RouteNLP: Closed-Loop LLM Routing with Conformal Cascading and Distillation Co-Optimization

Dongxin Guo[1], **Jikun Wu**[2], **Siu Ming Yiu**[1]
[1]The University of Hong Kong [2]Stellaris AI Limited
bettyguo@connect.hku.hk, hk950014@connect.hku.hk,
smyiu@cs.hku.hk

Abstract

Serving diverse NLP workloads with large language models is costly: at one enterprise partner, inference costs exceeded $200K/month despite over 70% of queries being routine tasks well within the capability of smaller models. We present RouteNLP, a closed-loop framework that routes queries across a tiered model portfolio to minimize cost while satisfying per-task quality constraints. The framework integrates three components: a difficulty-aware router with shared task-conditioned representations trained on preference data and quality signals; confidence-calibrated cascading that uses conformal prediction for distribution-free threshold initialization; and a distillation-routing co-optimization loop that clusters escalation failures, applies targeted knowledge distillation to cheaper models, and automatically retrains the router, yielding over twice the cost improvement of untargeted distillation. In an 8-week pilot deployment processing ∼5K queries/day at an enterprise customer-service division, RouteNLP reduced inference costs by 58% while maintaining 91% response acceptance and reducing p99 latency from 1,847 ms to 387 ms. On a six-task benchmark spanning finance, customer service, and legal domains, the framework achieves 40–85% cost reduction while retaining 96–100% quality on structured tasks and 96–98% on generation tasks, with human evaluation confirming that 74.5% of routed generation outputs match or exceed frontier-model quality.

1 Introduction

LLMs have proliferated quickly, and enterprise teams now face a sharp tradeoff: models spanning orders of magnitude in cost and capability are available, from lightweight distilled models costing fractions of a cent to frontier models costing dollars per thousand tokens (Chen et al., 2024). Financial institutions (Wu et al., 2023), travel platforms (Zhang et al., 2025), and customer service operations all face the same tension: *how to deliver consistent quality while minimizing inference costs under strict latency constraints*.

This paper arose from a concrete production need. Working with an enterprise partner in financial services, we observed NLP serving costs exceeding $200K/month, yet over 70% of queries were routine tasks that did not require frontier model capabilities. Across enterprise domains (finance, customer service, legal), only 25–35% of queries require frontier models, reflecting a heavy-tailed difficulty distribution that RouteNLP exploits (see Appendix A for detailed deployment scenarios). For example, extracting standard entities from templated SEC filings is straightforward for small models, while summarizing novel regulatory guidance demands frontier capabilities; similarly, in customer service, over 70% of queries are routine (order status, FAQ matching) where small models suffice.

Existing routing approaches have critical limitations for enterprise deployment: they are typically evaluated on single benchmarks (Ong et al., 2024; Ding et al., 2024), ignore production constraints such as latency SLAs (Chen et al., 2024), and most importantly decouple the router from the model portfolio, treating the set of available models as a fixed input rather than a learnable artifact. The recent unified framework of Dekoninck et al. (2025) provides theoretical foundations, but a gap remains between academic frameworks and production systems. RouteNLP's central contribution is to break the fixed-portfolio assumption by closing the loop between routing failures and the portfolio itself: escalation logs are clustered, used to generate targeted distillation data, folded back into cheaper-tier models, after which the router and conformal thresholds are recalibrated. This change yields over twice the cost reduction of untargeted distillation at equal data volume (21.7% vs. 9.4%, §5) and is, to our knowledge, the first such loop validated in a

multi-week production deployment.

Our specific contributions are:

- A **closed-loop distillation-routing co-optimization loop** (§3.3) that clusters escalation failures, applies targeted knowledge distillation to cheaper tiers, and automatically retrains the router. At equal data volume, this yields over 2× the cost gain of untargeted distillation.
- A **multi-task difficulty-aware router** (§3.1) with shared task-conditioned representations, trained jointly on preference data and per-task quality signals so that a single encoder can learn task-dependent difficulty patterns.
- **Confidence-calibrated cascading** (§3.2) that uses conformal risk control to initialize thresholds in a distribution-free manner. We are explicit about its caveats: the guarantee is marginal rather than per-query, and distribution shift can violate it.
- A **six-task multi-domain benchmark** (finance, customer service, legal) and an **8-week pilot deployment** (∼5K queries/day) that validates the simulation predictions to within a 4-point gap on cost reduction (62% benchmark, 58% pilot; §4–§6).

2 Related Work

LLM Routing and Cascading. Model routing was pioneered by Chen et al. (2020) and extended to LLMs by Chen et al. (2024). Subsequent work trained routers on human preference data (Ong et al., 2024), introduced tunable quality thresholds (Ding et al., 2024), formulated routing as a POMDP (Aggarwal et al., 2024), and used meta-modeling for per-query performance prediction (Sakota et al., 2024; Shnitzer et al., 2023; Nguyen et al., 2024). RouterBench (Hu et al., 2024) established systematic evaluation. Dekoninck et al. (2025) showed that optimal serving lies on a continuum between pure routing and cascading; RouteNLP's design is consistent with this analysis. Industry systems such as Microsoft's Copilot routing (Varangot-Reille et al., 2025) demonstrate production adoption. For cascading, Varshney and Baral (2022) demonstrated up to 88.9% savings, Gupta et al. (2024) proposed token-level uncertainty for deferral, and Yue et al. (2023) combined cascading with diverse reasoning. Our work extends this literature to multi-task enterprise settings with SLA awareness and distillation co-optimization. A detailed feature comparison with prior systems is in Appendix B.

Uncertainty, Distillation, and Efficient Serving. Principled deferral requires reliable confidence estimates; conformal prediction (Blot et al., 2025; Quach et al., 2024) and semantic entropy (Kuhn et al., 2023) provide calibrated or distribution-free approaches. Knowledge distillation (Hinton et al., 2015; Sanh et al., 2019) creates capable small models as routing targets; our co-optimization loop is conceptually related to Self-Refine (Madaan et al., 2023) and active learning, but operates at the system level across multiple models. Production routing depends on efficient infrastructure: vLLM (Kwon et al., 2023), continuous batching (Yu et al., 2022), speculative decoding (Leviathan et al., 2023; Cai et al., 2024), S-LoRA (Sheng et al., 2023), and quantization (Frantar et al., 2022; Lin et al., 2024). MoE architectures (Shazeer et al., 2017; Jiang et al., 2024) provide an intra-model analogue to inter-model routing.

