

A Systematic Survey of Large Language Model Routing

Jiangeng Chang¹ Xingyu Wu^{2*} Kay Chen Tan² Mengling Feng^{1,3}

¹Saw Swee Hock School of Public Health, National University of Singapore

²Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University

³Institute of Data Science, National University of Singapore

Abstract

Available LLMs differ substantially in cost, latency, and capability. Their strengths vary across queries, so the natural question becomes: which model for which query? This is the LLM routing problem. Routing is already used in production, and the academic literature has grown across cascading, bandit, generative, and benchmark threads, each with its own vocabulary, each largely ignoring the others. We argue the field is best read along a single axis: the *algorithmic operation* the router performs. We organise the recurring operations under four labels. Predictive routers score candidates before their answers are generated. Adaptive routers try a model, observe the outcome, and adjust. Generative routers let an LLM itself reason about the decision. Infrastructure work studies routing rather than performs it, via benchmarks and adversarial analysis. This survey formalises the routing decision under this four-paradigm lens, traces the methodological development of the field, compares assumptions and tradeoffs across paradigms, and identifies open research directions revealed by the taxonomy. A continuously updated reading list lives at <https://github.com/wuxingyu-ai/LLM-Routing>.

1 Introduction

Open and proprietary LLMs vary substantially in price, latency, and per-query capability. Llama 3 (Grattafiori et al., 2024), Mistral, Qwen, and other open families broaden the available design space. Because model strengths vary by task, a deployment can waste resources when it sends every query to the same high-cost model. This persistent heterogeneity motivates routing.

An LLM router treats model choice as a per-query decision. Given query q and pool $\mathcal{M} = \{m_1, \dots, m_K\}$, it selects the model that balances expected quality $Q(q, m)$ against inference cost

$C(q, m)$. This allows a deployment to approach the achievable quality and cost frontier instead of remaining at the single operating point imposed by a static rule. Such routing is already used in production. Recent robustness studies evaluate commercial routers such as OpenRouter and GPT 5 Auto (Shafran et al., 2025; Kassem et al., 2026), so the problem is no longer hypothetical.

At a deeper level, LLM routing is the classical *algorithm-selection* problem of Rice (1976) ported to foundation-model candidates (Zhou and Ye, 2025; Wu et al., 2024, 2025b). Four things make the LLM setting different from Rice’s. Costs differ by orders of magnitude, so cost-blind selection breaks down. Per-query evaluation of every candidate is prohibitive, so a pre inference router has to commit before seeing alternative outputs. LLM strengths are emergent and resist a clean feature list. And the candidate pool itself churns: yesterday’s frontier model becomes tomorrow’s mid-tier, so routing rules need to generalise zero-shot to LLMs the router has never seen.

In response, the LLM routing literature has expanded along multiple methodological axes: offline scorers over (q, m) features picking the argmax (Ong et al., 2025; Feng et al., 2025b; Zhuang et al., 2025; Ding et al., 2024; Tsiourvas et al., 2026); cheap-then-escalate cascades (Chen et al., 2023; Aggarwal et al., 2024); bandit and online-learning policies (Wei et al., 2025; Li, 2025; Sikeridis et al., 2025); LLM-as-router architectures that produce routing decisions through generation (Shen et al., 2023; Zhang et al., 2026a; Tran et al., 2025; Zheng et al., 2025); and a parallel infrastructure thread that studies routing rather than performing it (benchmarks, Hu et al., 2024; Lu et al., 2026; Huang et al., 2025b; attacks and defences, Shafran et al., 2025; Tang et al., 2026a; Kassem et al., 2026).

The resulting literature is difficult to compare. Two related surveys provide useful starting points.

* Corresponding author.

Varangot-Reille et al. (2025) treat routing as one item on a broader resource-optimisation menu, while Moslem and Kelleher (2026) zoom in on dynamic cascading. Slicing by supervision signal or by architecture works unevenly: many methods inherit conflicting labels, and the real differentiator; what the router *does* at decision time, gets buried. Our claim is that one axis cleanly separates the core literature. Read by *algorithmic operation*, a core routing method has one of four dominant moves. A router predicts utility offline and takes the argmax (Predictive); or tries a model, observes the result, and adjusts (Adaptive); or hands the decision to an LLM that reasons about it (Generative); or it does not route at all but studies routing from the outside (Infrastructure). The four labels identify the dominant operation. Hybrid and boundary cases receive an explicit secondary or adjacent scope note. Together, the taxonomy and these scope labels provide a consistent account of the representative literature reviewed here and put each paradigm’s trade-offs in dialogue with the others. They do not imply that the inventory is exhaustive or that future systems must fit exclusively within one paradigm. Figure 1 sketches how a user query traverses the routing pipeline under each of the three operating paradigms; Infrastructure is shown separately in Figure 2 since it studies routing rather than performing it.

Our contributions are as follows.

1. **A taxonomy of four paradigms** shown in Figure 2 and applied consistently to the representative literature reviewed in this survey. The labels record the dominant operation, while hybrid and boundary cases retain secondary scope labels. The taxonomy is an organising lens rather than a closed enumeration of possible routing designs.
2. **A paradigm-by-paradigm exposition** (§3) of representative methods within each paradigm, with a master comparison table classifying the surveyed works (Appendix Tables 2 and 3). Selection criteria, search strategy, and borderline-case judgments are documented in Appendix §A.
3. **A trend analysis** (§4) identifying four empirical trends, including a deployment gap between academic (predominantly GPU-bound) and industrial (predominantly API-callable) routing.
4. **Per-paradigm open problems and cross-paradigm frontiers**, including joint routing with query reformulation, certified defences against routing attacks, and routing beyond LLM candidates.

2 Preliminaries

2.1 The Multi-LLM Routing Problem

Let $\mathcal{Q}$ denote a distribution over user queries and $\mathcal{M} = \{m_1, \dots, m_K\}$ a pool of K candidate models. Each model m_k has a quality function $Q(q, m_k) \in [0, 1]$ over query $q \sim \mathcal{Q}$ and a cost function $C(q, m_k) \in \mathbb{R}_{\geq 0}$ (in dollars or tokens). A router is a function $R : \mathcal{Q} \rightarrow \mathcal{M}$ that maps each query to a model. The standard routing objective is:

$$\max_R \mathbb{E}_{q \sim \mathcal{Q}} [Q(q, R(q)) - \lambda \cdot C(q, R(q))], \quad (1)$$

with $\lambda \geq 0$ trading quality for cost. Variants include hard-budget constraints (Huang et al., 2025a), multi-turn settings (Zhang et al., 2026e), and personalized objectives (Wei et al., 2025).

Throughout the survey we default to *LLM-as-candidate* routing: each m_k is an independently trained LLM and q is a user query. Related selection problems can reuse one of the four operations when $\mathcal{M}$ is something else, but we mark them as extensions rather than core LLM routing. Examples include reward models during alignment (Wu and Lu, 2025); draft models inside a speculative-decoding loop (Liu et al., 2026); or the vision and language backbones assembled into a multimodal model (He et al., 2025). We surface such cases alongside the LLM-routing literature where they are directly relevant, and revisit them as a research direction of their own in §5.

2.2 The Four Routing Paradigms

We organize the core surveyed work under four labels defined by their relation to the routing function R in Eq. (1). The three operating paradigms construct R , while Infrastructure studies it. Each paragraph gives a verbal characterisation and a representative formalisation. Figure 2 provides a visual summary.

Predictive. One common implementation fits a parametric scorer $f_\phi : \mathcal{Q} \times \mathcal{M} \rightarrow \mathbb{R}$ on labelled $\mathcal{D}_{\text{train}} = \{(q_i, m_i, y_i)\}_{i=1}^N$ (with y_i preference, gold correctness, or observational reward), and applied at decision time as the cost-adjusted argmax:

$$\phi^* = \arg \min_{\phi} \frac{1}{N} \sum_i \mathcal{L}(f_\phi(q_i, m_i), y_i) + \Omega(\phi), \quad (2)$$

$$R_{\text{pred}}(q) = \arg \max_{m \in \mathcal{M}} f_{\phi^*}(q, m) - \lambda C(q, m). \quad (3)$$

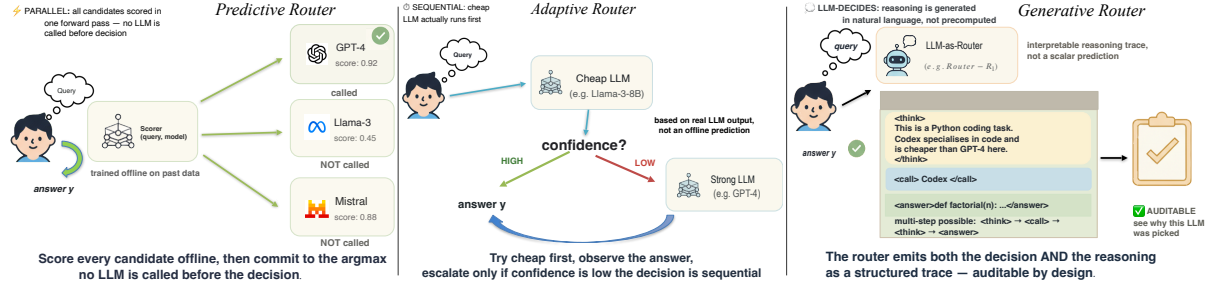


Figure 1: Anatomy of a routing decision across the three operating paradigms (Infrastructure is parallel rather than a routing operator, so it is omitted here). **Predictive** (left): an offline scorer $f(q, m)$ evaluates every candidate in a single forward pass; the router picks the $\arg\max$ and *no LLM is called before the decision*. **Adaptive** (middle): a cheap LLM is actually invoked first; a confidence check either commits its answer (HIGH) or escalates to a stronger model (LOW), so the decision uses real LLM output rather than a precomputed score. **Generative** (right): the router is itself an LLM that emits a structured reasoning, routing, and answer trace containing both the chosen candidate *and* a natural-language justification. Methods that retain this trace can expose evidence for later inspection, although a generated rationale is not itself a guarantee of faithful explanation.

Other implementations construct the score through retrieval, model profiles, or unsupervised estimation rather than the training objective in Eq. (2). The defining property is that the decision is made before candidate answers are generated. Whether the router transfers to a new pool depends on how it represents models and how well its evidence covers $\mathcal{M}$.

Adaptive. The router maintains a state s_t that summarises information observed so far and selects the next action conditional on s_t . Two instantiations cover the sub-area. *Cascade.* Order candidates by cost $C(q, m_1) \leq \dots \leq C(q, m_K)$; at stage j , observe a deferral signal $s_j = \sigma(q, m_j(q))$ and apply

$$R_{\text{cascade}}^{(j)}(q) = \begin{cases} m_j & s_j \geq \tau_j, \\ \text{defer to } j+1 & \text{otherwise.} \end{cases} \quad (4)$$

Bandit. Maintain reward estimates $\hat{Q}_t(q, m)$ and exploration bonuses $U_t(q, m)$ (e.g., LinUCB, Thompson) over history $\mathcal{H}_t$, and choose

$$m_t = \arg \max_{m \in \mathcal{M}} \hat{Q}_t(q, m) + \beta_t U_t(q, m) - \lambda C(q, m). \quad (5)$$

The defining property is sequential information acquisition; cascades observe within a query, bandits across queries.

Generative. The router itself is a language model π_θ . Given q and an encoding $\rho(\mathcal{M})$ of the pool (model cards, in-context demonstrations, or fine-tuned implicit representations), it samples a routing trajectory $\tau = (a_1, o_1, \dots, a_T)$ with actions $a_t \in$

$$\mathcal{A} = \{\langle \text{think} \rangle, \langle \text{call } m_i \rangle, \langle \text{answer} \rangle\}:$$

$$\tau \sim \pi_\theta(\cdot \mid q, \rho(\mathcal{M})), \quad R_{\text{gen}}(q) = \text{Read}(\tau). \quad (6)$$

Training θ uses supervised fine-tuning on (q, dec) pairs or RL with a routing reward $r(\tau, q)$: $\theta^* = \arg \max_\theta \mathbb{E}_{q, \tau \sim \pi_\theta} [r(\tau, q)]$. The single-shot specialization ($T = 1$) recovers an LLM-as-classifier router; multi-step ($T > 1$) supports chain-of-thought reasoning. The defining property is that routing is a generation problem. A prompt based implementation may transfer by updating $\rho(\mathcal{M})$, while a fine tuned or distributed implementation may still require training.

Infrastructure. The fourth paradigm differs methodologically: it does not produce R , it studies R . Given a router family $\mathcal{R}$ and a benchmark distribution $\mathcal{Q}_{\text{bench}}$, the operative objects include the quality–cost frontier $\mathcal{F}(\mathcal{R}) = \{(\bar{Q}(R), \bar{C}(R)) : R \in \mathcal{R}\}$ with $\bar{Q}(R) = \mathbb{E}[Q(q, R(q))]$ and $\bar{C}(R) = \mathbb{E}[C(q, R(q))]$. One robustness metric is the routing flip rate under perturbation Δ ,

$$\text{ASR}_{\text{flip}}(R, \Delta) = \mathbb{P}_q[R(\Delta(q)) \neq R(q)], \quad (7)$$

For attacks that seek cost inflation, a stricter metric is $\text{ASR}_{\text{cost}} = \mathbb{P}_q[C(q, R(\Delta(q))) > C(q, R(q))]$. The first metric counts every changed decision, while the second counts only changes that serve the stated attack objective. Infrastructure work also studies the deployment surface, including the workload, serving topology, latency budget, and KV cache reuse. The defining property is that the dependent variable is a property of R as a *function* (its frontier, robustness, or deployability) rather than

the decision $R(q)$ for any single q , so the paradigm has its own evaluation methodology that does not reduce to Eq. (1).

3 Taxonomy

Rationale and scope of the four paradigms.

Across the representative literature reviewed here, the utility of m for q is estimated from offline evidence, from feedback observed during operation, or from a decision generated by an LLM. We call these Predictive, Adaptive, and Generative routing. Infrastructure work instead studies the routing process. These labels capture the recurring dominant operations in our sample rather than exhaust the design space. Hybrid systems and adjacent extensions are marked explicitly when a single label would hide a material part of the method.

Cross-paradigm patterns. Predictive decisions usually have the lowest serving overhead, while Generative decisions usually have the highest. Pool transfer depends on the model representation and training procedure rather than on the paradigm label alone. Academic and industrial preferences also differ. Predictive work dominates publication counts, while cascades are common in deployment. Generative routing is structurally tied to the LLM-as-agent trend, so its growth tracks that broader movement rather than forces intrinsic to routing.

