Pandora's AI Model Routing Box: Efficient Allocation with Costly Value Estimation

Adam Fisch*, **Shubhendu Trivedi, Fantine Huot, William W. Cohen, Michael Kaisers, Mirella Lapata, Kate Larson, Jacob Eisenstein***
Google DeepMind

Heterogeneous AI systems composed of multiple models, architectures, harnesses, or inference-time settings can improve quality and efficiency by routing queries to the *specialist* who can answer most effectively at the lowest cost. Routing requires estimating each specialist's expected return, but this value estimation has a cost. Cheap estimators (e.g., embedding-based predictors) are fast but noisy, while accurate estimators (e.g., fine-tuned models with access to retrieval results or partial reasoning traces) are expensive. We formalize this tradeoff as an instance of Pandora's Box, the classical problem of optimal search with costly inspection. Under a Gaussian signal model, the resulting policies have closed-form value-of-information expressions that determine, for each specialist and input, whether refining the value estimate is worth its cost. We call the centralized policy Pandora's Router. We extend this to a decentralized setting, Pandora's Bidder, where specialists independently decide whether to invest in self-assessment before accepting an offered price to claim a query. Experiments across three domains—a standard multi-LLM benchmark, retrieval-augmented specialists, and LLMs with variable inference-time reasoning—show that Pandora's Router matches the routing quality of exhaustive estimation, while querying the expensive estimator far less often. In the decentralized setting, value-of-information reasoning improves allocative efficiency when competing estimates are accurate; when competing estimates are noisy, however, it can increase the strategic specialist's utility at the expense of others.

1. Introduction

AI model providers now often offer heterogeneous model families spanning a wide range of costs and capabilities: small and fast models for simple prompts, large and expensive ones for complex tasks, and augmented variants with tools, retrieval, or extended reasoning for more specialized tasks. This raises the question of how to allocate each input to the model best suited for it, for a given level of cost sensitivity. This is the *routing* problem, and a growing body of work addresses it by estimating each model's expected return on the input and selecting the maximizer (Feng et al., 2026; Hu et al., 2024; Shnitzer et al., 2023). But *value estimation* is neither free nor particularly easy. An embedding-based predictor is cheap but noisy; a fine-tuned scoring model is more accurate but more expensive; and computing partial solutions, executing tool calls, or running retrieval pipelines can be more expensive still. The routing decision itself thus involves a cost-accuracy tradeoff, which is typically ignored. We explore this tradeoff, asking: when is it worth paying for a better value estimate?

The question has a clean analogy to the *Pandora's Box* problem from search theory (Weitzman, 1979). Pandora is presented with M boxes, each with a hidden value which is known to her only in distribution. For a cost c_m she can open any box m and observe its value. At any point, she can stop searching and claim the best value seen thus far. Her goal is to maximize the difference between the value obtained minus the costs paid. Weitzman (1979) showed that the optimal policy has a remarkably simple structure, based on computing, for each box, its *reservation price*—the outside-option value at which the expected benefit of opening the box balances out the cost of opening it. Once these prices are calculated, Pandora opens the boxes in descending order of their reservation prices, stopping when she has found a value that exceeds the maximum of the reservation prices of the remaining unopened boxes.

*Project leads. Correspondence to {fisch, jeisenstein}@google.com.

In this paper, we cast routing with costly value estimation as an instance of Pandora's Box. Here, each specialist can be viewed as a box. The router always has access to a cheap value estimate $f_m(x)$, and can optionally pay c_m to query a more accurate but costly estimate $g_m(x)$; i.e., pay to "open the box", and gain more information about the quality of specialist m. The question of which estimates to compute, and when to stop computing and commit to a specialist, is the same as Pandora's problem. Under a Gaussian signal model for the relationship between f and g, the reservation prices and the associated value-of-information expressions have closed forms. To allow Pandora to select a box without making *any* queries to g, we consider the *non-obligatory* variant of the Pandora's Box problem, in which she can select a box that she has not opened (Beyhaghi and Kleinberg, 2019; Doval, 2018).

Next, we consider the generalization from routing to decentralized forms of prompt allocation. In a routing system, a single centralized router controls all value estimation. In practice, however, the specialists themselves may be better positioned to estimate their own value. For example, a retrieval-augmented specialist has access to its own corpus, and can measure how relevant it is to the query; a math specialist can start to reason about or plan what computations will be required; a domain expert can know its own performance on internal benchmarks. None of these resources need be available to the router, and in some scenarios, specialists might even want to conceal information such as whether there are matches to a query in a private retrieval corpus. With this in mind, we extend the concept of value estimation for routing, to value estimation for *bidding*, where decentralized agents learn how to leverage costly value estimation when claiming queries, in a way that maximizes their profit. Concretely, we propose *Pandora's Bidder*: each specialist faces a posted price—the current best competing offer—and must decide whether to invest in a more accurate self-assessment before claiming the query. This corresponds to a single stage of the ascending-price mechanism with costly preference elicitation studied by Parkes (2005); analyzing this simple setting isolates core components of decentralized prompt allocation under private value estimation with profit incentives.

Core contributions of this work

As AI systems with diverse capabilities proliferate, reasoning about the economics of model selection becomes increasingly relevant. To our knowledge, this is the first paper to explicitly connect model routing with the classical Pandora's Box problem from economics and operations research. Specifically, this paper makes three main contributions:

- We formalize model routing with costly value estimation as an instance of Pandora's Box with non-obligatory inspection (Doval, 2018), yielding a reservation-price-based policy. Under a Gaussian signal model, this policy has a closed form that is straightforward to compute (§4).
- We extend to a decentralized setting where specialists use value-of-information reasoning to decide whether to refine their self-assessments before accepting a posted price (§5).
- We evaluate both frameworks (see Figure 1) across three domains: a multi-LLM routing benchmark (EmbedLLM), retrieval-augmented specialists for factoid QA, and inference-time computation for mathematical reasoning. In each setting, we find that both Pandora's Router and Pandora's Bidder are able to gracefully trade-off expensive and cheap estimates to achieve strong allocative performance relative to baselines across a range of estimation costs.

2. Setting

We begin with a more precise introduction to the problem of efficient allocation with costly value estimation. Suppose that a router must assign each prompt to the best specialist, but estimating specialist quality is itself costly. We assume that each specialist m has a cheap value estimate $f_m(x)$ by default, while a more accurate but expensive estimate $g_m(x)$ can be queried selectively; we make these estimators concrete in Section 2.1. The goal is to maximize the expected reward of the selected specialist

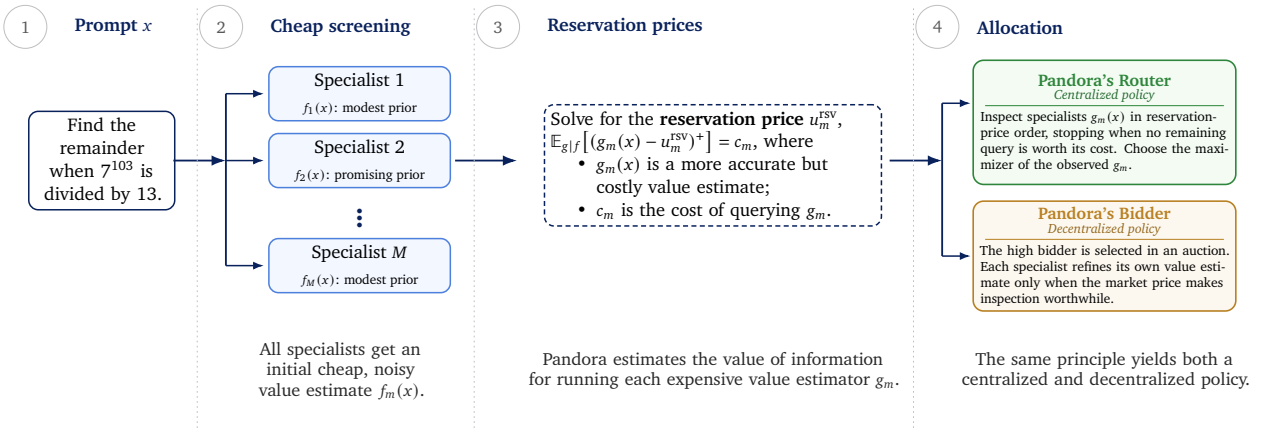

Figure 1 | **Overview of the PANDORA framework.** The system balances cheap screening and costly reasoning for routing. **(1)** A concrete prompt x is received. **(2)** Multiple specialists $m \in \{1, \ldots, M\}$ generate cheap, noisy value estimates $f_m(x)$. **(3)** The system computes **reservation prices** that quantify the expected value-of-information for querying a more costly and accurate value estimate $g_m(x)$, e.g., by running the specialist for a few reasoning tokens and checking if it is on the right track. **(4)** This unified principle supports allocation by centralized routing and decentralized auctions.

minus total estimation cost by deciding which expensive estimates $g_m(x)$ are worth computing.

Formally, let $X \in \mathcal{X}$ be a random input prompt, and let x denote a realization. Conditional on the input $X = x$, specialist m samples an output $Y_m \sim P_m(\cdot \mid x)$, $Y_m \in \mathcal{Y}$, and receives a cost-adjusted reward $R_m(x, Y_m)$. For notational convenience, we will write the cost-adjusted reward $R_m(x, Y_m)$ simply as R_m. The reward may combine, for example, the graded quality of the output with the cost of producing it (which could be the inference cost, or API fees for tools called), and is a random variable through both the randomness of Y_m and any noise in the evaluation itself (e.g., from a human annotator). The input-specific oracle *routing* problem is to select the specialist that maximizes the conditional expected reward, that is, $\hat{m}(x) \in \arg\max_m \rho_m(x)$, with $\rho_m(x) := \mathbb{E}[R_m \mid X = x]$. However, since the true reward is typically not available at routing time, the router must instead apply a *value estimator* to predict it, and then pick the best specialist based on their estimated values.

Of course, there are many ways to do value estimation, and these estimators may themselves incur widely varying computational costs. At one extreme, a cheap value estimate could just be a constant, such as the average cost-adjusted reward on a calibration set (and not conditional on x). At the other extreme, we might actually materialize the candidate output $Y_m = y$ for each model m, and then score y with an estimate of R_m (with an LLM-based auto-rater for example, if R_m involves a human judgment), or even observe R_m directly (if the reward is a known function of y). In practice, a straightforward approach to value estimation is to apply a trained model to the text of the prompt, or its embedding (Hu et al., 2024; Lu et al., 2024; Ong et al., 2025, *inter alia*). These estimators can move along a cost-accuracy curve depending on their complexity, or depending on any other auxiliary input information that is gathered by the estimator (e.g., by invoking tools or computing partial solutions, such as plans (Liang et al., 2025; Yao et al., 2022)). We denote the additional information revealed from such a costly inspection of specialist m by Z_m. Note that the costly information Z_m need not be private to the specialist—it may simply reflect additional computation applied to the prompt, such as from using a more expressive encoder.

Under this view, the routing process separates naturally into two phases: gathering information and making a final choice. We consider a minimal setting in which there are just two available value estimators: f_m (cheap) and g_m (costly), with costs $c_f < c_g$, respectively. For simplicity, we set $c_f = 0$, so that

f_m is always computed. We will use uppercase letters for the corresponding random variables. Let $F_m = f_m(X)$ be the cheap estimate for the specialist m, and let $\mathbf{F} = (F_1, \ldots, F_M)$. Inspecting specialist m reveals the costly information Z_m needed to compute the refined estimate $G_m = g_m(X, Z_m)$; a process we simply call querying g_m. We denote the cost of this process as c_g. For the additional cost of G_m to be justified, it should be more accurate, in the sense that on calibration data we have $\mathbb{E}[(R_m - G_m)^2] < \mathbb{E}[(R_m - F_m)^2]$. The router must dynamically decide which specialists to inspect (by querying g_m) in order to maximize the expected reward of the final assignment, minus the total cost of the realized G_m estimates.