Position of RouteNLP. The literature above establishes routing, cascading, conformal calibration, and distillation as well-studied building blocks. The gap that motivates our work is that no prior system combines all four into a single pipeline that treats the model portfolio itself as a learned artifact rather than a fixed input. Table 8 (Appendix B) makes this gap concrete along four axes (*multi-task* evaluation, *formal calibration*, *co-optimization*, and *SLA awareness*) that jointly characterize what an industrial routing system must offer. The next section presents the framework that fills this gap.

3 The RouteNLP Framework

The system operates over a model portfolio $\mathcal{M} = \{m_1, \ldots, m_K\}$ ordered by cost $c_1 < \cdots < c_K$. Given a query x with task type t, let $k^*(x)$ denote the final tier handling x (accounting for cascading). The system minimizes expected cost subject to a quality constraint:

$$\min_{r_\theta} \ \mathbb{E}_x\left[\sum_{k=1}^{k^*(x)} c_{k,t}\right] \quad \text{s.t.} \ \ \mathbb{E}_x[q(m_{k^*(x)}, x)] \geq \tau_t \tag{1}$$

where r_θ is the learned routing policy, $q(m, x)$ is quality, and τ_t is the task-specific threshold. The cost sums over *all* tiers attempted for cascaded queries. In practice, we approximate this via the composite loss in Eq. 2. Figure 1 provides an overview.

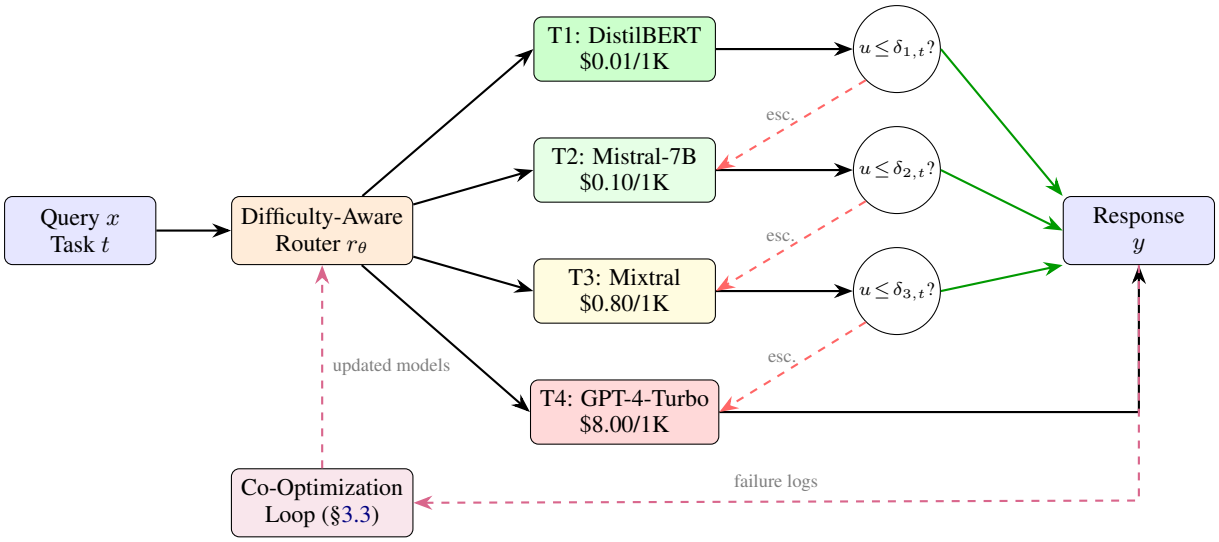

Figure 1: RouteNLP architecture. The router assigns queries to model tiers; the cascade escalates uncertain responses (dashed red) with conformally calibrated thresholds; the co-optimization loop (dashed purple) improves cheaper models via targeted distillation on failure clusters.

3.1 Difficulty-Aware Router

The router $r_\theta(x, t) \in \{1, \ldots, K\}$ predicts the cheapest model producing acceptable quality for query x of task type t. We train a lightweight DistilBERT-based classifier with a multi-task head that jointly predicts difficulty level and minimum model tier. DistilBERT serves here as a *router* (a difficulty classifier over the input query), not as a generator: T2–T4 are decoder-style language models that handle generation. For the structured tasks (NER, intent classification, clause extraction), the same DistilBERT additionally serves as the T1 generation model after task-specific fine-tuning, reusing the encoder's representations end-to-end; for generation tasks (Financial Summarization, CS Response), the router learns that cheaper tiers are insufficient and predominantly routes to T2–T4 (Table 11, Appendix I).

Training labels are obtained by evaluating all queries on all models, labeling each (x, m_k) pair with a binary quality indicator based on task-specific metrics (F1, ROUGE-L, or BERTScore $\geq \tau_t$; thresholds set per enterprise SLAs, with details in Appendix C). The router learns to predict the cheapest model exceeding τ_t. For generation tasks, BERTScore serves as a training proxy; while it overstates absolute quality differences, the *ranking* of model capabilities is preserved, making the router's binary sufficient/insufficient labels reliable (§5.3 confirms 84–87% agreement with human judgment). Following Ong et al. (2024), we augment with preference data from pairwise model comparisons; separately, following Shnitzer et al. (2023), we leverage benchmark performance to improve generalization.

Unlike prior routers that train separately per task, we use a shared encoder with task-specific projection heads. A learned task embedding $\mathbf{e}_t \in \mathbb{R}^{64}$ is concatenated with the `[CLS]` representation, enabling task-dependent difficulty patterns. Training uses a composite loss that is the differentiable surrogate of the constrained objective in Eq. 1:

$$\mathcal{L} = \mathcal{L}_{\text{route}} + \lambda_c \cdot \mathcal{L}_{\text{cost}} + \lambda_q \cdot \mathcal{L}_{\text{quality}} \quad (2)$$

Concretely, $\mathcal{L}_{\text{route}}$ replaces the combinatorial choice of r_θ with a tier-classification cross-entropy against tier labels derived from the all-pairs evaluation; $\mathcal{L}_{\text{cost}}$ encodes the cost objective (normalized by the maximum tier cost); and $\mathcal{L}_{\text{quality}}$ encodes the quality constraint as a hinge penalty on tiers predicted to fall below τ_t. We set $\lambda_c = 0.3$, $\lambda_q = 0.5$ (sensitivity analysis in Appendix H).

