VOL. 12 NO. 9,
MAY, 2026
E-ISSN 3026-958X
P-ISSN 3027-1169

African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Science Innovation & Technology Research (JSITR)

Contamination-Free LLM Routing on LiveBench Reasoning Tasks: Accuracy-Cost-Latency Tradeoff Learning

Grace Xu
Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA
Corresponding Author: g.xu.cs@outlook.com
DOI: https://doi.org/10.70382/ajsitr.v12i9.088

Abstract

Dynamic benchmarks make model selection harder and more useful at the same time: a router must identify when a low-cost policy is reliable without learning benchmark-specific artifacts. This study evaluates prompt-level routing on 200 LiveBench reasoning questions (100 zebra puzzles, 50 spatial questions, and 50 Web-of-Lies questions). Two reproducible, deterministic reasoning back ends are compared. FastLite uses exact reasoning only for recognized low-complexity cases and bounded heuristics otherwise; SolvePro uses full constraint, Boolean, and analytic spatial procedures. A router predicts whether FastLite is safe from task, length, release, constraint, and structural features available before an answer is generated. SolvePro reaches 100.0% exact accuracy, whereas FastLite reaches 67.0% exact and 69.9% answer-component accuracy. Under five-fold out-of-fold evaluation, a logistic router reaches 97.0% exact accuracy with a 32.5% strong-call rate and a 55.0% normalized cost saving relative to always using SolvePro. A transparent complexity rule preserves 100.0% exact accuracy with a 36.5% strong-call rate and 52.7% cost saving. Errors are concentrated in spatial and zebra items, confirming that prompt length alone is insufficient. The findings show that fresh, objectively scored benchmark items can support auditable accuracy-cost-latency routing when features encode verifiable computational structure. Because both back ends are deterministic policies rather than hosted LLM APIs, the reported latency and normalized cost values are controlled experimental quantities, not vendor performance claims.

Keywords: LiveBench, large language model routing, contamination-free evaluation, cost-sensitive inference, uncertainty calibration.

Introduction

Large language model deployment increasingly involves more than selecting a single model. A production router can send structurally simple prompts to a low-cost back end, reserve expensive reasoning for prompts with a higher estimated failure risk, and still enforce a target quality level. This differs from ordinary leaderboard ranking: a leaderboard asks which system is strongest on average, whereas a router asks which system is adequate for one prompt before its answer is known. The decision therefore depends on task structure, uncertainty, answer verifiability, cost, and latency.

Fresh evaluation data are central to that decision. LiveBench was designed around frequently updated questions and objective scoring to reduce contamination risk (White et al., 2024), while LiveCodeBench applies a related time-separated and execution-grounded design to code (Jain et al., 2024). These benchmarks address a practical weakness of static test sets: as public questions, solutions, and model outputs circulate, observed accuracy can increasingly reflect exposure rather than generalization. A router trained on such artifacts may learn an unsafe estimate of when the inexpensive path is reliable.

The present study also draws on the broader movement toward objective and multifaceted evaluation. MMLU covers knowledge and reasoning across many subjects (Hendrycks et al., 2021), GSM8K supports exact checking of mathematical answers (Cobbe et al., 2021), and IFEval uses verifiable instruction constraints (J. Zhou et al., 2023). Transformer-based representation learning (Devlin et al., 2019; Vaswani et al., 2017) and frontier-model scaling (OpenAI, 2023) make such breadth necessary, but they do not eliminate the need for task-level safety decisions. For routing, the label of interest is not generic difficulty; it is whether the cheaper back end is correct under the benchmark's own scoring rule.

Cost-aware model selection has been studied through cascades, sparse expert selection, and budgeted policies. FrugalGPT demonstrates the value of combining model calls under a cost objective (L. Chen et al., 2023), while mixture-of-experts systems route computation inside a model (Fedus et al., 2022; Jacobs et al., 1991; Shazeer et al., 2017). At the application level, budgeted retrieval likewise treats each additional evidence hop as a decision with a measurable resource cost (Su et al., 2025). Forecasting work for new inference tenants shows why the operating context can itself be uncertain (He et al., 2025). These studies motivate a prompt-level gate whose output can be audited in the same units as its accuracy, cost, and latency consequences.

A safe gate must do more than rank examples. Probability calibration is important when a threshold determines whether the low-cost path is accepted (Guo et al., 2017). Selective refusal in calibrated recruiter screening provides one example of converting a score into an action under uncertainty (Jin, 2025a), and evidence-calibrated retrieval shows the related value of abstaining when coverage is inadequate (Zhong et al., 2025). In the current setting, the router predicts whether FastLite will be exactly correct and invokes SolvePro when the predicted probability does not meet

a fold-specific safety threshold.

The empirical scope is deliberately narrow and fully stated. The evaluated split contains 200 LiveBench reasoning questions: 100 zebra puzzles, 50 spatial questions, and 50 Web-of-Lies questions. It contains no coding or instruction-following items, so the numerical findings apply only to these reasoning tasks. Moreover, FastLite and SolvePro are deterministic, reproducible reasoning policies rather than hosted commercial LLM APIs. They isolate the routing decision and permit exact reruns, but their millisecond latencies and normalized costs must not be interpreted as vendor measurements.

The operational hypothesis is that prompt structure is more informative than length alone. A long truth-maintenance prompt can be inexpensive once its statements are converted into a Boolean graph; a short spatial prompt can remain risky if it requires a geometric template outside the bounded back end. The experiment therefore extracts pre-answer features such as task family, prompt length, constraint count, number of people, truth statements, cuts, and spatial composition flags. Neither the reference answer nor SolvePro's output is available to the router.

Accuracy, normalized cost, and measured latency are reported as separate outputs. The always-strong policy defines the accuracy ceiling and cost denominator, the always-cheap policy shows the maximum call reduction and its risk, and an oracle shows how much saving is theoretically available with these two back ends. Simple rules and learned routers are evaluated between those endpoints. This design prevents an attractive cost number from obscuring where accuracy is lost.

Three research questions organize the study: (1) How large is the empirical performance gap between the bounded and deliberative back ends? (2) Can transparent structural features predict when FastLite is safe? (3) What accuracy-cost-latency tradeoffs are obtained by rules, logistic regression, and random forests under five-fold out-of-fold evaluation? The contributions are a contamination-aware routing formulation, a reproducible two-back-end evaluation on the stated split, and an operating-point analysis that reports endpoints, learned policies, rules, ablations, errors, and the oracle bound.

Literature Review

Dynamic and Objective Evaluation

Contamination-resistant evaluation is particularly important when benchmark outcomes become training labels for a router. LiveBench couples' fresh questions with deterministic ground truth (White et al., 2024), and LiveCodeBench adds time-aware, test-executed code assessment (Jain et al., 2024). Objective formats also appear in mathematical verification (Cobbe et al., 2021) and explicit instruction constraints (J. Zhou et al., 2023). Together, these designs reduce evaluator variance and make repeated threshold sweeps feasible.

A second strand concerns whether evidence and model outputs can be audited. Evidence-chain RAG links claims to deterministic provenance (Jin, 2025b), while machine-learned RAG quality prediction treats retrieval quality as an explicit variable (R. Zhang et al., 2025). Offline

financial-search reranking similarly separates query rewriting, retrieval, and listwise ranking so that failure can be localized (Meng et al., 2026). Numerical-reasoning guardrails for SEC and FRED data extend this principle to verifiable quantitative outputs (Z. Li et al., 2026). Accounting-aware evidence retrieval likewise makes source support explicit in institutional due diligence (Y. Chen et al., 2025). These studies support the present decision to use exact answers and pre-answer features instead of subjective post hoc ratings.

Cost-Aware Routing and Resource Allocation

Inference routing is related to both model cascades and operational resource allocation. FrugalGPT optimizes which model calls to make under cost constraints (L. Chen et al., 2023), whereas sparse expert architectures learn internal gates (Fedus et al., 2022; Shazeer et al., 2017). At the infrastructure level, multi-horizon GPU demand forecasting links workload semantics to operational risk (Zhao et al., 2025), and GPU-memory forecasting estimates the resource footprint of deep-learning jobs before execution (C. Wang et al., 2025). The common idea is that a prediction is useful only when it changes a resource-allocation decision.

