = \frac{\sigma^2}{2}\operatorname{Tr}(\mathcal{F}_h) + o(\sigma^2)$. This identity connects average KL response to local predictive sensitivity in the monitored coordinates. The practical setting $(N, \sigma) = (4, 0.05)$ gives a finite-sample, finite-scale approximation, with sampling variability controlled by N and approximation accuracy influenced by perturbation scale and numerical precision.

A.4 Interpreting Local Predictive Sensitivity

Let a be the unique top-scoring candidate and $b \ne a$, with margin $m_{ab} = s_a(h) - s_b(h) > 0$ and gradient $g_{ab} = \nabla_h m_{ab}$.

Proposition 3 (affine pairwise crossing). For $m_{ab}^{\mathrm{aff}}(h+\Delta) = m_{ab} + g_{ab}^\top \Delta$, $\sigma > 0$, and $g_{ab} \ne 0$,

$$\Pr[m_{ab}^{\mathrm{aff}}(h+\sigma V) < 0] = \Phi\left(-\frac{m_{ab}}{\sigma\|g_{ab}\|_2}\right), \quad V \sim \mathcal{N}(0, I_d). \tag{23}$$

For $g_{ab} = 0$ the probability is zero; Φ denotes the standard normal CDF.

Proof. When $g_{ab} \neq 0$, the projection $g_{ab}^{\top} V / \|g_{ab}\|_2$ is standard normal, and standardizing the crossing inequality gives the formula. When $g_{ab} = 0$, the margin remains at its positive initial value. □

The formula describes the probability that candidate b overtakes candidate a under the local affine approximation. Becoming the overall winner additionally requires b to exceed every other candidate.

Proposition 4 (predictive sensitivity and correctness). The same predictive maps and sensitivity values can yield opposite correctness orderings under different gold-label assignments.

Proof. Consider $q_{\beta,h}(1) = (1 + \exp[-(\log 9 + \beta h)])^{-1}$ and $q_{\beta,h}(2) = 1 - q_{\beta,h}(1)$. At $h = 0$, every $\beta \geq 0$ predicts candidate one with probability 0.9, while $\mathrm{Tr}(\mathcal{F}_0) = 0.09\beta^2$. The constant map with $\beta = 0$ has $F_{\mathrm{KL}} = 0$, whereas any map with $\beta > 0$ has $F_{\mathrm{KL}} > 0$ almost surely for $\sigma > 0$ and $N \geq 1$ in exact arithmetic. Assigning gold candidate two to the constant map and candidate one to the nonconstant map makes the less sensitive prediction incorrect and the more sensitive prediction correct. Swapping these gold labels reverses the correctness ordering while preserving both predictive maps and their sensitivity values. □

A.5 Geometry Features and Rank Normalization

Let $\overline{q} = N^{-1} \sum_k q^{(k)}$, $\overline{\kappa} = N^{-1} \sum_k \kappa_k$, and $\varepsilon_{\mathrm{num}} = 10^{-12}$. Define generalized JS divergence (Lin, 1991) and response variation by

$$D_{\mathrm{JS}}(x) = \frac{1}{N} \sum_{k=1}^{N} D_{\mathrm{KL}}(q^{(k)}(x) \| \overline{q}(x)), \tag{24}$$

$$\kappa_{\mathrm{tail}}(x) = \max\left\{0, \frac{\max_k \kappa_k(x) - \mathrm{median}_k\, \kappa_k(x)}{\max\{\overline{\kappa}(x), \varepsilon_{\mathrm{num}}\}}\right\}, \tag{25}$$

$$D_{\mathrm{choice}}(x) = \frac{1}{C} \sum_{j=1}^{C} \mathrm{Var}_{k=1:N}[s_j^{(k)}(x)], \tag{26}$$

where variance uses divisor N and an even-sample median averages its middle pair. Then

$$g_{\mathrm{raw}} = [F_{\mathrm{KL}}, D_{\mathrm{JS}}, \kappa_{\mathrm{tail}}, \overline{\kappa}, D_{\mathrm{choice}}]^{\top} \in \mathbb{R}_{+}^{5}. \tag{27}$$

Trajectory-dependent common score shifts preserve the softmax distribution but can alter D_{choice}. Therefore, we treat candidate-score variance as an auxiliary score-space statistic rather than an intrinsic distributional metric.

For a nonempty reference pool $\mathcal{A}$, define the empirical percentile operator

$$\mathrm{Rank}_{\mathcal{A}}[v](x) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathbf{1}[v(x_i) \leq v(x)]. \tag{28}$$

Writing $\mathrm{Rank}_{\mathrm{cal}} = \mathrm{Rank}_{\mathcal{I}_{\mathrm{cal}}}$, we normalize each response coordinate as

$$g_r(x) = \mathrm{Rank}_{\mathrm{cal}}[\log(1 + g_{\mathrm{raw},r})](x), \qquad \mathbf{g}(x) = [g_1(x), \ldots, g_5(x)]^{\top} \in [0,1]^5. \tag{29}$$

The logarithm preserves ordering. At fixed $\sigma > 0$, $F_{\mathrm{KL}} = 2\overline{\kappa}/\sigma^2$, so their ECDF ranks coincide in exact arithmetic. Retaining both gives mean KL sensitivity two of the five equally weighted coordinates in $H(x)$. Evaluation scores within the same calibration ECDF step receive the same rank.

Feature weighting. Let r_{KL} denote the common rank of F_{KL} and $\overline{\kappa}$, and let $r_{\mathrm{JS}}, r_{\mathrm{tail}}, r_{\mathrm{choice}}$ denote the remaining ranked summaries. The equal-weight aggregate is

$$H(x) = \frac{1}{5} \sum_{j=1}^{5} g_j(x) = \frac{2}{5} r_{\mathrm{KL}}(x) + \frac{1}{5} \left[r_{\mathrm{JS}}(x) + r_{\mathrm{tail}}(x) + r_{\mathrm{choice}}(x) \right]. \tag{30}$$

The fixed $2:1:1:1$ weighting prioritizes average predictive displacement as the core sensitivity signal, while the other summaries characterize disagreement, tail behavior, and score variation. This aggregation choice complements the Fisher interpretation of mean KL sensitivity. The expression specifies the weighting in H; learned components use the original five-coordinate representation.

Numerical implementation. We use stable softmax and evaluate $D_\varepsilon(p\|q) = \sum_j p_j^\varepsilon \log(p_j^\varepsilon/q_j^\varepsilon)$ with $p_j^\varepsilon = \mathrm{clip}(p_j, \varepsilon_{\mathrm{num}}, 1)$, $q_j^\varepsilon = \mathrm{clip}(q_j, \varepsilon_{\mathrm{num}}, 1)$, and $\varepsilon_{\mathrm{num}} = 10^{-12}$. Clipped entries are used directly without renormalization. KL responses and averaged JS values are lower-bounded by zero, and entropy uses the same probability floor. These conventions stabilize the numerical evaluation; the Fisher analysis describes the underlying exact KL divergence.

A.6 Label-Free Geometry-First Routing Score

We construct the label-free default from two complementary bases: an aggregate of perturbation responses and a two-stage ordering that combines output ambiguity with distributional disagreement. The response aggregate is

$$H(x) = \frac{1}{5}\sum_{r=1}^{5} g_r(x), \qquad G_H(x) = \mathrm{Rank}_{\mathrm{cal}}[H](x). \tag{31}$$

The second base prioritizes queries whose leading candidates have similar probabilities. Define margin uncertainty as $u_{\mathrm{margin}} = 1 - (q_{(1)}^{(0)} - q_{(2)}^{(0)})$, where $q_{(1)}^{(0)} \geq q_{(2)}^{(0)}$ are the two largest candidate probabilities. For a pool $\mathcal{P}$, let $r_{\mathrm{JS}}^{\mathcal{P}} = \mathrm{Rank}_{\mathcal{P}}[D_{\mathrm{JS}}]$, $r_{\mathrm{margin}}^{\mathcal{P}} = \mathrm{Rank}_{\mathcal{P}}[u_{\mathrm{margin}}]$, and $\mathcal{U}_{\mathcal{P}} = \mathrm{Top}_{b_{\min}}(u_{\mathrm{margin}}; \mathcal{P})$, with $b_{\min} = 0.10$ in the reported budget grid. The two-stage score is

$$T_{\mathcal{P}}(x_i) = \begin{cases} 2 + r_{\mathrm{margin}}^{\mathcal{P}}(x_i), & i \in \mathcal{U}_{\mathcal{P}}, \\ r_{\mathrm{JS}}^{\mathcal{P}}(x_i), & i \notin \mathcal{U}_{\mathcal{P}}. \end{cases} \tag{32}$$

The offset promotes the most uncertain queries above the rest, ordering the promoted group by margin uncertainty and the remaining queries by JS disagreement.