3.1 Paradigm 1: Predictive Routing

Predictive routers construct a score before candidate answers are generated and make one cost adjusted choice per query. Four sub-families differ by how that score is constructed: (i) *classification-based* treats routing as supervised “which-candidate-wins” prediction (RouteLLM (Ong et al., 2025), GraphRouter (Feng et al., 2025b), OptLLM (Liu et al., 2024), Hybrid LLM (Ding et al., 2024), and SkewRoute, a model router for knowledge graph RAG (Wang et al., 2025a)); (ii) *performance-regression-based* predicts continuous quality and cost per (q, m) pair (CARROT (Somerstep et al., 2025), Causal LLM Routing (Tsiourvas et al., 2026), IPR (Feng et al., 2025a)); (iii) *query-similarity-based* matches queries to candidates in a shared embedding space, including work that provides excess-risk bounds for unseen LLMs (EmbedLLM (Zhuang et al., 2025), UniRoute (Jitkrittum et al., 2025), RadialRouter (Jin et al., 2025)); (iv) *recommender-system-based* imports collaborative filtering and Item Response Theory (IRT-

Router (Song et al., 2025), RADAR (Fernandez et al., 2025), LLMRec/Let-the-LLM-Stick (Zhang et al., 2026c)).

Recent work extends this paradigm in two useful directions. Beyond GPT 5 introduces Avengers Pro, which optimises model choice for performance and efficiency together (Zhang et al., 2025b). ICL Router builds in context capability profiles from query vectors and model correctness labels, allowing new models to be added without retraining the router (Wang et al., 2026a). Both methods choose a candidate before an answer is generated, so they fall within the scope of predictive routing rather than aggregation.

Trade-off. The predictive paradigm’s selling point is decision-time cheapness. For conventional scorers, each routing call requires only a small forward pass and a cost lookup. The price is paid up front and again whenever $\mathcal{M}$ changes: a new candidate LLM usually means retraining, because (q, m) associations are pool-specific by construction. Probe vectors (Jitkrittum et al., 2025) and past-performance retrieval (Varangot-Reille et al., 2026) chip away at this, but a real zero-shot scorer over arbitrary unseen pools has not yet been established (Appendix §C.1).

3.2 Paradigm 2: Adaptive Routing

Adaptive routers try a model, observe feedback, and decide whether to commit, defer, or update. Three sub-families partition the area: (i) *cascade* escalates within a single query when a cheap model’s confidence is low (FrugalGPT (Chen et al., 2023), AutoMix (Aggarwal et al., 2024), SATER (Shen et al., 2025)); (ii) *bandit and online* update reward estimates across queries with formal regret guarantees (BaRP (Wei et al., 2025), MixLLM (Wang et al., 2025b), StageRoute (Li and Li, 2025)); (iii) *confidence-deferral* uses native model confidence as the trigger (Self-REF (Chuang et al., 2025)). The same operation extends beyond LLM candidates to reward-model selection (Wu and Lu, 2025) and drafter selection inside speculative decoding (Liu et al., 2026).

Trade-off. Adaptive routing is often straightforward to deploy. Cascade and bandit policies need only a candidate’s response plus a verifier or scalar reward, so commercial LLM endpoints can serve as arms when the required feedback is observable. This helps explain why cascades are common in industry despite predictive methods getting more

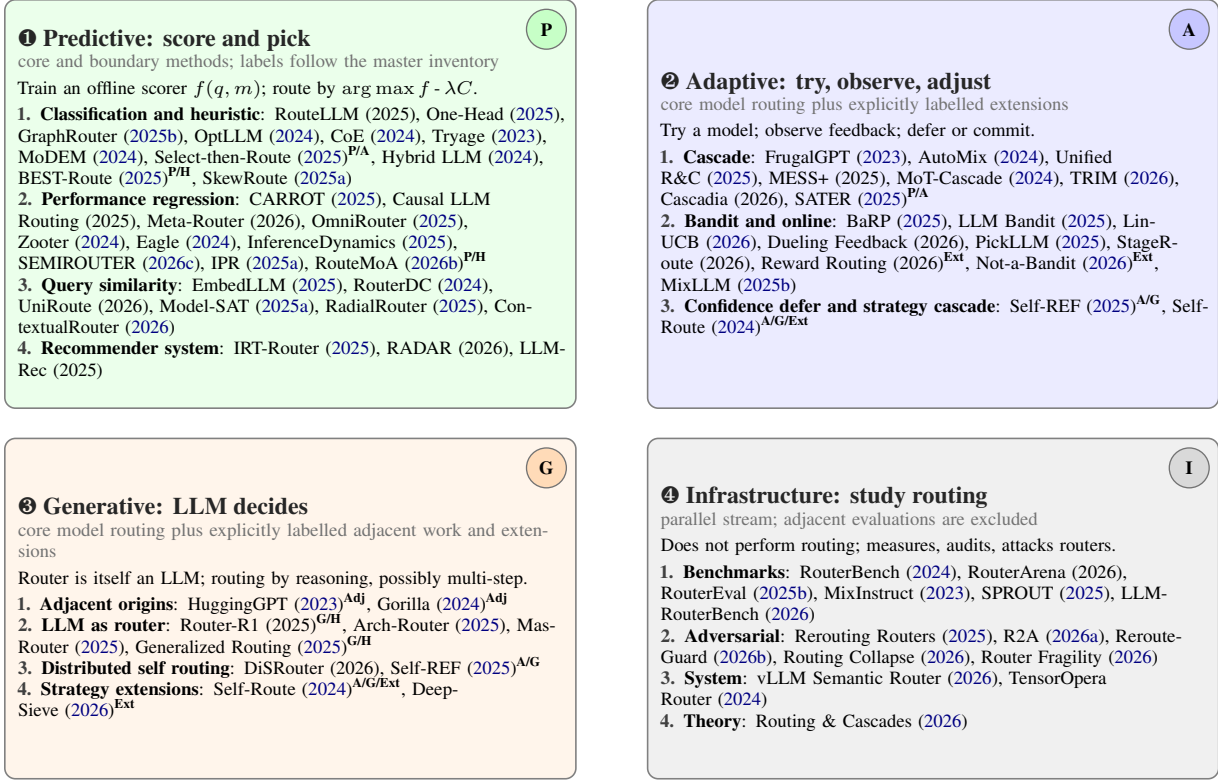


Figure 2: The four paradigms of LLM routing. The badges identify paradigms rather than paper counts. Cross paradigm labels indicate methods with substantive operations from two paradigms. Hybrid labels indicate routing combined with aggregation or another downstream mechanism. Extension and adjacent labels distinguish broader model selection targets and related systems from core routing methods.

academic attention. Online policies can absorb distribution shift through repeated updates. Fixed cascades do not have this property and still require threshold calibration when the workload changes. The trade-off is additional latency and a harder analysis problem. An escalated query has already incurred an earlier candidate call whose output may be discarded. In a two-stage cascade, this can approach the sum of both call latencies. Moreover, regret bounds for contextual bandits over LLMs with order-of-magnitude cost asymmetry remain loose; in practice, exploration constants end up hand-tuned (Appendix §C.2).

3.3 Paradigm 3: Generative Routing

The router itself is a language model that emits a routing trajectory of *think*, *call*, and *answer* actions. Tool selection systems such as HuggingGPT and Gorilla are historical precursors rather than core multi-LLM routers (Shen et al., 2023; Patil et al., 2024). Model routing was later formalised in two main branches: (i) *LLM-as-router* fine-tunes a single LLM to emit decisions (Router-

R1 (Zhang et al., 2026a), Arch-Router (Tran et al., 2025), MasRouter (Yue et al., 2025)); (ii) *distributed self-routing* lets each candidate decide whether to answer or forward (DiSRouter (Zheng et al., 2025)). Self Route is an Adaptive strategy extension that uses Generative self-assessment to escalate from retrieval augmented generation to full context processing (Li et al., 2024). DeepSieve is a Generative knowledge source extension that routes subquestions across tools and corpora (Guo et al., 2026).

Trade-off. Two properties make the generative paradigm distinctive. Prompt based routers can sometimes admit a new LLM by updating the pool description. Fine tuned and distributed routers may still need new training. Methods that emit and retain a structured trace offer material for later inspection, but that trace should not be treated as a faithful explanation without validation. Multi step protocols and agentic orchestration can be represented in the same action space. The trade-off is additional router inference. A generative router is more expensive than a small offline scorer, and a locally

served fine-tuned router LLM requires accelerator resources. The router prompt also becomes an attack surface distinct from the downstream LLM. This risk is only beginning to be characterised (Appendix §C.3).

3.4 Paradigm 4: Infrastructure

The fourth paradigm does not perform routing; it studies the routing process. Four sub-areas have emerged: *benchmarks* provide cross-router comparison surfaces (RouterBench (Hu et al., 2024), RouterArena (Lu et al., 2026), RouterEval (Huang et al., 2025b), and the more recent LLMRouterBench (Li et al., 2026)); *adversarial robustness* exposes router-specific vulnerabilities and proposes defences (Shafran et al. (2025), Tang et al. (2026a), Zhang et al. (2026b), Kassem et al. (2026)); *systems* sketches production architecture (Chen et al. (2026); TensorOpera Router (Stripelis et al., 2024) for production multi-model serving); *theory* formalises provider-versus-user welfare in cascades (Mahmood, 2026).

Trade-off. The infrastructure paradigm has unusual leverage: a shared benchmark can evaluate methods from all three operating paradigms against the same yardstick, while an attack can expose vulnerabilities across several router families. Coverage, on the other hand, is lopsided. Quality–cost benchmarks are dense, while safety benchmarks remain sparse. RerouteGuard is an early dedicated defence and has yet to face an adaptive attacker (Zhang et al., 2026b). The theory and production-systems strands remain less developed than benchmark research, with few links to the attack literature (Appendices §C.4, §D).

3.5 Adjacent Sub-areas

Several adjacent sub-areas share machinery with LLM routing and are frequently conflated with it; we fix the scope by noting where each fits. *Model merging and model soups* (Fisher-weighted merging, task-arithmetic) collapse $\mathcal{M}$ into a single model rather than choosing per query and are out of scope, though they share the predictive paradigm’s pool assumption and sometimes sit at the endpoint of a routing pipeline. *Automatic algorithm selection* poses a similar matching problem at a different level by choosing a solver for a problem instance before execution. Large Language Model Enhanced Algorithm Selection uses an LLM to encode algorithm code, selects task relevant di-

mensions, and matches the resulting representation with problem features (Wu et al., 2024). Towards Robustness and Explainability of Automatic Algorithm Selection uses a directed acyclic graph to model the conditional mechanisms linking problem and algorithm features. This improves robustness to marginal distribution shifts and supports model level and counterfactual explanations (Wu et al., 2025b). LLM driven Evolutionary Algorithm Selection represents optimisation problems through objective function syntax trees and evolutionary algorithms through LLM code embeddings, then derives a distributionally robust sampling strategy for negative instances (Wu et al., 2025a). These methods are adjacent rather than part of our inventory because they select general algorithms rather than language models. They are nevertheless informative for router design when candidate pools change and selection must rely on representations of both the query and the candidate. *Speculative-decoding drafter selection* (Liu et al., 2026) slots into the adaptive paradigm (§3.2) with drafters standing in for $\mathcal{M}$. *Adapter, LoRA, and module routing* (MoE-style gating) operates inside a single forward pass at the modular level rather than across whole LLMs; we treat this as a distinct problem, though the four paradigms still apply if “adapter” replaces “LLM” as the routed object (§5). *Knowledge source and RAG routing* changes the routed object from an LLM to a retriever or processing strategy. DeepSieve is a generative extension that routes subquestions across knowledge sources (Guo et al., 2026). SkewRoute instead uses retrieval score skewness to route questions between smaller and larger language models before generation. It is therefore a core Predictive model router for knowledge graph RAG (Wang et al., 2025a). *Tool selection in agentic systems* includes HuggingGPT and Gorilla (Shen et al., 2023; Patil et al., 2024). These works helped establish language generation as a control mechanism, but their candidate pools are tools and task models rather than a pool of independently trained LLMs. We therefore treat them as historical adjacent work. *Multi-agent orchestration frameworks* (LangChain, LangGraph, DSPy) include routing primitives but are framework documentation rather than routing research; we include only works (e.g., MasRouter (Yue et al., 2025)) whose primary contribution is routing methodology.

Team formation under structural constraints. Team Formation in Large Organizations offers a more distant analogue to selecting a set of agents.

It represents employees and communication links as a graph, then applies deep reinforcement learning to construct a team that covers the required skills while limiting disruption to the organisation (Lv et al., 2024). The work is not an LLM routing method because the selected entities are people and no query is dispatched to a language model. Its formulation is nevertheless relevant to future multi-agent routers that must select a compatible group rather than a single model.

Routing with aggregation and collective intelligence. Two recent systems move beyond selecting a single model. The Avengers learns a routing recipe that chooses candidate models and may combine repeated samples through voting (Zhang et al., 2026d). Beyond Gemini 3 Pro presents JiSi, which uses evidence from queries and candidate responses to switch between a direct routed answer and an aggregation strategy learned from a support set (Tang et al., 2026b). We classify both as hybrid routing and aggregation systems. Their routing components fit the taxonomy, but the full systems are adjacent because aggregation changes the computation and the source of the final answer. By contrast, Router R1 and RouteMoA remain core hybrid entries because learning or optimising the routing operation is central to their methodological contribution (Zhang et al., 2026a; Wang et al., 2026b). Broader collective intelligence frameworks that coordinate several models without making model selection their main technical contribution remain outside the core scope of this survey. The five recent routing and collective intelligence papers added during final revision are discussed for recency and are not included in the frozen statistics for the coded inventory of 75 works.

