HHAI 2026
M. Alimardani et al. (Eds.)

doi:10.3233/FAIA260493

Learning and Reasoning for Legal AI via Intelligent Adaptive Multi-Model Routing with a Smart Evaluation Layer

Aditya MUKHOPADHYAY [a], Sara GHANE [b,c], Marco DI GENNARO [b,c]
[a] *Manipal Institute of Technology, Udupi-Karkala Rd, Eshwar Nagar, Manipal, Karnataka 576104, India*
[b] *DT Services and Consulting, Avenue des Cerfs 11, 4031 Liège, Belgium*
[c] *The Belgian Research Lab for AI in INdustry (BRAIN), Cantersteen 12, 1000 Bruxelles, Belgium, E-mail: brain@dtsc.be*

Abstract. AI is increasingly used in legal workflows, but trusted deployment requires reliable performance across heterogeneous tasks. Single-model deployment is brittle: smaller models can be efficient on retrieval-style extraction, while larger models are stronger on interpretive reasoning but costly. We present Legal Adaptive Multi-Model Router (LAMMR), a routing architecture that combines rule-based signals (intent cues, document-length heuristics and confidence fallbacks) with data-driven routing, plus a task-aware evaluation layer that scores extraction with overlap-oriented metrics (F2/Jaccard), and reasoning with LegalBench-aligned correctness scoring.

Across 1,093 tasks (999 reasoning, 94 extraction) and 2,186 model-level evaluations, a single model architecture based on Llama Maverick attains extraction/reasoning scores of 0.2988/0.6425, while learning-only LAMMR reaches 0.3978/0.6578. This corresponds to +33.15% extraction, +2.40% reasoning, and +3.70% overall improvement versus Llama Maverick. These results indicate that routing policy is associated with improved legal-AI accuracy and reliability, with cost treated as a secondary operational consideration.

Keywords. Legal AI, adaptive routing, multi-model routing, legal reasoning, contract extraction, task-aware evaluation

1. Introduction

Current legal AI systems are widely used for document review, compliance checks, clause extraction, and legal question answering. In practice, many deployments rely on a single general-purpose model or fixed model assignment, with limited adaptation to task type, document length, or legal intent.

These designs often fall short for heterogeneous legal workloads. Extraction-heavy tasks reward high recall and robust span selection, while reasoning-heavy tasks demand stronger inferential consistency and legal interpretation. A static single-model strategy therefore creates a persistent quality-control tradeoff: one

model may be efficient but brittle on interpretation, while another may be stronger on reasoning but slower and more expensive. In legal operations, missed obligations, incorrect clause interpretation, or unsupported recommendations can trigger compliance violations, contract leakage, rework costs, and dispute exposure. Reliability, traceability, and transparency are therefore essential operational requirements, not optional features.

Our work focuses on model allocation as the primary design problem. We present *Legal Adaptive Multi-Model Router (LAMMR)*, a deployable architecture that combines interpretable rule routing with learned routing to assign legal requests to specialized models. Route quality is evaluated primarily by task-level accuracy, with failure concentration (zero-score rate), estimated cost, and large-model usage reported as secondary deployment indicators. This paper makes four contributions: a legal-domain routing architecture that combines deterministic controls with learned selection; a dual-task protocol that evaluates extraction and reasoning separately; an empirical study covering 1,093 tasks and 2,186 model-level evaluations using a multi-seed setup; and a deployment analysis addressing robustness, auditability, and governance. Section 2 reviews related work on inference-time routing, collaborative orchestration, and legal-domain benchmarking. Section 3 defines the objective and the research questions. Section 4 describes the LAMMR architecture and its components. Section 5 presents the experimental setup. Section 6 discusses results, and Section 7 covers future work.

2. Related Work

Research on multi-model language systems has moved from static model choice to adaptive routing policies that optimize quality under cost and latency constraints. Prior work spans inference-time routing, collaborative orchestration, trusted deployment, and legal-domain benchmarking.