To make things more concrete before presenting our efficient routing algorithms, we now describe three real-world domains where the two-tier estimation between f_m versus g_m arises organically, each with a qualitatively different source of costly information that is useful for deciding the best allocation.

2.1. Experimental domains and value estimators

We explore several empirical settings, each featuring different types of specialists and value estimators. In all cases, the cheap estimator f embeds the prompt with a pretrained encoder, retrieves the k nearest prompts from a calibration set (measured by cosine similarity in embedding space), and returns the average reward of the retrieved neighbors as the value estimate. The costly estimator is then constructed by fine-tuning a small language model encoder to predict the reward from a tailored, domain-specific input context. We describe these below. Note that in each domain, we partition the data into three splits: training (for fitting value estimators), calibration (for estimating the value-of-information model parameters in Sections 4 and 5), and test. Table 1 reports the MSE of each estimator on held-out calibration data. Additional dataset and implementation details are included in Appendix B.

MATH: Inference-time scaling in mathematical reasoning tasks. In inference-time scaling, extended reasoning improves accuracy on hard problems but increases latency and cost. The costly estimator g treats the early tokens of a reasoning trace as private information: it receives the prompt along with the first t tokens of a model's chain-of-thought, effectively peeking at a partial solution to judge whether the model's reasoning looks promising before committing to it. This is passed with the prompt to a small language model (Gemini 2.5 Flash-Lite) with a regression loss to predict the specialist's reward directly from the prompt text. The router chooses between Gemma3-4B (Gemma Team et al., 2025) (low-cost) and Gemini-3.1-Flash-Lite (Google DeepMind, 2026) (cost 0.66, so that $R_m = \mathbf{1}\{Y_m \text{ is correct}\} - 0.66$). This high cost is chosen so that it is beneficial to route to Gemini only when confident that it can answer correctly and Gemma cannot. We evaluate on a corpus of 16,512 mathematical problems spanning MATH (Hendrycks et al., 2021), Omni-Math (Gao et al., 2025), AIME (He et al., 2024; Mathematical Association of America), and HMMT (Harvard-MIT Mathematics Tournament).

RAG: Retrieval-augmented generation with specialized corpora. In retrieval-augmented generation (RAG; Lewis et al., 2020), retrieval results can improve answer quality but incur costs from retrieval computation, longer contexts, and potentially licensing fees for specialized corpora. The costly estimator g runs retrieval and passes the results alongside the prompt to a language model fine-tuned with a regression loss, as in Math. Notably, the retrieval results themselves do carry some signal: if the retrieved documents are relevant to the query, the specialist is more likely to answer correctly, whereas irrelevant retrievals may hurt. The router selects between three specialists: a low-cost model with no retrieval (the costly estimator g here is just the more expensive LLM-based estimator), a Wikipedia RAG model, and a PubMed RAG model, where the RAG models each incur a cost of 0.05 (again, subtracted from correctness in R_m). Prompts are a mixture of general-knowledge factoid questions and biomedical questions, following the same experimental setup as Eisenstein et al. (2025).

EmbedLLM: Large-scale language model selection. Finally, we consider EmbedLLM (Zhuang et al., 2025), a standard routing benchmark without auxiliary information, where the value of costly estimation comes purely from spending more compute to better distinguish among a large pool of

Setting	Estimator	MSE	c_g/c_f	Description
MATH	f: KNN g: SFT-CoT-20	.154 .096	5.8	k-nearest neighbors on prompt embeddings Fine-tuned transformer on prompt + first 20 tokens of the model's chain-of-thought (CoT) reasoning trace
RAG	f: KNN g: SFT-retrievals	.175 .109	> 7000	k-nearest neighbors on prompt embeddings Fine-tuned transformer on prompt + retrieval results
EmbedLLM	f: KNN g: SFT-prompt	.266 .198	1.6	k-nearest neighbors on prompt embeddings Fine-tuned transformer on prompt

Table 1 | Value estimators across experimental domains. The cheap estimator f is always computed; the costly estimator g is queried selectively. The mean-squared error (MSE) is measured on a held-out calibration set. The cost ratios c_g/c_f are discussed in Section 2.2 and derived in Section D.3.

candidates. As above, g is a fine-tuned language model; even without private information, this is more expensive at inference time than the KNN baseline, but is significantly more accurate than the KNN baseline, since the model can learn prompt-level features that the embedding space misses. EmbedLLM incorporates more than 100 open-weights models as routing targets; following Jitkrittum et al. (2025), we assign each model a cost proportional to its parameter count. Queries are drawn from standard benchmarks like MMLU (Hendrycks et al., 2020) and GSM8K (Cobbe et al., 2021).

While the MATH and RAG settings have only two and three target models to choose from respectively, both routing and efficient value estimation are still challenging and important problems even when the number of target models is small. The strongest frontier models can be up to five times as expensive as cheaper models from the same provider,[1] so there are strong financial incentives to reserve the most expensive models for situations where they yield meaningful improvements. Similarly, as shown in Table 1, more computationally-intensive value estimation strategies can yield significant improvements, but at high cost. The difficulty of the information acquisition problem depends not primarily on the number of targets, but on how discriminable the options are: we can have many options, but this doesn't matter if there is always a clear best box to use based on cheap signals alone; or we can have a few boxes that are hard to choose between, and it is expensive to know more. We picked experimental settings that try to cover key aspects of each of these considerations.

2.2. Value estimator costs

To estimate monetary costs for the value estimators in Table 1, we use prices from `https://ai.google.dev/gemini-api/docs/pricing` (retrieved August 1, 2026), because it offers a single source of prices for LLM inference, embedding, and retrieval. The resulting estimates are meant only to show that approaches to value estimation can incur vastly different costs; we do not claim that these specific prices are optimal or even typical. Because the cost-accuracy tradeoff is a *user* characteristic rather than an objectively-measurable property of the domain or method, we focus on the *ratio* between the prices of f and g. Please see Section D.3 for the derivation of these ratios.

3. Preliminaries: The Pandora's Box Problem

We begin by briefly reviewing the Pandora's Box problem (Weitzman, 1979), which we will connect to model routing and model bidding in the following sections. In the Pandora's Box problem, a decision-maker (i.e., Pandora) is presented with M options, called "boxes", each with an unknown reward (but with known distribution). For a price she can inspect any box to observe its reward, and

[1] Compare, e.g. Fable 5 vs Sonnet 5 at `https://platform.claude.com/docs/en/about-claude/pricing` (retrieved August 13, 2026).

at any time she can terminate the search and take the best reward that she has observed. The optimal policy depends on a mathematical object called the *reservation price*, which is a single scalar that encodes the value of inspecting a box. The reservation prices alone can be used to construct a simple, but optimal, priority-based search. Notably, this optimal search must be sequential, as the information acquired so far will help determine whether the remaining inspections still justify their cost.

To gain intuition for the reservation price, suppose Pandora has already found a box with value v for the input x, and must decide whether to open box m, whose hidden value is yet unknown, but known to be drawn from a distribution $P_m(\cdot \mid x)$. If Pandora opens the box and finds $V_m > v$, she takes V_m; otherwise, if $V_m < v$, she keeps v. Given an inspection cost of c_m, her expected payoff at this step from opening the box is therefore $\mathbb{E}[\max\{v, V_m\}] - c_m$; the payoff for *not opening* m is v. The net value is $\mathbb{E}[(V_m - v)^+] - c_m$, which is positive when the expected upside exceeds the cost. The *reservation price* u_m^{rsv}, defined by Weitzman (1979), is the outside-option value at which this net value is exactly zero:

$$\mathbb{E}[(V_m - u_m^{\text{rsv}})^+] = c_m. \tag{1}$$

When the current best value v exceeds u_m^{rsv}, the cost of opening box m outweighs the expected gain. The resulting decision rule is therefore quite simple: we open box m if $v < u_m^{\text{rsv}}$, and skip it otherwise. The full selection strategy, however, requires more than a single "open-or-skip" decision. Pandora must sequence her inspections, since each opened box updates the current best value v, which in turn changes. She may also skip inspection entirely: if the prior P_m strongly favors box m, she can save c_m and commit sight-unseen. These choices define the two classical variants of the problem.

Pandora-OI (obligatory inspection). In the obligatory-inspection variant, Pandora must open every box she eventually selects. This is the classical Pandora's Box setting, and admits a simple solution. Specifically, Weitzman (1979) showed that reservation prices alone determine the optimal policy: initialize a best-seen value $v^* := -\infty$, open boxes in descending order of u_m^{rsv}, and stop as soon as v^* exceeds the highest remaining reservation price. The box that yielded v^* is selected. The remaining boxes can be safely ignored because $v^* > u_m^{\text{rsv}}$ implies $\mathbb{E}[(V_m - v^*)^+] < c_m$, by the definition of u_m^{rsv}.

Pandora-NI (non-obligatory inspection). When inspection is non-obligatory, Pandora may select any unopened box, committing to it without paying the inspection cost, but accepting the risk that its realized value may disappoint (Doval, 2018). The optimal adaptive policy for this variant is NP-hard (Fu et al., 2023), so we use a tractable restriction to *committing policies* (Beyhaghi and Kleinberg, 2019). A committing policy designates a single box m as "held out": this box will not be opened, but may be selected as the final choice. The remaining $M - 1$ boxes are searched via Pandora-OI, with the held-out box providing the initial outside option.[2]

To use an unopened box as an outside option, we need to assign it a deterministic value. We use the *backup price* u_m^{backup}, which like the reservation price in Equation (1), is defined as the solution to:

$$\mathbb{E}[(u_m^{\text{backup}} - V_m)^+] = c_m. \tag{2}$$

Note that the backup price is related to, but different from, the reservation price. The reservation price is the outside option that makes Pandora indifferent to inspecting a box or skipping it; the backup price is the outside option that makes Pandora indifferent to committing to the sealed box sight-unseen versus paying to inspect it. Any opened box must exceed this threshold to justify discarding the sealed option. The relationship between the backup price and the reservation price depends on the expected shortfall below the mean of the held-out box, specifically, $u_m^{\text{backup}} \le u_m^{\text{rsv}}$ if and only if $c_m \le \mathbb{E}[(\mathbb{E}[V_m] - V_m)^+]$. For costs exceeding this threshold (very expensive boxes), the inequality reverses.

[2]The value of the best committing policy that holds out multiple boxes cannot exceed that of the best single-box policy (Beyhaghi and Kleinberg, 2019), so it suffices to evaluate all M single-holdout policies, in addition to the obligatory inspection policy (no holdouts). These M + 1 policies are compared by simulating their performance via MC sampling.