Resource policies must also be evaluated under distribution shift and asymmetric loss. Power-aware inventory planning connects job-level forecasts to capacity decisions (He et al., 2026), while constrained budgeting integrates demand forecasts with response models (Y. Lu et al., 2025). Conservative off-policy selection provides a way to reject an apparently efficient policy when logged evidence is weak (Ye et al., 2026). The decision rule should remain interpretable when observations are incomplete or shift away from the logged operating conditions. These concerns motivate the safety constraint and transparent baselines used here.

Calibration, Uncertainty, and Selective Behavior

Calibration converts a classifier score into a risk-bearing operating point (Guo et al., 2017). Calibration-light subject-independent BCI demonstrates the need to preserve confidence behavior under subject shift (Wu, Mi, et al., 2025), and macro-market fusion explicitly incorporates news uncertainty into a directional forecast (H. Zhou & K. Zhang, 2025). At a more theoretical level, spectral rank-aggregation guarantees characterize when an estimator can be trusted (Zhong & Ling, 2024a). Although these applications differ from LLM routing, each treats uncertainty as part of the decision rather than as a decorative output.

The same pattern appears in high-stakes multimodal systems. Uncertainty-aware medical vision-language classification links prediction with an explicit risk signal (S. Lu & Zou, 2026), while ultrasound explanation cards translate uncertainty into a human-readable diagnostic interface (Ye et al., 2025). Such work does not provide a direct baseline for LiveBench, but it clarifies two requirements adopted here: the router's probability must be evaluated at a chosen threshold, and residual errors must be reported by task rather than hidden inside an aggregate AUC.

Structural Features, Transfer, and Compact Predictors

Transparent feature design offers a practical middle ground between hand-written rules and opaque end-to-end gates. Language-guided feature selection for intrusion detection uses semantics to identify a compact decision representation (Y. Li & S. Lu, 2025). Interpretable CloudTrail stage detection combines linear models with sequence smoothing (Bai et al., 2026), and VMAF distillation shows how a costly quality signal can be approximated by an edge-deployable predictor (B. Zhou et al., 2025).

Transfer studies further caution against treating surface length as general difficulty. Wearable activity recognition can require cross-dataset adaptation even when label semantics appear stable (J. Zhang, 2025). For prompt routing, the analogous goal is to identify features tied to the computational structure of a task—constraints, truth graphs, or spatial composition—rather than to a release-specific lexical signature.

Evidence Presentation, Governance, and Human Oversight

Reliability also depends on how a system exposes evidence and constraints. Dark-pattern detection compares encoder and decoder families while generating interface-level explanations (Xu et al., 2025); evidence-constrained financial agents make accounting support part of the workflow (S. Zhou et al., 2026). Federated topic-preference learning adds differential privacy to knowledge-grounded chat (Kuo et al., 2025). These examples motivate separating route selection, execution, and audit evidence.

Human-facing explanation research emphasizes structured, decision-relevant presentation rather than unrestricted rationales. Recommendation cards calibrate product evidence and trust (B. Zhang et al., 2025); advertising brief cards translate intentions into editable design choices (Tu et al., 2025); and counseling recommendation cards link suggested support to source evidence (Y. Zhang & H. Zhang, 2025b). Financial risk dashboards likewise organize uncertainty, chart comprehension, and explanation hierarchy (Z. Li et al., 2025). These studies support the use of compact tables, threshold curves, and error breakdowns in the present evaluation.

Interface transparency is especially important when inputs are multilingual, creative, or security-sensitive. Trust-calibrated multilingual RAG evaluates access to migration information (Chen & Xu, 2026); text-grounded design-rationale interfaces expose which layout metadata supports a recommendation (Wu, Meng, et al., 2025); and an LLM design critic is benchmarked against human judgment (Y. Li et al., 2025). Privacy and data-integrity risk cards extend this approach to prompt-injection oversight (Su et al., 2026), while structured visual briefs preserve human editability in advertising workflows (Mu et al., 2026).

The literature also shows why routing evaluation should remain connected to the downstream user. A therapist-facing copilot retrieves and ranks support from multi-turn dialogue (Y. Zhang & H. Zhang, 2025a); teacher-facing early-warning explanations connect predictions to instructional decisions (J. Zhang, 2026); and a DevOps copilot evaluates both runbook retrieval and command safety (B. Zhang et al., 2026). Across these domains, a low-cost route

is useful only when the resulting action remains verifiable and appropriately reviewable. Materials and Methods.

Routing Formulation

The evaluation treats routing as a binary decision. For each prompt x, the router chooses FastLite when the estimated probability of a correct low-cost answer is at least a threshold and SolvePro otherwise. Let y_c(x) equal one when FastLite is exactly correct and zero otherwise. A probabilistic router estimates p(y_c = 1 | x) from structural features, and the decision is FastLite when p(y_c = 1 | x) ≥ τ. The threshold τ is selected within each training fold to minimize normalized cost subject to a 99% training safe-accuracy target. This makes calibration operational rather than descriptive (Guo et al., 2017), consistent with selective refusal (Jin, 2025a) and evidence-coverage abstention (Zhong et al., 2025).

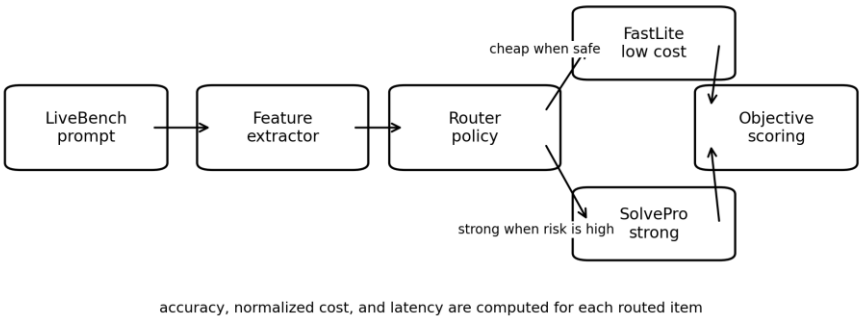


Fig. 1. Routing pipeline used in the experiment.

Dataset and Task Structure

The data contain objective ground truth and a single prompt turn for each item. Table 1 summarizes the split. Zebra puzzles are line-order constraint satisfaction problems with 2 to 5 people, multiple attributes, and questions that ask for positions or attribute values. Web-of-Lies items describe people at locations and truth-telling relations, then ask yes-or-no questions about selected locations. Spatial items ask for counts or shapes after cuts, tangencies, or geometric constructions. Prompt lengths are uneven: spatial questions average 83.9 whitespace tokens, while zebra and Web-of-Lies prompts average about 318 tokens. This heterogeneity is useful for routing because prompt length alone is not enough; Web-of-Lies prompts are long but cheap to solve, while some shorter spatial prompts require strong geometric rules.

Each row is evaluated as a single independent decision. The router cannot look at the answer from SolvePro before deciding, and it cannot use ground-truth strings as features. It only receives prompt-derived statistics and metadata that would be known at routing time. After the route is selected, the chosen back end produces an answer, and the answer is normalized

for punctuation, spacing, and case before scoring. This separation is essential: the router is a pre-answer decision rule, not an after-the-fact selector between two completed answers.

The routing objective is asymmetric. Sending a safe prompt to SolvePro wastes cost, but sending an unsafe prompt to FastLite creates an accuracy error. The threshold selection therefore minimizes cost subject to a safe-accuracy constraint rather than optimizing raw accuracy alone. This mirrors how model cascades are used in practice: the goal is not to maximize the number of cheap calls, but to use cheap calls where they do not violate a target quality level. The empirical tables report exact accuracy, component accuracy, cheap-call rate, strong-call rate, normalized cost, cost saving, and measured latency so that no single metric dominates interpretation.