Inference-time routing. Prior work studies cost-aware and adaptive model selection under deployment constraints. FrugalGPT introduced cost-aware LLM cascades and showed that adaptive escalation can substantially reduce spending while maintaining quality [1]. Like FrugalGPT, LAMMR treats cost as a deployment constraint; however, it routes requests using intent signals rather than cascade thresholds and incorporates legal-domain rules absent in FrugalGPT. RouteLLM learned request-level routing from preference data and demonstrated strong cost–quality tradeoffs with lightweight routers [2]. Unlike RouteLLM's purely learned approach, LAMMR integrates deterministic rules to ensure auditability for high-risk legal reasoning. Hybrid LLM, TREACLE, and SkyLLM extend routing to difficulty-aware, policy-based, and cross-API settings [3,4,5]. Other approaches improve generalization (Universal Routing [6]), reduce labeling needs (Smoothie [7]), or optimize selection via contrastive learning, bandits, and meta-models [8,9,10]. Model selection efficiency and training without task-specific labels are explored in [11,12], while RouterEval and MMR-Bench provide large-scale and multimodal benchmarks [13,14]. Ensemble methods such as LLM-Blender and recent surveys unify routing, ranking, and fusion strategies [15,16]. LAMMR builds on this work by combining intent-based routing with determinis-

tic legal rules, enabling cost-aware routing with stronger auditability in regulated settings.

Collaborative and hybrid orchestration. Prior work explores coordinated use of multiple models and tools. Mixture-of-Agents and RMoA study layered collaboration and diversity-aware aggregation [17,18]. AutoMix and related approaches use confidence and self-verification for adaptive escalation [19], and ZOOTER distills reward signals into efficient routing policies [20]. Mixture-of-Thought further leverages reasoning consistency for selective escalation [21]. LAMMR differs by avoiding per-request self-verification, instead using lightweight rules and a trained classifier to enable low-overhead, explainable routing suitable for legal governance.

Trusted deployment and governance. Cascaded human-AI decision frameworks made deferral, abstention, and human escalation explicit in the routing policy, which is directly relevant for high-risk settings [22]. This direction reinforces that routing should be treated as an auditable control mechanism, not only an efficiency trick.

Legal-domain evidence. LegalBench highlights the heterogeneity of legal tasks and serves as a broad benchmark for legal reasoning [23]. ContractEval focuses on clause-level risk analysis in commercial contracts and reveals performance differences across models [24], while recent surveys systematize legal NLP task taxonomies and datasets [25]. These findings motivate LAMMR's dual-task routing design to handle both extraction and reasoning efficiently.

Taken together, prior work strongly supports adaptive routing, but most studies remain domain-general and do not jointly optimize legal-task-aware evaluation with interpretable policy control. In this paper, the gap is addressed by proposing LAMMR, which combines deterministic rule transparency with learned adaptation in a legal deployment setting.

3. Problem Formulation and Positioning

We model routing as a decision function $R(q, d) \rightarrow m$, where q is a user query, d is task/document context, and m is the selected model. The objective is to maximize task-level accuracy subject to a cost budget B:

$$\max_{R} \mathbf{E}\left[\text{Accuracy}(m \mid q, d)\right] \quad \text{subject to} \quad \text{Cost} \leq B. \tag{1}$$

In Equation (1), task-level accuracy is the primary optimization target and cost is the hard operational constraint. LAMMR additionally enforces auditability as a design constraint by logging all route decisions and exposing route rationale for inspection; auditability is a governance requirement rather than an optimization variable in Eq. (1). One could choose latency as a optimization parameter. In this study, we prioritized accuracy over response time, and real-time latency profiling will be treated in a future study.

In this work, we address three research questions:

- **RQ1:** Does adaptive routing outperform a single-model strategy?
- **RQ2:** Which signals matter most: rules, learned predictors, or hybrids?

- **RQ3:** Can mixed legal workloads be evaluated without masking failures?

LAMMR combines hybrid AI control, inference routing, and legal AI benchmarking (e.g., LegalBench and ContractEval), with a focus on deployment needs such as transparency and control. LAMMR sits at the intersection of *hybrid AI control* (combining symbolic policies and learned components), *inference routing* (selecting models based on request properties), and *legal AI benchmarking* (task-grounded evaluation frameworks such as LegalBench [23] and clause-level contract risk benchmarks such as ContractEval [24]). Our focus is purely operational: we treat routing transparency and controllability as deployment requirements, rather than optional enhancements.