The algorithm proceeds as follows. For each candidate holdout $m \in \{0, 1, \ldots, M\}$, define the expected payoff of the committing policy that reserves box m:

$$\nu_m = \mathbb{E}\left[V_{\hat{m}} - \sum_{j \in O_m} c_j\right], \tag{3}$$

where $V_{\hat{m}}$ is the value of the (random) box that the committing policy selects, and O_m is the (random) set of boxes opened by Pandora-OI on $\{1, \ldots, M\} \setminus \{m\}$ with initial outside option $v^* = u_m^{\text{backup}}$. The case $m = 0$ corresponds to running Pandora-OI on all M boxes with no holdout ($u_0^{\text{backup}} = -\infty$). Because the held-out box is unobserved, we estimate ν_m via Monte Carlo (MC) samples $\tilde{V}_m \sim P_m(\cdot \mid x)$; we denote the MC estimate $\hat{\nu}_m$. We select m^* that maximizes $\hat{\nu}_m$, using $S = 100$ samples.[3] At test time, we run Pandora-OI on $\{1, \ldots, M\} \setminus \{m^*\}$ with initial value $v^* = u_{m^*}^{\text{backup}}$, opening the actual boxes. We default to m^* if we open no boxes or if no realized value clears $u_{m^*}^{\text{backup}}$; otherwise, we select the best opened box.[4]

4. Pandora's Router

A value-based router tries to select the specialist with the highest expected return. We connect routing to Pandora's Box by treating each specialist as a box and the costly value estimate as the value revealed by opening that box. One subtlety is that the box value is not the realized downstream reward R_m itself. The router never observes R_m before choosing a specialist. The relevant decision value is thus the expected reward conditional on the information the router can acquire. In practice, our learned costly estimate G_m is a learned plug-in approximation to this posterior decision value, and our proxy objective is to find the specialist for which G_m is largest. In Proposition 1 and Corollary 1 (Appendix A), we show that this is equivalent in expectation to optimizing R_m, assuming conditional independence.

Gaussian signal model. The Pandora's Box algorithm requires an estimate of the distribution of the box values. We model the distribution of G_m conditional on the cheap estimates as

$$G_m \mid \mathbf{F} = \mathbf{f} \;\sim\; \mathcal{N}\left(\mu_m,\ \sigma_m^2\right), \qquad \mu_m = h_m(\mathbf{f}) \tag{4}$$

where $\mathbf{f}$ collects the realized cheap estimates $(f_1(x), \ldots, f_M(x))$ for all specialists and μ_m is a function $h_m(\cdot)$ of this vector, where h_m and σ_m^2 are estimated on calibration data. The choice of h_m is flexible; we explore both gradient boosted decision trees and linear regression depending on the domain (see Section C). μ_m captures the predictable component of g_m given $\mathbf{f}$, and σ_m captures the residual uncertainty, which is approximated as normally distributed per Equation (4). Under this model, the value of opening box m against an outside option v is the expectation of a Gaussian censored at v:

$$\mathbb{E}[(G_m - v)^+] = (\mu_m - v)\,\Phi(\alpha_m) + \sigma_m\,\phi(\alpha_m), \tag{5}$$

with $\alpha_m = (\mu_m - v)/\sigma_m$, and Φ and ϕ the standard normal CDF and PDF. The reservation price u_m^{rsv} satisfies $\mathbb{E}[(G_m - u_m^{\text{rsv}})^+] = c_m$ and can be obtained by simple root-finding on Equation (5). Of course, while convenient, the Gaussian model is only an approximation, and real box values are rarely strictly Gaussian. We also explore a non-Gaussian signal model in Section D.6 (for an alternative model based on Gaussian processes, see Xie et al. (2024)), but find that while it is possible to improve fit to the empirical distribution, it does not yield an improvement in overall routing success and inspection cost.

An extension for handling correlated values. In routing, the Pandora-OI and Pandora-NI policies described above must contend with an additional complication: the hidden values of the boxes are

[3] Pilot experiments showed similar results with $S = 30$, with performance degrading significantly only at $S = 10$. We chose $S = 100$ because the overall computational costs of this operation are cheap relative to value estimation itself.

[4] As a small technical detail, Beyhaghi and Kleinberg (2019) use the *expected value* of the heldout box as the initial value v^*, rather than our choice of the backup price, which, empirically, we found to perform slightly better. See Section D.5.

Algorithm 1 Pandora's Router (with non-obligatory inspection)

Require: Cheap value estimates $\mathbf{f} = (f_1(x), \ldots, f_M(x))$, multi-variate Gaussian signal model parameters (h, Σ), costs $\{c_m\}$, number of Monte Carlo samples S.

Compute predicted means $\mu_m \leftarrow h(\mathbf{f})_m$ for all m
Compute reservation prices u_m^{rsv} via root-finding on Equation (1)
Compute backup prices u_m^{backup} for all m via root-finding on Equation (2)
for each candidate held-out box $m = 0, 1, \ldots, M$ **do**
    // Note: $m = 0$ corresponds to Pandora-OI on $\{1, \ldots, M\}$ with $u_0^{\text{backup}} = -\infty$.
    // $\hat{\nu}_m$ is an MC estimate of the expected committing-policy payoff ν_m (Equation (3)).
    Estimate $\hat{\nu}_m$ by running Pandora-OI on $\{1, \ldots, M\} \setminus \{m\}$ with initial value $v^* = u_m^{\text{backup}}$, averaged over S samples of realized $\tilde{\mathbf{G}} = (\tilde{G}_1, \ldots, \tilde{G}_M)$ where $\tilde{\mathbf{G}} \sim \mathcal{N}(\mu, \Sigma)$.
Select held-out box $m^* = \arg\max_m \hat{\nu}_m$
Run Pandora-OI on $\{1, \ldots, M\} \setminus \{m^*\}$ with initial value $v^* = u_{m^*}^{\text{backup}}$, using actual g-queries
if no boxes were opened **or** the best realized value $\le u_{m^*}^{\text{backup}}$ **then**
    return sealed holdout m^*
else
    return the best opened box

not independent. For some difficult prompts, all specialist scores G_m will tend to fall below the prediction μ_m; for easy prompts, they will cluster above it. This correlation is particularly strong in the EmbedLLM domain, where there are > 100 routing targets, some of which are extremely similar (e.g., different fine-tunings of the same base model). To handle this, we propose a heuristic approximation that recomputes reservation prices at each step of the sequential policy (inspired by Gergatsouli and Tzamos (2023), and also similar to the Gaussian process updates in Xie et al. (2024)). On calibration data we estimate the parameters of a multivariate Gaussian model $\mathbf{G} \sim \mathcal{N}(\mu, \Sigma)$. After each box is opened, we condition on the observed values $\mathbf{G}_{\text{observed}}$ and compute $P(\mathbf{G}_{\text{unobserved}} \mid \mathbf{f}, \mathbf{G}_{\text{observed}})$ via the standard multivariate Gaussian posterior. We then make a mean-field approximation to this posterior, yielding updated marginal distributions for each remaining box, from which we recompute the reservation prices. The sequential policy then proceeds as before with these new parameters.

Experimental setup. We evaluate Pandora's Router on the three domains described in §2.1 (MATH, RAG, and EmbedLLM), and compare with the following methods:

- **f-only**: Route using the cheap estimator f_m only (never opening any box).
- **g-always**: Route using the expensive estimator g_m only (always opening all boxes).
- **Top-2**: Always queries g_m for the two models with highest f_m (open the two expected best boxes).
- **Coin Flip**: Query g_m with probability $\frac{1}{2}$; route to the maximum observed (randomly open boxes).

We also evaluate two ablations that use the same inspection budget selected by Pandora's Router, but use different methods to decide which boxes to open (that is, instead of reservation prices). Specifically, we first run Pandora's Router to count how many times, N_{pr}, in total g_m was queried over a test set (in hindsight), and then reallocate those N_{pr} inspections according to the following rules:

- **Random-N$_{\text{pr}}$:** A simple control where g_m is randomly queried up to N_{pr} times over the entire test set; for every example we route to the best candidate among the randomly inspected options. We route to $\arg\max_m f_m$ as a default if no inspections were made for that example.
- **Margin-N$_{\text{pr}}$**: An uncertainty-based heuristic that queries g_m for specialists whose f-scores are close to being best: letting $m' = \arg\max_j f_j$, if $f_{m'} - f_m$ is small, we inspect both g_m and $g_{m'}$. Again, we inspect g_m for at most N_{pr} specialists over the entire test set, and route to m' as a default.

Method	MATH			RAG			EmbedLLM		
	Regret	**Cost**	**Total**	**Regret**	**Cost**	**Total**	**Regret**	**Cost**	**Total**
f-only	0.117	0.000	0.117	0.150	0.000	0.150	0.393	0.000	0.393
g-only	0.090	0.038	0.128	0.084	0.057	0.141	0.370	1.986	2.356
Top-2	0.090	0.038	0.128	0.107	0.038	0.146	0.402	0.036	0.438
Coin Flip	0.107	0.019	0.127	0.122	0.029	0.151	0.398	0.992	1.390
Random-N_{pr}	0.164	0.011	0.175	0.135	0.027	0.162	0.363	0.075	0.438
Margin-N_{pr}	0.094	0.011	0.105	0.101	0.027	0.128	0.314	0.075	0.389
Pandora's Router	0.094	0.011	**0.105**	0.091	0.027	**0.118**	0.311	0.075	**0.386**

Table 2 | Routing evaluation results averaged across query costs c_g. Lower is better on all metrics; the lowest average regret + cost measured per setting is **bolded**. See Section D.4 in the Appendix for a breakdown of results per cost level c_g, together with paired statistical significance tests.

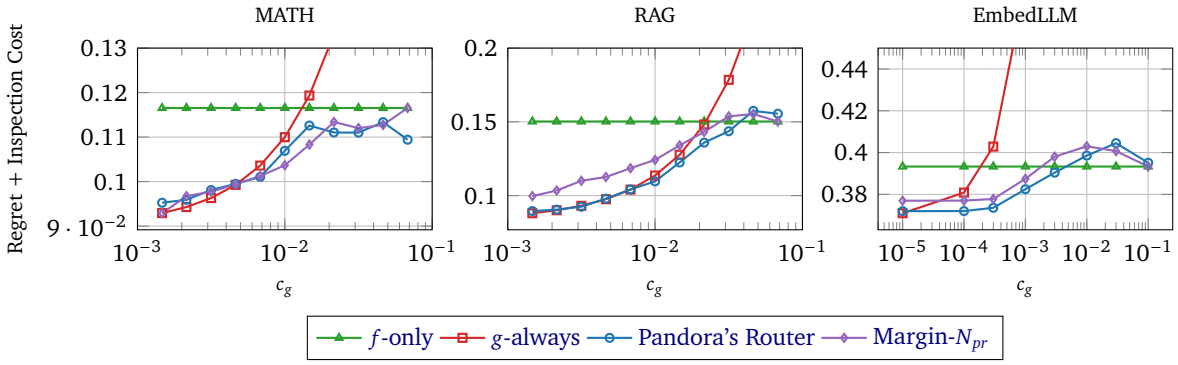

Figure 2 | Routing performance on the MATH, RAG, and EmbedLLM domains for varying costs of querying g, with $f = \text{KNN}_3$. The ideal value estimation policy would minimize regret + inspection cost at every cost level. Pandora's Router achieves near-minimal total cost across the full range of c_g (where $c_m = c_g$ for all m), querying g frequently when it is cheap and rarely when it is expensive.

The primary metric is *regret + inspection cost*: routing regret (the gap between the selected specialist's true, realized reward and the oracle best specialist in hindsight) plus the total cost of all g-queries. Although our method generalizes to heterogeneous costs, for simplicity we use a uniform inspection cost $c_m = c_g \; \forall m \in \{1, \ldots, M\}$, and sweep c_g to trace out the cost versus performance frontier.