The split has enough task diversity to expose several routing failure modes. Zebra prompts vary in the number of people, attributes, constraints, and release-level annotations. Web-of-Lies prompts are textually long but algebraically simple once the truth relations are parsed. Spatial prompts include both direct formulas and cases requiring template recognition. These differences make the split appropriate for studying prompt-level safety prediction even though it is not a full cross-category LiveBench sample. The method deliberately keeps the same routing interface for every task family so that additional categories can be added without changing the cost objective.

Table 1. Dataset composition and prompt statistics.

Task	N	Release dates	Mean words	Mean ?	Mean answer parts	Level count
spatial	50	2024-07-26	83.90	1.14	1	0
web_of_lies_v2	50	2024-06-24	318.60	3	3	0
zebra_puzzle	100	2024-06-24, 2024-11-25	318.10	2.50	2.50	50

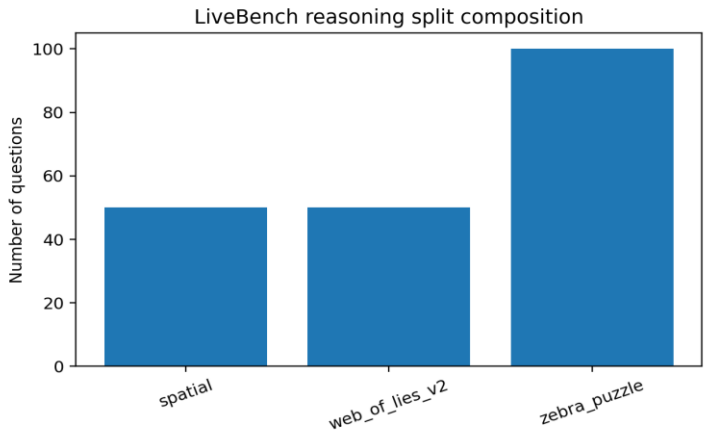


Fig. 2. Number of questions per task in the evaluated LiveBench reasoning split.

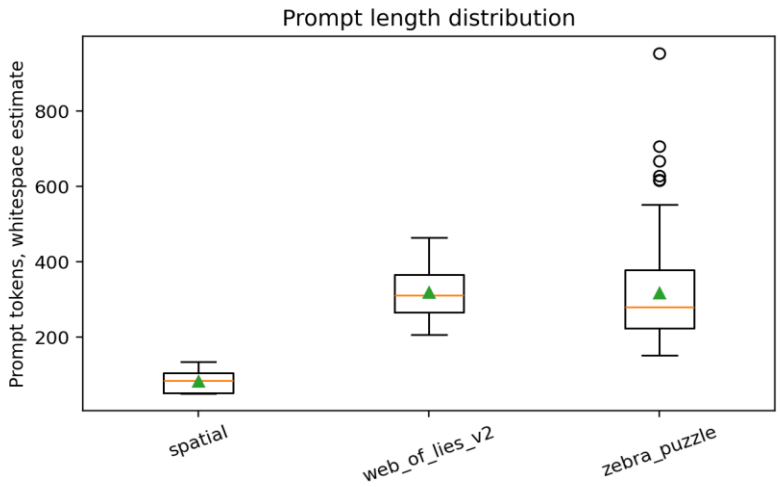


Fig. 3. Prompt length distribution by task.

Inference Back Ends and Structural Features

The two inference back ends are deterministic reasoning policies. FastLite represents a low-cost bounded policy. For zebra puzzles, it solves instances exactly only when the number of people is at most three; otherwise it applies shallow position and attribute heuristics. For Web-of-Lies, it uses exact Boolean truth maintenance because those graphs are linear in the number of statements and are inexpensive. For spatial tasks, it applies simple two-dimensional cut formulas and a small set of shallow geometric rules, then falls back to direct count heuristics. SolvePro is the full deliberative policy. It uses Z3-style integer constraint solving for zebra puzzles, Boolean satisfiability for Web-of-Lies, and analytic spatial template rules for the spatial prompts. Both back ends consume the same natural-language prompts and are scored only by exact comparison with the provided ground truth.

The router features are deliberately transparent. They include task indicators, prompt word and character counts, question count, release indicators, LiveBench level, zebra people and attributes, constraint count, parity and negation flags, Web-of-Lies statement and direct-fact counts, spatial cuts, and flags for cubes, spheres, composite shapes, shape questions, and maximum-piece requests. Table 2 summarizes selected means. The design follows work on semantic feature selection (Y. Li & S. Lu, 2025) and interpretable sequence features (Bai et al., 2026), but excludes answer text, ground truth, and any information unavailable before routing.

Cost Model and Evaluation Protocol

Normalized cost is defined as prompt tokens multiplied by a back-end multiplier: 1 for FastLite and 8 for SolvePro. The scale is a reproducible experimental assumption, not a vendor price. Measured latency is the local wall-clock time in milliseconds, averaged over three runs per item. Exact accuracy requires the full normalized answer to match; component accuracy gives partial credit for comma-separated lists. Learned routers use five-fold stratified cross-

validation (Kohavi, 1995). Logistic regression provides a calibrated linear baseline (Bishop, 2006), and a random forest provides a nonlinear alternative (Breiman, 2001). Chain-of-thought and self-consistency can add reasoning at inference time (Wei et al., 2022; X. Wang et al., 2023), but the router uses only inexpensive pre-answer structure.

FastLite is intentionally incomplete. Its purpose is to represent the behavior of an inexpensive model or tool chain whose reliability depends on recognizable structure. For zebra puzzles with at most three people, exact enumeration is cheap and stable, so FastLite solves the full constraint system. For larger zebra puzzles, it avoids the full combinatorial search and applies only local clues and answer-format heuristics. This produces successes on simple large cases and failures on tightly coupled cases, which is exactly the type of boundary a router should learn. The strong policy, in contrast, converts the zebra prompt into an integer constraint program and searches the feasible assignments until the requested answer fields are determined.

For Web-of-Lies prompts, both policies use exact Boolean truth maintenance. The prompts describe who is truthful and who lies through nested statements. After extraction, the truth state can be propagated as a small system of Boolean equations. Because this computation is cheap, sending these prompts to the strong policy is usually unnecessary. Including this task family prevents the router from learning a naive rule that long text should always go to the strong back end. It also illustrates why objective answer decomposition matters: the Web-of-Lies answers contain multiple yes-or-no components, and component accuracy confirms that the cheap policy is not merely matching one part of the output.

For spatial prompts, FastLite handles direct counting patterns such as simple cuts and some two-dimensional shape questions. SolvePro adds a broader library of analytic templates for cubes, spheres, tangencies, composite arrangements, and maximum-piece formulas. The spatial task therefore tests whether the router can distinguish between a short prompt with a covered formula and a short prompt requiring the strong template library. This distinction is especially important for latency interpretation because text length is low for spatial items, yet the cost of a wrong geometric shortcut can be high in accuracy terms.

Router Training and Baselines

The learned routers are evaluated only on held-out predictions. For each of five folds, the model is fit on four folds, a threshold is selected on those four folds, and predictions are recorded for the fifth. Metrics are computed from the concatenated 200 held-out decisions. This prevents reuse of the fitted examples at evaluation time and makes the operating point auditable. It also parallels conservative off-policy selection (Ye et al., 2026), where an apparently efficient policy must still satisfy evidence and risk constraints.

The rule-based baselines serve different purposes. The task-only safe router sends all Web-of-Lies items to FastLite and all others to SolvePro. The complexity rule adds transparent conditions from the cheap solver: Web-of-Lies is cheap, zebra puzzles with at most three people are cheap, and simple spatial formulas are cheap. The oracle uses FastLite correctness

after scoring and is therefore not deployable; it is an upper bound. As in constrained budgeting (Y. Lu et al., 2025), the comparison distinguishes the resource objective from the evidence needed to authorize an action.