4. Methodology and Architecture

4.1. System Overview

LAMMR is a request-level routing architecture that selects between two specialized models: an extraction model (`Llama-4-Maverick-17B-128E-Instruct-FP8`) and a reasoning model (`Llama-3.3-70B-Instruct`). For each input pair (q, d), where (q) represents a query, and (d) represents a legal document, the system computes lightweight intent and complexity signals, runs both symbolic and learned routing logic, chooses a model according to policy, and then executes a task-family-aware evaluation pass. Model allocation is thus treated as a control layer rather than a fixed setting.

Llama-series open-weight models are selected for two reasons. First, they support reproducibility and deployment transparency: routing decisions, model outputs, and evaluation scores can be logged and audited without reliance on proprietary APIs or version-lock risk, directly satisfying LAMMR's auditability requirement. Second, using a fixed candidate-model pair lets the evaluation focus on routing-policy behavior within this controlled model set, rather than broad cross-vendor model comparison. We do not claim that these are the best possible legal-domain models; evaluating additional open and proprietary legal LLMs remains future work.

4.2. Rule-Based Router

The rule-based router provides deterministic, explainable routing via a fixed precedence: reasoning-intent detection, extraction-intent detection, a long-document heuristic, and a reasoning default. It ensures stable, human-readable behavior using explicit triggers and thresholds that domain experts can review and adjust. Each route is tied to observable rule activations, enabling inspection, policy updates, and validation for high-risk cases. Figure 1 shows this flow. The policy is fully auditable and serves as a transparent baseline against learning-based and hybrid routers.

For an input pair (q, d) (query and document/context), let m_E denote the extraction model and m_R the reasoning model, the rule-base router is defined in Eq. (2).

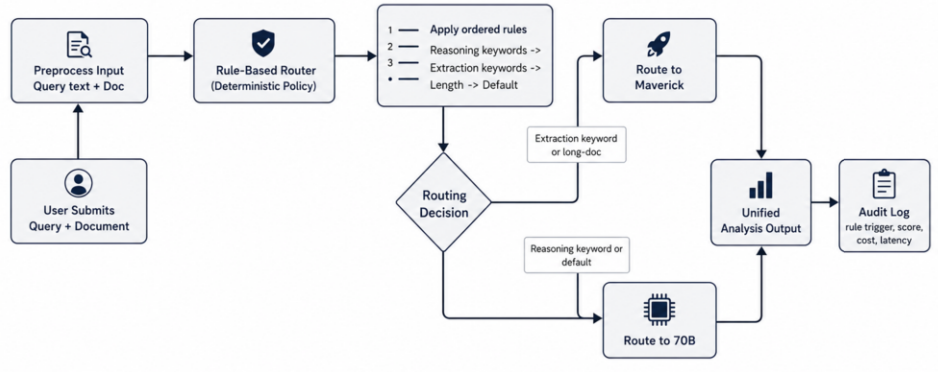

Figure 1. Deterministic rule-based routing baseline. Per query-document pair, the policy applies ordered checks (reasoning keywords, extraction keywords, long-document heuristic, default), routes to Maverick or 70B, returns a unified logged output for auditability.

$$R_{\text{rule}}(q,d) = \begin{cases} m_R, & \text{if } \phi_{\text{reason}}(q) = 1, \\ m_E, & \text{if } \phi_{\text{extract}}(q) = 1, \\ m_E, & \text{if } \ell(d) > \tau, \\ m_R, & \text{otherwise}, \end{cases} \tag{2}$$

where ϕ_{reason} and ϕ_{extract} are keyword-intent indicators, and $\ell(d)$ is a length heuristic with threshold τ. We use Eq. (1) as cost function to solve this equation.

4.3. LAMMR Router

The LAMMR router complements symbolic logic by modeling patterns that are not easily captured by fixed rules. It is trained on historical benchmark instances with model-preference labels derived from observed task outcomes, and it predicts which model is likely to produce the better response for a given input. In the current pipeline, text features are based on Term Frequency-Inverse Document Frequency (TF-IDF) n-grams from the query, while numeric features include query and document length.

A Random Forest (RF) classifier is used as the current predictor because it performs well on mixed sparse+dense features, supports robust non-linear decisions, and integrates cleanly with confidence-based policy. This learned layer allows routing behavior to adjust when workload shifts, while remaining simple enough to retrain and inspect during evaluation.