4 Comparison and Trends

Four cross cutting trends emerge when the frozen coded inventory is read through the paradigm lens. All counts, percentages, and temporal trends in this section describe this inventory and should not be interpreted as estimates of the complete routing literature. Figure 3 shows the publication growth and the original primary labels assigned during the frozen coding pass. The revised master tables retain secondary and adjacent scope labels, so those labels should be used for any paper level interpretation. Appendix Tables 2 and 3 report the supervision signal, representational intervention, and primary

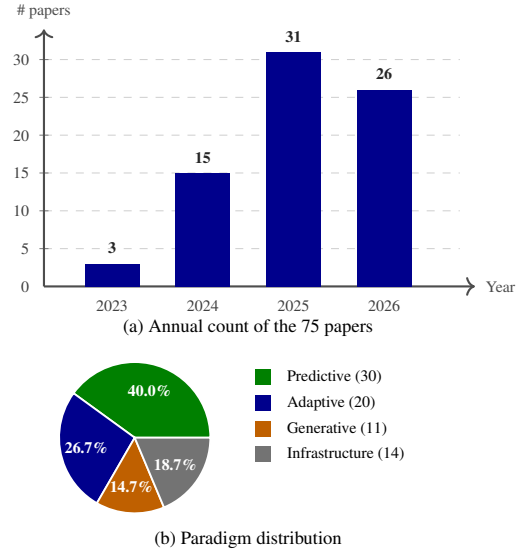


Figure 3: Publication trends and the original primary labels in the frozen coded inventory of 75 works. The master tables refine this initial coding with secondary and adjacent scope labels. The chart is therefore a record of the frozen counting pass rather than a count of methodologically pure core classes.

benchmark. A glossary of signal labels appears in Table 1.

Trend 1: Predictive dominates the count and the methodological palette. The original coding assigned roughly 40% of the inventory to the Predictive paradigm, and its internal range is wider than any single category label suggests. Classifier and embedding scorers (RouteLLM, EmbedLLM, RouterDC) sit next to causal estimators (Causal LLM Routing, Meta-Router) and structured capability profiles (UniRoute, IRT-Router, Inference-Dynamics); reward-distilled and label-free variants such as Zooter and SEMIROUTER coexist with cost aware optimisers such as CARROT, OptLLM, and IPR. Smoothie and RouteMoA use predictive components but are marked as output aware or hybrid in the master tables. Generative routing does not compete with this breadth so much as round it out. Router-R1, Arch-Router, MasRouter, and DiSRouter add reasoning-based decisions that predictive scorers do not naturally express. These four only cohere as a class when read against the predictive baseline.

Trend 2: Adaptive routing aligns well with API deployment. Cascade routing is common in industrial systems even though academic attention skews predictive. Cascade and bandit policies oper-

ate at the decision layer: they consume only an LLM’s response and a verifier or reward, without requiring the LLM’s weights. An HTTP endpoint can therefore be used as an arm when the required feedback is available. Contextual dueling bandits (Chiang et al., 2025) and preference conditioned bandits (Wei et al., 2025) add algorithmic depth without breaking the API deployment model. Causal LLM Routing is Predictive because it learns its policy from observational logs before deployment (Tsiourvas et al., 2026). LLM Selector is an adjacent active labelling method rather than an online routing policy (Durmazkeser et al.).

Trend 3: Infrastructure makes visible what the other three paradigms cannot. Treating infrastructure as a paradigm rather than as an appendix buys clarity. Benchmarks (RouterBench, RouterArena, RouterEval; full inventory in Appendix §E) turn quality–cost into a shared comparison surface that any of the three operating paradigms can be plotted on. Adversarial work (Shafran et al., 2025; Tang et al., 2026a; Zhang et al., 2026b) surfaces robustness properties no Predictive, Adaptive, or Generative method addresses internally. Theory (Mahmood, 2026) adds a normative axis (provider versus user welfare) that the empirical literature rarely addresses. The fragility audit of Kassem et al. (2026) flags a safety property orthogonal to all three. The most pressing gap is that the one published defence has not yet been evaluated under an adaptive attacker (§D.7).

Trend 4: Academia and industry route in different quadrants. Each paradigm has its own deployment shape, and the shapes do not line up across the academic–industrial divide. Predictive methods split between hosted classifiers such as RouteLLM CLS and Causal LLM Routing and local scorers such as GraphRouter, EmbedLLM, and UniRoute that need GPUs. Most Adaptive methods in the inventory are API callable because cascade and bandit logic needs decision-level feedback rather than access to model weights. Generative methods land at the opposite extreme: the router itself is an LLM, so local deployment usually requires a GPU. In the reviewed sample, academic publications are more concentrated in GPU-bound quadrants, while reported industrial deployments are more concentrated in API-callable quadrants. One way to narrow this *deployment gap* is to offer fine tuned router LLMs as commercial endpoints, making Generative routing API callable.

5 Future Directions

Open problems organise by paradigm, although progress in one paradigm can support work in another. Predictive representations can supply features for Adaptive bandits, while formal guarantees for multi step Generative routers could inform certified Infrastructure defences.

Predictive: zero-shot pool transfer. The open question is whether a single predictive scorer can transfer zero-shot to a new pool, as a sentence encoder transfers across NLP tasks without re-fitting the head. Hybrid representations that combine dense embedding, graph, and probe-vector signals, such as RouterDC (Chen et al., 2024) and GraphRouter (Feng et al., 2025b), provide a basis for reducing the performance gap on unseen pools. Research should also test whether methods such as UniRoute (Jitkrittum et al., 2025) and ContextualRouter (Varangot-Reille et al., 2026) remain effective across larger and more diverse candidate pools. Compositional capability profiles could eventually let a router score a novel fine tune by combining base model and adapter profiles. This possibility remains lightly studied in the reviewed literature.

Adaptive: principled exploration over heterogeneous LLMs. Empirical adaptive methods have developed faster than their theoretical analysis. An important open problem is contextual bandit exploration when the available LLMs have substantially different costs, as in BaRP (Wei et al., 2025) and MixLLM (Wang et al., 2025b). Regret bounds should represent this cost asymmetry explicitly. SLA conditioned cascades with guarantees under latency and quality constraints could also reduce reliance on heuristic thresholds. Multi turn off policy evaluation is another open direction because it would allow practitioners to estimate cascade quality from production logs without rerunning the underlying LLMs.

Multi turn routing also changes the unit of decision. MTRouter jointly embeds dialogue history and each candidate model, then uses terminal supervision from logged trajectories to estimate the current choice under a conversation budget (Zhang et al., 2026e). It remains an in scope routing method rather than a multiagent framework. We classify it as a stateful Predictive router because its policy is learned offline and does not update from online feedback. The method suggests defining

future objectives over complete interactions rather than isolated prompts.

Generative: deployment cost and prompt-injection robustness. Locally hosted LLM-as-router architectures require accelerator resources. Wider adoption therefore depends on compact router LLMs that can be offered as practical endpoints, following the direction explored by Arch-Router (Tran et al., 2025). We did not identify regret bounds for multi step generative routers. Establishing such guarantees would address an important theoretical gap and provide a basis for analysing more complex routing trajectories. Multi-modal and agentic dispatch also offer a path toward cross modal routing. Generative routers inherit a prompt injection attack surface that intersects the infrastructure agenda.

Infrastructure: adaptive-attack defences and deployment-aware metrics. The dedicated defence reviewed here (Zhang et al., 2026b) has not been evaluated against adaptive attackers who know the defence. Such evaluation is a priority for establishing whether its reported protection remains effective under a stronger threat model. Research should also develop certified routing defences with formal robustness guarantees. Benchmarks should incorporate deployment cost, safety, and attack success alongside accuracy so that defensive performance can be evaluated under realistic operating conditions. The limited evidence available for these questions shows that routing defence remains at an early stage.

Beyond model routing: adaptive module selection. Current routers treat each LLM as indivisible, but model reuse suggests routing reusable blocks, layers, or experts. Building LLMs Like LEGO reassembles Transformer blocks across depth and within layers (Wu et al., 2026). HM3 jointly optimises parameter and architecture spaces when merging pretrained models (Zhou et al., 2026). Towards Adaptive Continual Model Merging uses manifold based subspace affinity to evolve expert sets and activates them through feature subspace alignment without a learned gate (Qiu et al., 2026b). Fine Grained Model Merging via Modular Expert Recombination builds an expert library and uses a lightweight router to assemble a model for each input under a storage constraint (Qiu et al., 2026a). These systems study merging and reassembly rather than whole LLM routing, yet they move

the candidate set to internal modules. Future work should adapt routing objectives to this setting while modelling component compatibility and the cumulative cost of an assembled path.

Cross-paradigm directions. Five frontiers between the four paradigms are developed in Appendix §F: (i) joint routing and query reformulation; (ii) routing beyond LLM candidates (reward models (Wu and Lu, 2025), drafters (Liu et al., 2026), vision backbones (He et al., 2025), knowledge sources (Guo et al., 2026)); (iii) cross-modal routing (Moslem and Kelleher, 2026); (iv) privacy-preserving routing under formal (ϵ, δ) budgets; and (v) multi-objective routing returning Pareto fronts over quality, cost, latency, safety, and fairness.

6 Conclusion

LLM routing is now an established research area. We organise the field by what the router does. Predictive routing scores candidates before inference. Adaptive routing uses observed feedback. Generative routing delegates the decision to an LLM. Infrastructure studies the routing process itself. Scope labels preserve hybrid and adjacent cases. This view makes the main tradeoffs in overhead, deployment, pool transfer, and robustness explicit. It also exposes shared frontiers in query reformulation, certified defence, and cross modal routing.

Limitations

A few caveats apply to this survey. The frozen coded inventory contains 75 representative works from 2023 through 2026 after the final reconciliation pass. Additional adjacent papers discussed for context are not included in the quantitative counts. Industry systems without an academic counterpart (Martian, Portkey, and similar routing layers) are deliberately out of scope. The four-paradigm classification is lossy in places. Self-REF is the cleanest example: it acts as Adaptive in deferral mode and as Generative in native mode, so rather than forcing a single label we dual-list it. Numbers reproduced from the underlying papers, such as attack success rates and label budget savings, were not independently re-evaluated. Multimodal routing falls outside our scope (see Moslem and Kelleher (2026) for that direction). Finally, our treatment of robustness lives inside the Infrastructure paradigm rather than being broken out as a primary focus of its own.

Ethics Statement

This survey discusses adversarial attacks on production LLM routing systems for defensive understanding. Where attacks have been demonstrated against commercial systems, as in [Tang et al. \(2026a\)](#) and [Shafran et al. \(2025\)](#), we follow the original authors’ responsible-disclosure framing. Practitioners deploying LLM routers should treat the attacks surveyed in §3.4 as live threats and consider the design guidelines emerging from [Shafran et al. \(2025\)](#), [Tang et al. \(2026a\)](#), and [Zhang et al. \(2026b\)](#).

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62606442 and by The Hong Kong Polytechnic University under Project ID P0061084.

References

- Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, and 1 others. 2024. Automix: Automatically mixing language models. *Advances in Neural Information Processing Systems*, 37:131000–131034.
- Huamin Chen, Xunzhuo Liu, Bowei He, Fuyuan Lyu, Yankai Chen, Xue Liu, Yuhao Liu, and Junchen Jiang. 2026. The workload-router-pool architecture for llm inference optimization: A vision paper from the vllm semantic router project. *arXiv preprint arXiv:2603.21354*.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.
- Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. 2024. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328.
- Chao-Kai Chiang, Takashi Ishida, and Masashi Sugiyama. 2025. Llm routing with dueling feedback. *arXiv preprint arXiv:2510.00841*.
- Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. 2025. Learning to route llms with confidence tokens. In *International Conference on Machine Learning*, pages 10859–10878. PMLR.
- Jasper Dekoninck, Maximilian Baader, and Martin Vechev. 2025. A unified approach to routing and cascading for llms. In *International Conference on Machine Learning*, pages 12987–13010. PMLR.
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, volume 2024, pages 41348–41366.
- Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks VS Lakshmanan, Qingyun Wu, and Victor Rühle. 2025. Best-route: Adaptive llm routing with test-time optimal compute. In *International Conference on Machine Learning*, pages 13870–13884. PMLR.
- Mengsu Ding, Rong Zhu, Bolin Ding, and Jingren Zhou. 2026. [MixLLM: Selecting large language models with high-quality results and minimum inference cost for multi-stage complex tasks](#).
- Yavuz Durmazkeser, Patrik Okanovic, Andreas Kirsch, Torsten Hoefler, and Nezihe Merve Gürel. Active model selection for large language models. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*.
- Aosong Feng, Balasubramaniam Srinivasan, Yun Zhou, Zhichao Xu, Kang Zhou, Sheng Guan, Yueyan Chen, Xian Wu, Ninad Kulkarni, Yi Zhang, and 1 others. 2025a. Ipr: Intelligent prompt routing with user-controlled quality-cost trade-offs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2484–2498.
- Tao Feng, Yanzhen Shen, and Jiaxuan You. 2025b. Graphrouter: A graph-based router for llm selections. In *International Conference on Learning Representations*, volume 2025, pages 26186–26203.
- Nigel Fernandez, Branislav Kveton, Ryan A Rossi, Andrew S Lan, and Zichao Wang. 2025. Radar: Reasoning-ability and difficulty-aware routing for reasoning llms. *arXiv preprint arXiv:2509.25426*.
- Yichao Fu, Siqi Zhu, Runlong Su, Aurick Qiao, Ion Stoica, and Hao Zhang. 2024. [Efficient LLM scheduling by learning to rank](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Neel Guha, Mayee F Chen, Trevor Chow, Ishan S Khare, and Christopher Re. 2024. Smoothie: Label free language model routing. *Advances in Neural Information Processing Systems*, 37:127645–127672.