All figures are generated from the same result tables. Figure 1 gives the routing pipeline; Figures 2 and 3 describe task counts and prompt lengths; Figure 4 compares back-end accuracy; Figures 5 and 6 show policy tradeoffs and the logistic threshold sweep; and Figure 7 reports feature importance. Tables 1–3 define the data, features, and back ends, while Tables 4–10 report outcomes, folds, ablations, errors, importance, and selected thresholds.

Table 2. Mean structural features by task.

Task	Words	People	Attrs	Constr.	Says	Cuts	Shape q.	Cube	Sphere
spatial	83.88	0	0	0	0	2	0.12	0.20	0.24
web_of_lies_v2	318.60	0	0	0	9.76	0	0	0	0
zebra_puzzle	318.10	3.45	3.80	9.19	0	0	0	0	0

Table 3. Inference back ends and router role.

Back end	Role	Main reasoning budget	Cost multiplier
FastLite	Low-cost candidate	Exact Boolean Web-of-Lies; bounded zebra; shallow spatial formulas	1
SolvePro	Strong candidate	Full zebra CSP; Boolean truth maintenance; analytic spatial rules	8
Router	Gate	Logistic or random-forest estimate of FastLite safety	feature-only

Results

Back-End and Policy Performance

The first result is the quality gap between the two back ends. As Table 4 and Figure 4 show, SolvePro answers all 200 questions exactly, while FastLite answers 134 exactly. FastLite reaches 67.0% exact and 69.9% component accuracy. It is strongest on Web-of-Lies, where Boolean truth maintenance is inexpensive and exact, and weaker on spatial and larger zebra prompts. Long prompts therefore do not necessarily require the strong back end.

Table 5 shows why a router is preferable to either endpoint. Always using FastLite saves 87.5% normalized cost but reduces exact accuracy to 67.0%; always using SolvePro reaches 100.0% but pays full cost. The task-only policy reaches 100.0% with 26.8% saving. The complexity rule preserves 100.0% with a 36.5% strong-call rate and 52.7% saving, close to the oracle's 54.6% saving and 33.0% strong-call rate. Figure 5 makes these endpoints and intermediate policies directly comparable.

Prompt length explains only part of the outcome. Web-of-Lies and zebra prompts have similar mean lengths, but FastLite is exact on Web-of-Lies and only 57.0% exact on zebra. Spatial prompts are much shorter, yet FastLite exact accuracy is 54.0%. The router must therefore learn that length interacts with task structure. A long prompt can be cheap if it encodes a simple

Boolean graph; a short prompt can be risky if it asks for a geometric formula outside the low-cost library. This pattern is visible in the length-only ablation, which attains high accuracy by routing almost everything to SolvePro but loses nearly all cost saving.

Component accuracy gives additional insight into partial progress. FastLite component accuracy is higher than exact accuracy for zebra puzzles because some multi-part answers contain one correct component even when another component is wrong. This is useful for diagnosing solver behavior, but the routing decision is trained on exact safety because deployment normally needs the full answer to be correct. Reporting both metrics avoids hiding partial reasoning while still keeping the primary objective strict. SolvePro reaches 100.0% on both exact and component measures across all task families.

Table 4. Back-end accuracy, component accuracy, latency, and prompt size.

Task	N	FastLite acc.	SolvePro acc.	FastLite comp.	SolvePro comp.	FastLite ms	SolvePro ms
spatial	50	54.0	100.0	54.0	100.0	0.05	0.04
web_of_lies_v2	50	100.0	100.0	100.0	100.0	2.38	2.36
zebra_puzzle	100	57.0	100.0	62.8	100.0	5.61	16.81
overall	200	67.0	100.0	69.9	100.0	3.41	9.00

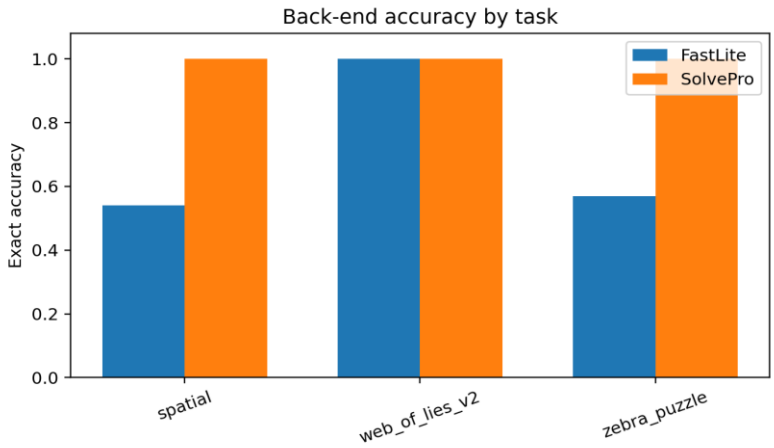


Fig. 4. FastLite and SolvePro exact-answer accuracy by task.

Table 5. Router performance under normalized cost and measured latency.

Method	Exact acc.	Comp. acc.	Strong rate	Cheap rate	Norm. cost	Saving	Mean ms	OOF AUC
Always cheap	67.0	69.9	0.0	100.0	259.670	87.5	3.413	
Always strong	100.0	100.0	100.0	0.0	2077.360	0.0	9.003	
Task-only safe	100.0	100.0	75.0	25.0	1519.810	26.8	9.006	
Complexity rule	100.0	100.0	36.5	63.5	983.295	52.7	9.011	

Method	Exact acc.	Comp. acc.	Strong rate	Cheap rate	Norm. cost	Saving	Mean ms	OOF AUC
Logistic router	97.0	97.2	32.5	67.5	934.120	55.0	8.821	0.962
Random-forest router	95.5	95.8	31.5	68.5	922.325	55.6	8.780	0.976
Oracle router	100.0	100.0	33.0	67.0	942.310	54.6	8.848	

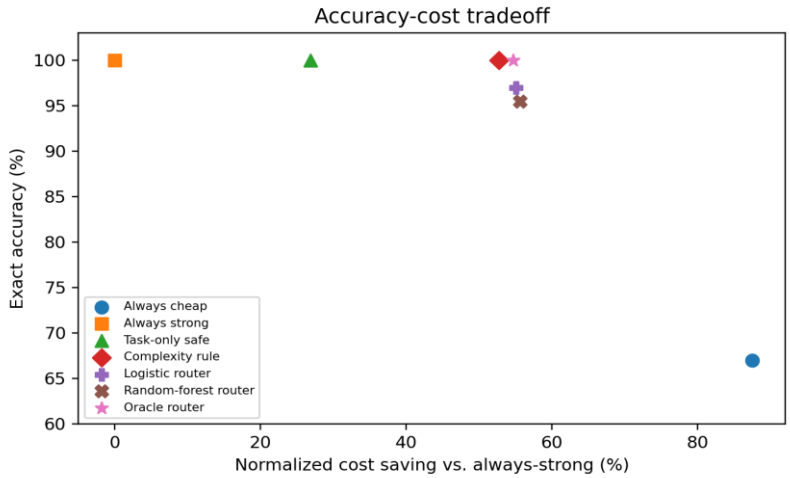


Fig. 5. Exact accuracy versus normalized cost saving for all routing policies.

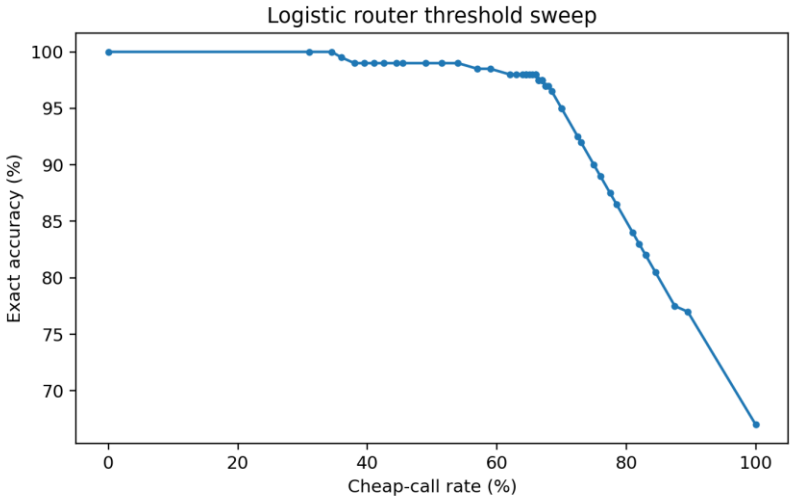


Fig. 6. Out-of-fold logistic threshold sweep showing accuracy as cheap-call rate changes.