Figure 2 shows the deployment flow of the LAMMR router policy. The LAMMR router estimates model preference from features $x(q,d)$ as defined in Eq. (3), where p_θ is the trained classifier score and δ is a decision threshold:

$$R_{\text{learn}}(q,d) = \begin{cases} m_R, & \text{if } p_\theta(m_R \mid x(q,d)) \geq \delta, \\ m_E, & \text{otherwise}, \end{cases} \tag{3}$$

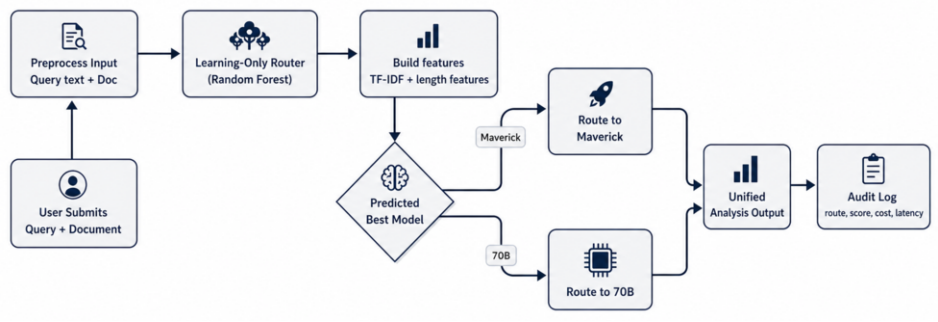

Figure 2. LAMMR router policy used in the final validated run. The policy predicts routes from query/document features, selects Maverick or 70B without rule overrides, and emits unified logged outputs for score, cost, and latency auditing.

4.4. Hybrid Router Policy

The hybrid policy combines rules with learning. The learned router first produces a candidate route for the input. A reasoning-keyword confidence gate then decides whether to override it: if the rule confidence $c_{\text{rule}} \geq \gamma$ (i.e., strong reasoning-intent keywords are detected in q), the system routes to the reasoning model; otherwise, the learned candidate route is accepted. This structure keeps the learned router as the default path while reserving deterministic rule authority for high-confidence reasoning-intent signals.

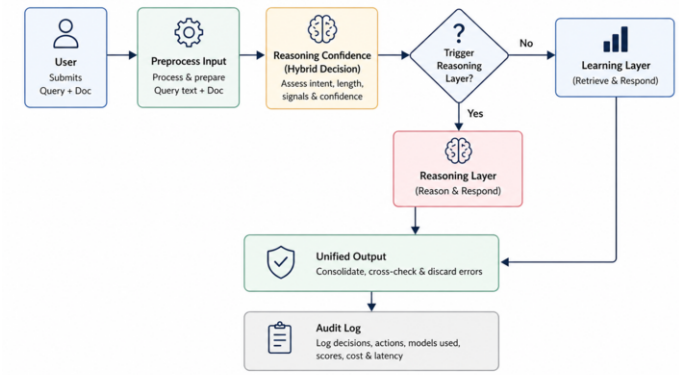

Figure 3. Hybrid routing policy high-level visualization. The rule based learned route from query/document features may be overridden by a deterministic reasoning-keyword rule. The final output is unified and fully logged for auditability.

The hybrid policy combines learned prediction with deterministic rule overrides, improving interpretability and auditability for high-risk reasoning. It is evaluated against the LAMMR router and static baselines in Section 6.

Operationally, hybrid routing also supports controlled deployment tuning. By adjusting the confidence threshold, practitioners can move along an interpretability–adaptation spectrum without changing the base models. The same framework can be paired with uncertainty-triggered fallbacks, enabling con-

servative behavior for cases where routing confidence is weak or risk tolerance is low. The hybrid router composes rule (Eq 2) and learning policies (Eq 3) through a confidence gate $c_{\text{rule}}(q, d)$:

$$R_{\text{hybrid}}(q, d) = \begin{cases} R_{\text{rule}}(q, d), & \text{if } c_{\text{rule}}(q, d) \geq \gamma, \\ R_{\text{learn}}(q, d), & \text{otherwise,} \end{cases} \tag{4}$$

In Eq. (4), $c_{\text{rule}}(q, d) \in \{0, 1\}$ is a binary confidence indicator: it equals 1 when the reasoning-keyword detector fires on query q (i.e., $\phi_{\text{reason}}(q) = 1$), and 0 otherwise. The override threshold is $\gamma = 1$, so the rule overrides the learned route exclusively when the reasoning-keyword signal is present and fires for reasoning-intent inputs only.