- Minghao Guo, Qingcheng Zeng, Xujiang Zhao, Yanchi Liu, Wenchao Yu, Mengnan Du, Haifeng Chen, and Wei Cheng. 2026. Deepsieve: Information sieving via llm-as-a-knowledge-router. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3054–3077.
- Xiyu Guo, Shan Wang, Chunfang Ji, Xuefeng Zhao, Wenhao Xi, Yaoyao Liu, Qinglan Li, Chao Deng, and Junlan Feng. 2025. Towards generalized routing: Model and agent orchestration for adaptive and efficient inference. *arXiv preprint arXiv:2509.07571*.
- Guy Hadad, Shadi Iskander, Sofia Tolmach, Oren Kalinsky, Haggai Roitman, and Ran Levy. 2026. Hotelquest: Balancing quality and efficiency in agentic search. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 209–225.
- Surya Narayanan Hari and Matt Thomson. 2023. Tryage: Real-time, intelligent routing of user prompts to large language models. *arXiv preprint arXiv:2308.11601*.
- Shiqi He, Insu Jang, and Mosharaf Chowdhury. 2025. Mordal: Automated pretrained model selection for vision language models. *arXiv preprint arXiv:2502.00241*.
- Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Router-bench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.
- Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. 2025a. Thriftllm: On cost-effective selection of large language models for classification queries. *Proceedings of the VLDB Endowment*, 18(11):4410–4423.
- Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025b. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*.
- Swayambhoo Jain, Ravi Raju, Bo Li, Zoltan Csaki, Jonathan Li, Kaizhao Liang, Guoyao Feng, Urmish Thakkar, Anand Sampat, Raghu Prabhakar, and 1 others. 2024. Composition of experts: A modular compound ai system leveraging large language models. *arXiv preprint arXiv:2412.01868*.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. 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 (Volume 1: Long Papers)*, pages 14165–14178.
- Youhe Jiang, Fangcheng Fu, Wanru Zhao, Stephan Rabanser, Jintao Zhang, Nicholas D Lane, and Binhang Yuan. 2025. Cascadia: An efficient cascade serving system for large language models. *arXiv preprint arXiv:2506.04203*.
- Ruihan Jin, Pengpeng Shao, Zhengqi Wen, Jinyang Wu, Mingkuan Feng, Shuai Zhang, and Jianhua Tao. 2025. Radialrouter: Structured representation for efficient and robust large language models routing. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14587–14600.
- Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, and 1 others. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.
- Vansh Kapoor, Aman Gupta, Hao Chen, Anurag Beniwal, Jing Huang, and Aviral Kumar. 2026. Trim: Hybrid inference via targeted stepwise routing in multi-step reasoning tasks. *arXiv preprint arXiv:2601.10245*.
- Aly M Kassem, Bernhard Schölkopf, and Zhijing Jin. 2026. How robust are router-llms? analysis of the fragility of llm routing capabilities. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7496–7507.
- Guannan Lai and Han-Jia Ye. 2026. When routing collapses: On the degenerate convergence of llm routers. *arXiv preprint arXiv:2602.03478*.
- Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, and 1 others. 2026. Llmrouter-bench: A massive benchmark and unified framework for llm routing. *arXiv preprint arXiv:2601.07206*.
- Shaoang Li and Jian Li. 2025. Near-optimal online deployment and routing for streaming llms. *arXiv preprint arXiv:2506.17254*.
- Yang Li. 2025. Llm bandit: Cost-efficient llm generation via preference-conditioned dynamic routing. *arXiv preprint arXiv:2502.02743*.
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893.
- Hongyi Liu, Jiayi Huang, Zhen Jia, Youngsuk Park, and Yu-Xiang Wang. 2026. Not-a-bandit: Provably no-regret drafter selection in speculative decoding for LLMs. In *The Fourteenth International Conference on Learning Representations*.
- Yueyue Liu, Hongyu Zhang, Yuantian Miao, Van-Hoang Le, and Zhiqiang Li. 2024. Optllm: Optimal assignment of queries to large language models. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 788–798. IEEE.

- Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. 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 (Volume 1: Long Papers)*, pages 1964–1974.
- Yifan Louie Lu, Rixin Liu, Jiayi Yuan, Xingqi Cui, Shenrun Zhang, Hongyi Liu, and Jiarong Xing. 2026. Routerarena: An open platform for comprehensive comparison of llm routers. In *International Conference on Learning Representations*, volume 2026, pages 44385–44406.
- Bing Lv, Junji Jiang, Likang Wu, and Hongke Zhao. 2024. Team formation in large organizations: A deep reinforcement learning approach. *Decision Support Systems*, 187:114343.
- Rafid Mahmood. 2026. Routing, cascades, and user choice for llms. *arXiv preprint arXiv:2602.09902*.
- Geoff McDonald and Jonathan Bar Or. 2025. Whisper leak: a side-channel attack on large language models. *arXiv preprint arXiv:2511.03675*.
- Kai Mei, Wujiang Xu, Minghao Guo, Shuhang Lin, and Yongfeng Zhang. 2025. Omnirouter: Budget and performance controllable multi-llm routing. *ACM SIGKDD Explorations Newsletter*, 27(2):107–116.
- Yasmin Moslem and John D Kelleher. 2026. Dynamic model routing and cascading for efficient llm inference: A survey. *arXiv preprint arXiv:2603.04445*.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. 2025. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pages 34433–34448.
- Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.
- Manhin Poon, XiangXiang Dai, Xutong Liu, Fang Kong, John CS Lui, and Jinhang Zuo. 2026. Online multi-llm selection via contextual bandits under unstructured context evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24855–24863.
- Roshini Pulishetty, Mani Kishan Ghantasala, Keerthy Kaushik Dasoju, Niti Mangwani, Vishal Garimella, Aditya Mate, Somya Chatterjee, Yue Kang, Ehi Nosakhare, Sadid Hasan, and 1 others. 2025. One head, many models: Cross-attention routing for cost-aware llm selection. *arXiv preprint arXiv:2509.09782*.
- Haiyun Qiu, Xingyu Wu, Liang Feng, and Kay Chen Tan. 2026a. Fine-grained model merging via modular expert recombination. *arXiv preprint arXiv:2602.06552*.
- Haiyun Qiu, Xingyu Wu, and Kay Chen Tan. 2026b. Towards adaptive continual model merging via manifold-aware expert evolution. *arXiv preprint arXiv:2604.22464*.
- John R Rice. 1976. The algorithm selection problem. In *Advances in computers*, volume 15, pages 65–118. Elsevier.
- Avital Shafran, Roei Schuster, Thomas Ristenpart, and Vitaly Shmatikov. 2025. Rerouting llm routers. *arXiv preprint arXiv:2501.01818*.
- Soham Shah and Kumar Shridhar. 2025. Select-then-route: Taxonomy guided routing for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 425–441.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugging-gpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.
- Yuanzhe Shen, Yide Liu, Zisu Huang, Ruicheng Yin, Xiaoqing Zheng, and Xuan-Jing Huang. 2025. Sater: A self-aware and token-efficient approach to routing and cascading. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10526–10540.
- Haochen Shi, Tianshi Zheng, Weiqi Wang, Baixuan Xu, Chunyang Li, Chunkit Chan, Tao Fan, Yangqiu Song, and Qiang Yang. 2025. Inferencedynamics: Efficient routing across llms through structured capability and knowledge profiling. *arXiv preprint arXiv:2505.16303*.
- Dimitrios Sikeridis, Dennis Ramdass, and Pranay Pareek. 2025. Pickllm: Context-aware rl-assisted large language model routing. In *International Workshop on AI for Transportation*, pages 227–239. Springer.
- Toby Simonds, Kemal Kurniawan, and Jey Han Lau. 2024. Modem: Mixture of domain expert models. In *Proceedings of the 22nd Annual Workshop of the Australasian Language Technology Association*, pages 75–88.
- Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. 2025. Carrot: A cost aware rate optimal router. *arXiv preprint arXiv:2502.03261*.
- Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. 2025. Irt-router: Effective and interpretable multi-llm routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for*

- Computational Linguistics (Volume 1: Long Papers)*, pages 15629–15644.
- Dimitris Stripelis, Zhaozhuo Xu, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Jipeng Zhang, Tong Zhang, Salman Avestimehr, and Chaoyang He. 2024. Tensoropera router: A multi-model router for efficient llm inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 452–462.
- Haochun Tang, Yuliang Yan, Jiahua Lu, Huaxiao Liu, and Enyan Dai. 2026a. Route to rome attack: Directing llm routers to expensive models via adversarial suffix optimization. *arXiv preprint arXiv:2604.15022*.
- Shengji Tang, Weihao Lin, Peng Ye, Jingqi Ye, Hao Li, Yiqun Zhang, Xiaosong Wang, Bo Zhang, Shuyue Hu, Tao Chen, and 1 others. 2026b. Beyond gemini-3-pro: Revisiting llm routing and aggregation at scale. *arXiv preprint arXiv:2601.01330*.
- Co Tran, Salman Paracha, Adil Hafeez, and Shuguang Chen. 2025. Arch-router: Aligning llm routing with human preferences. *arXiv preprint arXiv:2506.16655*.
- Asterios Tsiourvas, Wei Sun, and Georgia Perakis. 2026. Causal llm routing: End-to-end regret minimization from observational data. *Advances in Neural Information Processing Systems*, 38:37289–37314.
- Clovis Varangot-Reille, Christophe Bouvard, and Antoine Gourru. 2026. Generalising llm routing using past performance retrieval: A few-shot router is sufficient. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 304–319.
- Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. 2025. Doing more with less—implementing routing strategies in large language model-based systems: An extended survey. *arXiv e-prints*, pages arXiv–2502.
- Chenxu Wang, Hao Li, Yiqun Zhang, Linyao Chen, Jianhao Chen, Ping Jian, Qiaosheng Zhang, and Shuyue Hu. 2026a. Icl-router: In-context learned model representations for llm routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33413–33421.
- Hairu Wang, Yuan Feng, Yukun Cao, Xike Xie, and S Kevin Zhou. 2025a. Skewroute: Training-free llm routing for knowledge graph retrieval-augmented generation via score skewness of retrieved context. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11324–11340.
- Jize Wang, Han Wu, Zhiyuan You, Yiming Song, Yijun Wang, Zifei Shan, Yining Li, Songyang Zhang, Xinyi Le, Cailian Chen, and 1 others. 2026b. Routemoa: Dynamic routing without pre-inference boosts efficient mixture-of-agents. *arXiv preprint arXiv:2601.18130*.
- Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2025b. Mixllm: Dynamic routing in mixed large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10912–10922.
- Zijie Wang, Xinyu Yan, Che Wang, Zihao Zeng, Lei Xiao, and Wei Yang Bryan Lim. 2026c. SEMROUTER: Sparse-data enhanced routing for adaptive multi-LLM system. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Rabat, Morocco.
- Wang Wei, Tiankai Yang, Hongjie Chen, Yue Zhao, Franck Dernoncourt, Ryan A Rossi, and Hoda Eldardiry. 2025. Learning to route llms from bandit feedback: One policy, many trade-offs. *arXiv preprint arXiv:2510.07429*.
- Herbert Woisetschläger, Ryan Zhang, Shiqiang Wang, and Hans Arno Jacobsen. 2026. Mess+: Dynamically learned inference-time llm routing in model zoos with service level guarantees. *Advances in Neural Information Processing Systems*, 38:54034–54067.
- Xingyu Wu, Yinglan Feng, and Jibin Wu. 2025a. Llm-driven evolutionary algorithm selection: bidirectional framework and sampling paradigm. In *2025 International Conference on Machine Intelligence and Nature-Inspired Computing (MIND)*, pages 1–7. IEEE.
- Xingyu Wu, Jibin Wu, Yu Zhou, Liang Feng, and Kay Chen Tan. 2025b. Towards robustness and explainability of automatic algorithm selection. In *Forty-second International Conference on Machine Learning*.
- Xingyu Wu, Yan Zhong, Jibin Wu, Bingbing Jiang, and Kay Chen Tan. 2024. Large language model-enhanced algorithm selection: towards comprehensive algorithm representation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 5235–5244.
- Xingyu Wu, Yu Zhou, and Kay Chen Tan. 2026. Building llms like lego: Two-dimensional architecture reassembly of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44939–44956.
- Xinle Wu and Yao Lu. 2025. Reward model routing in alignment. *arXiv preprint arXiv:2510.02850*.
- Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2024. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *International Conference on Learning Representations*, volume 2024, pages 21691–21728.

- Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. 2025. Masrouter: Learning to route llms for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15549–15572.
- Haozhen Zhang, Tao Feng, and Jiaxuan You. 2026a. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:141233–141265.
- Wenhui Zhang, Huiyu Xu, Zhibo Wang, Zhichao Li, Zeqing He, Xuelin Wei, and Kui Ren. 2026b. Rerouteguard: Understanding and mitigating adversarial risks for llm routing. *arXiv preprint arXiv:2601.21380*.
- Yi-Kai Zhang, Shiyin Lu, Qingguo Chen, Weihua Luo, De-Chuan Zhan, and Han-Jia Ye. 2026c. Let the llm stick to its strengths: Learning to route economical llm. *Advances in Neural Information Processing Systems*, 38:33556–33575.
- Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. 2025a. Capability instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25958–25966.
- Yichi Zhang, Fangzheng Xie, Shu Yang, and Chong Wu. Meta-router: Bridging gold-standard and preference-based evaluations in llm routing. In *The Fourteenth International Conference on Learning Representations*.
- Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. 2025b. Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing. In *Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence*, pages 122–129.
- Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Xinrun Wang, Jia Xu, Lei Bai, and 1 others. 2026d. The avengers: A routing recipe for collective intelligence in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34870–34878.
- Yiqun Zhang, Hao Li, Zihan Wang, Shi Feng, Xiaocui Yang, Daling Wang, Bo Zhang, Lei Bai, and Shuyue Hu. 2026e. Mtrouter: Cost-aware multi-turn llm routing with history–model joint embeddings. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44206–44226.
- Zesen Zhao, Shuwei Jin, and Z Morley Mao. 2024. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.
- Hang Zheng, Hongshen Xu, Yongkai Lin, Shuai Fan, Lu Chen, and Kai Yu. 2025. Disrouter: Distributed self-routing for llm selections. *arXiv preprint arXiv:2510.19208*.
- Xinyao Zheng, Husheng Han, Shangyi Shi, Qiyang Fang, Zidong Du, Xing Hu, and Qi Guo. 2024. Inputsnoatch: Stealing input in llm services via timing side-channel attacks. *arXiv preprint arXiv:2411.18191*.
- Da-Wei Zhou and Han-Jia Ye. 2025. A unifying perspective on model reuse: From small to large pretrained models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25, International Joint Conferences on Artificial Intelligence Organization*, pages 10826–10835.
- Yu Zhou, Xingyu Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. 2026. Hm3: Hierarchical multi-objective model merging for pretrained models. *Advances in Neural Information Processing Systems*, 38:114876–114910.
- Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. 2025. Embedllm: Learning compact representations of large language models. In *International Conference on Learning Representations*, volume 2025, pages 76913–76926.

Appendix

Supplementary material for the main survey above.

A Paper Selection Methodology

To make the construction of the frozen coded inventory of 75 works transparent, we document the paper selection process in detail. Any survey reflects a particular sample of the literature, and its selection decisions form part of the methodology. We therefore report the sources searched, the queries used, the inclusion and exclusion criteria, and the treatment of borderline cases.

A.1 Search Strategy

The structured search ran in two passes between October 2025 and April 2026. Subsequent reconciliation passes through August 2026 incorporated newly published work and resolved duplicate or boundary entries. Four sources fed into the pipeline.