Learned Router Performance

The learned routers trade small accuracy losses for slightly fewer strong calls. As reported in Table 5, the logistic router reaches 97.0% exact accuracy, 97.2% component accuracy, and 55.0% cost saving while calling SolvePro on 32.5% of items. Its out-of-fold AUC is 0.962 and

average precision is 0.981. The random forest has higher AUC (0.976) and average precision (0.991), but its selected operating point reaches 95.5% exact accuracy. Figure 6 confirms that ranking quality and thresholded policy quality are not interchangeable.

Table 6 gives the fold thresholds and cheap-call rates. The logistic thresholds range from 0.35 to 0.40, and every training fold satisfies the 99% safe-accuracy constraint. On test folds, cheap-call rates range from 65.0% to 70.0%. The random forest uses similar thresholds but has a wider cheap-call range. The threshold sweep in Fig. 6 clarifies the tradeoff: at threshold 0 the router degenerates to always cheap, while higher thresholds move toward always strong. The selected operating region is near two-thirds cheap calls, where most Web-of-Lies, simple zebra, and simple spatial items are routed to FastLite.

The learned-router errors are small in number but informative. The logistic router makes six exact-answer errors across the 200 held-out predictions. Three occur on spatial prompts and three on zebra prompts. The random forest makes nine errors, again concentrated in spatial and zebra. None occur on Web-of-Lies. This shows that the most costly mistakes are not random noise spread uniformly across the benchmark. They appear where the cheap policy has a mixed success boundary and where structural features are not yet expressive enough to separate every safe and unsafe item.

The ablation results clarify the role of feature groups. Full structural features produce 55.0% cost saving at 97.0% exact accuracy. Removing task indicators has almost no effect because the remaining structure identifies each task family. Removing release and level features also has little effect, which is desirable because a router should not depend mainly on date metadata. The length-only model is the opposite: it is safe because it is conservative, not because it finds low-cost opportunities. It saves only 4.6% normalized cost, so it would be a poor production policy despite its high accuracy.

The random-forest importance ranking is consistent with the solver design. The number of people is the largest feature because it controls whether FastLite can solve zebra puzzles exactly. Character count, word count, and constraint count follow because they summarize puzzle complexity and extraction burden. Web-of-Lies features such as says-count and direct truth facts identify prompts that remain cheap despite length. Spatial composition flags identify cases where simple formulas are less likely to be sufficient. These importances are not causal proofs, but they match the engineered boundary between the cheap and strong policies.

Table 6. Five-fold learned-router thresholds and cheap-call rates.

Model	Fold	Tau	Train safe acc.	Test cheap rate
Logistic router	1	0.400	99.4	67.5
Logistic router	2	0.350	99.4	67.5
Logistic router	3	0.400	99.4	67.5
Logistic router	4	0.380	99.4	70.0
Logistic router	5	0.360	99.4	65.0
Random-forest router	1	0.340	99.4	70.0

Model	Fold	Tau	Train safe acc.	Test cheap rate
Random-forest router	2	0.340	99.4	67.5
Random-forest router	3	0.320	99.4	72.5
Random-forest router	4	0.350	99.4	70.0
Random-forest router	5	0.350	99.4	62.5

Table 7. Logistic-router ablation study.

Feature set	Exact acc.	Comp. acc.	Strong rate	Norm. cost	Saving	OOF AUC
Full feature set	97.0	97.2	32.5	934.120	55.0	0.962
No task indicators	97.0	97.2	32.0	931.215	55.2	0.961
No release/level	97.5	97.8	32.5	939.125	54.8	0.960
Complexity only	97.5	97.8	33.0	942.030	54.6	0.960
Length only	99.0	99.0	93.5	1982.790	4.5	0.716

Ablations, Errors, and Feature Importance

Table 7 shows that removing task indicators leaves exact accuracy unchanged because other structural features identify each family. Removing release and level features slightly improves accuracy, indicating that the router is not primarily using date metadata. The length-only model reaches 99.0% accuracy but calls SolvePro on 93.5% of items and saves only 4.5% cost. Tables 8 and 9, together with Figure 7, show why: errors concentrate in spatial and zebra items, and the most informative features describe people, constraints, truth statements, and spatial composition rather than length alone.

Table 8. Learned-router error breakdown by task.

Router	Task	N	Cheap	Strong	Errors	Acc.
Logistic router	spatial	50	27	23	3	94.0
Logistic router	web_of_lies_v2	50	50	0	0	100.0
Logistic router	zebra_puzzle	100	58	42	3	97.0
Random-forest router	spatial	50	28	22	5	90.0
Random-forest router	web_of_lies_v2	50	50	0	0	100.0
Random-forest router	zebra_puzzle	100	59	41	4	96.0

Table 9. Top random-forest feature importances for predicting FastLite safety.

Feature	Importance
n_people	0.2222
char_count	0.1154
word_count	0.1110
n_constraints	0.1081

Feature	Importance
num_says	0.0513
task_web	0.0432
num_direct_truth	0.0417
has_composite_shapes	0.0358
n_cuts	0.0336
release_2024_07	0.0244

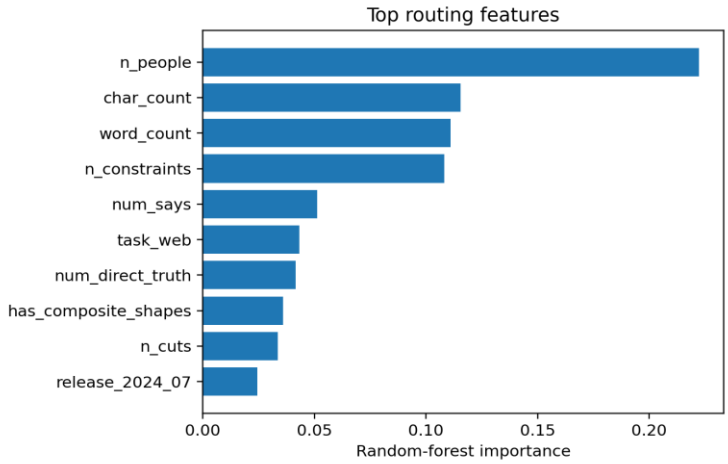


Fig. 7. Top feature importances for the safety prediction problem.

Table 10. Selected operating points from the logistic threshold sweep.

Tau	Exact acc.	Cheap rate	Strong rate	Norm. cost	Saving
0	67.0	100.0	0.0	259.670	87.5
0.200	89.0	76.0	24.0	798.670	61.6
0.280	95.0	70.0	30.0	900.415	56.7
0.340	97.0	68.0	32.0	931.215	55.2
0.360	97.0	67.5	32.5	934.120	55.0
0.500	97.5	66.5	33.5	951.725	54.2
0.800	99.0	49.0	51.0	1230.710	40.8
1	100.0	0.0	100.0	2077.360	0.0

Discussion

Interpretation of the Accuracy-Cost-Latency Tradeoff

The error analysis in Table 8 shows that learned-router failures are concentrated in spatial and zebra tasks: the logistic router makes three errors in each family, and the random forest makes five spatial and four zebra errors. No Web-of-Lies item is misrouted because FastLite solves all 50 exactly. The importance ranking in Table 9 is consistent with this boundary. The result should be read as an operating-point finding, not a general claim that one classifier dominates: formal uncertainty analysis likewise distinguishes estimator accuracy from confidence in an estimate (Zhong & Ling, 2024b).