Results. Results for the full set of baselines are shown in Table 2 for all three datasets, averaging over all values of c_g (see Section D.4 for results per c_g; the values tested correspond to those in Figure 2). Pandora's Router is reliably the best. The two inspection ablations, Margin-N_{pr} and Random-N_{pr}, borrow the query budget from Pandora's Router, but achieve higher regret because they do not use the budget as effectively. Differences in particular between Margin-N_{pr}, which is the most competitive comparison, and Pandora's Router at each specific cost level are shown in Figure 2 (for clarity in the figure, we compare Pandora's Router with f-only, g-only, and the margin baseline). Pandora minimizes regret plus inspection cost at nearly all cost levels c_g, in all settings. When the cost of g is low, Pandora's Router queries it for nearly every specialist, and all methods except f-only (which never queries g) perform similarly. As c_g increases, the g-always baseline continues to make a fixed number of queries regardless of cost, while f-only still never queries at all. Pandora's Router interpolates between these extremes, querying g only when the value of information exceeds the cost. As a result, the total regret + inspection cost of Pandora's Router tracks the lower envelope of the baselines across the full c_g range.

As a secondary analysis, Figure 6 in Section D.3 shows the cost-performance tradeoff offered by Pan-

dora's Router, in terms of monetary costs and routing regret. The costs are computed from the Gemini API prices described in Section 2.2. When $c_g = 0.001$, we are in a setting in which users are eager to pay for queries that improve routing; here Pandora's Router obtains regret that almost matches g-only, while incurring much lower inspection costs. When $c_g = 0.1$, we are in a setting in which users are unwilling to pay for queries to g; here Pandora's Router never queries g, and routing performance and cost match the f-only baseline. These tradeoffs are dynamically navigated by Pandora's Router.

Note that in the MATH experiment the aggregate results for Pandora's Router and Margin-N_{pr} are nearly identical. Since there are only two routing targets, regret is determined largely by the decision about how many g values to query: with two queries, both methods always select the g-maximizer; with zero queries both methods almost always select the f-maximizer (except in rare cases where the reserve price ordering is different from the ordering of f). Because Margin-N_{pr} inherits the query budget (N_{pr}) from Pandora's Router, it is unsurprising that its overall performance is very similar.

5. Pandora's Bidder

We now derive a simple but interesting extension of Pandora's Router to a decentralized setting in which the specialists control their own value estimates, and use them to participate in a marketplace. Pandora's Router assumes that a single decision-maker controls which boxes to open and which specialist to select. In many settings, however, the specialists themselves are better positioned to estimate their own value: a retrieval-augmented specialist has access to its own corpus, a math specialist can execute partial computations, and a domain expert may have proprietary benchmarks. None of these resources need be visible to a centralized router. Decentralization also has a number of practical advantages: new specialists can join without retraining the routing model, and the cost of value estimation is borne by the specialists rather than the platform.

As an alternative to a centralized router, consider a market-based mechanism where each specialist can purchase the right to answer a query. We formalize this via a posted-price mechanism corresponding to a single stage of the ascending-price framework with costly preference elicitation studied by Parkes (2005). We consider a leave-one-out setting with a single strategic specialist. A platform collects nonstrategic value estimates from $M - 1$ specialists and posts the best estimate as the price at which the remaining specialist can claim the query. The strategic specialist must then apply the same value-of-information (VoI) reasoning from Pandora's Router to decide whether this price is worth accepting, and whether or not to pay to observe a refined estimate before making this decision.

Concretely, for a given query x, the platform solicits G_j from each of the $M - 1$ nonstrategic specialists and sets the posted price as $p = \max_{j \neq m} G_j$. Note that the platform could incorporate a markup to extract profit, but here we simply use the best competing estimate. The strategic specialist m then faces a decision: accept the price (winning the right to answer the query, instead of the platform's choice), or decline (in which case the platform routes to the specialist that set the price). Specialist m begins with only the cheap estimate, which yields a predicted mean $\mu_m(x) = h(\mathbf{f})_m$ for G_m, as in Equation (4). The expected gain from paying c_m to observe G_m before responding to the price p is

$$\text{VoI}(p) = \mathbb{E}[(G_m - p)^+] - (\mu_m - p)^+. \tag{6}$$

This is the difference in expected profit between deciding after observing G_m and deciding based only on μ_m. The structure mirrors the reservation price computation for Pandora's Router in §4, but with the price p playing the role of the outside option. When p is low relative to μ_m, the specialist can confidently accept based on f alone; when p is high, the specialist can confidently decline. The value of information peaks when p is close to μ_m and the correct action is uncertain. The condition $\text{VoI}(p) = c_m$ defines an *interval* $[p_{\text{lo}}, p_{\text{hi}}]$ within which it is worth paying for the refined estimate (Parkes, 2005).

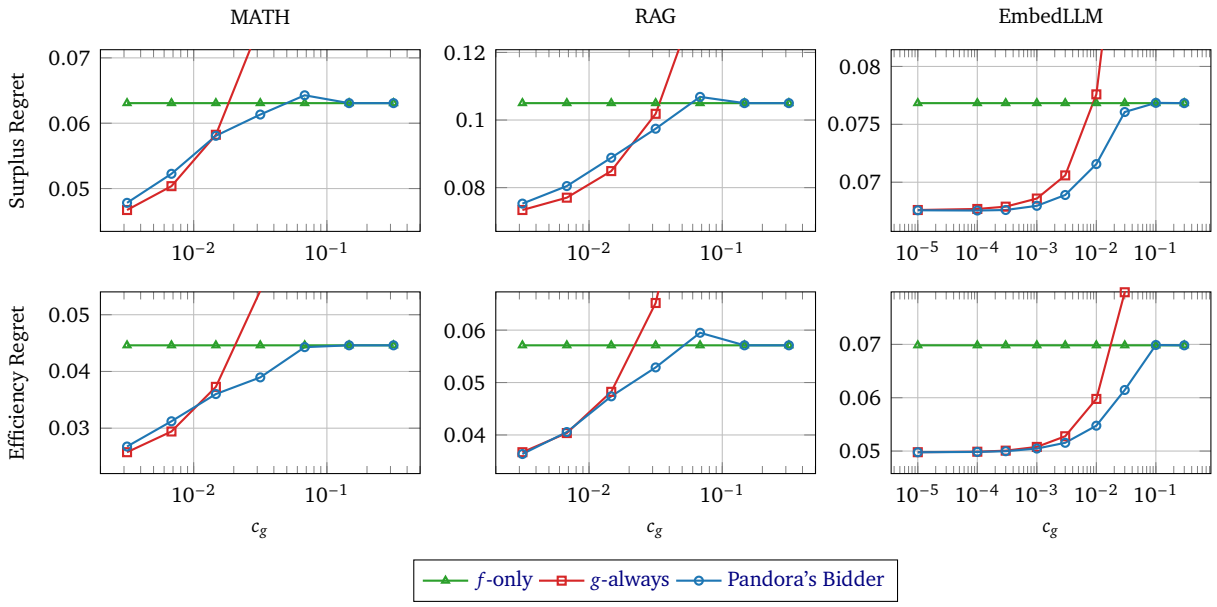

Figure 3 | Results for the posted-price auction across the three experimental domains for varying refinement cost c_g. The VoI-based bidding strategy interpolates between querying g when cheap and declining when expensive, while static baselines either over-invest or under-invest in refinement.

Using the same Gaussian model as in Equations (4) and (5), the boundaries satisfy

$$(\mu_m - p)\,\Phi(\alpha_m) + \sigma_m\,\phi(\alpha_m) = c_m + (\mu_m - p)^+ \tag{7}$$

with $\alpha_m = (\mu_m - p)/\sigma_m$, and Φ and ϕ the standard normal CDF and PDF. Like the reservation price, these values are obtained via standard root-finding techniques. The resulting bidding strategy pays c_m to observe G_m when $p_{\text{lo}} \le p \le p_{\text{hi}}$, accepting if $G_m > p$. Outside this interval, refinement is not worth it: the specialist simply accepts if $p < p_{\text{lo}}$ and declines if $p > p_{\text{hi}}$.

Experimental setup. We evaluate Pandora's Bidder in a leave-one-out setting across the three domains from Section 2.1. For each query, one specialist is designated as the strategic bidder; the remaining $M-1$ specialists submit their g estimates nonstrategically, and the platform posts the best as the price via $p = \max_{j \neq m} G_j$. The strategic bidder then applies the VoI-based acceptance strategy above. We rotate the held-out specialist across all M models and report the average. As in the centralized routing experiments, we sweep over c_g to trace out the cost versus performance frontier.We measure two quantities: (1) *specialist surplus*, which is $R_m - p$ when the strategic specialist m accepts the price and 0 when it declines, as well as any incurred estimation costs; and (2) *allocative efficiency*, which is the true reward $R_{\hat{m}}$ of the winning specialist (m if it accepts, otherwise the platform-chosen specialist), minus the m's estimation costs. We exclude competitors' estimation costs; by treating their g estimates as given, we isolate the contribution of the VoI-based policy. We compare against the g-always baseline (the bidder always pays for g) and the f-only baseline (the bidder always decides based on f). For both metrics, we measure the regret versus an oracle that knows G_m for free.

Results. Figure 3 plots allocative efficiency and bidder surplus for different c_g. Pandora's Bidder closely tracks the lower envelope of the regret of the two baselines across the full c_g range. When c_g is low, the bidder frequently refines its estimate, winning queries where it holds a true advantage. As c_g increases, the bidder selectively defaults to the cheap estimate or declines the price, avoiding the cost collapse of the g-always baseline. The f-only baseline, conversely, leaves efficiency on the table at low costs by never investing in refinement. With decentralization, however, maximizing local surplus does not al-

ways lead to better overall allocative efficiency. For example, when the price is inaccurate (set using the cheap estimate f rather than g for the $M-1$ competitors) the strategic bidder captures higher surplus at the cost of overall allocative efficiency (see additional results in Section D.2). Overpriced queries offer insufficient expected profit to incentivize the bidder; declining them protects the bidder's surplus but assigns the input to a less capable specialist. This tension is absent in the centralized setting, where the router internalizes all costs and optimizes global welfare directly. Extending Pandora's Bidder to multi-round ascending-price mechanisms (Parkes, 2005) may recover some of this efficiency loss.

6. Related Work

Information value theory. Many scenarios require agents to decide whether to spend effort learning more about the world or to exploit what they already know. An early formalization of this decision problem is *information value theory* (Howard, 1966), which provides the general framework for Equations (1) and (6). Bayesian Q Learning is an application of information value theory that quantifies the downstream value of actions that increase the precision of the agent's beliefs about state-action values (Dearden et al., 1998). While similar in spirit, the core ideas of Bayesian Q Learning cannot easily be applied to model routing because state-action pairs cannot be visited more than once; in this work, however, we show that the form of the model routing decision problem admits specialized efficient algorithms based on Pandora's Box (formalized as Pandora's Router in Section 4).

LLM routing. Routing user inputs within a collection of language models has been proposed as a way to reduce costs while maintaining high accuracy (Chen et al., 2023) and to leverage complementary capabilities of heterogenous models (Shnitzer et al., 2023). There are now several routing benchmarks (e.g., Feng et al., 2026; Hu et al., 2024), of which we use EmbedLLM (Zhuang et al., 2025). Most routing approaches require predicting the answer quality for each model on a given input, using pointwise (Jitkrittum et al., 2025) or pairwise (Ong et al., 2025) data. In all of this prior work, value estimation is treated as a cost-free operation with a fixed error profile. Our contribution in Pandora's Router in Section 4 is to treat value estimation as a decision involving a cost-performance tradeoff of its own.

Adaptive retrieval and tool use. Closely related to routing is the decision of whether to invoke external tools or retrievers. Adaptive RAG frameworks (Jeong et al., 2024) dynamically adjust retrieval strategies depending on the query complexity to save computational costs. Similarly, recent evaluations of adaptive retrieval (Moskvoretskii et al., 2025) highlight the tension between using internal LLM uncertainty estimation vs. lightweight external heuristics. We formalize this exact tradeoff—that is, deciding whether to spend computational power on self-assessment or tool-planning—through a value-of-information perspective that is both theoretically principled and empirically effective.