Conference proceedings. We scanned the ACL Anthology for ACL, EMNLP, NAACL, EACL, and COLM 2023–2026, plus the OpenReview portals for ICLR, ICML, and NeurIPS in the same window. Titles and abstracts were filtered with the keyword set {router, routing, LLM selection, model selection, cascading, multi-LLM, multi-model inference, query routing, model orchestration, multi-agent routing}. The set itself was iteratively refined. Our first pass omitted “cascading” and consequently missed FrugalGPT (Chen et al., 2023) and AutoMix (Aggarwal et al., 2024), two papers that effectively define the adaptive paradigm. The keyword went in for the second pass.

arXiv listings. We monitored listings under cs.CL, cs.LG, and cs.AI with the same keyword filter. arXiv coverage matters because routing methods often appear as preprints before formal publication. It also helped us track recent attack and defence work (Tang et al., 2026a; Zhang et al., 2026b; Lai and Ye, 2026).

Citation tracing. We then did two-hop forward and backward citation expansion from three anchor papers, each chosen to seed a different thread: RouteLLM (Ong et al., 2025) for preference-driven work, RouterBench (Hu et al., 2024) for benchmark work, and the prior survey of Moslem and Kelleher (2026) for methodological breadth. Citation

tracing added work that the keyword search alone had missed. Examples include Shafran et al. (2025) from the adversarial thread and Chiang et al. (2025) from the online-learning thread.

Community lists. Finally, we cross-checked our set against two community-maintained lists: the MilkThink-Lab Awesome-Routing-LLMs GitHub repository and the OpenRouter community wiki. Neither is formally curated and both mix industry tools in with academic papers, so we used them as completeness checks rather than primary sources. The cross-check surfaced additional candidates, which were then evaluated against the same inclusion and exclusion criteria.

A.2 Inclusion Criteria

Core operating methods had to make a query dependent choice among two or more independently trained LLMs. They also had to report an empirical routing evaluation. Infrastructure papers qualified when their primary object was a routing benchmark, attack, defence, theoretical analysis, or deployment architecture. Included papers appeared between 2023 and 2026 and were publicly accessible through proceedings, arXiv, or an open technical report.

The inventory also retains three kinds of boundary material with an explicit scope label. Output aware selection and aggregation systems are marked as hybrid or adjacent. Component and strategy selection systems are marked as extensions beyond an LLM candidate pool. Historical tool routing systems are marked as adjacent precursors. LLM Blender is retained because MixInstruct became a routing benchmark, but the full rank and fusion system is not treated as a core pre inference router (Jiang et al., 2023). This policy preserves the intellectual lineage without weakening the definition of the core class.

A.3 Exclusion Criteria

Four categories were excluded. Mixture-of-Experts methods such as Switch Transformer and GShard route inside a single trained model and form a separate research community with its own conventions and benchmarks. We therefore exclude them from the core inventory. Single-LLM prompt optimisation (OPRO, APE, EvoPrompt) and fine-tuning methods that do not involve multi-model selection are out of scope, though we revisit their potential combination with routing in §5. Pure systems pa-

pers covering load balancing, KV cache scheduling, or batching without a routing question are excluded. Routing aware serving studies are retained only as Infrastructure or adjacent resources. The Workload Router Pool vision qualifies because it explicitly makes routing part of the proposed architecture (Chen et al., 2026). Workshop papers without an accessible manuscript are excluded to ensure reproducibility and source verification.

A.4 Borderline Cases and Judgment Calls

Three categories sat right on the boundary and deserve an explicit explanation of how we handled them.

Industrial routing services. OpenRouter, GPT 5 Auto, Martian, and Portkey are commercial routing services without peer reviewed method papers. We discuss systems that are directly evaluated by the cited robustness literature, but commercial products do not become entries in the frozen coded inventory. Public documentation does not provide the methodological detail needed for the same comparison used for research papers.

Agent orchestration frameworks. Packages like LangChain, LangGraph, and DSPy ship routing primitives, but they are not themselves routing research. We admit only works whose primary contribution is routing methodology. MasRouter (Yue et al., 2025) qualifies; framework documentation does not.

Recommender-system papers using LLM-as-item. A handful of papers reframe LLM routing as a recommender-system problem, each candidate LLM becomes an item, each query a user. We include such papers when the paper itself positions its contribution as a routing method, and exclude broader recommender-system work that incidentally treats LLMs as items without addressing the quality–cost trade-off that is central to routing.

A.5 Limitations of the Sample

The frozen coded inventory of 75 works is not exhaustive, and three areas are honestly under-represented. The multimodal routing literature falls largely outside our scope; Moslem and Kelleher (2026) cover it separately, which we relied on rather than reproducing. Retrieval routing in RAG pipelines; choosing among retrievers rather than among LLMs, shares mechanisms with our subject but is studied by a different community

with different benchmarks, so we excluded it. And non-English-language routing research, particularly Chinese-language work on TagRouter and on capability-aware routing, is likely under-cited because we ran the keyword search in English only. Closing each of these three gaps is a natural next step for an extended version of this survey.

A.6 Final Set

The resulting frozen coded inventory contains 75 representative works. It includes core routing methods, infrastructure, and explicitly marked boundary material. Tables 2 and 3 classify them by paradigm and scope together with supervision signal and representational intervention; the trends visible in the table are discussed in §4, and the per-paradigm tables in §C provide additional metadata such as candidate-pool size, code release status, and deployment requirements for the core methods they cover.

B Master Method Inventory

This appendix consolidates the scope and paradigm classification of the 75 works in the frozen coded inventory. Table 2 covers 2023–2025 entries; Table 3 covers 2026 entries together with the adversarial sub-area. Table 1 translates the short labels in the **Signal** column into plain English. Per-paradigm sub-tables (with deployment shape and code availability) follow in §C.

For the **Repr. and Modify** column shared across both master tables: *Embedding* refers to a per-LLM learned vector; *Classifier* and *LLM classifier* refer to small versus LLM-sized scorers over (q, m) ; *LLM-as-router* means the router itself is an LLM; *Modify LLMs* means the method alters the candidate LLMs; *Cascade* and *Cascade-with-sample* refer to cheap-to-expensive escalation, optionally with adaptive sample count; *Recommender* refers to a collaborative-filtering style; *Cross-attention* refers to a single-attention-head router; *Probe vector* means an LLM fingerprint from a fixed probe set; *Capability*, *Skill profile*, and *Item-response* are structured factorings of LLM ability; *Multi-stage* means stage-wise model selection; *Ranker and Fuser* means a pairwise ranker plus a generative fuser.

C Extended Per-Paradigm Tables

This appendix provides per-paradigm summary tables that supplement the consolidated Tables 2

Signal	What the router learns from (plain English)
<i>Pairwise</i>	“Model A vs. model B, which one’s answer is better on this query?” (binary comparisons)
<i>Preference</i>	Same as Pairwise but specifically from human-preference platforms (e.g., Chatbot Arena) plus data augmentation
<i>Gold</i>	Ground-truth correctness labels: “Did model m get this query right?” Requires an oracle answer
<i>Contrastive</i>	Pulls matching (q, m) pairs together and pushes non-matching pairs apart in an embedding space
<i>Rank</i>	Treats routing as a ranking problem: order all candidate models by predicted quality
<i>Confidence</i>	The router uses an LLM’s self-reported confidence (token or verbalized) as the routing signal
<i>Bandit</i>	Online: try a model, observe its reward, update the policy. No labels needed in advance
<i>Full information</i>	Online: evaluate every candidate after each choice and update the selector from the complete feedback vector
<i>CDB</i>	Contextual Dueling Bandit, like Bandit but with pairwise preferences instead of scalar rewards
<i>Observational</i>	Passive deployment logs: which model was called and what happened, with no controlled experiment
<i>Causal</i>	Uses causal inference to remove the bias in observational data (which models got which queries)
<i>Active</i>	Uses active learning to pick which queries are worth labelling, under a labelling budget
<i>RL</i>	Reinforcement learning: a router LLM is trained by trial-and-error against a reward (e.g., GRPO, PPO)
<i>ICL</i>	In-context learning: no parameter updates. The router LLM is given few-shot examples in its prompt
<i>SFT</i>	Supervised fine-tuning: the router LLM is trained on $(query, correct\ model)$ labelled pairs
<i>Self-awareness</i>	Each candidate LLM is trained to judge its own competence and forward queries it cannot handle
<i>Unsupervised</i>	Label-free routing: the router learns from the candidate models’ own outputs (e.g., embeddings of generated answers) without any oracle labels

Table 1: Plain-English glossary of the **Signal** column of Tables 2 and 3. Each signal type is a different way the router learns “which model to pick for which query”.

and 3 in the main text. Each table records, for the methods within one paradigm, the candidate-pool size, the code release status, the deployment requirement, and a one-phrase summary of the primary contribution. The tables are intended for two uses. First, a reader who wants to adopt a method for production should be able to determine, at a glance, whether the method is reproducible (Code), deployable without local GPUs (Deploy), and tested at the scale they need (Pool). Second, a researcher who wants to position new work should be able to identify the closest existing point in the same paradigm by reading the contribution column. We discuss patterns that emerge from each table after the table itself.

C.1 Predictive Paradigm

For reference, we reproduce here the per-paper narrative previously rendered in the main body; the table that follows summarises the same set of methods on a per-method basis.

Classification-based. The most direct instantiation trains a classifier whose output is the chosen LLM’s identity. Ong et al. (2025) train RouteLLM

in four variants (similarity-weighted, matrix factorization, BERT classifier, and a causal-LLM classifier) from Chatbot Arena pairwise preferences augmented with golden labels and LLM-judge synthetic data; the BERT and causal-LLM variants are pure classifiers, while the SW and MF variants foreshadow the similarity- and recommender-based families discussed below. Pulishetty et al. (2025) use a single cross-attention head as the scoring classifier. GraphRouter (Feng et al., 2025b) represents tasks, queries, and candidate LLMs as nodes in a heterogeneous graph and trains an HGT-style network end-to-end as a router classifier. For routing, this suggests that serialising a candidate graph as text is not a substitute for topology aware message passing. Domain-classifier variants solve a coarser version of the same problem: Tryage (Hari and Thomson, 2023) dispatches each query to one of a pool of expert LLMs through a single forward pass, and MoDEM (Simonds et al., 2024) uses a lightweight DeBERTa-v3-large classifier to route queries to domain expert models (health, mathematics, science). Composition-of-Experts (Jain et al., 2024) chains two classifiers (input category, then expert LLM). OptLLM (Liu et al., 2024) pairs a multi-label classifier over (q, m) with iterative non-dominated solution generation (not a linear program) to solve the assignment problem under explicit cost-quality constraints. Hybrid LLM (Ding et al., 2024) is a binary classifier that predicts whether a query should be answered by a small or a large LLM, routing each query to a single candidate at inference time rather than escalating through a cascade (its authors explicitly distinguish their design from cascades and ensembles). Select-then-Route (Shah and Shridhar, 2025) stages the classification step explicitly: a taxonomy-guided selector first narrows the pool to task-appropriate candidates, after which a confidence-based cascade routes within the reduced pool. We mark it as Predictive and Adaptive because both operations affect the final path. Mordal applies a related selection template to model assembly over vision and language backbones rather than per query LLM routing, so it is an extension in the master table (He et al., 2025).

Performance-regression-based. A second family predicts continuous performance (quality, cost, or both) per (q, m) pair and routes by argmax. CARROT (Somerstep et al., 2025) predicts both cost and accuracy per query and proves a min-

Method	Year	Paradigm	Signal	Repr. / Modify	Benchmark
LLM-Blender (Jiang et al., 2023)	2023	Adj	Pairwise	Ranker+Fuser	MixInstruct
HuggingGPT (Shen et al., 2023)	2023	Adj	ICL	LLM planner	24 HF tasks
Tryage (Hari and Thomson, 2023)	2023	P	Gold	Neural classifier	code / text / clinical / patents
RouterBench (Hu et al., 2024)	2024	I	Gold	Benchmark	RouterBench
TensorOpera Router (Stripelis et al., 2024)	2024	I	–	Serving router	Multi-model serving traces
EmbedLLM (Zhuang et al., 2025)	2025	P	Gold	Embedding	MMLU/GSM8K family
RouterDC (Chen et al., 2024)	2024	P	Contrastive	Embedding	MMLU/GSM8K family
Eff. Scheduling (Fu et al., 2024)	2024	I/Adj	Rank	Scheduler	Serving traces
Hybrid LLM (Ding et al., 2024)	2024	P	Gold	Binary classifier	MixInstruct
AutoMix (Aggarwal et al., 2024)	2024	A	Confidence	Cascade	5 LLMs, 5 suites
Gorilla (Patil et al., 2024)	2024	Adj	SFT	LLM API caller	APIBench (1.6k+ APIs)
Zooter (Lu et al., 2024)	2024	P	Preference	Distilled router	MT-Bench, AlpacaEval
Eagle (Zhao et al., 2024)	2024	P	Pairwise	ELO ranker (training-free)	RouterBench
OptLLM (Liu et al., 2024)	2024	P	Gold	Classifier+heuristic search	12 LLMs (NLP), multi-objective
MoT-Cascade (Yue et al., 2024)	2024	A	Confidence	Cascade+CoT/PoT	GSM8K + 5 reasoning suites
Self-Route (Li et al., 2024)	2024	A/G/Ext	Confidence	Cascade + LLM judge	NarrativeQA, HotpotQA
MoDEM (Simonds et al., 2024)	2024	P	Gold	Domain classifier	MMLU domain split
Smoothie (Guha et al., 2024)	2024	Adj	Unsupervised	Latent quality model	AlpacaEval, SQuAD, MixInstruct, GSM8K
PickLLM (Sikeridis et al., 2025)	2025	A	RL	Q-learning router	4-LLM pool, prompt-response set
CoE (Jain et al., 2024)	2024	P	Gold	Two-step classifier	SambaNova suite
RouteLLM (Ong et al., 2025)	2025	P	Preference	SW, MF, BERT, and LLM scorers	MT-Bench, MMLU, GSM8K
GraphRouter (Feng et al., 2025b)	2025	P	Gold	Heterogeneous GNN	4 tasks (Alpaca, GSM8K, SQUAD, Multi-News)
Self-REF (Chuang et al., 2025)	2025	A/G	Confidence	Modify LLMs	4 suites (MMLU, OpenbookQA, GSM8K, MedQA)
Router-R1 (Zhang et al., 2026a)	2025	G/H	RL	LLM-as-router	7 QA suites (4 multi-hop + 3 general)
Arch-Router (Tran et al., 2025)	2025	G	SFT	LLM-as-router	CLINC-150, MANtIS, SGD, LMSYS-1M
BaRP (Wei et al., 2025)	2025	A	Bandit	Embedding	8 tasks
LLM Bandit (Li, 2025)	2025	A	Bandit	Embedding	HELM, OpenLLM
Causal LLM Routing (Tsiourvas et al., 2026)	2025	P	Observational	Causal head	RouterBench, SPROUT
BEST-Route (Ding et al., 2025)	2025	P/H	Gold	Model+sample choice	Custom + MT-Bench
Unified R&C (Dekoninck et al., 2025)	2025	A	Gold	Cascade	RouterBench
OmniRouter (Mei et al., 2025)	2025	P	Gold	RAG + Lagrangian	Custom
MESS+ (Woisetschlager et al., 2026)	2025	A	Gold	SLA scheduler	TinyLlama-1.1B + Llama-2-13B
MasRouter (Yue et al., 2025)	2025	G	Gold	Multi-agent router	5 suites
RouterEval (Huang et al., 2025b)	2025	I	Gold	Benchmark	RouterEval
Model-SAT (Zhang et al., 2025a)	2025	P	Gold	Skill profile	MMLU/GSM8K family
Generalized Routing (Guo et al., 2025)	2025	G/H	Gold	Agent orchestrator	5 suites
One-Head (Pulishetty et al., 2025)	2025	P	Gold	Cross-attention	RouterBench
InferenceDynamics (Shi et al., 2025)	2025	P	Gold	Capability	RouteMix
IRT-Router (Song et al., 2025)	2025	P	Gold	Item-response	12 datasets
CARROT (Somerstep et al., 2025)	2025	P	Gold	Cost+accuracy classifier	SPROUT, RouterBench
SATER (Shen et al., 2025)	2025	P/A	Confidence	Cascade+exit	6 suites
MixLLM (NAACL) (Wang et al., 2025b)	2025	A	Bandit	Contextual bandit	Mixed-LLM serving
RadialRouter (Jin et al., 2025)	2025	P	Gold	Radial Transformer	RouterBench
Select-then-Route (Shah and Shridhar, 2025)	2025	P/A	Gold	Taxonomy classifier + cascade	6 suites
IPR (Feng et al., 2025a)	2025	P	Gold	Quality-cost regressor	industrial cloud workload (11 LLMs)
SkewRoute (Wang et al., 2025a)	2025	P	Confidence	Model router (KG-RAG)	KG-RAG suites
LLM Selector (Durmazkeser et al.)	2025	Adj	Active	Embedding	6 benchmarks (151 LLMs)
LLMRec (Zhang et al., 2026c)	2025	P	Gold	Recommender (RecSys)	52 train / 10 eval LLMs