Latency requires careful interpretation. FastLite is faster overall, but SolvePro's spatial templates are slightly faster than the cheap fallback on some short items; the largest latency difference comes from zebra constraint solving. The economic benefit therefore appears mainly in normalized cost. Production deployments should replace the 1:8 multiplier with current service prices and measured tail latency. Progressive PDF rendering shows how a structural priority policy can target time to a functional first frame (H. Wang et al., 2025). Work on GPU demand forecasting (Zhao et al., 2025), capacity planning (He et al., 2026), and memory prediction (C. Wang et al., 2025) illustrates why resource models must be recalibrated when workloads or hardware change.

A notable finding is that the complexity rule is competitive with the oracle: it preserves 100.0% accuracy while saving 52.7% cost, only 1.9 percentage points below the oracle. The rule is auditable because it encodes known solver boundaries; the learned router is more adaptable but accepts several unsafe cheap calls. Conservative off-policy selection (Ye et al., 2026) suggests a useful compromise: a rule-defined guaranteed region can be combined with a learned policy on the unresolved cases.

The broader implication is that fresh benchmarks can support router training in a way that static benchmarks cannot. If FastLite succeeds because an item appeared in training data, the router learns a misleading safety label. LiveBench's update process and objective scoring reduce this risk (White et al., 2024), while LiveCodeBench provides a comparable time-aware principle for code (Jain et al., 2024). Evidence-chain provenance (Jin, 2025b) and retrieval-quality prediction (R. Zhang et al., 2025) add a complementary lesson: the evidence supporting a route should remain inspectable.

The comparison between rule and learned routing also clarifies the role of a compact proxy. Distilling an expensive quality metric into an edge predictor (B. Zhou et al., 2025) is useful only if the proxy is validated at the deployment threshold. Numerical guardrails (Z. Li et al., 2026) make the same point for calculation-intensive assistants. In this study, the logistic model offers a portable score, whereas the complexity rule offers a stronger guarantee on the observed split. A hybrid can preserve the rule's auditability and use the model to reduce unnecessary strong calls outside the certified region.

The cost-saving metric explains why the oracle saving (54.6%) is not equal to FastLite's accuracy rate (67.0%). Token-weighted savings are larger when FastLite handles long safe prompts and smaller when it handles short ones. Web-of-Lies is valuable because it combines long inputs with exact cheap solvability. Similar allocation problems arise when demand forecasting informs constrained budgets (Y. Lu et al., 2025) or when job-level forecasts drive power-aware inventory (He et al., 2026): the value of a correct decision depends on the resources attached to that case.

The local millisecond measurements add a second perspective but should not be collapsed with cost. SolvePro is slower overall because zebra search dominates its mean; FastLite avoids that search on many items. A hosted LLM study would also need queueing, output-token, and tail-latency measurements. Command-safe DevOps assistance (B. Zhang et al., 2026) shows

why workload prediction and execution safeguards should remain separate components of an operational router.

Objective answer formats make router analysis cleaner. Web-of-Lies lists, zebra attributes, and spatial counts or shape names can be normalized automatically. Budgeted multi-hop retrieval (Su et al., 2025), financial-search reranking (Meng et al., 2026), and evidence-constrained accounting agents (S. Zhou et al., 2026) similarly benefit from decomposing retrieval, decision, and verification. Free-form generation would require additional evaluators, and its router label would be less stable.

Logistic regression remains competitive because many safety boundaries are nearly rule-like: at most three zebra people, a recognizable truth graph, or a simple spatial formula. The random forest ranks items well but produces more errors at its selected threshold. Calibration-light transfer (Wu, Mi, et al., 2025), uncertainty-aware medical classification (S. Lu & Zou, 2026), and uncertainty-sensitive market fusion (H. Zhou & K. Zhang, 2025) all reinforce the distinction between ranking performance and a risk-bearing operating point.

The results suggest a practical protocol for multi-model systems: run cheap and strong back ends on a fresh, objectively scored benchmark; derive only pre-answer features; train a safety predictor out of fold; report both endpoints, transparent rules, learned routers, and the oracle; and publish task-level errors and threshold sweeps. Human-facing evidence designs (Su et al., 2026) and incident visualization (Nie et al., 2026) further suggest that unsafe or ambiguous routes should expose the reason for escalation rather than only the selected model name.

Cross-Domain Implications

The deterministic back ends should be interpreted as controlled policy surrogates, not as estimates of any named commercial model. Lightweight foundation-model transfer on medical data shows that compact back ends can remain useful when their training objective matches the target task (Mi et al., 2025); conversely, predictive DDoS mitigation illustrates how the cost of a false negative can change the preferred operating point (B. Wang et al., 2024). The present contribution is therefore the evaluation protocol—pre-answer features, out-of-fold safety labels, explicit endpoints, and task-level errors—rather than a universal claim about one cheap/strong pair.

Uncertainty should also be monitored after deployment. Formal uncertainty quantification for spectral estimators separates point performance from confidence in that performance (Zhong & Ling, 2024b). An LLM router needs the same distinction: a high AUC does not guarantee that a fixed probability threshold remains safe after a benchmark release or back-end update. Calibration curves, coverage, and the rate of unsafe cheap calls should therefore be recomputed whenever the data-generating process changes.

Finally, routing affects the form of downstream oversight. Evidence-linked counseling support demonstrates that selecting and presenting a response policy can be as consequential as selecting a model (Y. Zhang & H. Zhang, 2025b). Autonomous-driving detection work similarly highlights the sensitivity of visual pipelines to architectural changes (Ling et al.,

2024). These are not direct LiveBench baselines, but they reinforce a deployment principle: high-consequence routes should include a verifier, abstention path, or human review rather than relying on confidence alone.

Limitations and Future Work

The empirical scope is limited to the 200 reasoning questions in the evaluated split. The method can be extended to coding and instruction following, but the reported numbers must not be read as performance on those categories. Evidence from cross-dataset transfer and teacher-facing prediction shows why new task families require independent validation rather than assumed portability. The strong back end also reaches 100.0% here, so the study does not estimate cases in which both back ends fail.

The back ends are deterministic reasoning policies rather than proprietary API models. This choice makes every result reproducible, but the measured millisecond latencies are not cloud-model latencies and the 1:8 cost ratio is not a price table. Lightweight medical pretraining (Mi et al., 2025) and predictive DDoS mitigation (B. Wang et al., 2024) illustrate that an inexpensive policy's acceptable risk depends on its domain. A deployment study should substitute current prices, service latency, and application-specific loss, then reselect the threshold.

FastLite does not represent every small LLM, and SolvePro does not represent every frontier LLM. Neural back ends would add stochastic sampling, refusal behavior, formatting failures, and residual strong-model errors. Therapist-facing retrieval and ranking (Y. Zhang & H. Zhang, 2025a) emphasizes that a policy choice can alter downstream behavior even when model capability is unchanged. High-risk extensions should therefore add abstention, verification, repeated sampling, or human review.

The features are extracted by regular expressions and task-specific parsers. They are transparent but depend on prompt conventions. A future LiveBench release could add paraphrases or new structures that lower parser coverage. Multilingual RAG evaluation (Chen & Xu, 2026) and autonomous-driving detector optimization (Ling et al., 2024) exemplify settings in which language or visual distribution shift can change the reliability of a previously effective representation. The safe response is to version the extractor, report coverage, and rerun the same scoring pipeline on each release.

Cost is modeled only from input length and a fixed multiplier. Commercial systems may bill input and output tokens differently and can have minimum or queueing latencies unrelated to prompt length. GPU-memory prediction (C. Wang et al., 2025) and multi-horizon demand forecasting (Zhao et al., 2025) demonstrate that operational constraints can change with workload and hardware. The normalized cost here is therefore an experimental control, not a universal price estimate.

Finally, five-fold evaluation on one 200-item split cannot substitute for a longitudinal benchmark. A production router should be checked for drift in cheap-model correctness and recalibrated whenever a back end or release changes. Formal uncertainty quantification (Zhong & Ling, 2024b) provides the relevant conceptual distinction between an estimate and

confidence in that estimate. Larger releases would also support narrower confidence intervals, temporal holdouts, and stronger calibration diagnostics.