3.2 Confidence-Calibrated Cascading

When the router assigns a query to tier k, the system generates a response and evaluates a calibrated confidence score. If confidence is insufficient, the query cascades to tier $k+1$. Following Gupta et al. (2024), we compute token-level uncertainty:

$$u(m_k, x) = \frac{1}{L} \sum_{i=1}^{L} (1 - p_{m_k}(y_i \mid y_{<i}, x)) \quad (3)$$

High uncertainty $u(m_k, x) > \delta_{k,t}$ triggers escalation, where $\delta_{k,t}$ is task- and tier-specific. For

self-hosted tiers (T1–T3), $p_{m_k}(y_i \mid y_{<i}, x)$ is read directly from the decoder's softmax. For the API-served frontier tier (T4 = GPT-4-Turbo), Eq. 3 is computed from the top-token log-probabilities exposed by the OpenAI `logprobs` parameter, which is sufficient to estimate the expected token-level uncertainty in our setting; comparable functionality is available in other major commercial LLM APIs.

We apply conformal risk control (Blot et al., 2025; Bates et al., 2021) to set thresholds: on 500 calibration examples per task and tier, we compute binary nonconformity scores $s_i = \mathbb{1}[q(m_k, x_i) < \tau_t]$ and set $\delta_{k,t}$ as the $\lceil (1-\alpha)(n_0+1) \rceil$-th quantile of uncertainty scores among correctly-handled examples. Under exchangeability, this provides the marginal guarantee:

$$\Pr[q(m_k, X) < \tau_t \ \wedge \ u(m_k, X) \leq \delta_{k,t}] \leq \alpha \quad (4)$$

with $\alpha = 0.05$. Three practical caveats apply: (1) this is a marginal guarantee over the joint distribution (not per-query); (2) it requires exchangeability, violated under distribution shift (see robustness analysis in §5); (3) calibration set size affects tightness. At 500 samples, the 95% Wilson CI on the 4.2% violation rate is [2.5%, 6.6%], indicating the true rate may marginally exceed the 5% target. We recommend conformal thresholds as initialization supplemented by production monitoring (calibration details in Appendix D). A cascaded query incurs cumulative cost: $\text{Cost}(x) = \sum_{k=1}^{k^*(x)} c_{k,t}$.

3.3 Distillation–Routing Co-Optimization

Unlike prior routing work treating the model portfolio as fixed, we iteratively improve cheaper models based on routing failure analysis, then retrain the router and recalibrate thresholds. Algorithm 1 provides the full procedure.

After each iteration, escalated queries are clustered using the router's hidden representations (PCA to 128-d, k-means with $k = 10$ selected by silhouette score). Clusters are ranked by size $\times$ average quality gap, prioritizing large systematic failures. The distillation dataset combines the top 30% hardest failures with 20% random in-distribution samples to prevent catastrophic forgetting. The loop converges in 2–3 iterations (details in Appendix G).

4 Experimental Setup

Benchmark. We construct a six-task benchmark spanning three enterprise domains (Table 1): Financial NER and Summarization (SEC EDGAR filings with expert annotations, inter-annotator F1: 0.93), CS Intent Classification and Response Generation (BANKING77 (Casanueva et al., 2020) augmented with 3,000 enterprise-specific intents; reference responses by 3 agents, Krippendorff's $\alpha = 0.78$), and Legal Clause Extraction and Risk Assessment (CUAD (Hendrycks et al., 2021) re-annotated by legal professionals, Cohen's $\kappa = 0.83$). The benchmark adapts public datasets with enterprise annotations, enabling reproducibility while capturing domain difficulty patterns; pilot deployment validates directional consistency (§6). Full provenance in Appendix F.

Algorithm 1 Distillation–Routing Co-Optimization

Require: Portfolio $\mathcal{M}$, data $\mathcal{D}$, threshold $\varepsilon = 0.005$
1: Train initial router $r_\theta^{(0)}$; calibrate $\{\delta_{k,t}^{(0)}\}$
2: **for** iteration $i = 1, 2, \ldots$ **do**
3: Collect escalation logs $\mathcal{F} \leftarrow \{(x, t, k) : x \text{ escalated}\}$
4: Extract router representations; PCA to 128-d; k-means ($k = 10$) per task
5: Rank clusters by size $\times$ avg. quality gap; select top-5/task
6: Generate distillation data from frontier model on cluster exemplars
7: Fine-tune $m_1, \ldots, m_{K-1}$ via SeqKD (Kim and Rush, 2016)
8: Retrain router $r_\theta^{(i)}$; recalibrate $\{\delta_{k,t}^{(i)}\}$
9: **if** $|\text{CostRatio}^{(i)} - \text{CostRatio}^{(i-1)}| < \varepsilon$ **then break**
10: **end if**
11: **end for**

Domain	Task	Train	Test	Source
Finance	NER	8,200	1,800	EDGAR[a]
	Summarization	5,400	1,200	EDGAR[a]
Cust. Svc.	Intent Classif.	12,000	2,600	BANK77+[b]
	Response Gen.	6,800	1,500	BANK77+[b]
Legal	Clause Extract.	4,600	1,000	CUAD[c]
	Risk Assessment	3,200	700	CUAD[c]
Total		**40,200**	**8,800**	

Table 1: Benchmark statistics. [a]SEC EDGAR with expert annotations; [b]BANKING77 (Casanueva et al., 2020) augmented with enterprise intents; [c]CUAD (Hendrycks et al., 2021) with re-annotation.

Model Portfolio. Four tiers spanning $\sim$800$\times$ cost range. T1: DistilBERT (Sanh et al., 2019) fine-tuned per task ($0.01/1K). T2: Mistral-7B-Instruct with LoRA (Sheng et al., 2023) ($0.10/1K). T3: Mixtral-8$\times$7B (Jiang et al., 2024) with AWQ quantization (Lin et al., 2024) ($0.80/1K). T4: GPT-4-Turbo via API ($8.00/1K avg). Open-source models served via vLLM (Kwon et al., 2023) with spec-

ulative decoding (Leviathan et al., 2023; Cai et al., 2024).