Table 2: Representative routing literature inventory for 2023–2025 (Part 1 of 2; 2026 papers in Table 3). **Paradigm:** **P**=Predictive (offline scorer + argmax), **A**=Adaptive (try, observe, adjust), **G**=Generative (LLM-as-router), **I**=Infrastructure, **H**=Hybrid routing and aggregation, **Adj**=Adjacent work, and **Ext**=an extension beyond whole LLM routing. A slash preserves a material secondary label. Self REF, for example, is Adaptive in deferral mode and Generative in native mode. **Acronyms:** *RL*=reinforcement learning, *ICL*=in-context learning, *SFT*=supervised fine-tuning, *GNN*=graph neural network. A glossary appears before the table.

imax rate-optimality guarantee on the resulting router. Tsiourvas et al. (2026) treat routing as end-to-end regret minimisation from observational deployment data, combining a classification-based upper-bound surrogate with a softmax-weighted regret approximation under interval conditioning over the cost-sensitivity parameter. Zhang et al. use CATE estimation to combine gold-standard and biased preference signals. OmniRouter (Mei et al., 2025) fits a utility predictor under explicit budget and performance constraints. Zooter (Lu et al., 2024) distils a reward model into a lightweight scorer, regressing on the reward score rather than

on a discrete label. Eagle (Zhao et al., 2024) is training-free: a global plus local ELO ranker over pairwise outcomes serves as the quality-prediction function. InferenceDynamics (Shi et al., 2025) factors LLM ability into separate capability and knowledge dimensions combined as a weighted sum, and predicts query-level performance from the resulting score. SEMIROUTER (Wang et al., 2026c) attacks the data-scarcity side of the same regression problem, training the predictor under sparse query-by-model annotations and integrating new candidates under limited supervision through an adaptive architecture. IPR (Feng et al., 2025a)

Method	Year	Paradigm	Signal	Repr. / Modify	Benchmark
DiRouter (Zheng et al., 2025)	2026	G	Self-awareness	Modify LLMs	7 in-domain suites
UniRoute (Jitkrittum et al., 2025)	2026	P	Gold	Probe vector	30+ unseen LLMs
Meta-Router (Zhang et al.)	2026	P	Causal	CATE estimator	HealthBench, PRBench
Dueling Feedback (Chiang et al., 2025)	2026	A	CDB	Embedding	RouterBench, MixI
RouterArena (Lu et al., 2026)	2026	I	Gold	Benchmark	RouterArena (8.4k queries, 9 domains)
MixLLM (Ding et al., 2026)	2026	A	Gold	Multi-stage	Multi-stage tasks
LLMRouterBench (Li et al., 2026)	2026	I	Gold	Benchmark	Large-scale unified routing evaluation
Workload-Router-Pool (Chen et al., 2026)	2026	I	–	Vision	Systems
RADAR (Fernandez et al., 2025)	2026	P	Gold	Item-response	Reasoning suites
StageRoute (Li and Li, 2025)	2026	A	Bandit	Online + redeploy	Streaming traces
Online LinUCB (Poon et al., 2026)	2026	A	Bandit	Linear	AIME 2024/2025 + 4 others
TRIM (Kapoor et al., 2026)	2026	A	RL	Stepwise cascade	MATH-500, AIME
Cascadia (Jiang et al., 2025)	2026	A	Confidence	Cascade serving	Production traces
Routing & Cascades (Mahmood, 2026)	2026	I	–	Game-theory	Theoretical
Mordal (He et al., 2025)	2026	Ext	Gold	VLM-backbone scorer	VLM benchmarks
Reward Routing (Wu and Lu, 2025)	2026	A/Ext	Bandit	Reward-model bandit	DPO / online alignment
Not-a-Bandit (Liu et al., 2026)	2026	A/Ext	Full information	Drafter no-regret	Speculative decoding
SEMIROUTER (Wang et al., 2026c)	2026	P	Gold (sparse)	Adaptive scorer	Multi-LLM, sparse data
DeepSieve (Guo et al., 2026)	2026	G/Ext	Confidence	LLM-as-knowledge-router	Multi-hop QA, RAG
HotelQuEST (Hadad et al., 2026)	2026	Adj	Gold	Benchmark + audit	214 hotel queries
ContextualRouter (Varangot-Reille et al., 2026)	2026	P	Gold	Past-performance retrieval	5 suites, 1% data
RouteMoA (Wang et al., 2026b)	2026	P/H	Gold	Lightweight scorer + LLM-judge refinement	MoA pools (ACL 2026)
Infrastructure – adversarial sub-area (§3.4):					
Rerouting Routers (Shafran et al., 2025)	2025	I	–	Attack (WB+BB)	MT-Bench, MMLU, GSM8K
R2A (Tang et al., 2026a)	2026	I	–	Attack (BB)	9 routers, RouterArena
RerouteGuard (Zhang et al., 2026b)	2026	I	–	Defence	–
Routing Collapse (Lai and Ye, 2026)	2026	I	–	Analysis	–
Router Fragility (Kassem et al., 2026)	2026	I	–	Audit	EACL fragility audit

Table 3: Representative routing literature inventory for 2026 plus the adversarial sub-area (Part 2 of 2; 2023–2025 in Table 2). Column conventions match Table 2. Labels **H**, **Adj**, and **Ext** follow the scope rules defined in that caption. **Additional acronyms:** *CDB*=contextual dueling bandit, *WB/BB*= white-box / black-box attack, *MixI*=MixInstruct, *VLM*=vision-language model.

regresses on per-model response quality under a user-controlled cost-tolerance parameter, allowing deployment-time movement along the Pareto frontier without retraining the predictor. Two systems sit at the boundary of this family. Smoothie estimates latent quality from generated candidate answers, so it is output aware selection rather than pre inference routing (Guha et al., 2024). RouteMoA first applies a lightweight predictive scorer and then invokes LLM judges to refine the selected agent set, so the full method is a hybrid of routing and aggregation (Wang et al., 2026b).

Query-similarity-based. A third family encodes both queries and candidate LLMs into a shared embedding space and routes by similarity matching. EmbedLLM (Zhuang et al., 2025) learns a low-dimensional vector per LLM (default 64 dims) and computes a bilinear (q, m) score over the joint embedding. RouterDC (Chen et al., 2024) learns these vectors through dual contrastive learning over both query-to-LLM pairs (pulling a query toward its winning LLM) and query-to-query pairs (pulling queries with similar winning LLMs together). UniRoute (Jitkrittum et al., 2025) represents each LLM by its accuracy vector on a fixed

probe set, which lets the router admit a new candidate by running the probe once; the paper accompanies the construction with an excess-risk bound that generalises to 30+ unseen LLMs. The ModelSAT representation introduced by Capability Instruction Tuning (Zhang et al., 2025a) learns capability dimensions from per-model performance and matches against query-specific skill demands. RadialRouter (Jin et al., 2025) introduces RadialFormer, a lightweight Transformer with a radial structure that explicitly models the query-LLMs interaction graph and trains under a cost-aware objective. ContextualRouter (Varangot-Reille et al., 2026) foregoes a learned scorer entirely: it retrieves similar past queries and reweights candidate scores through lightweight attention, remaining effective with as little as 1% historical data and generalising to new candidates without retraining.

Recommender-system-based. The smallest cluster reframes routing as a recommender-system problem, casting each candidate LLM as an item and each query as a user, letting collaborative-filtering machinery rank candidates. The IRT-based router (Song et al., 2025) fits psychometric Item Response Theory parameters (difficulty per query, dis-

crimination per query, ability per LLM), producing a low-dimensional and interpretable LLM representation that admits new candidates without retraining the router. RADAR (Fernandez et al., 2025) extends the IRT framing to reasoning-specific routing by jointly modelling per-LLM reasoning ability and per-query difficulty. LLMRec (Zhang et al., 2026c) (also referred to as “Let-the-LLM-Stick”) casts routing as a full recommender-system task, training on 52 LLMs and evaluating on 10, and inheriting collaborative-filtering machinery directly from the RecSys literature rather than borrowing only the conceptual framing.

Sub-family trade-offs. Many core methods in the four sub-families share a practical virtue: once f_{ϕ^*} is fitted, a routing decision requires one scorer pass plus a cost lookup. This is usually less expensive than invoking an LLM as the router. Where the four diverge is in how they handle the things a flat scorer cannot. Classification-based methods offer a simple entry point. Cross-entropy on “which candidate won this query” can train an off-the-shelf classifier. The price is that the classifier head is tied to a specific candidate set, so adding an LLM means retraining, and a discrete output gives you no native lever on the quality–cost parameter λ . The regression family fixes the λ problem by predicting continuous quality and cost per (q, m) , which makes $f_{\phi} - \lambda C$ a first-class object and lets observational, causal and reward distilled signals share the same regression template (Tsiourvas et al., 2026; Zhang et al.; Lu et al., 2024). The catch is that per-model performance labels at RouterBench scale are expensive to collect and remain uncommon outside large deployments. Query-similarity methods reframe the new-model problem as a probe-set construction (Jitkrittum et al., 2025) and have produced explicit excess-risk bounds for unseen candidates. They are less developed on the cost-aware side, since similarity in embedding space does not compose naturally with λC . Finally, recommender-system methods import collaborative filtering and yield interpretable IRT parameters (Song et al., 2025; Fernandez et al., 2025). They inherit the classic cold-start problem when an LLM has no interaction history, and they treat LLMs as atomic items; a fresh fine-tune of the same base model gets a new entry rather than reusing its parent’s parameters. Across all four, changes to $\mathcal{M}$ remain challenging. Probe-vector and retrieval methods reduce retrain-

ing, but robust onboarding for arbitrary candidates remains unresolved.

The Predictive core table records methods whose routing score is available before candidate answers are generated. The shared algorithmic operator is a scorer $f(q, m)$ constructed from offline evidence. Methods differ in what input representation the scorer uses (raw query embeddings, learned LLM embeddings, graph nodes, in-context demonstrations, probe vectors), and in what utility signal trains the scorer (preference, gold-standard correctness, reward distillation, or observational deployment logs). Table 4 excludes the output aware, assembly, and aggregation boundary cases identified above.

Patterns. Table 4 shows that pool size alone does not determine deployment requirements. EmbedLLM and GraphRouter use learned scorers that may require local accelerator support, while Causal LLM Routing, OmniRouter, CARROT, and Eagle retain compact decision modules. Code availability also varies within both peer-reviewed and preprint work, so the Code column should be read per method rather than as a venue-level pattern. Recent methods add observational, causal, sparse-label, retrieval, and probe-based signals to the gold supervision used by many earlier systems. The supervision signal itself has diversified well beyond gold labels. Reward distillation showed up in 2024 (Zooter), as did training-free pairwise ranking (Eagle). The per query oracle label that characterised earlier work is therefore not the only viable signal.

C.2 Adaptive Paradigm

For reference, we reproduce here the per-paper narrative previously rendered in the main body; the table that follows summarises the same set of methods on a per-method basis.

Cascade and test-time compute. Dekoninck et al. (2025) unify routing and cascading as specializations of the same Pareto-optimal policy. Earlier work that established the cascade pattern includes FrugalGPT (Chen et al., 2023) and AutoMix (Aggarwal et al., 2024), which instantiate cheap-to-expensive escalation triggered by confidence or verification. (A close cousin, Hybrid LLM (Ding et al., 2024), is sometimes grouped with cascades but is in fact a single-call binary classifier that predicts whether a small or large LLM should answer; we place it in the predictive paradigm following the authors’ own framing.) Yue et al.