Pandora's Box and optimal search. Prior work on sequential search with costly inspection is reviewed in §3. The Pandora's Box problem is also used as a motivating framework for LLM reasoning about cost-uncertainty tradeoffs by Ding et al. (2026), as well as by Xie et al. (2024) for cost-aware Bayesian optimization in which the reservation price (for obligatory inspection) is used to drive an acquisition function. In contrast, we instantiate the non-obligatory variant of the Pandora's Box algorithm to drive LLM routing specifically, and we derive closed-form value-of-information expressions under a practical Gaussian signal model that make our algorithm simple to implement and execute. On a more technical level, we focus on a *committing policy* approximation to the non-obligatory inspection variant of the Pandora's Box problem (Beyhaghi and Kleinberg, 2019). Another class of approximations uses randomization (Beyhaghi and Cai, 2023; Scully and Doval, 2024), which we may consider in future work.

Decentralized AI and multi-agent systems. Decentralized coordination through market mechanisms has a long history in AI, from ContractNet (Davis and Smith, 1983), which introduced negotiation-based task allocation, to AgentNet (Yang et al., 2025), which builds dynamic graph topologies between LLM agents. Our work on Pandora's Bidder in Section 5 is most informed by auction-based approaches,

which were applied to ContractNet by Sandholm (1993) and linked to decentralized RL by Chang et al. (2020). The combination of auctions with cost-accuracy tradeoffs in value estimation is generally intractable, but Parkes (2005) offers empirically-validated heuristics. For a survey on the problem of delegation between AI components, see Tomašev et al. (2026). Auctions and related mechanisms have received increasing attention as an approach for decentralized orchestration of LLMs (Collina et al., 2025; Dütting et al., 2024; Nourzad et al., 2025; Tarasova et al., 2025; Zhao et al., 2025), but these frameworks generally assume that agents know their own valuations *a priori*, while we describe how agents can reason about costly value estimation when they are unknown.

7. Conclusion

Value-based routing is a natural approach to optimizing the allocation of computing power to AI model inputs. We have argued that value estimation itself brings cost-accuracy tradeoffs, and we show how the Pandora's Box problem offers a unified framework for value-based routing under heterogeneous value estimators. We also extend the framework to a decentralized setting, in which specialists reason about the value of information in an auction-based allocation mechanism.

Limitations. Several limitations suggest promising directions for future work. The Gaussian signal model, while tractable, may not capture the heavy tails or multimodality present in some domains (see Section D.6 for coverage statistics and a pilot study of a non-Gaussian signal model). The two-estimator restriction (f and g) could be relaxed to chains or trees of estimators with varying performance tradeoffs. And the myopic VoI computation in the leave-one-out auction does not account for strategic anticipation of future bids; an extension to a multi-round ascending-price auction (Parkes, 2005) is a natural next step. Furthermore, as discussed in Section 5, when bidding against weaker adversaries that place poor bids, Pandora's Bidder can improve its own utility at the expense of the overall welfare.

Acknowledgements

We thank Alekh Agarwal, Jonathan Berant, and Ian Gemp for helpful research discussions, and Chris Dyer and Artem Sokolov for helpful comments and feedback on the manuscript. This research also benefited from early-stage discussions with Will Dabney.

References

H. Beyhaghi and L. Cai. Pandora's problem with nonobligatory inspection: Optimal structure and a PTAS. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 803–816, 2023.

H. Beyhaghi and R. Kleinberg. Pandora's problem with nonobligatory inspection. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 131–132, 2019.

M. Chang, S. Kaushik, S. M. Weinberg, T. Griffiths, and S. Levine. Decentralized reinforcement learning: Global decision-making via local economic transactions. In *International Conference on Machine Learning*, pages 1437–1447. PMLR, 2020.

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

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

N. Collina, S. Goel, A. Roth, E. Ryu, and M. Shi. Emergent alignment via competition. *arXiv preprint arXiv:2509.15090*, 2025.

R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109, 1983.

R. Dearden, N. Friedman, S. Russell, et al. Bayesian q-learning. *Aaai/iaai*, 1998:761–768, 1998.

W. Ding, N. Tomlin, and G. Durrett. Calibrate-then-act: Cost-aware exploration in LLM agents. *arXiv preprint arXiv:2602.16699*, 2026.

L. Doval. Whether or not to open Pandora's box. *Journal of Economic Theory*, 175:127–158, 2018.

P. Dütting, V. Mirrokni, R. P. Leme, H. Xu, and S. Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, 2024.

J. Eisenstein, R. Aghajani, A. Fisch, D. Dua, F. Huot, M. Lapata, V. Zayats, and J. Berant. Don't lie to your friends: Learning what you know from collaborative self-play. *arXiv preprint arXiv:2503.14481*, 2025.

S. Feng, Y. Bai, Z. Yang, Y. Wang, Z. Tan, J. Yan, Z. Lei, W. Ding, W. Shi, H. Wang, et al. Moco: A one-stop shop for model collaboration research. *arXiv preprint arXiv:2601.21257*, 2026.

H. Fu, J. Li, and D. Liu. Pandora box problem with nonobligatory inspection: Hardness and approximation scheme. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 789–802, 2023.

B. Gao, F. Song, Z. Yang, Z. Cai, Y. Miao, Q. Dong, L. Li, C. Ma, L. Chen, R. Xu, Z. Tang, B. Wang, D. Zan, S. Quan, G. Zhang, L. Sha, Y. Zhang, X. Ren, T. Liu, and B. Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=yaqPf0KAlN.

Gemma Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, T. Mesnard, G. Cideron, J. bastien Grill, S. Ramos, E. Yvinec, M. Casbon, E. Pot, I. Penchev, G. Liu, F. Visin, K. Kenealy, L. Beyer, X. Zhai, A. Tsitsulin, R. Busa-Fekete, A. Feng, N. Sachdeva, B. Coleman, Y. Gao, B. Mustafa, I. Barr, E. Parisotto, D. Tian, M. Eyal, C. Cherry, J.-T. Peter, D. Sinopalnikov, S. Bhupatiraju, R. Agarwal, M. Kazemi, D. Malkin, R. Kumar, D. Vilar, I. Brusilovsky, J. Luo, A. Steiner, A. Friesen, A. Sharma, A. Sharma, A. M. Gilady, A. Goedeckemeyer, A. Saade, A. Feng, A. Kolesnikov, A. Bendebury, A. Abdagic, A. Vadi, A. György, A. S. Pinto, A. Das, A. Bapna, A. Miech, A. Yang, A. Paterson, A. Shenoy, A. Chakrabarti, B. Piot, B. Wu, B. Shahriari, B. Petrini, C. Chen, C. L. Lan, C. A. Choquette-Choo, C. Carey, C. Brick, D. Deutsch, D. Eisenbud, D. Cattle, D. Cheng, D. Paparas, D. S. Sreepathihalli, D. Reid, D. Tran, D. Zelle, E. Noland, E. Huizenga, E. Kharitonov, F. Liu, G. Amirkhanyan, G. Cameron, H. Hashemi, H. Klimczak-Plucińska, H. Singh, H. Mehta, H. T. Lehri, H. Hazimeh, I. Ballantyne, I. Szpektor, I. Nardini, J. Pouget-Abadie, J. Chan, J. Stanton, J. Wieting, J. Lai, J. Orbay, J. Fernandez, J. Newlan, J. yeong Ji, J. Singh, K. Black, K. Yu, K. Hui, K. Vodrahalli, K. Greff, L. Qiu, M. Valentine, M. Coelho, M. Ritter, M. Hoffman, M. Watson, M. Chaturvedi, M. Moynihan, M. Ma, N. Babar, N. Noy, N. Byrd, N. Roy, N. Momchev, N. Chauhan, N. Sachdeva, O. Bunyan, P. Botarda, P. Caron, P. K. Rubenstein, P. Culliton, P. Schmid, P. G. Sessa, P. Xu, P. Stanczyk, P. Tafti, R. Shivanna, R. Wu, R. Pan, R. Rokni, R. Willoughby, R. Vallu, R. Mullins, S. Jerome, S. Smoot, S. Girgin, S. Iqbal, S. Reddy, S. Sheth, S. Põder, S. Bhatnagar, S. R. Panyam, S. Eiger, S. Zhang, T. Liu, T. Yacovone, T. Liechty, U. Kalra, U. Evci, V. Misra, V. Roseberry, V. Feinberg, V. Kolesnikov, W. Han, W. Kwon, X. Chen, Y. Chow, Y. Zhu, Z. Wei, Z. Egyed, V. Cotruta, M. Giang, P. Kirk, A. Rao, K. Black, N. Babar, J. Lo, E. Moreira, L. G. Martins, O. Sanseviero, L. Gonzalez, Z. Gleicher, T. Warkentin, V. Mirrokni, E. Senter, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, Y. Matias, D. Sculley, S. Petrov, N. Fiedel, N. Shazeer, O. Vinyals,

J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, J.-B. Alayrac, R. Anil, Dmitry, Lepikhin, S. Borgeaud, O. Bachem, A. Joulin, A. Andreev, C. Hardin, R. Dadashi, and L. Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

E. Gergatsouli and C. Tzamos. Weitzman's rule for Pandora's box with correlations. *Advances in Neural Information Processing Systems*, 36:12644–12664, 2023.

Google DeepMind, March 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Flash-Lite-Model-Card.pdf.

Harvard-MIT Mathematics Tournament. HMMT past problems and solutions. https://www.hmmt.org/. Accessed: 2026-04-22.

C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, J. Liu, L. Qi, Z. Liu, and M. Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

R. A. Howard. Information value theory. *IEEE Transactions on systems science and cybernetics*, 2(1): 22–26, 1966.

Q. J. Hu, J. Bieker, X. Li, N. Jiang, B. Keigwin, G. Ranganath, K. Keutzer, and S. K. Upadhyay. Routerbench: A benchmark for multi-LLM routing system. In *International Conference on Machine Learning*, 2024.

S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

W. Jitkrittum, H. Narasimhan, A. S. Rawat, J. Juneja, C. Wang, Z. Wang, A. Go, C.-Y. Lee, P. Shenoy, R. Panigrahy, et al. Universal model routing for efficient LLM inference. *arXiv preprint arXiv:2502.08773*, 2025.

A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170, 2023.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

J. Lee, F. Chen, S. Dua, D. Cer, M. Shanbhogue, I. Naim, G. H. Ábrego, Z. Li, K. Chen, H. S. Vera, et al. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*, 2025.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

A. Liang, J. Berant, A. Fisch, A. Goyal, K. Krishna, and J. Eisenstein. Plantain: Plan-answer interleaved reasoning. *arXiv preprint arXiv:2512.03176*, 2025.

K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.109. URL https://aclanthology.org/2024.naacl-long.109/.

A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Mathematical Association of America. American invitational mathematics examination (AIME). https://maa.org/math-competitions/amc-1012. Accessed: 2026-04-22.

V. Moskvoretskii, M. Marina, M. Salnikov, N. Ivanov, S. Pletenev, D. Galimzianova, N. Krayko, V. Konovalov, I. Nikishina, and A. Panchenko. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6355–6384. Association for Computational Linguistics, 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.319. URL https://aclanthology.org/2025.acl-long.319/.