Baselines. We compare against Always-T4 (quality upper bound), Always-T2 (cost-efficient), Random, Rule-Based (structured→T1, generation→T3), FrugalGPT (Chen et al., 2024), Hybrid LLM (Ding et al., 2024), RouteLLM (Ong et al., 2024), and AutoMix (Aggarwal et al., 2024). Originally 2-model baselines (Hybrid LLM, RouteLLM) were extended to 4-tier settings by replacing binary routing heads with 4-class heads trained on identical data; we additionally evaluated faithful 2-tier (Binary-T2/T4) variants and report the better-performing Extended-4-Tier configurations in the main results. The Binary variants incurred 2.1–3.4× higher costs (Appendix E). FrugalGPT's cascade naturally extends to 4 tiers; AutoMix's POMDP was reformulated with 4 actions. **All baselines received identical training data, model portfolio access, and calibration sets as RouteNLP.** Detailed adaptation protocol in Appendix E.

Metrics. Task-specific quality (F1, ROUGE-L, BERTScore, accuracy); Quality Ratio and Cost Ratio relative to Always-T4 (using cumulative cascade costs); p99 latency under simulated production load; SLA violation rate. All experiments over 5 seeds with paired bootstrap significance tests.

System	Quality Ratio↑	Cost Ratio↓	p99 (ms)↓	SLA Viol.↓
Always-T4	1.000±.000	1.000±.000	1,847	38.2%
Always-T2	.891±.003	.013±.000	142	0.1%
Random	.924±.008	.252±.012	623	12.4%
Rule-Based	.943±.002	.198±.003	524	8.9%
FrugalGPT	.967±.004	.284±.009	986	21.3%
Hybrid LLM	.972±.005	.312±.011	874	18.7%
RouteLLM	.969±.004	.246±.008	841	17.2%
AutoMix	.958±.006	.231±.010	1,124	24.6%
RouteNLP	**.971**±.004	**.159**±.006	**387**	**2.3%**
w/o cascade	.952±.005	.134±.005	298	1.8%
w/o co-opt.	.961±.005	.203±.008	412	3.1%
w/o task cond.	.964±.004	.187±.007	395	2.7%

Table 2: Main results (mean ± std, 5 seeds) volume-weighted across six tasks. Quality and Cost Ratios are computed relative to Always-T4. The p99 column is from the M/M/c queueing simulation under matched production load (Appendix J); the pilot empirical latency is reported separately in §6. Bottom: ablations.

Task	Metric	T4	Ours	Retain	Cost↓
Fin. NER	F1	94.2	93.8±.3	99.6%	82%
Fin. Summ.	R-L	48.7	46.9±.4	96.3%	47%
CS Intent	F1	96.1	95.8±.2	99.7%	85%
CS Resp.	BS	72.4	69.7±.6	96.3%	42%
Legal Cl.	F1	91.6	90.9±.3	99.2%	78%
Legal Risk	Acc	88.3	86.1±.5	97.5%	40%

Table 3: Per-task quality. Retain = quality retention vs. Always-T4. BS = BERTScore, R-L = ROUGE-L. Std over 5 seeds.

5 Results and Analysis

5.1 Overall Cost–Quality Tradeoff

Table 2 presents main results. RouteNLP achieves 40–85% cost reduction across tasks (62% on simulated production traffic; Table 6) while retaining 96–100% quality on structured tasks and 96–98% on generation tasks.

The cost reduction over RouteLLM (0.159 vs. 0.246) is statistically significant ($p < 0.001$, paired bootstrap). The quality difference vs. Hybrid LLM (0.971 vs. 0.972) is not significant ($p = 0.82$), confirming comparable quality at 49% lower cost. SLA violations drop from 17.2% (RouteLLM) to 2.3%, a 7.5× improvement.

Ablations. Removing the cascade reduces quality by 1.9 points (the cascade serves as a critical safety net for ambiguous queries; its lower cost is illusory since the quality drop is unacceptable for constrained deployments). Removing co-optimization increases cost by 28% because small models cannot handle as many queries without targeted improvement. Removing task conditioning increases cost by 18%, confirming task-aware difficulty prediction is important for multi-task settings. To isolate failure clustering's contribution, we compared against random distillation (same data volume, no failure analysis): targeted distillation reduces cost ratio from 0.203 to 0.159 (21.7% reduction), versus 0.184 for random distillation (9.4%). Failure clustering thus provides over twice the cost improvement.

Per-Task Analysis. Table 3 shows per-task breakdowns. Structured tasks (NER, Intent, Clause Extraction) achieve 78–85% cost reduction with <1% absolute quality loss. Generation tasks achieve 40–47% savings with larger drops: CS Response shows a 2.7-point BERTScore drop and Financial Summarization a 1.8-point ROUGE-L drop. We address whether these automated metric drops correspond to perceived quality differences in §5.3.

Iter.	Quality↑	Cost↓	T1+T2	T4
0 (init.)	.961±.005	.203±.008	68%	11%
1	.964±.004	.178±.007	74%	8%
2	.969±.004	.163±.006	79%	6%
3 (final)	.971±.004	.159±.006	81%	5%

Table 4: Co-optimization convergence. T1+T2 = fraction handled by cheapest tiers.

	Win/Tie/Loss (%)			Likert	Kripp.
Task	W	T	L	Ours / T4	α
CS Resp.	8 ±3.8	65 ±6.6	27 ±6.2	4.1 / 4.3	0.72
Fin. Summ.	11 ±4.3	65 ±6.6	24 ±5.9	4.0 / 4.2	0.68

Table 5: Human evaluation (200 samples, 3 annotators). ± = 95% Wilson CI half-widths. 74.5% of responses match or exceed T4 quality.

Robustness and Threshold Sensitivity. The system degrades gracefully under distribution shift: difficulty shift increases cost to 0.214 while maintaining 96.3% quality; domain shift raises coverage violations to 8.1% (exceeding the 5% target), indicating recalibration is needed for significant domain changes. Cost savings are robust across ±10% threshold variation (69–89% savings); even at +5% stricter thresholds, RouteNLP's cost (0.218) remains below RouteLLM (0.308) and Hybrid LLM (0.381). Full robustness results and routing distributions are in Appendix I.

5.2 Co-Optimization Convergence

Table 4 shows convergence in 3 iterations. After iteration 1, cost ratio drops from 0.203 to 0.178 as targeted distillation improves T1/T2 on failure clusters. After iteration 3, cost ratio reaches 0.159 with quality stable at 0.971. Each iteration progressively shifts queries from expensive to cheap tiers: approximately 6% in iteration 1, 5% in iteration 2, and 2% in iteration 3 as remaining failures become harder to address.