Method	Year	Pool	Code	Deploy	Primary contribution
RouteLLM-SW/MF/CLS (Ong et al., 2025)	2025	2	✓	API	Preference-driven router family (Bradley-Terry similarity, matrix factorization, BERT classifier)
RouteLLM-LLM (Ong et al., 2025)	2025	2	✓	GPU	Causal-LLM scorer at full-LLM scale; same family as above but heavier
GraphRouter (Feng et al., 2025b)	2025	10+	✓	GPU	Heterogeneous-graph router with HGT message passing for inductive new-model attachment
EmbedLLM (Zhuang et al., 2025)	2025	112	✓	GPU	Learnable LLM embeddings; bilinear (q, m) scoring
RouterDC (Chen et al., 2024)	2024	7	✓	GPU	Dual contrastive learning over (query, LLM) and (query, query) pairs
UniRoute (Jitkrittum et al., 2025)	2026	30+	–	GPU	Probe-set fingerprints + cluster mapping; excess-risk bound for unseen models
One-Head (Pulishetty et al., 2025)	2025	5	–	GPU	Single cross-attention head over the candidate-LLM pool
Model-SAT (Zhang et al., 2025a)	2025	66	–	GPU	Capability-instruction-tuning encoder; learned skill profiles for matching
Causal LLM Routing (Tsiourvas et al., 2026)	2025	13	✓	API	End-to-end regret minimization via classification + softmax-weighted surrogates with cost-sensitivity conditioning
Meta-Router (Zhang et al.)	2026	2	–	API	CATE estimation combining gold-standard and biased preference signals (pairwise routing)
OmnRouter (Mei et al., 2025)	2025	12	✓	API	Joint budget and performance controls; explicit Pareto-frontier coverage
IRT-Router (Song et al., 2025)	2025	20	✓	GPU	Item-Response-Theory psychometric router; interpretable difficulty / discrimination
InferenceDynamics (Shi et al., 2025)	2025	8	–	API	Decomposes LLMs into capability and knowledge dimensions combined as a weighted score
Tryage (Hari and Thomson, 2023)	2023	22	–	API	Real-time prompt classifier dispatching to a pool of expert LLMs
Zooter (Lu et al., 2024)	2024	6	–	API	Distills reward-model signal into a lightweight router via knowledge distillation
Eagle (Zhao et al., 2024)	2024	11	–	API	Training-free ELO ranker using global plus local pairwise outcomes
OptLLM (Liu et al., 2024)	2024	12	✓	API	Multi-label classifier with uncertainty plus iterative non-dominated assignment
MoDEM (Simonds et al., 2024)	2024	5	–	API	DeBERTa-v3-large classifier routing to one of five domain expert LLMs (math/health/science/coding/other)
CoE (Jain et al., 2024)	2024	5+	–	API	Two-step routing: classify input into category, then map category to expert LLM
CARROT (Somerstep et al., 2025)	2025	14	–	API	Minimax rate-optimal cost-accuracy router; introduces SPROUT benchmark
RADAR (Fernandez et al., 2025)	2026	35	–	API	IRT-style joint modelling of per-LLM reasoning ability and per-query difficulty
SEMIROUTER (Wang et al., 2026c)	2026	–	–	API	Adaptive scorer trained under sparse query-by-model annotations; integrates new candidates with limited supervision (EACL 2026)
RadialRouter (Jin et al., 2025)	2025	11	–	GPU	RadialFormer Transformer with structured query-LLMs interaction; cost-aware evaluation (Findings EMNLP 2025)
Select-then-Route (Shah and Shridhar, 2025)	2025	22	–	API	Two-stage taxonomy classifier narrows pool, then confidence-based cascade routes within (EMNLP 2025 Industry)
BEST-Route (Ding et al., 2025)	2025	8	–	API	Joint pre-inference choice of model identity and sample count
SkewRoute (Wang et al., 2025a)	2025	2	–	API	Training-free score-skewness rule selects a smaller or larger LLM for KG-RAG
IPR (Feng et al., 2025a)	2025	6	–	API	Quality-cost regression with user-controlled tolerance; deployed on AWS workload (EMNLP 2025 Industry)
ContextualRouter (Varangot-Reille et al., 2026)	2026	5+	–	API	Past-performance retrieval with attention reweighting; effective with 1% historical data (EACL 2026 SRW)

Table 4: Predictive paradigm methods. ✓ = code released at submission time. *Deploy*: API = router runs as a service without local GPUs; GPU = router itself is GPU-bound. *Pool* = number of candidate LLMs in the largest reported experiment.

(2024) extend the cascade with a mixed Chain-of-Thought and Program-of-Thought consistency check as the escalation trigger, improving cost-efficiency on math reasoning. BEST-Route (Ding et al., 2025) jointly predicts model identity and the number of samples before generation. We therefore place it in the Predictive table with a hybrid test time compute label. MESS+ (Woiseschläger et al., 2026) provides service-level agreement guarantees over a model zoo. TRIM (Kapoor et al., 2026) pushes the cascade pattern inside a single chain-of-thought trace: a per-step router escalates only the uncertainty-critical reasoning steps to a stronger LLM under a budget. Cascadia (Jiang et al., 2025) ports cascade routing into a production serving system, scheduling queries through a chain of LLMs of increasing capability. SATER (Shen et al., 2025) unifies pre-generation routing and within-cascade exit decisions through a self-aware confidence signal, reporting over 50% cost and

80% latency reductions across three small models and six datasets.

Bandit and online learning. Wei et al. (2025) train a contextual bandit router over prompt and preference vectors under partial-feedback restrictions that match deployment. Li (2025) study a preference-conditioned multi-armed-bandit formulation. Poon et al. (2026) prove sublinear regret for an unstructured-context LinUCB variant. Pick-LLM (Sikeridis et al., 2025) uses Q-learning to adapt the routing policy on the fly to the runtime query distribution and observed reward. Chiang et al. (2025) formalize routing as a contextual dueling bandit and prove a sublinear regret bound for their FGTS.CDB algorithm. Wang et al. (2025b) introduce MixLLM, a contextual-bandit router that balances response quality, cost, and latency in a mixed-LLM deployment and supports continual learning from production traffic. StageRoute (Li and Li, 2025) attacks the streaming setting in which the candidate pool itself is periodically redeployed,

proving near-optimal regret for joint deployment-and-routing decisions.

Adaptive routing beyond the LLM pool. Online selection also appears in adjacent component-level settings. [Wu and Lu \(2025\)](#) route among multiple reward models during DPO and online alignment, casting reward-model choice as a contextual bandit. In speculative decoding, [Liu et al. \(2026\)](#) show that one verified target trajectory can provide feedback for all candidate drafters without additional target-model queries. Their HedgeSpec method is therefore a full-information no-regret online learner rather than a bandit. Both are Adaptive extensions, but they differ in feedback structure.

Confidence-deferral. A subset of methods uses an LLM’s own confidence as the deferral signal. Self-REF ([Chuang et al., 2025](#)), when used in deferral mode, treats its native confidence token as the trigger for escalation. Earlier two-tier work in the same vein includes uncertainty-aware classifier-routing methods that we group with this paradigm because they share the try-then-defer structure.

Sub-family trade-offs. What makes adaptive routing structurally different is that decisions condition on observed behaviour, not on weights or features. Cascade and bandit policies need just a candidate’s response, a verifier score, or a scalar reward. A commercial LLM endpoint can serve as an arm when the required feedback is observable. This helps explain why cascade methods such as FrugalGPT, AutoMix, and the serving-oriented Cascadia transfer naturally to industrial settings, and why StageRoute is able to fold pool redeployment into the same online-learning loop. Bandit estimators can respond to distribution shift by updating with each round of feedback. Fixed cascades do not adapt in this way and need threshold calibration when the workload changes. The same mechanics carry over outside the LLM-vs-LLM frame. Bandit-style routing among candidate reward models during alignment ([Wu and Lu, 2025](#)) receives partial feedback. By contrast, HedgeSpec evaluates all drafters from the verified trajectory and updates a full-information online learner ([Liu et al., 2026](#)). The downsides are theoretical opacity at scale and a real inference tax. Contextual-bandit regret bounds over heterogeneous LLMs are still loose, and empirical practice runs ahead of what the theory explains; thresholds and exploration constants end up hand-tuned. Cascades also pay a

structural inefficiency that grows with depth. An escalated query has already incurred an earlier candidate call whose response may then be discarded. Chaining stages increases both latency and the combinatorial space of thresholds. Bandit methods carry a complementary cold-start cost: early queries can be exploration-heavy and may underperform a comparably budgeted predictive scorer trained offline on the same amount of feedback.

The Adaptive core table records methods that share the operator “try a model, observe a feedback signal, then defer or commit.” Methods divide into three sub-families: cascading (triggered by an output-side confidence or verifier, including the within-CoT and serving-system variants TRIM and Cascadia), bandit and online learning (driven by long-run reward statistics, with StageRoute attacking the streaming-deployment regime and component-level extensions in reward-model routing ([Wu and Lu, 2025](#)) and drafter routing ([Liu et al., 2026](#))), and confidence-based deferral. SATER combines a pre-generation routing decision with a cascade and is therefore also marked Predictive in the master inventory. LLM Selector is omitted from this core table because active labelling selects training data rather than updating a deployed routing policy ([Durmazkeser et al.](#)).

Patterns. Table 5 shows that most whole-LLM adaptive methods can operate over APIs because escalation and bandit logic use observable feedback rather than model weights. The component-level Not-a-Bandit extension is different because it is integrated into a local speculative-decoding stack. Cascades appear from 2023 onward, while the reviewed bandit and online methods become more prominent in 2025 and 2026. Reported pool sizes vary, and several studies do not state a directly comparable candidate count.

C.3 Generative Paradigm

For reference, we reproduce here the per-paper narrative previously rendered in the main body; the table that follows summarises the same set of methods on a per-method basis.

Origins in adjacent tool selection. HuggingGPT prompts ChatGPT to plan a task graph and dispatch each subtask to a specialised HuggingFace model ([Shen et al., 2023](#)). Gorilla fine tunes LLaMA to retrieve and call APIs from a large pool ([Patil et al., 2024](#)). Both use generation as a control mechanism, but neither performs query level selection among

Method	Year	Pool	Code	Deploy	Primary contribution
FrugalGPT (Chen et al., 2023)	2023	12	✓	API	First confidence-based cascade; popularized the cheap-to-expensive escalation pattern
AutoMix (Aggarwal et al., 2024)	2024	5	✓	API	Self-verification cascade; cheaper LLM verifies its own answer before escalating
Unified R&C (Dekoninck et al., 2025)	2025	11	–	API	Theoretical unification of routing and cascading; Pareto-optimality proof
MESS+ (Woisetschlager et al., 2026)	2025	2	✓	API	Service-level-agreement guarantees; routes to satisfy SLA with high probability
BaRP (Wei et al., 2025)	2025	8	–	API	Trade-off-conditioned bandit; one policy covers entire Pareto frontier
LLM Bandit (Li, 2025)	2025	15+	–	API	Preference-conditioned multi-armed bandit; large LLM pool
LinUCB (Poon et al., 2026)	2026	6	–	API	Unstructured-context LinUCB; sublinear regret without context-distribution assumptions
Dueling Feedback (Chiang et al., 2025)	2026	11	–	API	Contextual Dueling Bandit formalization; sublinear regret bound for FGTS.CDB
Self-REF (deferral) (Chuang et al., 2025)	2025	3	–	GPU	Native confidence tokens drive a learned deferral cascade
MoT-Cascade (Yue et al., 2024)	2024	2	✓	API	Cascade with mixed Chain-of-Thought and Program-of-Thought consistency check as escalation trigger
PickLLM (Sikeridis et al., 2025)	2025	4	–	API	Q-learning router that adapts on the fly to the runtime query distribution and observed reward
StageRoute (Li and Li, 2025)	2026	–	–	API	Online deployment + routing for streaming queries; near-optimal regret under periodic pool redeployment
TRIM (Kapoor et al., 2026)	2026	2+	–	API	Per-step routing inside chain-of-thought; escalates only uncertainty-critical reasoning steps under a budget
Cascadia (Jiang et al., 2025)	2026	3	–	API	Cascade serving system; queries flow through a chain of LLMs of increasing capability
Reward Routing (Wu and Lu, 2025)	2026	–	–	API	Contextual bandit over candidate reward models during DPO and online alignment (component-level routing)
Not-a-Bandit (Liu et al., 2026)	2026	7	–	Local	HedgeSpec full-information no-regret drafter selection inside speculative decoding
SATER (Shen et al., 2025)	2025	3	–	API	Self-aware combined routing+cascade; >50% cost and >80% latency reduction (EMNLP 2025)
MixLLM (NAACL) (Wang et al., 2025b)	2025	–	–	API	Contextual-bandit dynamic routing balancing quality, cost, latency; supports continual learning (NAACL 2025)

Table 5: Adaptive paradigm methods. Most are API-deployable because the adaptive logic operates over decision-level feedback rather than over candidate-LLM internals. Local denotes integration inside a locally served decoding stack.

independently trained LLM candidates. They are historical adjacent work rather than members of the Generative core.

LLM-as-router. Subsequent work formalizes the operation for model selection. Zhang et al. (2026a) introduce Router-R1 using Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct policy models. It interleaves <think> reasoning with <search> calls, inserts candidate responses inside <info>, and emits the final result inside <answer>. Training uses PPO and a rule-based reward combining format, outcome, and cost components. Because the trajectory may call more than one model before producing the final answer, the full system is also marked Hybrid. Arch-Router (Tran et al., 2025) fine-tunes a 1.5B Qwen2.5 to map queries to operator-defined domain-action pairs. MasRouter (Yue et al., 2025) routes within a multi-agent system that it itself constructs through cascaded decisions over collaboration mode, role allocation, and LLM choice. Guo et al. (2025) extend the generative pattern to orchestration over models and agents. Self Route first attempts retrieval augmented generation and asks the LLM whether the retrieved evidence is sufficient. The system escalates to full context processing when the evidence is judged insufficient (Li et al., 2024). It is therefore an Adaptive strategy routing extension with a Generative self-assessment signal. The alternatives are processing

paths rather than candidate LLMs. SkewRoute instead uses score skewness computed before model generation to choose between smaller and larger language models. It is therefore listed as a core Predictive model router specialised for knowledge graph augmented generation (Wang et al., 2025a).

Distributed self-routing. DiRouter (Zheng et al., 2025) eliminates the central router entirely. Each candidate LLM is trained, through self-awareness SFT and utility-aware PPO, to decide whether to answer or forward, yielding a fully distributed routing protocol. Self-REF (Chuang et al., 2025) in its native confidence-token mode also belongs here, since each LLM emits a routing-relevant signal as part of its own generation.