We then map the two-stage score to a calibration percentile and combine the two bases with equal weights. Let $\mathcal{P}_i$ denote the calibration or evaluation pool containing example i:

$$G_{2\mathrm{st}}(x_i) = \frac{1}{n_{\mathrm{cal}}} \sum_{j \in \mathcal{I}_{\mathrm{cal}}} \mathbf{1}[T_{\mathcal{I}_{\mathrm{cal}}}(x_j) \leq T_{\mathcal{P}_i}(x_i)], \quad G_{\mathrm{rob}}(x) = \tfrac{1}{2}\big(G_H(x) + G_{2\mathrm{st}}(x)\big). \tag{33}$$

Calibration scores provide the common ECDF reference, while within-pool ranks incorporate the unlabeled query pool. This construction uses feature values without correctness labels and produces a batchwise default ordering.

To adjust upper-tail score spacing before adaptive fusion, we apply an order-preserving anchor to each base score $G(x) \in [0, 1]$:

$$G^+(x) = G(x) + \lambda_{\mathrm{anchor}} \sigma_\ell\left(\mathrm{clip}\left(\frac{G(x) - (1 - b_{\min})}{\tau_{\mathrm{anchor}}}, -30, 30\right)\right), \tag{34}$$

where $\sigma_\ell(z) = (1 + e^{-z})^{-1}$. We fix $\lambda_{\mathrm{anchor}} = 0.5$ and $\tau_{\mathrm{anchor}} = 0.08$, controlling the adjustment amplitude and transition width around rank $1 - b_{\min} = 0.90$.

Lemma 1 (order preservation under anchoring). For $\lambda_{\mathrm{anchor}} \geq 0$ and $\tau_{\mathrm{anchor}} > 0$, the anchor transformation is strictly increasing and preserves ties.

Proof. The clipped sigmoid adjustment is nondecreasing. For any $u < v$, the difference between the transformed scores is therefore at least $v - u > 0$. Equal inputs receive identical adjustments. □

Applying this transformation to both query and calibration scores preserves every comparison in the calibration ECDF. The label-free default is thus

$$R(x) = \mathrm{Rank}_{\mathrm{cal}}[G_{\mathrm{rob}}^+](x) = \mathrm{Rank}_{\mathrm{cal}}[G_{\mathrm{rob}}](x) \in [0, 1]. \tag{35}$$

The anchor preserves the default ranking while adjusting the relative score spacing used in additive adaptive combinations.

A.7 Calibration-Based Adaptive Routing

The terminal-uncertainty component is

$$u_{\text{entropy}}(x) = -\frac{1}{\log C}\sum_j q_j^{(0)}(x)\log q_j^{(0)}(x),$$
$$U(x) = \tfrac{1}{2}\big(\text{Rank}_{\text{cal}}[u_{\text{margin}}](x) + \text{Rank}_{\text{cal}}[u_{\text{entropy}}](x)\big) \in [0,1]. \tag{36}$$

These ranks are not calibrated correctness probabilities (Guo et al., 2017; Desai & Durrett, 2020).

A.7.1 Asymmetric escalation utility

For calibration labels y_i, let $a_{E,i} = \mathbf{1}[\widehat{y}_E(x_i) = y_i]$ and $d_i = a_{\text{S},i} - a_{\text{W},i}$. Define

$$\eta_i = \begin{cases} d_i, & d_i \geq 0, \\ \gamma d_i, & d_i < 0, \end{cases} \qquad \gamma = 1.25. \tag{37}$$

Rescue, equal correctness, and harm receive 1, 0, and -1.25. Both endpoint predictions are needed on calibration data; acquiring them is outside the evaluation call budget.

Proposition 5 (cost-ratio interpretation). Let $\mathsf{Res} = \{a_{\text{W}} = 0, a_{\text{S}} = 1\}$ and $\mathsf{Harm} = \{a_{\text{W}} = 1, a_{\text{S}} = 0\}$. The expected escalation utility is

$$\mathbb{E}[\eta \mid x] = \Pr(\mathsf{Res} \mid x) - \gamma \Pr(\mathsf{Harm} \mid x). \tag{38}$$

The harm penalty γ therefore sets the rescue-to-harm probability ratio required for positive expected utility: when $\Pr(\mathsf{Harm} \mid x) > 0$, escalation has positive utility exactly when this ratio exceeds γ.

Proof. Write $\eta = \mathbf{1}_{\mathsf{Res}} - \gamma \mathbf{1}_{\mathsf{Harm}}$ and take conditional expectations. Dividing by $\Pr(\mathsf{Harm} \mid x) > 0$ gives the ratio criterion. When $\Pr(\mathsf{Harm} \mid x) = 0$, expected utility equals the rescue probability and is positive whenever rescue is possible. □

A.7.2 Component-level cross-fitting

The shared learner combines ridge regression (Hoerl & Kennard, 1970) with a bounded correction for pairwise ordering (Burges et al., 2005) and averages predictions across bootstrap members (Breiman, 1996). Each member samples the training pool with replacement, retaining its original size, and standardizes features using the resample statistics. Its prediction is

$$\widehat{\beta} = (X^\top X + \lambda I)^{-1} X^\top (t - \bar{t}\mathbf{1}), \quad \ell(x) = \bar{t} + z(x)^\top \widehat{\beta} + 0.20 \tanh(z(x)^\top \widehat{w}), \tag{39}$$

where X and $z(x)$ are the standardized training matrix and query features, and $\bar{t}$ is the resampled target mean. The ridge term estimates target values, while the bounded correction incorporates pairwise ordering information. We use $\lambda = 8$ for geometry and $\lambda = 32$ for semantics. Feature standard deviations below 10^{-8} are replaced by one, and numerically constant targets yield the corresponding constant prediction.

To learn the pairwise correction $\widehat{w}$, each member samples up to 4096 pairs with non-negligible target differences. The standardized feature difference from the higher-target example to the lower-target example receives label one, and its negative receives label zero. For the resulting design matrix $D \in \mathbb{R}^{m \times d_\phi}$ and labels y^{pair}, we initialize $w = 0$ and apply

$$w \leftarrow w - \zeta\left(\frac{D^\top\left[\sigma_\ell(\text{clip}(Dw, -30, 30)) - y^{\text{pair}}\right]}{m} + \frac{w}{c}\right),$$
$$\zeta = \left(\frac{1}{4m}\sum_{j=1}^{m}\|D_{j,:}\|_2^2 + \frac{1}{c}\right)^{-1}, \qquad c = 0.25. \tag{40}$$

Optimization runs for at most 80 updates and stops when the update norm falls below 10^{-7}. When no eligible pair is sampled, we set $\widehat{w} = 0$. Ensemble predictions average 12 members for full fits and 4 for fold fits.

Geometry OOF uses at most five seeded folds. We stratify by target sign when at least two classes are present and each contains at least two examples, capping the fold count at the smallest class size; otherwise, we use seeded unstratified folds. For component features ϕ_i and targets t_i, define

$$\widetilde{t}_i^{\mathrm{OOF}} = \mathcal{L}_{-\nu}(\phi_i), \quad i \in \mathcal{V}_\nu, \qquad \widetilde{t}^{\mathrm{full}}(x) = \mathcal{L}_{\mathrm{all}}(\phi(x)), \tag{41}$$

where $\mathcal{L}_{-\nu}$ performs fitting and bootstrap standardization on the calibration examples outside $\mathcal{V}_\nu$, while $\mathcal{L}_{\mathrm{all}}$ uses the full calibration set. Feature ECDFs use all calibration feature values without using correctness labels. This component-level cross-fitting (Chernozhukov et al., 2018) provides two ranked score versions:

$$Q^{\mathrm{OOF}}(x_i) = \mathrm{Rank}_{\mathrm{cal}}[\widetilde{t}^{\mathrm{OOF}}](x_i), \qquad Q^{\mathrm{full}}(x) = \mathrm{Rank}_{\mathrm{cal}}[\widetilde{t}^{\mathrm{full}}](x). \tag{42}$$

The OOF version uses pooled calibration OOF predictions as its rank reference and supplies learned scores for selection and reliability estimation. The full-fit version uses full-fit calibration predictions as its reference and supplies evaluation scores. Inner geometry OOF follows the same fold construction; outer semantic folds use label-independent partitions.