N. Nourzad, H. Yang, S. Chen, and C. Joe-Wong. DR. WELL: Dynamic reasoning and learning with symbolic world model for embodied llm-based multi-agent collaboration. *CoRR*, abs/2511.04646, 2025. doi: 10.48550/ARXIV.2511.04646.

I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

D. C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and Artificial Intelligence*, 44(3):269–302, 2005.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *AAAI*, volume 93, pages 256–262, 1993.

C. Sciavolino, Z. Zhong, J. Lee, and D. Chen. Simple entity-centric questions challenge dense retrievers. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.496. URL https://aclanthology.org/2021.emnlp-main.496/.

Z. Scully and L. Doval. Local hedging approximately solves pandora's box problems with nonobligatory inspection. *arXiv preprint arXiv:2410.19011*, 2024.

T. Shnitzer, A. Ou, M. Silva, K. Soule, Y. Sun, J. Solomon, N. Thompson, and M. Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

K. Sun, Y. Xu, H. Zha, Y. Liu, and X. L. Dong. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.18/.

E. Tarasova, V. Erofeeva, O. Granichin, and K. Chernikov. Decentralized adaptive task allocation for dynamic multi-agent systems. *Scientific Reports*, 15(1):39226, 2025.

N. Tomašev, M. Franklin, and S. Osindero. Intelligent AI delegation. *arXiv preprint arXiv:2602.11865*, 2026.

M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979.

Q. Xie, R. Astudillo, P. I. Frazier, Z. Scully, and A. Terenin. Cost-aware bayesian optimization via the pandora's box gittins index. *Advances in Neural Information Processing Systems*, 37:115523–115562, 2024.

Y. Yang, H. Chai, S. Shao, Y. Song, S. Qi, R. Rui, and W. Zhang. AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems*, 2025.

S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2022.

C. Zhao, Q. Hu, S. Song, D. Chen, Z. Han, J. Xu, and B. Zheng. Llm-auction: Generative auction towards llm-native advertising. *ArXiv*, abs/2512.10551, 2025.

R. Zhuang, T. Wu, Z. Wen, A. Li, J. Jiao, and K. Ramchandran. EmbedLLM: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Contents

A. Pandora's Routing Objective

In Section 4, we connect routing to Pandora's Box by treating each specialist as a box and the costly value estimate as the value revealed by opening that box. The revealed value, however, is not R_m itself, but G_m, which is the costly, more accurate estimate of R_m. In this section we formally justify the objective of searching for the maximum value of G_m when R_m is unknown.

Let $\mathcal{I}_0$ denote the zero-cost information available before any costly queries, including the prompt and the cheap estimates $\mathbf{F}$. Inspecting specialist m reveals costly information Z_m at cost c_m. In the ideal calibrated case, the value revealed by opening box m is

$$G_m^\star = \mathbb{E}[R_m \mid \mathcal{I}_0, Z_m]. \tag{8}$$

Thus $G_m^\star$ is the posterior decision value of specialist m after Z_m has been acquired.

Admissible Policies. We restrict attention to sequential policies that only use information that has actually been observed. Let $\mathcal{F}_0 = \mathcal{I}_0$ be the zero-cost information state, and let $O_0 = \emptyset$ be the set of opened specialists. At each step t, a policy either stops, or selects an unopened specialist $A_{t+1} \in \mathcal{M} \setminus O_t$ to inspect. The stopping decision and the choice of A_{t+1} are required to be $\mathcal{F}_t$-measurable. If A_{t+1} is inspected, the policy pays cost $c_{A_{t+1}}$, observes $Z_{A_{t+1}}$, and the state updates to

$$O_{t+1} = O_t \cup \{A_{t+1}\}, \qquad \mathcal{F}_{t+1} = \mathcal{F}_t \vee \sigma(A_{t+1}, Z_{A_{t+1}}).$$

where $\sigma(A_{t+1}, Z_{A_{t+1}})$ denotes the sigma algebra generated by A_{t+1} and $Z_{A_{t+1}}$. Let τ be the stopping time, $O = O_\tau$ the final opened set, and

$$C_O = \sum_{i \in O} c_i$$

the total inspection cost. The final selected specialist $\widehat{M}$ must be measurable with respect to the terminal information $\mathcal{F}_\tau$. We call such a policy *admissible*. In the obligatory-inspection case we require $\widehat{M} \in O$ almost surely; in the non-obligatory case, $\widehat{M}$ may also be unopened.

Remark 1. *Here $\mathcal{I}_0$ should be interpreted as the information available at zero computational cost. Costly computations are represented by Z_m and are not included in the information state until specialist m is inspected, even when they are deterministic functions of the prompt.*

Proposition 1 (Opened-value reduction)**.** *Suppose the collection $\{(R_m, Z_m)\}_{m=1}^M$ is conditionally independent given $\mathcal{I}_0$. Let Π be any admissible policy that opens a random set O, incurs cost $C_O = \sum_{i \in O} c_i$, and selects an opened specialist $\widehat{M} \in O$ almost surely. Define*

$$G_m^\star = \mathbb{E}[R_m \mid \mathcal{I}_0, Z_m].$$

Then

$$\mathbb{E}[R_{\widehat{M}} - C_O] = \mathbb{E}[G^\star_{\widehat{M}} - C_O].$$

Consequently, among policies that select opened specialists, maximizing expected realized reward is equivalent to maximizing the opened posterior decision value net of inspection costs.

Proof. By admissibility, $\widehat{M}$ and C_O are $\mathcal{F}_\tau$-measurable. Hence the tower property gives

$$\mathbb{E}[R_{\widehat{M}} - C_O] = \mathbb{E}\left[\mathbb{E}[R_{\widehat{M}} \mid \mathcal{F}_\tau] - C_O\right].$$

Because $\widehat{M}$ is $\mathcal{F}_\tau$-measurable,

$$\mathbb{E}[R_{\widehat{M}} \mid \mathcal{F}_\tau] = \sum_{m=1}^{M} \mathbf{1}\{\widehat{M} = m\} \mathbb{E}[R_m \mid \mathcal{F}_\tau].$$

On the event $\{\widehat{M} = m\}$, the specialist m has been opened, so $\mathcal{F}_\tau$ contains Z_m. The remaining terminal information consists of zero-cost information and signals from other opened specialists, together with decisions that are measurable functions of those observed signals. Conditional independence implies that this additional information does not change the posterior mean of R_m once $(\mathcal{I}_0, Z_m)$ is known. Therefore

$$\mathbb{E}[R_m \mid \mathcal{F}_\tau] = \mathbb{E}[R_m \mid \mathcal{I}_0, Z_m] = G_m^\star$$

on the event that m has been opened. Substituting this into the previous equation yields

$$\mathbb{E}[R_{\widehat{M}} \mid \mathcal{F}_\tau] = G_{\widehat{M}}^\star,$$

and therefore

$$\mathbb{E}[R_{\widehat{M}} - C_O] = \mathbb{E}[G_{\widehat{M}}^\star - C_O].$$

□

Corollary 1 (Sealed value of an unopened specialist)**.** *Let $\mathcal{H}$ be any information state generated by an admissible policy before specialist m has been opened. Define the value that would be revealed by opening m at this information state as*

$$G_{m|\mathcal{H}}^\star = \mathbb{E}[R_m \mid \mathcal{H}, Z_m].$$

If specialist m is selected without being opened, then its decision value is

$$\mathbb{E}[R_m \mid \mathcal{H}] = \mathbb{E}[G_{m|\mathcal{H}}^\star \mid \mathcal{H}].$$

Thus an unopened specialist is evaluated by the posterior expectation of the value that would have been revealed by opening it.

Proof. This is the tower property:

$$\mathbb{E}[G_{m|\mathcal{H}}^\star \mid \mathcal{H}] = \mathbb{E}[\mathbb{E}[R_m \mid \mathcal{H}, Z_m] \mid \mathcal{H}] = \mathbb{E}[R_m \mid \mathcal{H}].$$

□

Proposition 1 explains why opened boxes can be searched using posterior decision values rather than realized rewards. Corollary 1 explains the corresponding role of unopened boxes in the non-obligatory inspection variant: an unopened specialist is a sealed outside option, whose value is the current posterior expectation of the value that opening would reveal. The backup-price construction in Pandora-NI is a way to compare this sealed option against the opened values of the other boxes.

The proposition and corollary are exact for the ideal posterior values. In practice, we do not observe $G_m^\star$ directly. Our costly estimator returns a learned score

$$G_m = g_m(X, Z_m),$$

which we use as a plug-in approximation to $G_m^\star$. If g_m were trained by squared loss with unlimited data and sufficient model capacity, its population regression target would be the conditional mean in Equation (8). With finite data and a restricted model class, G_m is only an approximation. The Pandora reduction should therefore be read as exact for calibrated posterior decision values, and approximate when implemented with learned scores. The empirical question is whether the learned G_m is accurate and calibrated enough to improve routing decisions for the original reward objective.

B. Dataset Details

MATH. For this evaluation, we compile a diverse corpus of mathematical problems spanning multiple levels of difficulty. This includes the standard Hendrycks MATH dataset (Hendrycks et al., 2021), which provides roughly 12500 problems across seven mathematical domains. We also incorporate the Omni-Math dataset (Gao et al., 2025), an Olympiad-level benchmark designed to assess advanced reasoning capabilities. From this dataset, we utilize a subset of 1625 hard problems, filtered by problems which have between zero and 16 derivation steps. Furthermore, we include 969 American Invitational Mathematics Examination (AIME) problems (He et al., 2024; Mathematical Association of America), filtered to preclude any overlap with the MATH dataset, alongside 1419 problems from the Harvard-MIT Mathematics Tournament (HMMT) (Harvard-MIT Mathematics Tournament) (filtered to avoid overlap with Omni-Math). For the MATH dataset, we retain its standard 5000-problem test split. The remaining datasets are divided using a balanced 50/50 train-test split, resulting in a final aggregate corpus of 9504 training and 7008 testing examples. Prompts are shown in Appendix C.2.

RAG. Our RAG evaluation setting builds on prior work that combines two retrieval specialists (Wikipedia and PubMed) along with a low-cost model without retrieval (Eisenstein et al., 2025). Following prior work, we train and evaluate on a collection of factoid questions from PopQA (Mallen et al., 2023), Entity Questions (Sciavolino et al., 2021), Natural Questions (Kwiatkowski et al., 2019), and BioASQ (Krithara et al., 2023). We use 18661 questions for training, 1600 for test, and 400 for calibration, drawing an equal proportion from each of the four datasets. We use the same retrieval pipelines as Eisenstein et al. (2025), and generate responses using Gemini-3.1-Flash-Lite. Prompts are shown in Appendix C.2. We use 4 few-shot examples per dataset (16 total) in order to prompt the LLM to generate answers in the right format. The responses are judged for correctness against the ground-truth answers given in each dataset. For each question, we sample 8 responses and score each for correctness. The reward is the average correctness score, less a constant cost penalty of 0.05 for systems that use retrieval (i.e., Wikipedia and Pubmed specialists). This cost value was chosen to make the routing problem non-trivial (i.e., the model without retrieval can compete when retrieval is not necessary). To avoid false negatives inherent in exact string matching, we use Gemini-3.1-Flash as an LLM-as-a-judge using the same annotation prompt as Sun et al. (2024) (Appendix A.1, Prompt 2), which was measured to have very strong agreement with human judgements (98%).