Conclusion

This study evaluated an accuracy-cost-latency router on 200 LiveBench reasoning questions. FastLite reached 67.0% exact accuracy and SolvePro reached 100.0%. A five-fold out-of-fold logistic router achieved 97.0% exact accuracy with a 32.5% strong-call rate and 55.0% normalized cost saving; a transparent complexity rule preserved 100.0% accuracy with a 36.5% strong-call rate and 52.7% saving. Structural features—people, constraints, truth-graph cues, and spatial composition—were more useful than prompt length alone.

The main practical conclusion is that routing should be evaluated as a decision policy. The cheap endpoint measures risk, the strong endpoint measures the ceiling, the selected operating point measures deployable savings, and the oracle measures unrecovered opportunity. Reporting all four, together with task-level errors and a threshold sweep, makes the tradeoff auditable and prevents aggregate cost savings from hiding unsafe cheap calls.

The numerical findings are limited to the stated reasoning split and to deterministic back ends; they are not vendor LLM latency or pricing claims. Future work should add temporal releases, coding and instruction-following tasks, service-specific prices, strong-model uncertainty, calibrated abstention, and verifier or human-review paths. The core requirement remains unchanged: the route must be selected before the answer is known, evaluated on fresh objective data, and reported with enough evidence to show where cost is saved and where errors remain.

References

Bai, J., Chen, S., Zheng, D., & Kuo, M.-J. (2026). Interpretable attack-chain stage detection from AWS CloudTrail event sequences via linear models and HMM smoothing. *Information, Electrical and Electronic Engineering, 6*(1), 28–43. https://doi.org/10.33474/infotron.v6i1.24923

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. https://doi.org/10.1007/978-0-387-45528-0

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, L., Zaharia, M., & Zou, J. (2023). *FrugalGPT: How to use large language models while reducing cost and improving performance [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2305.05176

Chen, Y., & Xu, H. (2026). Trust-calibrated multilingual RAG for humanitarian information platforms: Empirical evaluation on OMoS-QA for migration information access. *International Journal of Graphic Design, 4*(1), 141–164. https://doi.org/10.51903/ijgd.v4i1.3552

Chen, Y., Zhou, S., & Lin, E. (2025). Accounting-aware evidence retrieval for institutional due diligence of tokenized trade receivable RWA. *Journal of Technology Informatics and Engineering, 4*(3), 649–663. https://doi.org/10.51903/jtie.v4i3.542

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021). *Training verifiers to solve math word problems [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2110.14168

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *J. Burstein, C. Doran, & T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research, 23*(120), 1–39. https://www.jmlr.org/papers/v23/21-0998.html

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In D. Precup & Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 1321–1330). PMLR. https://proceedings.mlr.press/v70/guo17a.html

He, S., Li, C., & Rao, H. (2025). Few-shot cold-start workload forecasting for new AI inference tenants with time-series foundation models. *Journal of Technology Informatics and Engineering, 4*(1), 306–324. https://doi.org/10.51903/jtie.v4i1.546

He, S., Nie, J., & Li, C. (2026). Power-aware inventory planning for AI infrastructure using job-level forecasting and LLM workload explanations. *Journal of Technology Informatics and Engineering, 5*(1), 341–359. https://doi.org/10.51903/jtie.v5i1.548

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). Measuring massive multitask language understanding. In International Conference on Learning Representations. https://openreview.net/forum?id=d7KBjmI3GmQ

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*(1), 79–87. https://doi.org/10.1162/neco.1991.3.1.79

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., & Stoica, I. (2024). *LiveCodeBench: Holistic and contamination free evaluation of large language models for code [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2403.07974

Jin, J. (2025a). Calibrated resume-job matching for trustworthy LLM-assisted recruiter screening: Pairwise matching, probability calibration, and selective refusal on two public recruitment datasets. *Journal of Technology Informatics and Engineering, 4*(3), 625–648. https://doi.org/10.51903/jtie.v4i3.529

Jin, J. (2025b). Evidence-chain reliable RAG: Hallucination detection, source attribution, and deterministic provenance explanations. *Journal of Technology Informatics and Engineering, 4*(2), 520–533. https://doi.org/10.51903/jtie.v4i2.535

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. S. Mellish (Ed.), Proceedings of the 14th International Joint Conference on Artificial Intelligence (Vol. 2, pp. 1137–1145). Morgan Kaufmann. https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf

Kuo, M.-J., Zheng, D., & Hires, J. (2025). Federated topic-preference learning for knowledge-grounded chat with differential privacy. *Journal of Technology Informatics and Engineering, 4*(2), 385–401. https://doi.org/10.51903/jtie.v4i2.502

Li, Y., & Lu, S. (2025). Language-guided feature selection for DDoS and intrusion detection on CICIDS2017. *Journal of Technology Informatics and Engineering, 4*(1), 284–305. https://doi.org/10.51903/jtie.v4i1.531

Li, Y., Lu, S., & Zhao, L. (2025). LLM-as-design-critic: Aligning AI-generated UI feedback with human graphic design judgment. *International Journal of Graphic Design, 3*(1), 196–215. https://doi.org/10.51903/ijgd.v3i1.3661

Li, Z., Zhang, K., & Wong, A. (2026). Numerical-reasoning guardrails for a quant research assistant: A compact reproducible benchmark using SEC and FRED data. *Journal of Technology Informatics and Engineering, 5*(2), 75–90. https://doi.org/10.51903/jtie.v5i2.541

Li, Z., Zhou, S., & Zhou, Z. (2025). Financial risk dashboard design for institutional RWA investors: Visual hierarchy, chart comprehension, and explainability in FinChart-Bench. *International Journal of Graphic Design, 3*(1), 196–210. https://doi.org/10.51903/ijgd.v3i1.3715

Ling, Z., Xin, Q., Lin, Y., Su, G., & Shui, Z. (2024). Optimization of autonomous driving image detection based on RFAConv and triplet attention. *Applied and Computational Engineering, 77*, 210–217. https://doi.org/10.54254/2755-2721/77/2024MA0067

Lu, S., & Zou, T. (2026). Uncertainty-aware medical vision–language classification on a lightweight MedMNIST-compatible biomedical patch benchmark. *Journal of Technology Informatics and Engineering, 5*(2), 1–19. https://doi.org/10.51903/jtie.v5i2.530

Lu, Y., Zhou, H., & Zhang, Y. (2025). A constrained, data-driven budgeting framework integrating macro demand forecasting and marketing response modeling. *Journal of Technology Informatics and Engineering, 4*(3), 493–520. https://doi.org/10.51903/jtie.v4i3.466

Meng, S., Chen, J., & Zheng, I. (2026). LLM-inspired offline reranking for financial search: Query rewriting, hybrid retrieval, and listwise relevance ranking on FiQA. *Journal of Technology Informatics and Engineering, 5*(1), 361–378. https://doi.org/10.51903/jtie.v5i1.537

Mi, G., Ye, T., & Wood, D. (2025). A lightweight medical foundation model for cross-modal multi-task pretraining and parameter-efficient few-shot transfer on MedMNIST. *Journal of Technology Informatics and Engineering, 4*(3), 572–589. https://doi.org/10.51903/jtie.v4i3.492

Mu, J., Lu, Y., & Hwang, E. (2026). Structured visual brief interfaces for advertising design: A UI/UX framework for turning creative intentions into designer-editable graphic design cards. *International Journal of Graphic Design, 4*(1), 192–208. https://doi.org/10.51903/ijgd.v4i1.3702

Nie, J., Liu, G., Li, C., & Zou, T. (2026). Evidence-constrained incident visualization cards for distributed cloud logs: A UI/UX framework for turning Hadoop, OpenStack, and ZooKeeper logs into actionable SRE design interfaces. *International Journal of Graphic Design, 4*(1), 179–185. https://doi.org/10.51903/ijgd.v4i1.3703