4.5. Smart Evaluation Layer

The Smart Evaluation Layer enforces task-aware scoring instead of a single blended metric. Extraction outputs are evaluated with overlap-oriented measures (Jaccard/F2 family), while reasoning outputs are evaluated with correctness-based criteria aligned with LegalBench-style tasks [23]. Importantly, these scoring criteria are tied to the *task family* of the input, not to the model that produced the output: LegalBench-aligned correctness scoring applies to all reasoning-task outputs regardless of which model was routed to handle them, and ContractEval-aligned overlap scoring applies to all extraction-task outputs regardless of model assignment. Routing decisions and evaluation criteria are therefore independent. This separation is critical because extraction and reasoning errors have different semantics and different operational consequences.

5. Experimental Setup

5.1. Dataset

The benchmark comprises 1,093 tasks (999 reasoning, 94 extraction), combining control items with benchmark-aligned legal tasks. LegalBench, a collaboratively constructed benchmark spanning diverse legal reasoning forms, anchors the reasoning-layer evaluation [23]. ContractEval, a clause-level benchmark for legal risk identification in commercial contracts, anchors the extraction-layer evaluation where clause/span fidelity is central [24]. Together, they provide complementary coverage: LegalBench probes reasoning robustness, while ContractEval targets extraction reliability.

5.2. Baselines

We evaluate a static single-model baseline using Llama Maverick alongside LAMMR policies: the rule-based router (Eq. (2)), the learned router (Eq. (3)), the hybrid router (Eq. (4)), and two learned uncertainty-fallback variants. In Table 1, LAMMR router denotes the learned-only policy, while Hybrid router adds deter-

ministic rule overrides. For the uncertainty-fallback variants, the learned router first selects a candidate model using Eq. (3) and computes confidence as the larger of the two model probabilities. If this confidence is below $\delta_u = 0.7$, the candidate is replaced by a fixed fallback: **fallback (70B)** uses `Llama-3.3-70B-Instruct`, whereas **fallback (Maverick)** uses `Llama-4-Maverick-17B-128E-Instruct-FP8`. These variants differ from the rule-based router, which uses deterministic keyword and length checks without a learned confidence estimate.

5.3. Evaluation Metrics

Primary reporting is accuracy-centered at the task level. For reasoning, scoring follows LegalBench-aligned correctness criteria so that route quality reflects legal interpretation and inferential robustness rather than lexical overlap [23]. For extraction, scoring follows ContractEval-aligned overlap-oriented metrics (Jaccard/F2 family), which directly capture retrieval fidelity for contractual risk signals [24]. Accordingly, LegalBench and ContractEval are treated as complementary evaluation references: the former emphasizes robust reasoning, while the latter emphasizes clause/span extraction reliability. We additionally report zero-score rate to capture failure concentration, and treat estimated cost and 70B usage as secondary factors. Statistical comparisons against the single-model architecture using Llama Maverick are reported with paired bootstrap confidence intervals and p-values.

5.4. Cost Estimation

Monetary cost is estimated at the API level using the formula:

$$\text{cost} = (\text{input tokens} \times r_{\text{in}}) + (\text{output tokens} \times r_{\text{out}}) \tag{5}$$

where r_{in} and r_{out} are input and output rates per token for each model on the Azure AI Foundry deployment. As explained in Sec. 4.1, we use two different models accessible via Azure AI Foundry: for the extraction model we use `Llama-4-Maverick-17B-128E-Instruct-FP8` (lower per-token rate), while for the reasoning model, we use `Llama-3.3-70B-Instruct` (higher per-token rate). The energy cost was not measured; all cost figures in Tab. 1 are scoped to API-level pricing only. We prioritized accuracy over monetary cost, so cost is reported as a secondary operational indicator and is not the primary optimization target in Eq. (1).