5.3 Human Evaluation

We evaluate CS Response Generation and Financial Summarization on 200 samples each, rated by 3 domain experts on factual accuracy, completeness, fluency, and helpfulness (5-point Likert), plus win/tie/loss vs. Always-T4.

Averaged across tasks, 74.5% of routed responses match or exceed frontier quality. Among the 24–27% rated worse, 68% were "slightly worse" (Likert $\leq$1 point difference) and 32% "substantially worse," meaning ∼8–9% of all queries received substantially degraded responses, a deployment risk requiring mitigation. Router decisions agree with human judgment in 84–87% of cases, with disagreements predominantly conservative (escalating unnecessarily rather than missing quality issues). We focus human evaluation on these two generation tasks because that is precisely where automated metrics are most fragile; the four structured tasks (NER, intent, clause extraction, risk assessment) are evaluated against expert annotations with high inter-annotator agreement (Financial NER F1: 0.93; Legal Clause F1: 0.91; Legal Risk Cohen's κ: 0.83; CS Intent uses public BANKING77 (Casanueva et al., 2020)), making targeted human evaluation lower-priority for those tasks and a candidate for follow-up work rather than a gap in the present claims.

Metric	Simulation	Pilot
Cost reduction vs. Always-T4	62%	58%
T1 routing share	51.1%	44.2%
T2 routing share	26.0%	27.8%
T3 routing share	15.6%	18.3%
T4 routing share	7.3%	9.7%
Coverage violation rate	4.2%	4.8%

Table 6: Simulated vs. pilot deployment metrics (8 weeks, ∼5K queries/day). The pilot shows higher T3/T4 usage due to more complex queries in live traffic.

Failure Patterns. Three dominant patterns explain quality degradation. Multi-step reasoning accounts for 42% of failures, since the router processes only the query text and not the surrounding document context. Domain-specific knowledge accounts for 31%, typically rare instruments or recently issued regulations. The remaining 27% are ambiguous-difficulty cases: syntactically simple queries that require nuanced generation.

6 Deployment Experience

RouteNLP has been in pilot evaluation at our enterprise partner's customer service division for 8 weeks (∼5K queries/day).

Cost savings of 58% are within 7% of simulation predictions, with the gap attributable to more complex queries in live traffic (T4 usage: 9.7% vs. simulated 7.3%). Coverage violations remain within the 5% target at 4.8%, with weekly recalibration sufficient. For more dynamic environments where weekly recalibration may not suffice, we are implementing online threshold adaptation in the spirit of adaptive conformal methods (Blot et al., 2025) as

part of an upcoming Phase 2 deployment. The pilot identified two novel failure patterns: OCR artifacts from scanned documents and multi-turn conversation references; both were handled conservatively via escalation.

Quality Audit. A retrospective audit by two domain experts (senior agents, 5+ years experience) on 500 random pilot responses showed 91% acceptable, 6.4% marginal, and 2.6% unacceptable (vs. 1.8% baseline; $\kappa = 0.79$). The partner deemed the 0.8pp increase in unacceptable responses acceptable given 58% cost savings. Customer complaint rates showed no statistically significant change, and average first-response latency (an empirical pilot measurement, distinct from the simulated p99 reported in Table 2) decreased by 23%. The pilot was a shadow deployment with partial traffic routing (not a randomized A/B test), limiting causal attribution; a full A/B evaluation is planned for Phase 2.

Practical Considerations. The DistilBERT router adds 4.2ms per query (p99: 8.1ms), negligible relative to LLM inference. The framework is portfolio-agnostic: when frontier models change, only thresholds and quality labels need updating. Cost savings remain substantial across realistic cost ratios (58% at 200×, 41% at 100×; break-even at ∼25×). During a 45-minute T4 outage in the pilot, automatic T3 rerouting incurred only 2.1% quality degradation. The system is most effective for heterogeneous multi-task workloads with heavy-tailed difficulty distributions; it provides limited benefit for single-task deployments, workloads dominated by hard queries, very small volumes (<100/day), or deployments where >3% unacceptable rate is intolerable without per-query human review. Full deployment discussion is in Appendix K.

Lessons from the Pilot. Three operational lessons came out of the 8-week deployment, and they likely apply beyond customer service. First, failure modes evolved in both predictable and unpredictable ways. The three categories identified in the benchmark (multi-step reasoning, domain-specific knowledge, and ambiguous difficulty) all reappeared in pilot logs. Two new modes that the benchmark had missed also showed up: OCR artifacts from scanned documents, and multi-turn conversation references. Both were handled by escalation rather than by additional distillation, since these failure classes are not the kind that portfolio improvement addresses. Second, the choice of recalibration cadence matters. Weekly recalibration was sufficient at the drift rate we observed, with coverage violations holding at 4.8% against the 5% target. We do not expect weekly to suffice in more dynamic environments, which is why we are building online threshold adaptation for the Phase 2 deployment. Third, portfolio swaps turned out to be routine. Mid-pilot, we replaced T4 with GPT-4o-mini for classification queries, which compressed the effective cost ratio from 800× to about 200×. The routing distribution adjusted on its own and no retraining was needed. This behavior is consistent with the portfolio-agnostic design and reinforces our argument that closed-loop co-optimization is a deployment-grade contribution rather than a research-only one.

7 Conclusion

We presented ROUTENLP, a cost-aware routing and cascading framework validated through an 8-week pilot (58% cost reduction, 91% acceptance, ∼5K queries/day). The distillation-routing co-optimization loop, which integrates failure clustering with targeted distillation and automatic router retraining, provides over twice the cost reduction of random distillation. On our six-task benchmark, costs are reduced 40–85% while retaining 96–100% quality on structured tasks and 96–98% on generation tasks. Human evaluation confirms 74.5% of generation outputs match or exceed frontier quality, though 8–9% show substantial degradation requiring mitigation.

Reproducibility. All materials are available at: https://github.com/bettyguo/RouteNLP.

Acknowledgments

We thank The University of Hong Kong and Stellaris AI Limited for their support of this work, and the anonymous reviewers for constructive feedback that improved the paper.