A.7.3 Adaptive construction

Fitting $(g(x_i), \eta_i)$ gives $\widetilde{G}_{\mathrm{learn}}$ and its ranks G_{learn}. The base identities are

$$\mathfrak{G}^+ = \{G^+_{\mathrm{learn}}, G^+_H, G^+_{\mathrm{2st}}, G^+_{\mathrm{rob}}\}. \tag{43}$$

Let $e_0(x) \in \mathbb{R}^{d_e}$ be the unit-normalized category-and-question embedding from a fixed pretrained `all-MiniLM-L6-v2` sentence encoder (Reimers & Gurevych, 2019; Wang et al., 2020), excluding gold answers and correctness. If $d_e > 96$, set $e = P^\top e_0$ with independent $P_{uv} \sim \mathcal{N}(0, 1/96)$ and seed 1729; otherwise $e = e_0$. This label-independent projection reduces fitting cost (Dasgupta & Gupta, 2003).

For the full semantic fit, we define residual utility targets by subtracting raw geometry OOF predictions from calibration utilities:

$$r_i^{\mathrm{sem}} = \eta_i - \widetilde{G}^{\mathrm{OOF}}_{\mathrm{learn}}(x_i). \tag{44}$$

We fit the semantic learner on $(e(x_i), r_i^{\mathrm{sem}})$ for all $i \in \mathcal{I}_{\mathrm{cal}}$, yielding raw predictions $\widetilde{Z}^{\mathrm{full}}$.

For semantic OOF, a label-independent seeded permutation forms $\min\{5, n_{\mathrm{cal}}\}$ outer folds $\mathcal{V}_\nu^{\mathrm{sem}}$, with training pools $\mathcal{T}_\nu = \mathcal{I}_{\mathrm{cal}} \backslash \mathcal{V}_\nu^{\mathrm{sem}}$. Within each $\mathcal{T}_\nu$, we recompute inner geometry OOF predictions from $(g(x_j), \eta_j)_{j \in \mathcal{T}_\nu}$ and construct fold-specific residual targets:

$$\begin{aligned} r^{\mathrm{sem}}_{j,-\nu} &= \eta_j - \widetilde{G}^{\text{inner-OOF}}_{\mathcal{T}_\nu}(x_j), \quad && j \in \mathcal{T}_\nu, \\ \widetilde{Z}^{\mathrm{OOF}}(x_i) &= \mathcal{L}^{\mathrm{sem}}_{\mathcal{T}_\nu}(e(x_i)), && i \in \mathcal{V}_\nu^{\mathrm{sem}}, \end{aligned} \tag{45}$$

where $\mathcal{L}^{\mathrm{sem}}_{\mathcal{T}_\nu}$ is fitted on $(e(x_j), r^{\mathrm{sem}}_{j,-\nu})_{j \in \mathcal{T}_\nu}$. Both fitting procedures use the corresponding bootstrap ensemble sizes specified above. Equation 42 maps the raw predictions to Z^{OOF} and Z^{full} using their respective calibration references.

Lemma 2 (outer-fold semantic raw-prediction label isolation). Holding features, endpoint predictions, configuration, and random seeds fixed, changing labels in $\mathcal{V}_\nu^{\mathrm{sem}}$ leaves $\widetilde{Z}^{\mathrm{OOF}}(x_i)$ unchanged for every $i \in \mathcal{V}_\nu^{\mathrm{sem}}$.

Proof. With random seeds fixed, the label-independent outer partition leaves $\mathcal{T}_\nu$ unchanged. Inner geometry folds, fits, and residual targets depend on features and utilities within $\mathcal{T}_\nu$. The semantic training data, bootstrap draws, standardization, and fitted predictions on the fixed outer-fold features therefore remain unchanged. □

This establishes fold-wise label isolation for raw semantic OOF predictions. Pooled ECDF ranks can depend on a fold's labels through other folds' predictions. Selection and reliability estimation use calibration data, while the resulting pipeline is assessed on the separate evaluation pool.

Utility-guided selection. For a nonempty ordered calibration index sequence $\mathcal{I}$, let $\mathrm{Top}_b^{\mathrm{util}}(s;\mathcal{I})$ select the $\min\{|\mathcal{I}|, \max\{1, \mathrm{round}(b|\mathcal{I}|)\}\}$ highest-scoring occurrences, counting duplicates separately. Define

$$\mathcal{J}_{\mathcal{I}}(s) = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}}\frac{1}{|\mathcal{I}|}\sum_{i\in\mathrm{Top}_b^{\mathrm{util}}(s;\mathcal{I})}\eta_i. \tag{46}$$

This objective averages whole-pool harm-penalized utility equally across budgets; reported AUBC uses trapezoidal integration.

For $\mathcal{W} = \{0, 0.25, 0.5, 0.75, 1\}$, construct candidate scores

$$S^{\mathrm{sel}}_{G,\omega} = G^{+,\mathrm{sel}} + \omega_U U + \omega_Z Z^{\mathrm{OOF}},$$

where the learned base uses OOF values and label-free bases use direct calibration values. Select

$$(G^+_{\mathrm{sel}}, w_U, w_Z) = \underset{\substack{G^+\in\mathfrak{G}^+\\(\omega_U,\omega_Z)\in\mathcal{W}^2}}{\arg\max}\ \left\{\mathcal{J}_{\mathcal{I}_{\mathrm{cal}}}(S^{\mathrm{sel}}_{G,\omega}) - \lambda_c(\omega_U+\omega_Z)\right\}, \quad \lambda_c = 0.002. \tag{47}$$

Among these 100 candidates, ties are resolved by smaller total weight, larger unpenalized utility, the base priority $G^+_{\mathrm{rob}} > G^+_{\mathrm{learn}} > G^+_H > G^+_{\mathrm{2st}}$, and finally smaller ω_U.

Writing $A^{\mathrm{sel}}_{\mathrm{raw}}$ for the selected calibration score, we define its ranked version and the corresponding full-fit adaptive score:

$$\begin{aligned} A^{\mathrm{sel}}(x_i) &= \mathrm{Rank}_{\mathrm{cal}}[A^{\mathrm{sel}}_{\mathrm{raw}}](x_i), \qquad i\in\mathcal{I}_{\mathrm{cal}},\\ A^{\mathrm{full}}_{\mathrm{raw}}(x) &= G^{+,\mathrm{full}}_{\mathrm{sel}}(x) + w_U U(x) + w_Z Z^{\mathrm{full}}(x),\\ A(x) &= \mathrm{Rank}_{\mathrm{cal}}[A^{\mathrm{full}}_{\mathrm{raw}}](x) \in [0,1]. \end{aligned} \tag{48}$$

Here full denotes full-calibration learned fits and their calibration references; label-free bases retain their pool-dependent definitions. Reliability estimation uses A^{sel}, while inference uses A as the adaptive component of the final shrinkage blend.

Proposition 6 (finite-search comparison). The selected penalized calibration objective is at least that of G^+_{2st} with zero additive weights.

Proof. The zero-weight G^+_{2st} candidate belongs to the finite search and has zero penalty. Maximization therefore gives

$$\mathcal{J}_{\mathcal{I}_{\mathrm{cal}}}(A^{\mathrm{sel}}_{\mathrm{raw}}) - \lambda_c(w_U+w_Z) \geq \mathcal{J}_{\mathcal{I}_{\mathrm{cal}}}(G^+_{\mathrm{2st}}).$$

The proposition concerns the calibration selection objective only. □

A.8 Reliability-Shrunk Fusion

Let R^{cal} denote the default score on calibration examples. We compare it with A^{sel} using paired bootstrap resampling. For each $\mu = 1, \ldots, M$, draw an ordered sequence $\mathcal{I}^{(\mu)}_{\mathrm{cal}}$ of n_{cal} indices with replacement and compute

$$\begin{aligned} \delta^{(\mu)} &= \mathcal{J}_{\mathcal{I}^{(\mu)}_{\mathrm{cal}}}(A^{\mathrm{sel}}) - \mathcal{J}_{\mathcal{I}^{(\mu)}_{\mathrm{cal}}}(R^{\mathrm{cal}}),\\ p_+ &= \frac{1}{M}\sum_{\mu=1}^{M}\mathbf{1}[\delta^{(\mu)} > 0], \qquad M = 500. \end{aligned} \tag{49}$$

Both scores use the same resampled occurrences and multiplicities, with ties resolved by occurrence order; scores, fits, and ECDFs remain fixed. The fraction p_+ summarizes empirical bootstrap support for positive utility gain on the calibration data used for fitting and selection (Efron & Tibshirani, 1994), with resolution $1/M = 0.002$.