5.5. Experimental Setup

The protocol uses retry-safe evaluation with five seeds (42–46), 4000 bootstrap samples, task-balance power 1.5, and uncertainty threshold 0.7. Evaluation is performed separately for LegalBench-aligned reasoning and ContractEval-aligned extraction before aggregation, preserving distinct error semantics. We report model-level baselines (2,186 evaluations across 1,093 tasks) and multi-seed policy-level results.

6. Results and Discussion

All claims here are based on final validated main run defined in Experimental Setup: 1,093 tasks (999 reasoning, 94 extraction) and 2,186 model-level evaluations. Each evaluation corresponds to the general Eq. 1. Task-family scores are available for the single-model architecture and the proposed routers.

Table 1. Final validated policy results (mean over 5 seeds). Best values per metric are bold; higher is better for scores, lower for cost, 70B usage, and zero-score rate. Zero-score rate denotes complete failures. All LAMMR policies share the same extraction zero-score rate (26.3%) due to fixed routing, while reasoning zero-score rate varies. Avg score is the task-weighted mean over 1,093 tasks (91% reasoning, 9% extraction); extraction and reasoning metrics should be interpreted separately due to imbalance.

Policy	Avg score	Extraction score	Reasoning score	Avg cost (USD)	70B usage	Zero-score rate
LAMMR router (primary)	**0.6353**	0.3978	**0.6578**	0.2536	10.68%	**10.9%**
Uncertainty fallback (70B)	0.6305	**0.4259**	0.6500	0.2830	22.01%	12.0%
Uncertainty fallback (Maverick)	0.6258	0.3978	0.6475	0.2283	3.65%	11.9%
Hybrid router	0.6109	0.3978	0.6311	0.2778	28.13%	11.4%
Rule-based router	0.5748	0.2988	0.6011	0.2581	64.84%	13.7%
Single-model architecture using Llama Maverick	0.6126	0.2988	0.6425	**0.1607**	**0.00%**	12.3%

Interpreting zero-score rate. There is no universal acceptable zero-score rate; the threshold depends on the risk level of the legal workflow. In real deployment, a zero-score rate case means complete task failure, such as a missed extraction target or an unsupported reasoning answer. Such cases should trigger safeguards such as human review, abstention, retry, or rerouting rather than being delivered directly. Thus, Table 1 uses zero-score rate as a comparative reliability indicator: lower is better, but the observed rates are not claimed to be acceptable for autonomous legal deployment.

Overall accuracy and policy ordering. The strongest overall policy is the LAMMR router (0.6353), followed by uncertainty fallback (70B) (0.6305), uncertainty fallback (Maverick) (0.6258), the single-model architecture using Llama Maverick (0.6126), hybrid router (0.6109), and the rule-based router (0.5748). The deployed execution path for the best non-oracle policy is illustrated in Figure 2, while the deterministic baseline flow is shown in Figure 1.

Extraction behavior. The largest extraction improvement comes from uncertainty fallback (70B), which reaches 0.4259 versus 0.2988 for the single-model architecture using Llama Maverick (+0.1272). The LAMMR router, uncertainty fallback (Maverick), and hybrid router each reach 0.3978 (+0.0991 over baseline), showing that substantial extraction gains are achievable without always routing to the largest model.

Reasoning behavior. The LAMMR router also has the strongest reasoning score (0.6578), followed by uncertainty fallback (70B) (0.6500), uncertainty fallback (Maverick) (0.6475), the single-model architecture using Llama Maverick (0.6425), and hybrid router (0.6311). The smaller spread in reasoning scores among the top policies compared with extraction scores suggests that extraction routing decisions contribute most of the observed cross-policy separation in total score.

Operational footprint (secondary): cost and large-model usage. The lowest-cost policy is the single-model architecture using Llama Maverick (USD 0.1607), but it also has the weakest overall score. Among routed policies, uncertainty fallback (Maverick) is the most cost-efficient (USD 0.2283) and uses 70B only 3.65% of the time, while still improving average score over the single-model architecture using Llama Maverick. Uncertainty fallback (70B) provides the highest extraction score but at higher cost (USD 0.2830) and higher 70B usage (22.01%). The LAMMR router sits between these extremes, delivering the best total score at moderate cost (USD 0.2536) and moderate 70B usage (10.68%). Hybrid routing shows higher large-model dependence in this evaluation (28.13%).