OpenAI. (2023). *GPT-4 technical report [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2303.08774

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In International Conference on Learning Representations. https://openreview.net/forum?id=B1ckMDqlg

Su, W., Chen, S., & Zhao, C. (2025). Budgeted multi-hop retrieval agent for compositional question answering: A retrieval-policy evaluation on the official MultiHop-RAG benchmark. *Journal of Technology Informatics and Engineering, 4*(3), 649–662. https://doi.org/10.51903/jtie.v4i3.543

Su, W., Rao, H., & Ma, E. (2026). Privacy and data-integrity risk cards for LLM agents: A UI/UX design framework for secure human oversight under prompt-injection attacks. *International Journal of Graphic Design, 4*(1), 186–191. https://doi.org/10.51903/ijgd.v4i1.3699

Tu, H., Zhao, S., & Zhou, A. (2025). Visual brief cards for advertising design: A structured UI/UX framework for turning creative intentions into graphic design decisions. *International Journal of Graphic Design, 3*(1), 210–226. https://doi.org/10.51903/ijgd.v3i1.3714

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), Advances in neural information processing systems (Vol. 30, pp. 5998–6008). Curran Associates, Inc. https://proceedings.neurips.cc/paper/7181-attention-is-all-you-need

Wang, B., He, Y., Shui, Z., Xin, Q., & Lei, H. (2024). Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. *Applied and Computational Engineering, 64*(1), 89–94. https://doi.org/10.54254/2755-2721/64/20241350

Wang, C., Wen, Z., Zhang, R., Xu, P., & Jiang, Y. (2025). GPU memory requirement prediction for deep learning task based on bidirectional gated recurrent unit optimization transformer. In *2025 5th International Conference on Artificial Intelligence, Virtual Reality and Visualization (AIVRV)* (pp. 31–35). IEEE. https://doi.org/10.1109/AIVRV67401.2025.11350369

Wang, H., Ren, Y., & Chang, X. (2025). Layout-aware progressive PDF rendering: AI prioritization of PDF slices to reduce time-to-functional-first-frame on FUNSD. *Journal of Technology Informatics and Engineering, 4*(2), 425–446. https://doi.org/10.51903/jtie.v4i2.523

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. In International Conference on Learning Representations. https://openreview.net/forum?id=1PL1NIMMrw

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., & Goldblum, M. (2024). *LiveBench: A challenging, contamination-free LLM benchmark [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2406.19314

Wu, Q., Meng, S., & Zhao, J. (2025). Text-grounded LLM-assisted design rationale interfaces: Turning advertising layout metadata into explainable UI/UX decision cards. *International Journal of Graphic Design, 3*(1), 216–240. https://doi.org/10.51903/ijgd.v3i1.3713

Wu, Q., Mi, G., & Wood, D. (2025). Calibration-light subject-independent motor imagery BCI via self-supervised pretraining and Conformer. *Journal of Technology Informatics and Engineering, 4*(1), 239–262. https://doi.org/10.51903/jtie.v5i1.493

Xu, H., Chen, Y., & Med, A. (2025). Automatic detection and explanation of dark patterns from interface microcopy: Empirical comparison of BERT-style encoders, RoBERTa-style encoders, and LLM-style decoders on the ec-darkpattern dataset. *Journal of Technology Informatics and Engineering, 4*(3), 590–612. https://doi.org/10.51903/jtie.v4i3.491

Ye, T., Chang, X., & Zhong, E. (2025). Uncertainty-aware breast ultrasound explanation cards: A visual communication framework for image-based AI diagnostic support using BreastMNIST_224. *International Journal of Graphic Design, 3*(2), 365–380. https://doi.org/10.51903/ijgd.v3i2.3701

Ye, T., Mu, J., & Hunter, J. (2026). Off-policy evaluation and conservative policy selection for slot-level dynamic bidding and ranking on the Open Bandit Dataset (Small). *Journal of Technology Informatics and Engineering, 5*(1), 178–199. https://doi.org/10.51903/jtie.v5i1.503

Zhang, B., Ren, Y., & Zou, J. (2025). LLM-style explainable e-commerce recommendation cards: A UI/UX design framework for trust-calibrated product recommendation. *International Journal of Graphic Design, 3*(2), 381–396. https://doi.org/10.51903/ijgd.v3i2.3697

Zhang, B., Sun, X., Liu, G., & Zhou, B. (2026). LLM-style DevOps copilot for cloud-native troubleshooting: Retrieval-augmented runbook generation and command-safety evaluation. *Journal of Technology Informatics and Engineering, 5*(2), 104–118. https://doi.org/10.51903/jtie.v5i2.534

Zhang, J. (2025). From general human activity recognition to volleyball-oriented wearable transfer learning: Cross-dataset evidence from UCI HAR and WISDM for domain adaptation and edge deployment. *Journal of Technology Informatics and Engineering, 4*(1), 263–283. https://doi.org/10.51903/jtie.v4i1.524

Zhang, J. (2026). Early warning, grade prediction, and teacher-facing LLM-ready explanations toward an open volleyball course: Reproducible evidence from four public education datasets. *Journal of Technology Informatics and Engineering, 5*(2), 20–44. https://doi.org/10.51903/jtie.v5i2.525

Zhang, R., Wen, Z., Wang, C., Tang, C., Xu, P., & Jiang, Y. (2025). Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms. In *2025 5th International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)* (pp. 171–175). IEEE. https://doi.org/10.1109/CEI66465.2025.11398480

Zhang, Y., & Zhang, H. (2025a). A therapist-facing session copilot for live counseling support: Reasoning-guided retrieval and ranking from multi-turn counseling dialogues. *Journal of Technology Informatics and Engineering, 4*(2), 464–486. https://doi.org/10.51903/jtie.v4i2.547

Zhang, Y., & Zhang, H. (2025b). Visualizing the right counseling support: Evidence-linked recommendation cards for explainable mental health intake interfaces. *International Journal of Graphic Design, 3*(1), 214–229. https://doi.org/10.51903/ijgd.v3i1.3722

Zhao, S., Bai, J., & Roberson, D. (2025). Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba Clusterdata GPU Trace. *Journal of Technology Informatics and Engineering, 4*(3), 544–571. https://doi.org/10.51903/jtie.v4i3.498

Zhong, Z. S., Chen, J., Zhong, E., & Sun, X. (2025). Evidence-calibrated RAG for unanswerable question answering: Retrieval coverage, abstention calibration, and hallucination-proxy analysis on SQuAD 2.0. *Journal of Technology Informatics and Engineering, 4*(2), 502–520. https://doi.org/10.51903/jtie.v4i2.536

Zhong, Z. S., & Ling, S. (2024a). Improved theoretical guarantee for rank aggregation via spectral method. *Information and Inference: A Journal of the IMA, 13*(3), iaae020. https://doi.org/10.1093/imaiai/iaae020

Zhong, Z. S., & Ling, S. (2024b). *Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization*. arXiv. https://arxiv.org/abs/2408.05944

Zhou, B., Wang, H., & Chang, X. (2025). Distilling VMAF into an edge-deployable quality predictor: A pilot shot-level proxy with LLM-ready quality tokens. *Journal of Technology Informatics and Engineering, 4*(2), 447–463. https://doi.org/10.51903/jtie.v4i2.522

Zhou, H., & Zhang, K. (2025). News-based uncertainty and macro-market fusion for VIX direction forecasting: Evidence from the 2015–2024 FRED panel. *Journal of Technology Informatics and Engineering, 4*(2), 487–501. https://doi.org/10.51903/jtie.v4i2.540

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., & Hou, L. (2023). *Instruction-following evaluation for large language models [Preprint]*. arXiv. https://doi.org/10.48550/arXiv.2311.07911

Zhou, S., Chen, Y., & Lee, K. (2026). Accounting-aware evidence-constrained agents for disclosure, settlement, and secondary-market risk monitoring in tokenized. *Journal of Technology Informatics and Engineering, 5*(2), 60–74. https://doi.org/10.51903/jtie.v5i2.544