We quantify event support by counting calibration examples on which endpoint correctness differs, corresponding to nonzero utility:

$$n_{\mathrm{event}} = \sum_{i\in\mathcal{I}_{\mathrm{cal}}}\mathbf{1}[a_{\mathrm{W},i} \neq a_{\mathrm{S},i}]. \tag{50}$$

Event count and bootstrap support determine two reliability weights. For $\kappa_{\text{event}} > 0$ and $0 \le p_0 < p_1 \le 1$, define

$$w_{\text{event}}(n) = \frac{n}{n + \kappa_{\text{event}}}, \qquad w_{\text{stability}}(p) = \text{clip}\left(\frac{p - p_0}{p_1 - p_0}, 0, 1\right). \tag{51}$$

Proposition 7 (gate properties). The event gate is strictly increasing on $n \ge 0$, takes values in $[0, 1)$, and equals 0 at $n = 0$ and $1/2$ at $n = \kappa_{\text{event}}$. The stability gate equals zero for $p \le p_0$, increases linearly from zero to one on $[p_0, p_1]$, and equals one for $p \ge p_1$.

Proof. The event gate has derivative $\kappa_{\text{event}}/(n + \kappa_{\text{event}})^2 > 0$; its range and stated values follow directly from its definition. The stability properties follow from clipping with $p_1 > p_0$. □

We fix $(\kappa_{\text{event}}, p_0, p_1) = (200, 0.80, 0.95)$ and combine event support with bootstrap stability to determine the adaptive weight:

$$\alpha = w_{\text{event}}(n_{\text{event}}) w_{\text{stability}}(p_+), \qquad S_{\text{MAGE}}(x) = (1 - \alpha)R(x) + \alpha A(x). \tag{52}$$

These parameters control shrinkage, with a single α computed for each calibration split and shared across queries.

Proposition 8 (bounded fusion and label isolation). The construction satisfies $0 \le \alpha < 1$, $S_{\text{MAGE}}(x) \in [0, 1]$, and $|S_{\text{MAGE}}(x) - R(x)| \le \alpha$. Routing is invariant to changes in evaluation labels when inputs, endpoint predictions, calibration labels, configuration, and random draws are fixed.

Proof. Proposition 7 gives $0 \le \alpha < 1$. Since $R, A \in [0, 1]$, their convex combination lies in $[0, 1]$, and $|S_{\text{MAGE}} - R| = \alpha|A - R| \le \alpha$. Feature extraction and within-pool ranking use inputs and endpoint predictions, while fitting, residual construction, selection, and bootstrap reliability use calibration targets. Changing evaluation labels therefore preserves the scores and their stable ordering. □

When $n_{\text{event}} = 0$ or $p_+ \le 0.80$, $\alpha = 0$ and the final score equals R. For $\alpha > 0$, adaptation can reorder queries with close default scores; its effect on routing utility is assessed on held-out data.

A.9 Exact Budgeted Inference

MAGE uses batchwise selective routing based on a global ranking under a fixed strong-model call budget. For $n = n_{\text{eval}}$ and $K_b = \text{round}(bn)$, sort evaluation indices once by decreasing S_{MAGE} with stable ties to obtain π. Define

$$\begin{aligned} \mathcal{R}_b &= \{\pi(t) : 1 \le t \le K_b\}, \qquad z_b(x_i) = \mathbf{1}[i \in \mathcal{R}_b], \\ \widehat{y}_b(x_i) &= \begin{cases} \widehat{y}_{\text{W}}(x_i), & z_b(x_i) = 0, \\ \widehat{y}_{\text{S}}(x_i), & z_b(x_i) = 1. \end{cases} \end{aligned} \tag{53}$$

If $K_b = 0$, the routed set is empty.

Proposition 9 (cardinality and nesting). For every $b \in \mathcal{B}$, $|\mathcal{R}_b| = K_b$ and $|K_b/n - b| \le 1/(2n)$. If $b_1 \le b_2$, then $\mathcal{R}_{b_1} \subseteq \mathcal{R}_{b_2}$.

Proof. Permutation prefixes contain distinct indices. Nearest-integer rounding has error at most $1/2$ and is nondecreasing, giving the stated count bound and nesting of prefixes under a shared ordering. □

To summarize the pipeline, let $\mathcal{E}(x) = (\{(s^{(k)}(x), q^{(k)}(x))\}_{k=0}^{N}, e(x))$, and let Θ_{cal} contain fitted components, calibration references, selected weights, and α. The dependencies are

$$\begin{aligned} \left\{(\mathcal{E}(x_i), \widehat{y}_{\text{W}}(x_i), \widehat{y}_{\text{S}}(x_i), y_i)\right\}_{i \in \mathcal{I}_{\text{cal}}} &\longmapsto \Theta_{\text{cal}}, \\ \left(\{\mathcal{E}(x_i)\}_{i \in \mathcal{I}_{\text{eval}}}, \Theta_{\text{cal}}\right) &\longmapsto \{S_{\text{MAGE}}(x_i)\}_{i \in \mathcal{I}_{\text{eval}}} \longmapsto \{\mathcal{R}_b\}_{b \in \mathcal{B}}. \end{aligned} \tag{54}$$

Table 4: **Qwen endpoint diagnostics on full benchmark pools.** Accuracies are percentages; Gap is in percentage points, computed before rounding.

Dataset	Examples	3B	72B	Oracle	Gap
MMLU-Pro	12,032	35.21	55.04	60.51	19.83
ARC-Challenge	1,172	83.62	94.80	95.99	11.18
GPQA-Diamond	198	32.83	42.42	57.07	9.60
CommonsenseQA	1,221	80.92	86.57	92.22	5.65
MuSR	756	55.03	55.16	72.35	0.13
MedQA	1,273	50.67	77.22	82.48	26.55
MathQA	2,985	32.96	53.20	63.69	20.23
LogiQA	651	48.23	60.06	68.82	11.83
Total	20,288		*Accuracy in %*		

At inference, strong-model predictions are requested for routed queries. Call rate measures strong-model use, while end-to-end cost also includes weak-model scoring and perturbation passes, semantic encoding, and router computation. Estimated Transformer FLOPs cover the Transformer portion of this computation. Calibration fitting, including nested cross-fitting, adds computation on cached features and endpoint outcomes while reusing the extracted perturbation responses. The protocol uses white-box access to the weak model and pool-based ranking of unlabeled queries, with evaluation conducted within each benchmark distribution.

B Experiment Details

This appendix provides a detailed evaluation of MAGE, examining routing performance, component contributions, sensitivity, and computational efficiency, and connecting the local Fisher analysis in Appendix A.3 to observed routing behavior.

B.1 Benchmarks and Endpoint Diagnostics

We use 20,288 examples from the labeled splits of eight multiple-choice benchmarks: MMLU-Pro (Wang et al., 2024), ARC-Challenge (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024), CommonsenseQA (Talmor et al., 2019), MuSR (Sprague et al., 2024), MedQA (Jin et al., 2021), MathQA (Amini et al., 2019), and LogiQA (Liu et al., 2020), partitioned by our calibration/evaluation protocol below.

For correctness indicators $a_{\mathrm{W},i}, a_{\mathrm{S},i} \in \{0,1\}$, define

$$\mathrm{Oracle}(\mathcal{I}) = \frac{100}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \max\{a_{\mathrm{W},i}, a_{\mathrm{S},i}\}, \quad \mathrm{Gap}(\mathcal{I}) = \frac{100}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} (a_{\mathrm{S},i} - a_{\mathrm{W},i}). \tag{55}$$

Oracle is the label-informed accuracy ceiling without a call-budget constraint, reported solely as an evaluation diagnostic; Gap is strong-minus-weak accuracy in percentage points. In Table 4, MuSR combines a $0.13\,\mathrm{pp}$ gap with a 72.35% oracle, illustrating per-example complementarity despite similar endpoint accuracies.

B.2 Frozen Models and Implementation

We use Qwen2.5-3B-Instruct → Qwen2.5-72B-Instruct and repeat the evaluation with Llama-3.2-3B-Instruct → Llama-3.1-70B-Instruct. Methods share examples, prompts, candidate order, and cached endpoint likelihoods without length normalization, but receive adapter-specific inputs.

MAGE uses $(\rho, N, \sigma) = (0.78, 4, 0.05)$ and a fixed `all-MiniLM-L6-v2` sentence encoder (remaining constants: Appendix A). Hyperparameters are shared across benchmarks and endpoint pairs, while components, combination weights, and reliability estimates are fitted per calibration split. Cross-fitting operates at the component level: semantic OOF predictions use label-independent outer folds, with residual targets recomputed through inner geometry OOF fits within each outer training set. Candidate selection and reliability estimation share the resulting calibration scores; evaluation uses full-calibration component fits (Appendix A.7). Evaluation labels are reserved for performance assessment and post-hoc diagnostics.

White-box extraction adds four perturbation trajectories, batched within each weak-model candidate-scoring pass. Inference uses bfloat16 and PyTorch scaled dot-product attention. Weak inference runs on one GPU; strong endpoints use four NVIDIA A800 80 GB GPUs, with loader limits of 76 GiB per GPU and 256 GiB CPU memory.