EmbedLLM. The EmbedLLM dataset is composed of questions from benchmarks such as GPQA (Rein et al., 2023), GSM8K (Cobbe et al., 2021), and MMLU (Hendrycks et al., 2020), along with the scores from 123 models (Zhuang et al., 2025). We use this dataset without modification. Following Jitkrittum et al. (2025), we set the model costs to be a linear multiple of the number of parameters. Specifically we apply a cost of 1 per trillion parameters, so that, e.g., Qwen-1.5-7B-Chat has a cost of .00772 and Qwen-1.5-32B-chat has a cost of .0325. Seven models were excluded because the parameter count could not be determined (e.g., Claude). We use 16756 prompts for training, 2436 for test, and 609 for calibration.

C. Implementation details

C.1. Value estimators

We give additional details for the value estimators f_m and g_m described in Section 2.1.

Embedding-based estimation (KNN). The cheap value estimator computes an embedding $e(x)$ of the prompt and retrieves the k nearest neighbors $\mathcal{N}_k(x)$ from a calibration set, returning:

$$f_m^{\mathrm{KNN}}(x) = \frac{1}{k} \sum_{x' \in \mathcal{N}_k(x)} R_m(x'), \tag{9}$$

where proximity is measured by cosine similarity in the embedding space. This estimator is fast but

limited to patterns that are visible in the embedding space. We use Gemini Embedding 2 (Lee et al., 2025) as our embedder, and use $k = 3$ for the KNN computation.

Fine-tuned estimation (SFT). A small language model h_θ is fine-tuned with a regression loss to predict the cost-adjusted reward from an input context $z_m(x)$:

$$\min_\theta \sum_{(x,m)} \left(h_\theta(z_m(x)) - R_m(x)\right)^2. \tag{10}$$

The context $z_m(x)$ can be the prompt alone (SFT-prompt), the prompt with retrieval results (SFT-retrievals), or the prompt with partial reasoning traces (SFT-CoT-k). We use Gemini-2.5-Flash-Lite as the base model for fine-tuning. Targets are real-valued. Prompts are shown in Appendix C.2.

C.2. Prompts

Math task: prompt with reasoning.

Solve the following math problem. Show your work step-by-step, explaining your reasoning clearly. After your reasoning, provide the final answer enclosed in \boxed{} tags.

Problem:
{problem}

Step-by-step solution:
⟨reasoning here⟩

The final answer is \boxed{answer}.

Math task: SFT value estimator prompt using CoT.

Question: {question}
Chain-of-Thought: {CoT text}
Specialist: {specialist}
Based on the question and the chain of thought, will the specialist answer correctly? (Predict 1.0 for yes, 0.0 for no)

RAG task: QA prompt with retrievals.

You are a helpful agent whose job is to answer a question. Your answers should be short. For example, if the question is "What is the capital of France?", please answer "Paris", and not "Paris is the capital of France". If you are asked a yes/no question, you may only answer "yes" or "no".

To help you answer the question correctly, you will be given verified information from {corpus}. The verified information may not be necessary, and you can directly answer the question if you are confident that you have the correct answer. The verified information may also not be relevant or sufficient to answer the question. You should still always respond with your best guess.

{few_shot_examples}

QUESTION: {question}
{retrievals}
QUESTION: {question}
ANSWER:

RAG task: standard prompt without retrievals.

You are a helpful agent whose job is to answer a question. Your answers should be short. For example, if the question is "What is the capital of France?", please answer "Paris", and not "Paris is the capital of France". If you are asked a yes/no question, you may only answer "yes" or "no".

{few_shot_examples}

QUESTION: {question}
ANSWER:

RAG task: SFT value estimator prompt using retrievals.

INSTRUCTIONS: Your task is to predict how likely it is that a language model with {capability} correctly answers the following question.

QUESTION: {question}

INFORMATION: Here are the retrieved passages that the language model will have access to:

1. Title: {title_1}
Passage: {passage_1}
2. Title: {title_2}
Passage: {passage_2}
...

QUESTION: {question}

RESPONSE:

RAG task: SFT value estimator prompt without retrievals.

INSTRUCTIONS: Your task is to predict how likely it is that a language model with {capability} correctly answers the following question.

QUESTION: {question}

RESPONSE:

EmbedLLM: SFT value estimator prompt (no additional info).

INSTRUCTIONS: Your task is to predict how likely it is that the model {actor} gives a satisfactory response to the following prompt.

PROMPT: {prompt}

How likely is it?

C.3. Experiment compute resources

Much of our work is based on pre-computed outputs from existing models (e.g., the EmbedLLM data). The main exception is in value estimation, for which we finetuned small Transformer-based encoders ourselves, and inference for the RAG and math domains. Specifically, in the evaluation on the math domain, where we made 15k calls to the Gemini-2.5-Flash-Lite and Gemma3-4B models to compute answers, where the models are served on a TPU cluster with 32 TPUs. This process took less than 1 hour. For the RAG domain, we made ~500k calls to Gemini-2.5-Flash-Lite to generate 8 responses per question per retrieval setting, and then another ~500k calls to Gemini-2.5-Flash to score all the responses for correctness. We also fine-tuned small Transformers as value estimators for each setting, requiring 1-3 hours of training time. The training time was higher for the math domain, needing up to 6 hours, while running on 64 Google TPUs utilizing up to 436.69 GiB of Memory. The Pandora's box algorithm itself was run on CPU at minimal cost.

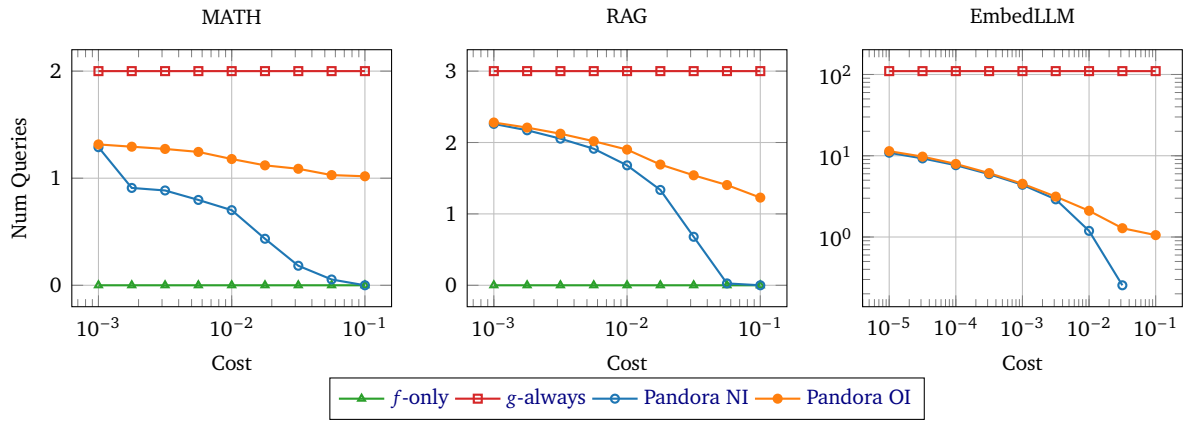

Figure 4 | Number of queries to g during model routing.

D. Supplemental Results

D.1. Number of queries

Figure 4 shows the number of queries executed by each algorithm. Note that Pandora-OI, the obligatory-inspection variant, must always execute at least one query. At higher costs, most of the savings for Pandora-NI derives from identifying prompts for which it is not necessary to query g.

D.2. Auctions against weak opponents

In Section 5, the leave-one-out auction computes a posted price from the costly value estimates $g_{j\neq m}$. Since the optimal value-of-information-based strategy for the strategic specialist m depends on the price it faces (i.e., if it is in the interval $[p_{\text{lo}}, p_{\text{hi}}]$), changing the posted price will also affect the total allocative efficiency and the specialist's surplus. In particular, we will see that when the price does not accurately represent the true *value* of the best competing specialist, the strategic specialist can exploit this the maximize its own profit at the expense of the overall allocative efficiency of the joint system. Figure 5 shows what happens if the posted prices are computed from the weak estimator f_m instead. In both the RAG and EmbedLLM settings, efficiency regret actually increases, even while individual surplus regret improves at most cost levels. This indicates that the leave-one-out bidder is able to improve its individual utility at the expense of overall welfare, by exploiting poor bids from the other participants. This can happen both when the posted price is too low (the specialist will accept the price, even though a competitor might be better), or when the posted price is too high (the specialist will decline the price, even though it is better than all competitors).

D.3. Monetary costs for value functions

As noted in Section 2.2, we estimate monetary costs from prices listed at `https://ai.google.dev/gemini-api/docs/pricing` (retrieved August 1, 2026). In all evaluations, the f value estimator is a small MLP applied to a question embedding. The MLP itself is very cheap to run, and can be assumed to be cost-free. The embedding costs \$0.20 per million tokens, and this can be amortized across target models, although to be conservative about c_g/c_f, we do not account for this amortization here. The g value estimators are priced as follows:

- **SFT-CoT-20**, used in the MATH domain, requires running the target model LLM reasoning process for 20 tokens. The cheapest frontier LLM (Gemini 3.5 flash-lite) costs \$2.50 per million output tokens, and \$0.25 per million input tokens. In the MATH domain, there are approximately 43 tokens per question. This gives $c_f \approx 8.6 \times 10^{-6}$ and $c_g \approx 2.5 \times 20 \times 10^{-6} = 5 \times 10^{-5}$, yielding $c_g/c_f \approx 5.8$.
- **SFT-retrievals**, used in the RAG domain, requires running retrieval, which costs $1.4{\times}10^{-2}$ per model

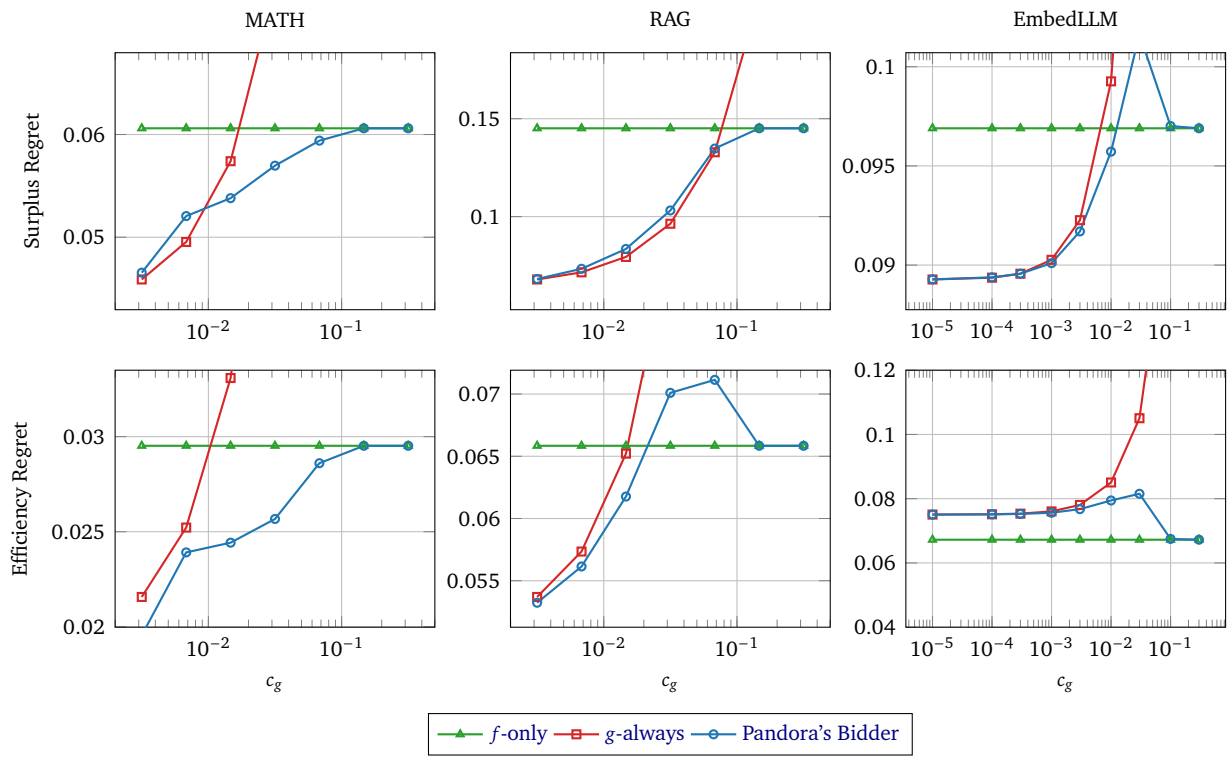

Figure 5 | Specialist surplus regret and total allocative efficiency regret of the posted-price auction, when bidding against opponents who have access only to f, rather than g as in Figure 3.

evaluation. At 10.1 tokens per question in the RAG domain, we have $c_f \approx 10.1 \times .2 \times 10^{-6} \approx 2 \times 10^{-6}$. We conservatively estimate $c_g/c_f > 7000$.