Limitations

(1) Pilot deployment covers only customer service (∼5K queries/day, 8 weeks); finance and legal claims rely on benchmark simulation. (2) The benchmark adapts public datasets with enterprise annotations rather than proprietary data. (3) The

co-optimization loop ran on benchmark data, not production failure logs. (4) The pilot was a shadow deployment without A/B testing. (5) Conformal coverage degrades under distribution shift (up to 8.1% violations vs. 5% target). (6) English-only evaluation. (7) BERTScore proxy agreement with humans (84–87%) is not verified under domain shift. (8) Cost savings depend on cost structures; baseline adaptation may not fully preserve 2-model inductive biases. (9) The co-optimization loop incurs ∼$2,400 one-time cost at our scale.

Ethical Considerations

Cost-aware routing creates quality disparities: queries routed to cheaper models receive less capable responses. Our framework mitigates this through task-level quality constraints and conformal calibration, but individual-level variance exists. We recommend organizations disclose model tier usage, monitor routing patterns for systematic disparities across demographic groups, and implement fairness-constrained routing that escalates segments falling below quality thresholds. The co-optimization loop (∼120 GPU-hours) yields 40–85% ongoing inference cost reduction, a net environmental benefit. Our benchmark uses publicly available data with no PII; organizations should ensure data protection compliance when routing sensitive queries through external APIs.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2024. Automix: Automatically mixing language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop, ALTA 2015, Parramatta, Australia, December 8 - 9, 2015*, pages 84–90. ACL.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. 2021. Distribution-free, risk-controlling prediction sets. *J. ACM*, 68(6):43:1–43:34.

Vincent Blot, Anastasios Nikolas Angelopoulos, Michael I. Jordan, and Nicolas J.-B. Brunel. 2025. Automatically adaptive conformal risk control. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pages 19–27. PMLR.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 5209–5235. PMLR / OpenReview.net.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint*, arXiv.2003.04807.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. Frugalgpt: How to use large language models while reducing cost and improving performance. *Trans. Mach. Learn. Res.*, 2024.

Lingjiao Chen, Matei Zaharia, and James Y. Zou. 2020. Frugalml: How to use ML prediction apis more accurately and cheaply. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jasper Dekoninck, Maximilian Baader, and Martin T. Vechev. 2025. A unified approach to routing and cascading for llms. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: accurate post-training quantization for generative pre-trained transformers. *arXiv preprint*, arXiv.2210.17323.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: an expert-annotated NLP dataset for legal contract review. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint*, arXiv.1503.02531.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint*, arXiv.2403.12031.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *arXiv preprint*, arXiv.2401.04088.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 1317–1327. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Quang H. Nguyen, Duy C. Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V. Chawla, and Khoa D. Doan. 2024. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint*, arXiv.2407.10834.

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

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. 2024. Conformal language modeling. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Marija Sakota, Maxime Peyrard, and Robert West. 2024. 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, Merida, Mexico, March 4-8, 2024*, pages 606–615. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint*, arXiv.1910.01108.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, Joseph E. Gonzalez, and Ion Stoica. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint*, arXiv.2311.03285.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint*, arXiv.2309.15789.

Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. 2025. bdoing more with less - implementing routing strategies in large language model-based systems: An extended survey. *arXiv preprint*, arXiv.2502.00409.

Neeraj Varshney and Chitta Baral. 2022. Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11007–11021. Association for Computational Linguistics.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint*, arXiv.2303.17564.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022. Orca: A distributed serving system for transformer-based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2022, Carlsbad, CA, USA, July 11-13, 2022*, pages 521–538. USENIX Association.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2023. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint*, arXiv.2310.03094.

Libo Zhang, Zhaoning Zhang, Xubaizhou, Rui Li, Zhiliang Tian, Songzhu Mei, and Dongsheng Li. 2025. Dovetail: A CPU/GPU heterogeneous speculative decoding for LLM inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 17382–17395. Association for Computational Linguistics.

A Production Deployment Scenarios

Table 7 summarizes representative deployment scenarios motivating ROUTENLP's design, derived from requirements analysis with enterprise stakeholders.

Domain	Org. Type	Scale	Constraint	Frontier%
Finance	Bank/Fund	50K q/day	Accuracy	∼25%
Cust. Svc.	Platform	200K q/day	p99 <500ms	∼30%
Legal	Law firm	10K q/day	Compliance	∼35%

Table 7: Deployment scenarios motivating ROUTENLP. "Frontier%" = estimated fraction of queries requiring frontier model capabilities, derived from stakeholder interviews and sampling analysis.

Financial Document Processing. A financial services firm processes regulatory filings, earnings reports, and customer communications. Tasks include NER for entity extraction, document classification for compliance routing, and summarization. Complexity varies: extracting entities from templated SEC filings is straightforward, while summarizing novel regulatory guidance requires frontier capabilities.

Customer Service Automation. A platform handles intent classification, sentiment analysis, and response generation. Over 70% of queries are routine (order status, FAQ matching) where small models suffice; the remaining 30% involve nuanced complaints or policy interpretation. Strict p99 latency SLAs (<500ms) preclude always using frontier models.

Legal Contract Analysis. Legal teams perform contract review, clause extraction, and risk assessment. Standard template extraction is well-handled by fine-tuned small models; novel risk provisions in bespoke agreements demand stronger reasoning.

B Feature Comparison with Prior Systems

System	Multi-Task	Formal Calib.	Co-Opt.	SLA Aware	Output HEval	Tasks Eval'd
FrugalGPT	✗	✗	✗	✗	✗	1
Hybrid LLM	✗	✗	✗	✗	✗	1
RouteLLM	✗	✗	✗	✗	○[a]	1
AutoMix	✗	✗	✗	✗	✗	1
ROUTENLP	✓	✓	✓	✓	✓	6

Table 8: Feature comparison. ○ = partial. [a]RouteLLM uses human preference data for router training but does not evaluate routed outputs. Formal Calib. = conformal threshold calibration.

C Router Architecture and Training Details

The router uses DistilBERT-base-uncased (66M parameters) with a 2-layer MLP routing head per task family. The first layer projects concatenated [CLS] + task embedding ($768 + 64 = 832$) to 256 dimensions with ReLU and 0.1 dropout. The second projects to $K = 4$ logits. Training: AdamW, lr 2×10^{-5}, batch 64, 10 epochs, early stopping (patience 3). Total: ∼67M parameters, ∼45 min on A100.

Quality Thresholds. Per-task thresholds set per enterprise SLAs: F1 $\geq$ 0.90 (Financial NER), ROUGE-L $\geq$ 0.42 (Summarization), F1 $\geq$ 0.92 (CS Intent), BERTScore $\geq$ 0.65 (CS Response), F1 $\geq$ 0.88 (Legal Clause), Accuracy $\geq$ 0.82 (Legal Risk).