B.3 Baseline Adaptations

All adapters share a two-endpoint candidate-likelihood task, with fixed endpoint predictions and identical integer strong-call counts at each budget. This protocol compares their query rankings under a common routing task. Target-pair fitting and model selection use the shared calibration split; internal training/validation partitions are formed within that split. Each adapter produces an escalation score, with larger values receiving higher routing priority.

Adapters. RouteLLM-BERT uses the official `bert_gpt4_augmented` checkpoint without target-pair retraining (Ong et al., 2025). It takes the query prompt as input and ranks queries by the predicted strong-model win probability. FrugalGPT uses a scorer that receives the query prompt concatenated with the weak model's selected answer and predicts weak-model correctness. We reserve 55% of the calibration split for validation and use one minus the predicted correctness probability as the escalation score. HybridLLM uses a classifier that predicts weak-model correctness from the query prompt. Its predicted error probability determines escalation priority. We reserve 20% of calibration examples for validation and select the checkpoint with the lowest validation loss. GraphRouter retains the published graph architecture under the Performance First setting, using fixed `all-MiniLM-L6-v2` embeddings and calibration-visible response information (Feng et al., 2025). Training targets select the endpoint with higher correctness, breaking ties in favor of the weak model. We reserve 20% of calibration examples for validation and select the checkpoint with the highest validation macro-F1. Queries are ranked by the predicted strong-minus-weak endpoint score. UniRoute adapts the paper's K-means/error-profile construction to the fixed endpoint pairs, using `all-mpnet-base-v2` in place of the proprietary encoder (Jitkrittum et al., 2026). We select $K \in \{1, 2, 4, 8, 16\}$ by mean calibration routing accuracy across budgets, using leave-one-out cluster error estimates and favoring smaller K in ties. Evaluation queries are assigned to the nearest centroid and ranked by the cluster's estimated weak-minus-strong error rate.

Uncertainty control. *Candidate entropy* computes $-\sum_{j=1}^{C} q_j^{(0)} \log q_j^{(0)}$, with each answer option defining a class. Semantic entropy instead groups free-form responses by meaning (Kuhn et al., 2023; Farquhar et al., 2024). We use unnormalized entropy, so varying candidate counts can affect cross-query rankings.

Adapters are compared at matched strong-call counts; their total computational costs are analyzed in Section B.11.

B.4 Evaluation Metrics and Statistical Analysis

We use routing seeds $\mathcal{S} = \{42, 43, 44, 45, 46\}$ and nominal budgets $\mathcal{B} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Unless otherwise specified, methods share the same 25% calibration split and disjoint evaluation remainder for each benchmark–seed pair, with integer sizes defined in equation 14. Realized strong-call counts follow Appendix A.9.

For held-out accuracy $\mathrm{Acc}_{d,s}(b)$ on benchmark d, seed s, and ordered budgets $b_1 < \cdots < b_5$, normalized trapezoidal integration gives

$$\mathrm{AUBC}_{d,s} = \frac{1}{b_5 - b_1} \sum_{r=1}^{4} \frac{b_{r+1} - b_r}{2} \left[\mathrm{Acc}_{d,s}(b_r) + \mathrm{Acc}_{d,s}(b_{r+1})\right]. \tag{56}$$

The resulting budget weights are $(1/8, 1/4, 1/4, 1/4, 1/8)$. Macro results weight benchmarks and seeds equally. Accuracy and AUBC are reported as percentages; differences are method minus the stated reference, in percentage points.

Primary AUBC comparisons. For macro-AUBC inference, we conduct a conditional example-clustered bootstrap with 10,000 replicates. Within each benchmark, unique held-out example IDs

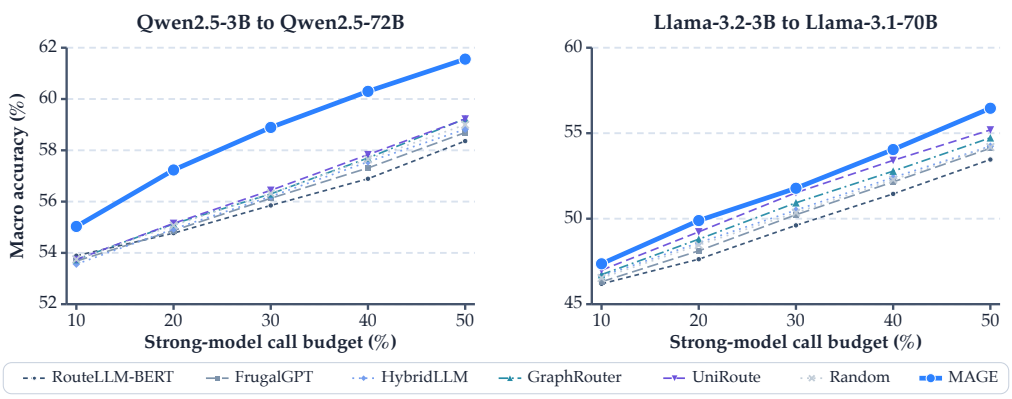


Figure 3: **Accuracy across strong-model call budgets.** Curves show macro accuracy for the Qwen and Llama endpoint pairs.

are resampled with replacement, sharing each ID's multiplicity across its observed seed and method records. Weighted accuracies are aggregated with equal weights across the fixed seeds and eight benchmarks. This estimates held-out example-sampling variability conditional on the fitted routers, realized routed sets, and observed benchmark–seed configurations. We report pointwise basic 95% intervals and approximate two-sided null-centered bootstrap p-values with a plus-one correction. For each endpoint pair, Holm adjustment applies to p-values across ten contrasts, including contrasts not shown in the main tables: the five adapters, candidate entropy, Random, Always-Weak, Always-Strong, and Oracle. The latter three serve as endpoint and ceiling references.

Diagnostic intervals. Calibration-size analyses use 5,000 bootstrap replicates and 95% percentile intervals (Efron & Tibshirani, 1994), resampling benchmarks with replacement and drawing one seed per occurrence. Feature/layer probe intervals use the same benchmark-resampling scheme but draw and average five seeds with replacement per occurrence. Differences are paired by benchmark and seed, and by budget where applicable. These exploratory intervals are reported without multiplicity adjustment. Component ablations, extraction sweeps, frozen-score comparisons, and perturbation-repeat agreement are summarized descriptively, with post-hoc diagnostics examining empirical associations.

AUROC assigns half credit to ties; AUPRC is computed as average precision (AP). Frozen-score AUROC and AP group numerically tied scores using `numpy.isclose`, while probe AP uses stable item order. Undefined metrics are omitted. For nonempty $\mathcal{R}_b$, define

$$\mathrm{HarmRate}(b) = \frac{100 \sum_{i \in \mathcal{R}_b} \mathbf{1}[a_{\mathrm{W},i} = 1, a_{\mathrm{S},i} = 0]}{|\mathcal{R}_b|}, \qquad \mathrm{Gain}(b) = \mathrm{Acc}(b) - \mathrm{Acc}_{\mathrm{W}}\,. \tag{57}$$

Higher AUROC, AUPRC, and gain are better; lower harm rate is better.

B.5 Performance Across Budgets

MAGE achieves the highest macro accuracy among the five displayed adapters at every evaluated budget on both endpoint pairs (Figure 3; Table 1). Candidate entropy scores slightly higher at several Llama budgets.

Uncertainty-control comparison. On Qwen, macro AUBC is 58.679% for MAGE, 56.475% for UniRoute, and 57.625% for candidate entropy. The gain over entropy is $1.054\,\mathrm{pp}$ (basic 95% interval $[0.524, 1.560]\,\mathrm{pp}$; approximate Holm-adjusted $p \approx 0.001$). On Llama, the corresponding values are 51.907%, 51.311%, and 51.899%. Gains over UniRoute and entropy are $0.596\,\mathrm{pp}$ ($[-0.122, 1.309]\,\mathrm{pp}$; adjusted $p \approx 0.207$) and $0.008\,\mathrm{pp}$ ($[-0.829, 0.829]\,\mathrm{pp}$; adjusted $p \approx 0.986$), respectively.

Table 5: **GPQA-Diamond component case study.** Five-seed AUBC using nested semantic OOF; Δ is relative to Full MAGE and computed before rounding. Bold marks the highest AUBC.

Variant	AUBC	Δ
Full MAGE	39.966	0.000
w/o reliability shrinkage	36.824	-3.142
Geometry core only	39.932	-0.034
w/o geometry	33.868	-6.098
w/o semantic residual	39.899	-0.068
w/o rank-space fusion	**39.983**	$+0.017$
Conventional uncertainty only	34.003	-5.963
Raw-feature classifier	35.236	-4.730
Geometry-feature classifier	35.084	-4.882
Euclidean simplex distance	35.203	-4.764
Fisher–Rao geodesic	34.088	-5.878

Table 6: **Prespecified extraction sensitivity.** Three-benchmark AUBC; Δ is relative to the daggered frozen value in each sweep.