- **SFT-prompt**, used in EmbedLLM, requires LLM inference to generate a single token. As above, this costs 2.5×10^{-6} per output token and 2.5×10^{-7} per input token; the f estimator costs 2×10^{-7} per input token. At 126 tokens per query, this yields a ratio of $c_g/c_f \approx 1.6$. However, this does not account for amortization of the embedding cost c_f across target models, which would be most significant in this domain because it has the largest number of models.

D.4. Numerical results and significance tests

Numerical results and significance tests for the routing evaluations are shown in Tables 3, 4, and 5. In each table, bold indicates the lowest regret + inspection cost, asterisk indicates a statistically significant improvement over all alternatives at $p < .05$ by a paired bootstrap test. These results present a consistent picture: Pandora's router is in the argmin of regret at nearly every cost level. At low costs, it matches g-always; at high costs it matches f-only; at intermediate costs, it often offers the best regret, although this difference is usually not statistically significant with respect to all alternatives.

D.5. Alternative algorithm

We build on prior work on non-obligatory inspection Beyhaghi and Kleinberg (2019), which differs slightly from Algorithm 1: instead of using u_m^{backup} in lines 6, 8, and 9, they use μ_m. In low-cost settings, we have $u_m^{\text{backup}} < E[V_m] = \mu_m$, which means that in these settings, our variant of the algorithm is less likely to commit to non-inspection and is therefore more aligned with Pandora-OI. In high-cost settings, $u_m^{\text{backup}} > E[V_m] = \mu_m$, which means that in these settings, our variant is more likely to commit to non-inspection and is therefore more similar to g-always. Empirically, our variant is slightly

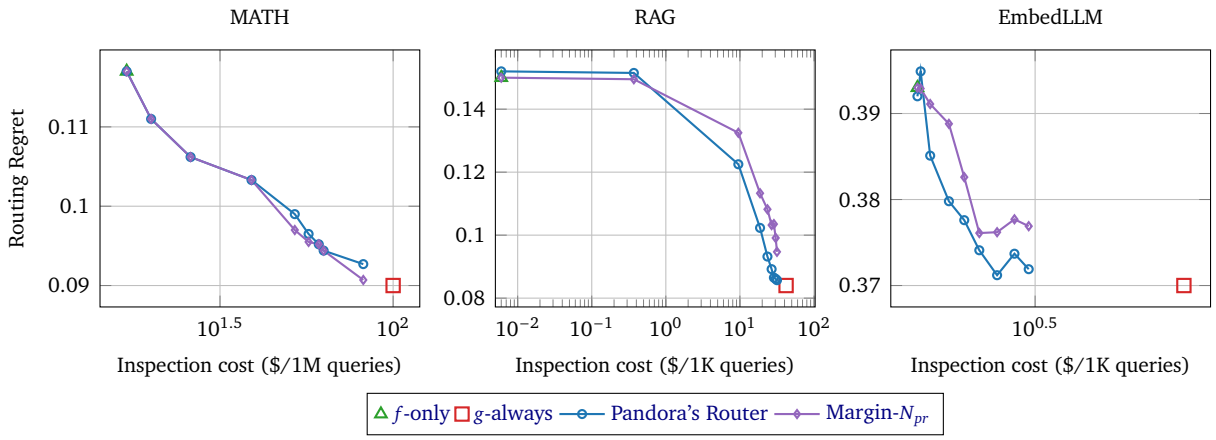

Figure 6 | Routing performance versus monetary inspection cost, as estimated in Section D.3.

Cost	f-only	g-only	Top-2	Coin Flip	Random-N_{pr}	Margin-N_{pr}	Pandora's Router
1.5e-03	0.117	0.093	0.093	0.109	0.194	**0.093**	0.095
2.1e-03	0.117	**0.094**	0.094	0.109	0.193	0.097	0.096
3.2e-03	0.117	**0.096**	0.096	0.110	0.198	0.098	0.098
4.6e-03	0.117	**0.099**	0.099	0.112	0.197	0.099	0.099
6.8e-03	0.117	0.104	0.104	0.114	0.193	0.101	**0.101**
1.0e-02	0.117	0.110	0.110	0.117	0.199	**0.104***	0.107
1.5e-02	0.117	0.119	0.119	0.122	0.175	**0.108***	0.113
2.2e-02	0.117	0.133	0.133	0.129	0.177	0.113	**0.111**
3.2e-02	0.117	0.153	0.153	0.139	0.147	0.112	**0.111**
4.6e-02	0.117	0.183	0.183	0.154	0.130	**0.113**	0.113
6.8e-02	0.117	0.226	0.226	0.176	0.117	0.117	**0.109***

Table 3 | Numerical results on MATH routing. Bold indicates lowest regret + inspection cost, asterisk indicates significance at $p < .05$ (paired bootstrap).

Cost	f-only	g-only	Top-2	Coin Flip	Random-N_{pr}	Margin-N_{pr}	Pandora's Router
1.5e-03	0.150	**0.088**	0.110	0.124	0.129	0.100	0.090
2.1e-03	0.150	0.090	0.112	0.125	0.134	0.103	0.090
3.2e-03	0.150	0.093	0.114	0.127	0.146	0.110	0.092
4.6e-03	0.150	0.098	0.117	0.129	0.155	0.113	0.098
6.8e-03	0.150	0.104	0.121	0.132	0.155	0.119	0.104
1.0e-02	0.150	0.114	0.127	0.137	0.170	0.124	0.110
1.5e-02	0.150	0.128	0.137	0.144	0.182	0.134	0.122
2.2e-02	0.150	0.148	0.150	0.155	0.198	0.143	0.136
3.2e-02	0.150	0.178	0.171	0.170	0.196	0.154	**0.144**
4.6e-02	**0.150***	0.223	0.200	0.192	0.167	0.155	0.157
6.8e-02	**0.150**	0.288	0.244	0.225	0.150	0.150	0.155

Table 4 | Numerical results on RAG routing. Bold indicates lowest regret + inspection cost, asterisk indicates significance at $p < .05$ (paired bootstrap).

Cost	f-only	g-only	Top-2	Coin Flip	Random-N_{pr}	Margin-N_{pr}	Pandora's Router
1.0e-05	0.393	0.371	0.402	0.398	0.412	0.377	0.372
1.0e-04	0.393	0.381	0.402	0.403	0.421	0.377	0.372
3.0e-04	0.393	0.403	0.403	0.414	0.436	0.378	0.374
1.0e-03	0.393	0.480	0.404	0.453	0.454	0.387	0.382
3.0e-03	0.393	0.700	0.408	0.563	0.469	0.398	0.390
1.0e-02	**0.393**	1.470	0.422	0.947	0.484	0.403	0.399
3.0e-02	**0.393***	3.670	0.462	2.047	0.439	0.401	0.405
1.0e-01	**0.393**	11.370	0.602	5.894	0.393	0.393	0.395
3.0e-01	0.393	33.370	1.002	16.887	0.393	0.393	**0.391**

Table 5 | Numerical results on EmbedLLM routing. Bold indicates lowest regret + inspection cost, asterisk indicates significance at $p < .05$ (paired bootstrap).

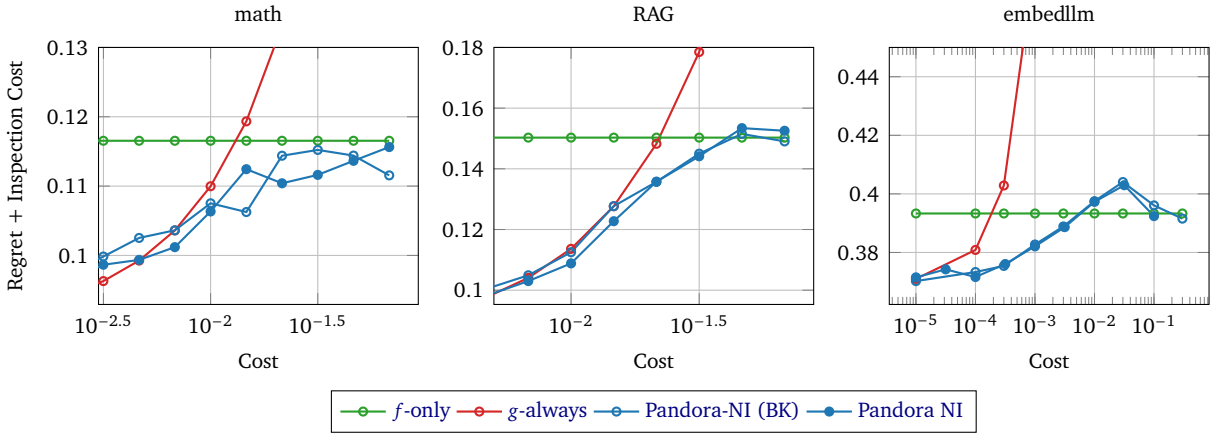

Figure 7 | Evaluation of the Beyhaghi and Kleinberg (2019) algorithm, which uses μ_m instead of u_m^{backup} for evaluating committing policies.

superior to the original algorithm, as shown in Figure 7.

D.6. Non-Gaussian signal models

Our implementation models $G_m \mid \mathbf{F} = \mathbf{f}$ as a conditional Gaussian, but real value distributions are bounded in $[0, 1]$, so a Gaussian may not be an appropriate model. For this reason, we also experimented using KNN on the vector of f-scores to estimate local conditional mean and local standard deviation directly from the neighborhood of f. For $K = 16$ this indeed improved calibration metrics somewhat: empirical coverage of the $\pm 1\sigma$ interval ranged between 58.7% (RAG) – 78.1% (MATH) for Gaussian, and 63.0% (RAG) – 74.8% (MATH) for KNN (theoretical target: 68.27%). For the $\pm 2\sigma$ interval, the range was 89.8% (RAG) – 94.5% (EmbedLLM) for the Gaussian signal model, and 91.2% (MATH) – 94.7% (EmbedLLM) for KNN (theoretical target: 95.45%).

We apply this signal model to the routing task, with results shown in Table 6. Despite the better calibration of the KNN signal model, we did not see a significant improvement in regret + cost. For this reason, we focus on the Gaussian signal model in the main implementation.

Setting	Method	Gaussian		KNN	
		Regret + Cost	Queries	Regret + Cost	Queries
MATH	Pandora's Router	0.1049	0.58	0.1090	0.78
RAG	Pandora's Router	0.1181	1.40	0.1218	1.22
EmbedLLM	Pandora's Router	0.3867	3.71	0.3913	0.40

Table 6 | Comparison between Gaussian and KNN approximations for Pandora's Router.