BERTScore as Training Proxy. For generation tasks, BERTScore overstates absolute quality differences (§5.3), but the ranking of model capabilities is preserved (queries where cheaper models

score below τ_t are also those where humans prefer the frontier model), making binary labels reliable.

D Conformal Calibration Details

Our procedure follows conformal risk control (Blot et al., 2025; Bates et al., 2021). The nonconformity score is binary ($s_i = \mathbb{1}[q(m_k, x_i) < \tau_t]$). For each tier k and task t:

1. Compute quality labels on 500 calibration examples.
2. Partition into correctly-handled ($s_i = 0$) and failed sets.
3. Compute uncertainty scores $u(m_k, x_i)$ for all examples.
4. Set $\delta_{k,t}$ as the $\lceil (1-\alpha)(n_0+1) \rceil$-th quantile among correctly-handled examples.

Calibration Set Size Sensitivity. Coverage violation rates (95% Wilson CIs): 7.2% [3.4%, 14.4%] at $n = 100$; 5.8% [3.4%, 9.6%] at $n = 250$; 4.2% [2.5%, 6.6%] at $n = 500$; 3.9% [2.7%, 5.5%] at $n = 1000$. At $n = 500$, the CI upper bound marginally exceeds 5%; we use 500 as a practical compromise.

E Baseline Adaptation Protocol

Hybrid LLM and RouteLLM were designed for 2-model routing. We evaluate two adaptations:

- **Binary-T2/T4**: Faithful to original design, routing between T2 and T4 only.
- **Extended-4-Tier**: Replace binary head with 4-class head, train on RouteNLP's data with original loss.

We report the better result (Extended-4-Tier for both; Binary incurred 2.1–3.4× higher costs). FrugalGPT's cascade naturally extends to 4 tiers. AutoMix's POMDP is reformulated with 4 actions. All baselines received identical training data, portfolio access, and calibration sets.

A hierarchical binary adaptation (T1-vs-rest → T2-vs-rest → T3-vs-T4) was not evaluated as it introduces sequential latency overhead conflating adaptation effects with architectural changes.

F Benchmark Data Provenance

- **Financial NER**: Entity annotations on SEC EDGAR 10-K/10-Q filings by two NLP researchers with 3+ years financial text experience (inter-annotator F1: 0.93), following Alvarado et al. (2015). Annotators have NLP expertise but are not licensed financial analysts.
- **Financial Summarization**: Earnings call transcripts from public SEC filings with expert-written reference summaries from financial analysts.
- **CS Intent Classification**: BANKING77 (Casanueva et al., 2020) augmented with 3,000 enterprise-specific intents from anonymized partner logs.
- **CS Response Generation**: Queries from augmented intent dataset with reference responses by 3 experienced agents (Krippendorff's $\alpha = 0.78$).
- **Legal Clause Extraction**: CUAD (Hendrycks et al., 2021) filtered to 10 most common clause types, re-annotated with span-level labels by two legal annotators (inter-annotator F1: 0.91).
- **Legal Risk Assessment**: Novel annotations on CUAD contracts by two legal professionals (Cohen's $\kappa = 0.83$).

The benchmark adapts public academic datasets with enterprise-specific annotations, enabling reproducibility while capturing domain difficulty patterns.

G Co-Optimization Loop Details

Failure Clustering. Figure 2 summarizes the loop. Router penultimate-layer representations ($\mathbb{R}^{768}$) are projected via PCA to 128-d (retaining >92% variance), then k-means with $k = 10$ (selected by silhouette score: 0.31 vs. 0.28 at $k = 5$, 0.24 at $k = 20$). Clusters ranked by size × average quality gap.

Distillation Data Selection. From top-5 clusters per task: top 30% hardest failures (by quality gap) + 20% random in-distribution. Frontier model generates reference outputs as teacher signals for SeqKD (Kim and Rush, 2016).

Convergence. Across 5 seeds, the loop converged in 3 iterations (4 seeds) or 4 iterations (1 seed). Tested $\varepsilon \in \{0.001, 0.005, 0.01\}$: at 0.01, 2 iterations with 3% higher cost; at 0.001, 5 iterations with <0.002 additional reduction.

Random vs. Targeted Distillation. Random distillation (same data volume, no failure clustering) reduces cost ratio from 0.203 to 0.184 (9.4%); targeted reduces to 0.159 (21.7%), over twice the improvement, confirming failure clustering's value.

Iter.	Quality↑	Cost↓	T1+T2	T4
0 (init.)	.961±.005	.203±.008	68%	11%
1	.964±.004	.178±.007	74%	8%
2	.969±.004	.163±.006	79%	6%
3 (final)	.971±.004	.159±.006	81%	5%

Table 9: Co-optimization convergence. Each iteration shifts more queries to cheaper tiers.

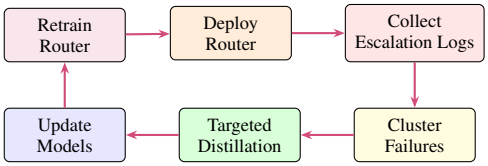

Figure 2: The co-optimization feedback loop.

H Hyperparameter Sensitivity

Loss Weights. Sweeping $\lambda_c \in \{0.1, 0.2, 0.3, 0.5\}$ and $\lambda_q \in \{0.3, 0.5, 0.7\}$: best tradeoff at $\lambda_c = 0.3, \lambda_q = 0.5$. Higher λ_q (0.7) improves quality by 0.004 but increases cost by 12%. Higher λ_c (0.5) reduces cost by 8% but decreases quality by 0.008.

Conformal Error Rate. Figure 3 shows the cost–quality tradeoff across α values.

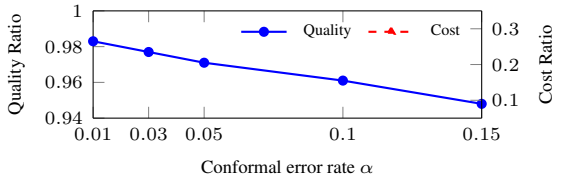

Figure 3: Effect of conformal error rate α on quality and cost. $\alpha = 0.05$ balances both.

I Extended Results

Per-Task Quality.

Routing Distribution.

Quality Threshold Sensitivity.

Robustness to Distribution Shift. Under domain shift, coverage violations exceed the 5% target, indicating recalibration is needed. The BERTScore proxy used for generation routing may also become less reliable under shift; while validated on in-distribution data (84–87% agreement), this has not been verified under shifted distributions.