Parameter	Value	AUBC	Δ
Target depth	0.50	53.920	-0.629
Target depth	0.65	53.709	-0.840
Target depth	$0.78^{\dagger}$	**54.549**	0.000
Target depth	0.90	53.278	-1.270
Trajectories	1	52.940	-1.609
Trajectories	2	53.310	-1.238
Trajectories	$4^{\dagger}$	**54.549**	0.000
Trajectories	8	54.275	-0.274
σ	0.02	53.907	-0.642
σ	$0.05^{\dagger}$	54.549	0.000
σ	0.10	**54.734**	$+0.185$

B.6 Component Analysis

Table 5 reports a post-hoc GPQA-Diamond case study. *w/o reliability shrinkage* uses the unshrunk adaptive score A by setting $\alpha = 1$, while *Geometry core only* uses the default R, including margin uncertainty (Appendix A.6). *w/o geometry* replaces geometry with uncertainty-based components, retaining semantic learning, selection, and shrinkage; semantic residuals follow the same nested OOF construction with an uncertainty base learner. *w/o semantic residual* fixes the semantic weight at zero and reruns selection and reliability estimation. *w/o rank-space fusion* combines pre-fusion scores using the full method's reliability weight.

Classifier controls fit the shared learner on $\log(1+g_{\mathrm{raw}})$ or its ECDF features. Scalar controls use the same learner with normalized distances from the unperturbed candidate distribution q to uniform:

$$d_{\mathrm{E}}(q) = \frac{\|q - \mathbf{1}/C\|_2}{\sqrt{1 - 1/C}}, \qquad d_{\mathrm{FR}}(q) = \frac{\arccos\left(\sum_j \sqrt{q_j/C}\right)}{\arccos(C^{-1/2})}. \tag{58}$$

Removing geometry lowers AUBC by $6.098\,\mathrm{pp}$. Full MAGE exceeds the unshrunk adaptive score A by $3.142\,\mathrm{pp}$ and the default R by $0.034\,\mathrm{pp}$. Removing the semantic residual and rank-space fusion changes AUBC by $-0.068\,\mathrm{pp}$ and $+0.017\,\mathrm{pp}$, respectively. These descriptive comparisons use 148 evaluation examples per seed.

B.7 Sensitivity Analysis

We vary one parameter at a time, holding the others fixed, using prefixes of at most 1,000 examples each from MMLU-Pro and ARC-Challenge and all 198 GPQA examples (Table 6; Figure 4).

The default depth ($\rho = 0.78$) and trajectory count ($N = 4$) achieve the highest AUBC in their respective sweeps. Relative to the default, eight trajectories change AUBC by $-0.274\,\mathrm{pp}$, while $\sigma = 0.10$ gives $+0.185\,\mathrm{pp}$. We retain the main-experiment default, $\sigma = 0.05$, which probes a smaller neighborhood for the local second-order Fisher approximation.

Calibration-Set Size Sensitivity. Each benchmark–seed permutation fixes a 50% evaluation set. Nested prefixes of the disjoint remainder form calibration sets containing 5%, 10%, 25%, or 50% of the full benchmark. Unlabeled ranking uses all noncalibration examples, while performance is measured on the fixed evaluation subset. Under this protocol, the 25% reference AUBC is 58.59%, compared with 58.679% in the main protocol.

Across the tested 5%–50% range, AUBC changes relative to the 25% reference range from $-0.08\,\mathrm{pp}$ to $+0.03\,\mathrm{pp}$, with all nonreference intervals including zero (Table 7). These results show limited observed sensitivity to calibration-set size.

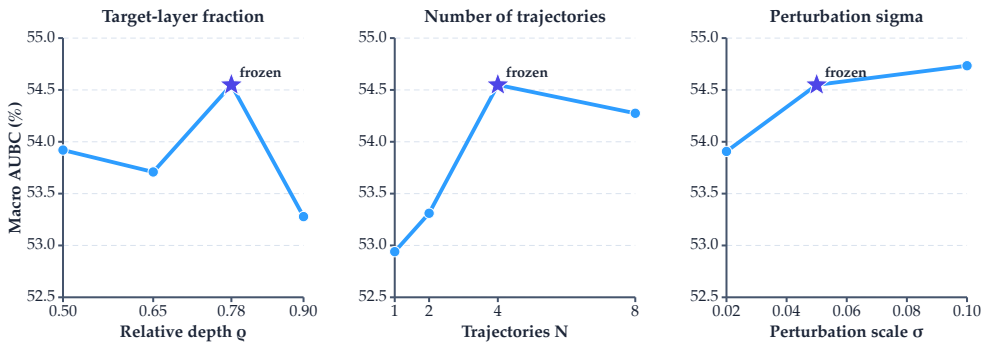


Figure 4: **One-factor-at-a-time sensitivity on the fixed three-benchmark subset.** Stars mark frozen values.

Table 7: **Calibration-size sensitivity on fixed evaluation sets.** Differences and paired exploratory 95% percentile intervals use the 25% reference and the hierarchical bootstrap in Section B.4.

Calibration	Macro AUBC	Δ	CI low	CI high
5%	58.51	−0.08	−0.38	0.27
10%	58.52	−0.07	−0.30	0.11
25%	58.59	0.00	0.00	0.00
50%	58.62	+0.03	−0.11	0.17

B.8 Routing Hyperparameter Sensitivity

We vary six routing hyperparameters: the harm penalty γ, anchor strength λ_{anchor}, anchor temperature τ_{anchor}, event scale κ_{event}, and reliability thresholds p_0, p_1. Post-hoc, one-factor-at-a-time sweeps cover eight Qwen benchmarks and five routing seeds, with 15 configurations including the default. All configurations share perturbation caches, calibration/evaluation splits, and call budgets. Changes to γ refit the learned components with nested semantic OOF; anchor and gate sweeps reuse component predictions and recompute selection and reliability.

Macro AUBC ranges from 58.669% to 58.702%, with all paired changes from the default below 0.024 pp in magnitude (Table 8). Symmetric utility ($\gamma = 1$) yields 58.699%, compared with 58.679% at $\gamma = 1.25$; harm rates show small, nonmonotonic changes with γ across the displayed budgets. The fixed default $\gamma = 1.25$ expresses the chosen rescue–harm trade-off. Removing the anchor changes AUBC by $+0.009$ pp, while varying the event scale and reliability thresholds changes it by at most 0.011 pp. Overall, Qwen performance varies little across the tested settings.

Exploratory 95% percentile intervals use 5,000 hierarchical bootstrap replicates, resampling benchmarks and drawing one routing seed per sampled benchmark occurrence, with settings paired within each draw. Fits remain fixed, and intervals are reported without multiplicity adjustment.

B.9 Post-Hoc Routing Diagnostics

We group held-out outcomes by $(a_{\text{W}}, a_{\text{S}})$ into rescue $(0,1)$, harm $(1,0)$, both wrong $(0,0)$, and both correct $(1,1)$. Pooling Qwen example–seed observations gives mean MAGE scores of 0.574, 0.532, 0.522, and 0.421, respectively.

Score and endpoint benefit. Within each benchmark and seed, stable score ordering forms ten near-equal deciles $\mathcal{Q}_k$. Their empirical endpoint benefit is

$$B_k = \frac{100}{|\mathcal{Q}_k|} \sum_{i \in \mathcal{Q}_k} (a_{\text{S},i} - a_{\text{W},i}). \tag{59}$$

Positive benefit indicates more rescues than harms. This ranking diagnostic shows an overall increase in macro-averaged benefit with score decile, with local reversals (Figure 5).

Table 8: **Routing hyperparameter sensitivity on Qwen.** Results are macro-averaged over eight benchmarks and five seeds. Δ is the AUBC difference from the default, computed before rounding. Harm rates report the percentage of escalated queries whose correct weak prediction becomes incorrect. Daggers indicate default settings.