Statistical robustness. Paired bootstrap tests versus single-model architecture using Llama Maverick show significant improvements for LAMMR router ($\Delta = +0.02265$, 95% confidence interval (CI) [0.01438, 0.03168], $p < 0.001$), uncertainty fallback (70B) ($\Delta = +0.01789$, 95% CI [0.00663, 0.02965], $p = 0.002$), and uncertainty fallback (Maverick) ($\Delta = +0.01316$, 95% CI [0.00767, 0.01945], $p < 0.001$). For hybrid router, hybrid-only rerun reports $\Delta = +0.00717$ (+1.19%), 95% CI [-0.00277, 0.01762], and $p = 0.142$. In hybrid-only rerun, extraction gains persist, reasoning is near parity, and overall gain over Always-Maverick is not significant at $\alpha = 0.05$.

Evaluation without masking high-risk failure modes (RQ3). Results should be interpreted with the benchmark scope in mind. The dataset is imbalanced (999 reasoning vs. 94 extraction), transfer to other legal domains is not yet established [23], extraction diversity is limited, costs are API-level estimates, and borderline outputs received limited human adjudication. Public benchmark contamination, including unequal model exposure, also cannot be ruled out; therefore, we interpret results as routing-policy comparisons under a shared evaluation protocol rather than uncontaminated estimates of standalone model capability. These constraints motivate our RQ3 design: extraction and reasoning are scored separately, zero-score rate is reported to surface complete failures, and cross-family averages are treated as secondary so that the dominant reasoning split does not mask extraction failures.

6.1. Deployment Considerations and Future Work

LAMMR is intended for legal decision support, not automated legal advice. Its deployment value lies in inspectable routing, explicit fallbacks, and logged routes/outputs, with human review, abstention, drift monitoring, and recalibration for high-impact use. Future work will expand extraction and adversarial tasks, add reasoning-aware routing, run task-family-separated significance tests, include expert error analysis, and evaluate temporal and domain shift.

7. Conclusion

Legal Adaptive Multi-Model Router (LAMMR) provides a practical and effective framework for improving legal-AI performance under deployment constraints. The method combines interpretable rules and learning based router with a task-aware

evaluation layer that separately measures reasoning quality and clause/span extraction quality. In our experiment, routed policies outperform the single-model architecture using Llama Maverick, with the LAMMR router achieving the significant scores at moderate operational cost. Overall, the findings reinforce adaptive routing as an architecture for improving legal-AI accuracy and cost effectiveness in real Legal Workflows.

References

[1] Chen L, et al.. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance; 2023. Available from: https://arxiv.org/abs/2305.05176.

[2] Ong I, et al. RouteLLM: Learning to Route LLMs from Preference Data. In: The Thirteenth International Conference on Learning Representations. ICLR 2025; 2025. Available from: https://openreview.net/forum?id=8sSqNntaMr.

[3] Ding D, et al. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. In: The Twelfth International Conference on Learning Representations. ICLR 2024; 2024. Available from: https://openreview.net/forum?id=02f3mUtqnM.

[4] Zhang X, et al. Efficient Contextual LLM Cascades through Budget-Constrained Policy Learning. In: Advances in Neural Information Processing Systems. NeurIPS 2024; 2024. Available from: https://openreview.net/forum?id=aDQlAzO9dS.

[5] Zhao H, et al. SkyLLM: Cross-LLM-APIs Federation for Cost-Effective Query Processing. In: Findings of the Association for Computational Linguistics: ACL 2025. Vienna, Austria: Association for Computational Linguistics; 2025. p. 20864-73.

[6] Jitkrittum W, et al.. Universal Model Routing for Efficient LLM Inference; 2025. Available from: https://arxiv.org/abs/2502.08773.

[7] Guha N, Chen MF, Chow T, Khare IS, Re C. Smoothie: Label free language model routing. In: Advances in Neural Information Processing Systems (NeurIPS); 2024. Available from: https://proceedings.neurips.cc/paper_files/paper/2024/hash/e6b57a990462df5afa58d64ce2709db9-Abstract-Conference.html.

[8] Shirkavand R, et al. Cost-Aware Contrastive Routing for LLMs. In: Advances in Neural Information Processing Systems. NeurIPS 2025; 2025. Available from: https://openreview.net/forum?id=4Qe2Hga43N.