Task	Metric	T4	Ours	Retain	Cost↓
Fin. NER	F1	94.2	93.8±.3	99.6%	82%
Fin. Summ.	R-L	48.7	46.9±.4	96.3%	47%
CS Intent	F1	96.1	95.8±.2	99.7%	85%
CS Resp.	BS	72.4	69.7±.6	96.3%	42%
Legal Cl.	F1	91.6	90.9±.3	99.2%	78%
Legal Risk	Acc	88.3	86.1±.5	97.5%	40%

Table 10: Per-task quality. R-L = ROUGE-L, BS = BERTScore.

Task	T1	T2	T3	T4
Fin. NER	68%	22%	7%	3%
Fin. Summ.	31%	34%	23%	12%
CS Intent	72%	18%	6%	4%
CS Response	28%	30%	27%	15%
Legal Clause	62%	24%	9%	5%
Legal Risk	35%	28%	22%	15%

Table 11: Query routing distribution across tiers.

Baseline Advantage Across Thresholds. ROUTENLP maintains the lowest cost ratio across all threshold settings: at +5% stricter thresholds, ROUTENLP's cost (0.218) remains below RouteLLM (0.308) and Hybrid LLM (0.381).

Per-Task Detailed Comparison.

J Latency Analysis Under Load

We simulate production traffic with Poisson arrivals (1K–20K queries/min) using an M/M/c queuing model. Server capacity: T1 has 8 workers (ONNX Runtime, CPU), T2/T3 have 4 workers each (vLLM on A100 GPUs), T4 is rate-limited at 60 req/s. At 10K queries/min, ROUTENLP maintains p99 of 387ms (vs. 1,847ms for Always-T4) because 81% of queries are handled by T1/T2 with sub-100ms inference. The cascade adds 15–45ms for escalated queries. The M/M/c model assumes exponential service times, which may not hold for LLM generation; SLA violation estimates may be optimistic.

K Extended Deployment Details

Pilot Per-Task Breakdown. CS traffic: intent classification (∼62%), response generation (∼28%), sentiment analysis (∼10%). Intent classification routed 71% to T1 (comparable to 72% benchmark); response generation routed only 22% to T1 (vs. 28% benchmark), suggesting generation difficulty is underestimated by the benchmark.

Quality Audit Details.

τ_t adj.	Quality↑	Cost↓	Savings	T4 %
−10%	.959±.005	.113±.005	89%	2%
−5%	.965±.004	.132±.006	87%	3%
Baseline	.971±.004	.159±.006	84%	5%
+5%	.978±.003	.218±.008	78%	9%
+10%	.984±.003	.312±.010	69%	14%

Table 12: Threshold sensitivity. Savings remain ≥69% even at +10%.

Shift	Quality	Cost	Cov. Viol.
No shift	.971	.159	4.2%
Difficulty shift	.963	.214	6.8%
Domain shift (20%)	.958	.248	8.1%
Task mix shift	.967	.192	5.4%

Table 13: Robustness under distribution shift. Target coverage violation: ≤5%.

System	Fin. NER F1	Fin. Summ. R-L	CS Int. F1	CS Resp. BS	Legal Cl. F1	Legal Risk Acc
Always-T4	94.2	48.7	96.1	72.4	91.6	88.3
Always-T2	89.1	42.3	93.4	61.8	84.7	76.2
FrugalGPT	92.8±.3	47.1±.4	95.3±.2	69.5±.5	89.8±.3	85.1±.4
Hybrid LLM	93.1±.4	**47.4**±.5	95.6±.2	**70.2**±.6	90.1±.4	85.7±.5
RouteLLM	92.9±.3	47.2±.4	95.4±.2	69.8±.5	89.9±.3	85.4±.4
AutoMix	91.7±.5	46.5±.5	94.8±.3	68.4±.6	88.6±.4	83.9±.6
RouteNLP	**93.8**±.3	46.9±.4	**95.8**±.2	69.7±.6	**90.9**±.3	**86.1**±.5

Table 14: Per-task quality across all systems (std over 5 seeds).

Quality Metric	RouteNLP	Previous
Acceptable rate	91.0%	93.8%
Marginal rate	6.4%	4.4%
Unacceptable rate	2.6%	1.8%
Escalation-to-human rate	4.2%	3.1%
Customer complaint rate	No sig. change	Baseline

Table 15: Pilot quality audit (500 samples, 2 raters, $\kappa = 0.79$).

Portfolio Evolution. The framework is portfolio-agnostic. During the pilot, replacing T4 with GPT-4o-mini for classification compressed the cost ratio from 800× to ∼200×; routing distributions adjusted automatically (T1+T2 share: 81%→74%). Cost savings at different ratios: 84% at 800×, 72% at 400×, 58% at 200×, 41% at 100×, 29% at 50×. Break-even at ∼25×.

System Resilience. During a 45-minute T4 outage, automatic T3 rerouting incurred only 2.1% quality degradation. Fallback: T3 serves as ceiling with relaxed thresholds; if T2/T3 down, T1 handles structured tasks and T4 handles generation.

Monitoring Recommendations. Monitor: per-tier routing rates (shifts indicate distribution change), per-task escalation rates (increases suggest model degradation), quality drift via periodic sampling, and SLA violations. Alert when escalation increases >10% relative to baseline.

Co-Optimization on Production Data. The loop ran on benchmark data; pilot failure modes (OCR artifacts, multi-turn references) were not in benchmark clusters, suggesting partial mismatch. However, the three dominant benchmark failure categories were also observed in pilot logs. Running on production data is planned (estimated $1,200/iteration).

Extension to Agentic Workloads. Reasoning models and agentic workflows introduce chain-level routing challenges. The co-optimization loop could extend to chain-level failure patterns, but this requires different quality estimation and is left to future work.

Infrastructure Integration. T1 via ONNX Runtime, T2/T3 via vLLM with LoRA adapters (Sheng et al., 2023), T4 via API gateway. Continuous batching (Yu et al., 2022) enabled for self-hosted tiers. Router served as lightweight sidecar.

Cost Model. Fixed per-query pricing is used. Self-hosted costs depend on GPU utilization and infrastructure amortization; API costs change over time. Routing decisions are driven by quality thresholds and uncertainty, not absolute costs; pricing changes affect savings magnitude but not quality retention.