Parameter	Value	AUBC (%)	Δ (pp)	Harm@10%	Harm@30%	Harm@50%
γ	1.00	58.699	+0.020	8.76	9.77	9.39
γ	1.10	58.702	+0.023	8.86	9.82	9.35
γ	1.25†	58.679	0.000	8.86	9.79	9.39
γ	1.50	58.699	+0.020	8.89	9.75	9.32
λ_{anchor}	0.00	58.688	+0.009	8.83	9.84	9.37
λ_{anchor}	0.25	58.685	+0.006	8.89	9.81	9.38
λ_{anchor}	0.50†	58.679	0.000	8.86	9.79	9.39
λ_{anchor}	1.00	58.679	0.000	8.93	9.79	9.37
τ_{anchor}	0.04	58.684	+0.005	8.96	9.82	9.34
τ_{anchor}	0.08†	58.679	0.000	8.86	9.79	9.39
τ_{anchor}	0.16	58.684	+0.005	8.87	9.83	9.35
κ_{event}	100	58.677	−0.002	8.74	9.63	9.37
κ_{event}	200†	58.679	0.000	8.86	9.79	9.39
κ_{event}	400	58.673	−0.006	8.82	9.76	9.40
p_0	0.70	58.682	+0.002	8.90	9.78	9.40
p_0	0.80†	58.679	0.000	8.86	9.79	9.39
p_0	0.90	58.683	+0.004	8.86	9.84	9.37
p_1	0.90	58.669	−0.011	8.78	9.76	9.40
p_1	0.95†	58.679	0.000	8.86	9.79	9.39
p_1	1.00	58.681	+0.002	8.77	9.83	9.39

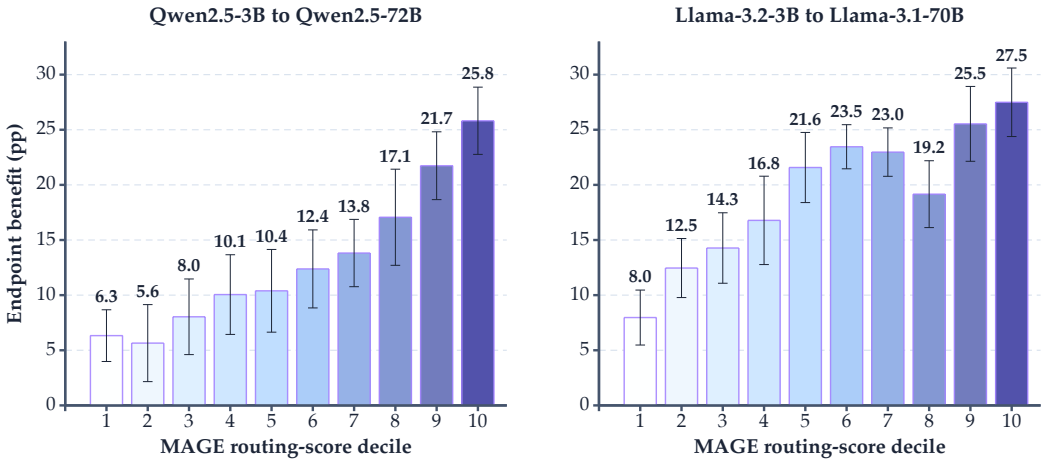


Figure 5: **Endpoint benefit by within-benchmark/seed score decile.** Bars show macro means in percentage points; error bars show descriptive ± 1.96 standard errors across correlated benchmark–seed summaries.

Table 9 evaluates rescue detection, treating rescue as positive and all other outcomes as negative. Margin uncertainty is $1 - (q^{(0)}_{(1)} - q^{(0)}_{(2)})$, and the uncertainty control U averages margin and normalized-entropy ECDF ranks (equation 36). Among the four displayed scores, MAGE achieves the highest rescue AUPRC, while margin yields slightly higher AUROC. At 10% calls, MAGE gives 2.59 pp accuracy gain and an 8.86% harm rate, compared with 1.93 pp and 12.98% for U. Compared with MAGE, the default R differs by 0.19 pp in rescue AUPRC, by at most 0.20 pp in harm rate, and by at most 0.15 pp in routing gain across the reported budgets.

Table 9: **Frozen-score diagnostics on Qwen.** Benchmark/seed macro detection and harm percentages, and routing gains in percentage points.

	Rescue detection		Harm rate ↓			Routing gain ↑		
Method	AUROC	AUPRC	10%	30%	50%	10%	30%	50%
MAGE (ours)	64.24	31.86	8.86	9.79	9.39	2.59	6.45	9.11
Geometry Core	64.04	31.67	9.06	9.85	9.43	2.56	6.30	8.99
Margin uncertainty	64.40	30.78	11.83	11.78	10.46	2.13	5.29	8.25
Uncertainty rank mean	64.17	30.63	12.98	11.66	10.42	1.93	5.20	8.33

Table 10: **Feature probes: three-benchmark macro percentages.** Raw geometry is log-transformed before ECDF ranking; full features form the 12-dimensional diagnostic representation.

Representation	Rescue AUROC	Rescue AUPRC	Harm AUROC	Harm AUPRC	Probe AUBC
Output uncertainty	54.08	30.15	57.61	12.00	45.24
Raw hidden state (PCA-5)	52.83	29.49	56.70	11.33	45.77
Raw geometry	53.54	31.76	53.49	10.63	**46.05**
ECDF geometry	53.92	31.80	58.33	11.57	45.96
Full MAGE features	**54.15**	**32.07**	**60.10**	**14.41**	45.95
Shuffled-label control	52.02	30.68	53.24	11.96	45.21

B.10 Representation Probes and Routing Stability

We evaluate logistic probes separately from MAGE on deterministic samples of 1,000 MMLU-Pro, 198 GPQA, and 1,000 MedQA examples (base seed 1729 plus dataset-specific offsets), using 25% calibration and five routing seeds. Preprocessing transformations and probes are fitted on the calibration split.

Probe fitting. For each representation, we fit separate L_2-regularized logistic models for rescue and harm:

$$y_i^{\mathrm{R}} = \mathbf{1}[a_{\mathrm{W},i} = 0, a_{\mathrm{S},i} = 1], \qquad y_i^{\mathrm{H}} = \mathbf{1}[a_{\mathrm{W},i} = 1, a_{\mathrm{S},i} = 0]. \tag{60}$$

Features are standardized on calibration data, replacing scales below 10^{-8} by one. Single-class targets yield constant probabilities. Otherwise, weights start at zero and the unpenalized intercept β at the logit of calibration prevalence clipped to $[10^{-5}, 1-10^{-5}]$. For standardized calibration design X and $\lambda_{\mathrm{probe}} = 1$, the step size is

$$\tau_{\mathrm{probe}} = \min\left\{0.20, \left(\frac{0.25\|X\|_2^2}{n_{\mathrm{cal}}} + \lambda_{\mathrm{probe}}\right)^{-1}\right\}, \tag{61}$$

where $\|X\|_2$ is the spectral norm. Optimization uses at most 600 iterations, clips logits to $[-30, 30]$, and stops when the updates satisfy $\|\Delta w\|_2 + |\Delta\beta| < 10^{-8}$. Diagnostic AUBC uses MAGE's fixed harm penalty, ranking examples by

$$S_{\mathrm{probe}}(x) = \widehat{p}_{\mathrm{R}}(x) - 1.25\widehat{p}_{\mathrm{H}}(x). \tag{62}$$

Feature probes. We compare five standardized representations: two uncertainty ECDF ranks; hidden-state PCA-5; $\log(1 + g_{\mathrm{raw}})$; its five-dimensional ECDF transform; and a 12-dimensional combination of five geometry ECDF coordinates, two uncertainty ranks, and semantic PCA-5. The geometry-based shuffled-label control independently permutes rescue and harm calibration targets, while evaluation uses the original targets.

Raw geometry achieves the highest probe AUBC (46.05%), while full features lead rescue AUPRC and harm AUROC/AUPRC. Full-feature harm AUPRC exceeds the uncertainty control by 2.41 pp (Table 10).

Layer-wise probes. We re-extract geometry with fresh perturbations at relative depths $\{0.15, 0.30, 0.50, 0.65, 0.78, 0.90\}$ (equation 16) and fit identical ECDF probes. Each depth, including 0.78, uses a new perturbation cache. Probe AUBC ranges from 45.25% to 45.70%, with depth 0.78 yielding the highest observed harm AUPRC (13.18%).

Table 11: **Layer probes (left) and perturbation stability (right).** AUBC and its within-benchmark standard deviation use percentage units; agreement is dimensionless. Daggers mark frozen settings.

Depth	Rescue AUROC	Rescue AUPRC	Harm AUPRC	Probe AUBC
0.15	53.63	30.55	11.05	45.46
0.30	54.43	32.16	11.09	45.70
0.50	53.40	31.35	11.04	45.25
0.65	55.35	32.12	11.91	45.70
0.78†	54.64	31.96	**13.18**	45.50
0.90	54.98	31.37	11.43	45.51

Traj.	Score corr.	Jac.@10%	Jac.@30%	Jac.@50%	AUBC	AUBC std.
1	0.394	0.208	0.376	0.534	46.83	1.019
2	0.481	0.211	0.365	0.542	46.72	0.551
4†	0.711	0.270	0.468	0.651	**47.61**	0.515
8	0.805	0.342	0.543	0.700	47.22	0.571

Perturbation stability. With routing seed fixed at 42, we repeat extraction and full router fitting with nested semantic OOF ten times for each $N \in \{1, 2, 4, 8\}$. Repeat- and N-specific perturbation seeds generate separate trajectory sets. Within each benchmark, we average Spearman correlation and Jaccard agreement $|\mathcal{R}_b^{(u)} \cap \mathcal{R}_b^{(v)}|/|\mathcal{R}_b^{(u)} \cup \mathcal{R}_b^{(v)}|$ over the 45 dependent repeat pairs. AUBC standard deviations are computed across the ten repeats within each benchmark and then macro-averaged.