[9] Wang W, et al.. Learning to Route LLMs from Bandit Feedback: One Policy, Many Trade-Offs; 2025. Available from: https://arxiv.org/abs/2510.07429.

[10] Šakota M, et al. Fly-Swat or Cannon? Cost-Effective Language Model Choice via Meta-Modeling. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. WSDM 2024; 2024. Available from: https://arxiv.org/abs/2308.06077.

[11] Zhang YK, Huang TJ, Ding YX, Zhan DC, Ye HJ. Model Spider: Learning to Rank Pre-Trained Models Efficiently. In: Advances in Neural Information Processing Systems (NeurIPS); 2023. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/hash/2c71b14637802ed08eaa3cf50342b2b9-Abstract-Conference.html.

[12] Shnitzer T, et al.. Large Language Model Routing with Benchmark Datasets; 2023. Available from: https://arxiv.org/abs/2309.15789.

[13] Huang Z, et al. RouterEval: A Comprehensive Benchmark for Routing LLMs to Explore Model-Level Scaling Up in LLMs. In: Findings of the Association for Computational Linguistics: EMNLP 2025. Suzhou, China: Association for Computational Linguistics; 2025. p. 3860-87. Available from: https://aclanthology.org/2025.findings-emnlp.208/.

[14] Ma H, et al.. MMR-Bench: A Comprehensive Benchmark for Multimodal LLM Routing; 2026. Available from: https://arxiv.org/abs/2601.17814.

[15] Jiang D, et al. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. ACL 2023; 2023. p. 14165-78.

[16] Chen Z, Li J, Chen P, Li Z, Sun K, Luo Y, et al.. Harnessing Multiple Large Language Models: A Survey on LLM Ensemble; 2025. Available from: https://arxiv.org/abs/2502.18036.

[17] Wang J, et al. Mixture-of-Agents Enhances Large Language Model Capabilities. In: The Thirteenth International Conference on Learning Representations. ICLR 2025; 2025. Available from: https://openreview.net/forum?id=h0ZfDIrj7T.

[18] Xie Z, et al. RMoA: Optimizing Mixture-of-Agents through Diversity Maximization and Residual Compensation. In: Findings of the Association for Computational Linguistics: ACL 2025. Vienna, Austria: Association for Computational Linguistics; 2025. p. 6575-602.

[19] Aggarwal P, et al. AutoMix: Automatically Mixing Language Models. In: Advances in Neural Information Processing Systems. NeurIPS 2024; 2024. Available from: https://proceedings.neurips.cc/paper_files/paper/2024/hash/ecda225cb187b40ea8edc1f46b03ffda-Abstract-Conference.html.

[20] Lu K, et al. Routing to the Expert: Efficient Reward-Guided Ensemble of Large Language Models. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL 2024; 2024. Available from: https://aclanthology.org/2024.naacl-long.109/.

[21] Yue M, et al. Large Language Model Cascades with Mixture of Thought Representations for Cost-Efficient Reasoning. In: The Twelfth International Conference on Learning Representations. ICLR 2024; 2024. Available from: https://openreview.net/forum?id=6okaSfANzh.

[22] Fanconi C, et al. Cascaded Language Models for Cost-Effective Human–AI Decision-Making. In: Advances in Neural Information Processing Systems. NeurIPS 2025; 2025. Available from: https://openreview.net/forum?id=4Q1vA6P9J9.

[23] Guha N, et al. LEGALBENCH: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NeurIPS 2023. Red Hook, NY, USA: Curran Associates Inc.; 2023. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/hash/89e44582fd28ddfea1ea4dcb0ebbf4b0-Abstract-Datasets_and_Benchmarks.html.

[24] Liu S, et al. ContractEval: Benchmarking LLMs for Clause-Level Legal Risk Identification in Commercial Contracts. In: Proceedings of the Natural Legal Language Processing Workshop 2025. Suzhou, China: Association for Computational Linguistics; 2025. Available from: https://aclanthology.org/2025.nllp-1.19/.

[25] Singh A, Joshi A, Jiang J, young Paik H. A Survey of Classification Tasks and Approaches for Legal Contracts; 2025. Available from: https://arxiv.org/abs/2507.21108.