Four trajectories achieve the highest observed AUBC (47.61%), while eight improve agreement at twice the perturbation cost (Table 11). At $N = 4$, top-10% Jaccard is 0.270, indicating greater repeat sensitivity in query selection than in aggregate AUBC.

B.11 End-to-End Efficiency

Profiling protocol. We profile endpoint components on 32 examples per benchmark with CUDA synchronization after model loading and warm-up. Weak base inference and four-trajectory batches are timed separately; semantic encoding is batch-amortized, and router fitting/scoring is amortized over benchmark size. Accuracies correspond to the nested-OOF router. Runtime estimates are based on the component profiles described above, and exclude nested-OOF calibration overhead.

Estimated per-query wall time is

$$\widehat{T}_d(b) = \bar{t}_{\mathrm{W},0} + \bar{t}_{\mathrm{pert}} + \bar{t}_{\mathrm{sem}} + t^{\mathrm{amort}}_{\mathrm{router},d} + \widehat{b}_d\, \bar{t}_{\mathrm{S}}, \tag{63}$$

where $\widehat{b}_d$ is the realized call fraction and the router term includes routing decisions. Strong-model time uses the benchmark-wide profile mean. GPU-seconds weight GPU component runtimes by occupied device counts; CPU router time contributes to wall time. This additive profile model covers pipeline computation, with serving-level queueing and concurrency outside its scope.

Transformer computation. We approximate Transformer FLOPs as twice the loaded parameter count times sequence-token work. For each benchmark, cost relative to one direct strong-model candidate-likelihood evaluation is

$$\widehat{C}_d(b) = \frac{\widehat{F}_{\mathrm{W},0} + \widehat{F}_{\mathrm{pert}} + \widehat{F}_{\mathrm{sem}} + \widehat{b}_d \widehat{F}_{\mathrm{S}}}{\widehat{F}_{\mathrm{S}}}. \tag{64}$$

The numerator covers Transformer computation for weak scoring, perturbations, semantic encoding, and routed strong inference. We macro-average these benchmark-level ratios. Nested OOF affects router fitting but not the extraction settings or realized call counts used in these estimates.

Across the evaluated budgets, relative Transformer FLOPs range from 31.3% to 71.3% on Qwen (Table 12) and from 32.8% to 72.8% on Llama. Figure 6 separates fixed weak-side overhead from budget-dependent strong-model costs.

Accuracy–compute trade-offs. Figure 7 combines the nested-OOF router accuracies with estimated Transformer FLOPs under a common weak-first cost model. Baselines incur one weak pass plus budgeted strong inference; MAGE also includes perturbation and semantic-encoding costs. At matched strong-model call budgets, MAGE achieves higher accuracy than the five adapted routing baselines, with larger gains on Qwen than on Llama. The plotted configurations show the additional computation associated with these gains.

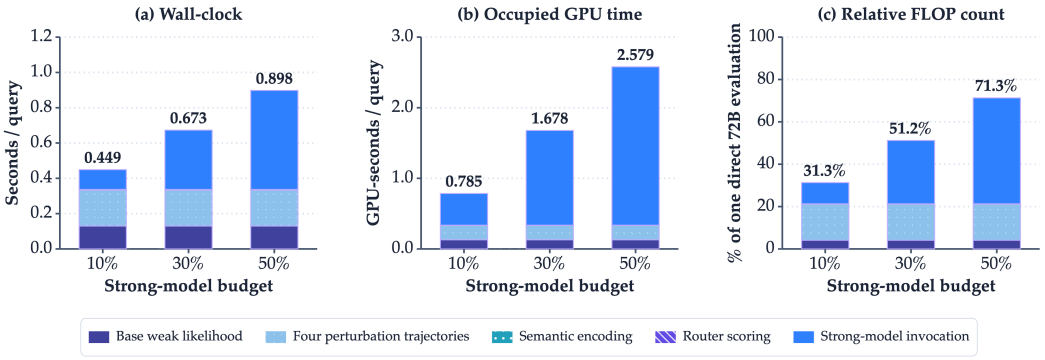


Figure 6: **Qwen cost decomposition.** Profile-based component-time estimates and estimated Transformer FLOPs, with the FLOP panel normalized to one direct 72B likelihood evaluation. Timing excludes nested-OOF calibration overhead.

Table 12: **Profile-based pipeline cost and routing accuracy on Qwen.** Reported accuracies use the nested-OOF router. Runtime estimates are derived from the component profiles and fitting costs. Relative FLOPs measure Transformer computation normalized to direct Qwen2.5-72B likelihood scoring.

Budget	Macro Acc. (%)	Wall (s/query)	GPU time (GPU-s/query)	Rel. FLOPs (×72B)
10%	55.03	0.449	0.785	0.313
20%	57.23	0.562	1.233	0.413
30%	58.89	0.673	1.678	0.512
40%	60.30	0.785	2.128	0.612
50%	61.55	0.898	2.579	0.713

C Discussion and Future Directions

Our results support the use of local perturbation responses as a routing signal that complements output confidence. The component analysis highlights a strong label-free default, with reliability shrinkage controlling the effect of calibration-based refinement.

Limitations. MAGE requires white-box access to the weak model and four additional perturbation trajectories per candidate-scoring pass, together with calibration and router computation. Our evaluation focuses on multiple-choice tasks with within-benchmark calibration and batchwise routing; broader deployment settings remain to be explored.

Adaptive Probing. Future work could allocate trajectories according to the stability of successive probe responses, stopping early when routing decisions stabilize and investing more computation in ambiguous cases. Skipping semantic encoding at inference when its selected weight is zero could further reduce overhead. These optimizations should be assessed jointly for routing quality and total computation.

Generalization Across Models and Domains. The shared hyperparameter configuration yields higher observed macro AUBC than UniRoute on both endpoint pairs, with stronger evidence on Qwen. Evaluation across additional architectures and capability gaps could clarify when local sensitivity is most informative. Cross-benchmark calibration and distribution-shift experiments would test the portability of fitted routing components, while extension to free-form generation would require defining predictive responses beyond fixed candidate sets.

Online Routing and Efficiency. Streaming quantiles or adaptive thresholds could extend the batchwise rule to online budget control. The variation in routing gains across endpoint pairs mo-

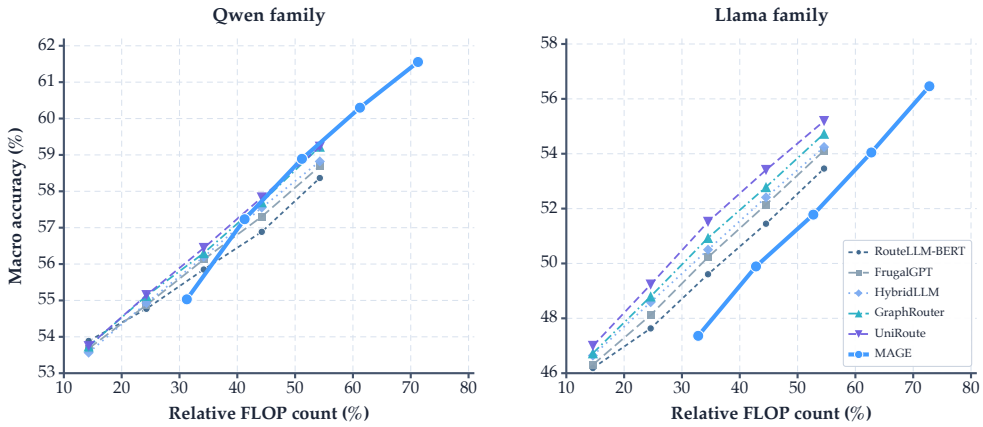


Figure 7: **Accuracy–compute curves across endpoint pairs.** Macro accuracy versus estimated Transformer FLOPs for MAGE and the five adapted routing baselines at strong-model call budgets of 10%–50%. FLOPs are normalized to one direct strong-model candidate-likelihood evaluation. All methods use common weak-first accounting; baseline-router overhead is set to zero, while MAGE includes perturbation and semantic-encoding costs.

tivates jointly choosing model pairs, probing effort, and escalation thresholds. Future evaluation should account for calibration and inference costs under realistic batching and concurrency.