Sub-family trade-offs. The distinctive thing about generative routing is that the same language-model machinery used to *describe* candidates is also used to *pick* among them. For prompt based methods, onboarding an unseen LLM can be an update to the pool encoding $\rho(\mathcal{M})$. Fine tuned and distributed methods may still require training. When a router emits and retains a structured decision trace, operators gain evidence for inspection. The trace remains a generated claim and requires separate fidelity validation before it can support an explanation. The same architecture can express multi-step protocols such as Router-R1’s reason-

ing, search, information, and answer sequence, as well as agentic orchestration over tools and sub-agents. These settings still require task-specific interfaces and training. Self Route shows that Adaptive cascading can also operate *below* the model layer through processing strategy selection, with an LLM providing the self-assessment signal. The price is computational. A typical routing decision invokes a language model. Local deployment therefore needs accelerator resources, although an externally hosted router can expose the same operation through an API. Per-query latency and cost are higher than for a small offline scorer, with the magnitude depending on the router model and serving configuration. Most model-level methods in our set require RL or SFT rather than prompt-only operation. A strategy routing extension exists in Self Route. It combines an Adaptive cascade with Generative self-assessment, while HuggingGPT is an adjacent tool controller. Neither establishes performance on the same core model routing setting as the fine tuned variants. That leaves the combination of low deployment cost and competitive quality as a clean open opportunity. There is also a paradigm-specific attack surface: prompt injection can manipulate the instruction channel used by an LLM router. Predictive and adaptive routers have different input manipulation surfaces, so robustness comparisons should separate these mechanisms.

The Generative core consists of LLM routers for model selection and distributed self routing. Tool controllers and strategy selectors are retained in the master inventory with adjacent or extension labels. Locally served core methods require accelerator resources because the routing decision is produced by a language model.

Patterns. Three observations are worth flagging. First, the core table separates model routing from its adjacent history. HuggingGPT and Gorilla treat tool control as task planning, while Router R1, Arch Router, MasRouter, Generalized Routing, and DiSRouter address choices involving LLMs or LLM based agents. Self Route extends the operation to processing strategies rather than models. Second, each model-routing entry can be read as attacking a weakness of one of the other paradigms. Arch-Router targets unseen-model onboarding (Predictive’s pain point); Router-R1 targets multi-step reasoning (which neither Predictive scoring nor Adaptive cascading expresses directly);

DiSRouter targets the central-point-of-failure of a conventional router; and MasRouter pushes routing into the multi-agent regime. Third, the GPU requirement is structural for routers that serve their own LLM but optional when a router is available through an external endpoint. Self Route demonstrates this option in a strategy selection setting. Hosted router endpoints or compact prompt based routers are therefore plausible paths off the locally hosted side of the deployment gap discussed in §4.

C.4 Infrastructure Paradigm

For reference, we reproduce here the per-paper narrative previously rendered in the main body; the table that follows summarises the same set of methods on a per-method basis.

Benchmarks. Hu et al. (2024) provide RouterBench (405k responses, 11 LLMs, 8 suites). MixInstruct (Jiang et al., 2023) and SPROUT (Somerstep et al., 2025) extend coverage. RouterEval (Huang et al., 2025b) provides more than 200 million performance records covering over 8,500 LLMs and 12 evaluations for model-level scaling analysis. LLMRouterBench (Li et al., 2026) provides more than 400K instances from 21 datasets and 33 models, together with ten routing baselines and metrics for both performance and performance-cost evaluation. RouterArena (Lu et al., 2026) provides a multi-domain (9 top-level domains, 44 categories) open platform with three empirical difficulty bands and Bloom-taxonomy cognitive levels, continuously updated. The EACL 2026 router-fragility benchmark (Kassem et al., 2026) measures router robustness across query categories, including the finding that jailbreak prompts are disproportionately routed to weaker, less safety-aligned candidates. HotelQuEST evaluates agentic hotel search under retrieval quality and tool call cost, but it does not benchmark selection among LLMs. It is therefore listed as adjacent evaluation rather than core routing Infrastructure (Hadad et al., 2026).

Adversarial robustness. Shafran et al. (2025) introduce confounder gadgets, universal token sequences that cause routers to upgrade to expensive models. Tang et al. (2026a) extend the attack to a query-limited black-box setting with universal suffixes evaluated across seven open routers and two commercial services, including OpenRouter and GPT-5-Auto. Zhang et al. (2026b) present an early dedicated defence, an embedding-based anomaly detector with adaptive thresholding. Lai and Ye

Method	Year	Pool	Code	Deploy	Primary contribution
Router-R1 (Zhang et al., 2026a)	2025	6	✓	GPU	PPO-trained 3B LLM-as-router with multi-round <think>, <search>, <info>, and <answer> actions
Arch-Router (Tran et al., 2025)	2025	6	✓	GPU	SFT 1.5B model for domain-and-action matching; intent-aligned routing
MasRouter (Yue et al., 2025)	2025	5	✓	GPU	Constructs a multi-agent system through cascaded decisions over collaboration mode, role allocation, and LLM choice
Generalized Routing (Guo et al., 2025)	2025	5+	–	GPU	Mixture of Models and Agents (MoMA) routing layer for unified orchestration over LLMs and agents
DiSRouter (Zheng et al., 2025)	2026	5	–	GPU	Distributed self-routing via PPO and self-awareness SFT; no central router
Self-REF (native) (Chuang et al., 2025)	2025	4	–	GPU	Native confidence-token decoding as the routing primitive

Table 6: Core Generative methods. Router R1 and Generalized Routing also receive a Hybrid label in the master inventory because their full systems combine routing with multi model or agent execution.

Resource	Year	Code	Type	Primary contribution
RouterBench (Hu et al., 2024)	2024	✓	Benchmark	Early widely used routing benchmark; 405k responses across 11 LLMs and 8 task suites
RouterArena (Lu et al., 2026)	2026	✓	Benchmark	Multi-domain (9 top-level, 44 categories); three empirical difficulty bands plus Bloom-taxonomy cognitive labels; continuously updated open platform
RouterEval (Huang et al., 2025b)	2025	✓	Benchmark	More than 200 million performance records from over 8,500 LLMs across 12 evaluations
LLMRouterBench (Li et al., 2026)	2026	✓	Benchmark	More than 400K instances from 21 datasets and 33 models; unified evaluation of ten routing baselines
MixInstruct (Jiang et al., 2023)	2023	✓	Benchmark	Pairwise preference benchmark seeded from LLM-Blender; widely reused
SPROUT (Somerstep et al., 2025)	2025	✓	Benchmark	Approximately 45K prompts from six suites evaluated across 14 LLMs
Rerouting Routers (Shafran et al., 2025)	2025	–	Attack	White-box and transfer attacks using query-independent confounder gadgets
R2A (Tang et al., 2026a)	2026	✓	Attack	Query-limited black-box attack evaluated on seven open routers and two commercial services
RerouteGuard (Zhang et al., 2026b)	2026	–	Defence	Embedding-based anomaly detection with adaptive thresholding
Routing Collapse (Lai and Ye, 2026)	2026	✓	Analysis	Diagnoses training-time pathology where routers collapse to a single dominant model
Router Fragility (Kassem et al., 2026)	2026	–	Audit	Demonstrates that off-the-shelf routers route jailbreaks to weakly aligned models
vLLM Semantic Router (Chen et al., 2026)	2026	✓	System	Vision paper (references open-source components: vLLM Semantic Router, OpenClaw); workload–router–pool architecture
TensorOpera Router (Stripelis et al., 2024)	2024	✓	System	Predictive multi-model interface evaluated on throughput, monetary cost, and answer performance
Routing & Cascades (Mahmood, 2026)	2026	–	Theory	Game-theoretic analysis of provider-side routing/cascades vs. user choice

Table 7: Infrastructure paradigm: benchmarks, attack and defence studies, analyses, and production systems.

(2026) analyse a training-time pathology in which routers collapse to repeatedly selecting the single strongest or most-expensive model, leaving cheaper candidates unused. Kassem et al. (2026) demonstrate that off-the-shelf routers naturally route jailbreak queries to weaker safety-trained models, opening a passive routing-jailbreak surface.

Systems. The vLLM Workload-Router-Pool vision (Chen et al., 2026) sketches the production system architecture, with the workload, router, and candidate pool as the three orthogonal design dimensions. TensorOpera Router (Stripelis et al., 2024) implements a single interface over multiple experts and evaluates predictive routing against query throughput, monetary cost, and answer performance.

Theoretical analysis. Mahmood (2026) provide a game-theoretic analysis of provider-side routing and cascading versus user choice, characterising when a provider’s optimal router actually benefits the user and when it diverges from user welfare. The result formalizes a tension that production

routing services implicitly confront and that the operating paradigms above do not directly address.

Sub-family trade-offs. Infrastructure work has an unusual multiplier effect. It need not produce a router because its outputs can apply across router families. A well-designed benchmark provides a shared metric for Predictive, Adaptive, and Generative methods. A published attack can expose a vulnerability shared by several methods. RouterBench, RouterArena, RouterEval, and LLMRouterBench play this role on the quality–cost axis; the adversarial sub-area has done the same for cost-inflation attacks; and Mahmood (2026)’s game-theoretic analysis of provider-side routing adds a normative dimension that links operator behaviour to user welfare. The vLLM Workload-Router-Pool vision (Chen et al., 2026) closes the loop on the systems side by giving academic routers a concrete deployment target to write to. The main limitation is uneven coverage. Quality–cost benchmarks are comparatively dense, while safety, multimodal capability, and router-side deployment cost are sparsely represented. The routing-fragility audit (Kassem et al., 2026) provides a safety-oriented

evaluation but does not yet constitute a broad safety benchmark suite. Defensive work is thinner still. RerouteGuard has not been evaluated against an adaptive attacker, an important requirement for robustness claims. Theory and production systems have also received less coverage and remain weakly connected to the empirical attack literature.

The Infrastructure core table contains resources that do not themselves perform routing but directly support routing research: benchmarks, attack and defence studies, training-time analyses, and production systems. We list them with a slightly different schema than the previous three subsections because the shared metadata (Pool, Deploy) is not meaningful for resources of this kind.

Patterns. Table 7 surfaces three regularities. Benchmark resources usually release data or code. Availability in the adversarial sub-area is mixed, as shown by the Code column, which limits exact cross-paper reproduction for some studies. The paradigm has grown rapidly since 2025, with new benchmarks, attacks, a defence, a systems vision, and theoretical analysis. The gap between published academic infrastructure and production services also remains wide. TensorOpera Router and the vLLM Workload-Router-Pool vision provide systems-oriented evidence, while commercial services such as OpenRouter, Martian, and Portkey expose additional operational questions that are not fully represented in the academic infrastructure literature.

D Adversarial Routing: Extended Survey

In the body, routing-decision robustness sits as one strand inside the Infrastructure paradigm (§3.4). The direct literature remains small and is complemented by adjacent work on serving side channels. This section provides a threat-model decomposition, reviews the published attacks and defences, and identifies evaluation gaps for future work.

D.1 Threat-Model Decomposition

A threat model for routing has to specify three things at once. What does the adversary want the router to do (objective)? What can the adversary observe and modify about the router and its inputs (access)? And what is the structure of the perturbations the adversary can introduce (capability)? The published routing-attack literature is already organised along these three axes, even if the papers

themselves don't state it that way. The rest of this subsection works each one in turn.

D.2 Attack Objectives

Cost-inflation attacks. Shafran et al. (2025) introduce *rerouting* attacks that append a query-independent confounder gadget and increase the probability that an easy query is sent to a strong model. They evaluate both direct access to an open router and transfer to black-box targets. The economic effect depends on the billing policy. It may increase the operator's inference cost or allow a user to obtain a stronger model than the routing policy intended. Tang et al. (2026a) study a query-limited black-box setting and evaluate seven open routers together with OpenRouter and GPT-5 Auto. Cost inflation is convenient to measure when the selected model or the resulting API cost is observable.

Quality-deflation attacks. The reverse objective sends queries to a weaker candidate. Shafran et al. (2025) evaluate this objective by reversing their router score and report substantial downgrade rates for several open routers. Zhang et al. (2026b) subsequently include quality hijacking in their threat taxonomy. These results establish the feasibility of decision-level downgrading, although the effect on users still depends on whether the weaker model produces an incorrect or misleading answer. Evaluations should therefore report both routing changes and downstream answer quality.

Jailbreak-routing attacks. The third direction routes unsafe queries to candidate models with weak safety alignment. Kassem et al. (2026) document this as a passive vulnerability rather than an active attack: routers trained on a cost-quality objective may lack an explicit safety term, so some jailbreak prompts are sent to weaker or less aligned candidates. Zhang et al. (2026b) also evaluate safety-bypass scenarios within a broader rerouting threat model. This line of evaluation remains limited because it requires both router-level measurements and candidate-specific safety outcomes.

Discussion. The three objectives concern operator cost, answer quality, and user safety, but the evidence is uneven. Cost inflation is easier to measure because model choice and API cost can provide direct signals. Quality and safety evaluations require an additional assessment of the selected model's

response and are more sensitive to the composition of the candidate pool.

D.3 Adversary Access

White-box. In the white-box setting, the attacker can inspect the router and use its score to optimise a confounder gadget. [Shafran et al. \(2025\)](#) apply this setting to open RouteLLM variants. Full score access is plausible for a locally deployed router but is unavailable for a hosted commercial service. The same paper also studies transfer from a surrogate to targets whose internal score is hidden.

Black-box. [Tang et al. \(2026a\)](#) use routing decisions collected under a limited query budget to train a hybrid ensemble surrogate. They then aggregate gradients across the surrogate components to optimise a universal adversarial suffix. This setting is relevant to services that expose the selected model. GPT-5 Auto does not expose that decision, so the paper evaluates it indirectly through response comparisons, model-style fingerprinting, and changes in reasoning time.

Transfer. Both [Shafran et al. \(2025\)](#) and [Tang et al. \(2026a\)](#) report transfer across router targets. R2A evaluates seven open routers and two commercial systems. Direct attack success rates are reported for targets whose routing decisions can be observed, whereas GPT-5 Auto is assessed through indirect evidence. These results show that universal routing triggers can transfer, but they do not establish that all router architectures share the same failure mechanism.

D.4 Adversary Capability

Query-specific perturbations. A perturbation is computed per query. This is the standard adversarial-example setting in image classification and is applicable to routing, although it has not been the primary focus of the published literature.

Universal triggers. [Shafran et al. \(2025\)](#) introduce confounder gadgets, universal token sequences that can be appended to any query to flip the router’s decision toward a target model. Universality makes the attack practically dangerous because the attacker incurs the optimisation cost only once and can then weaponise the gadget at scale. Both Rerouting Routers and R2A primarily study universal gadgets.

Low-perplexity variants. A naive perplexity filter can detect random adversarial tokens because

they have very low natural-language likelihood. [Shafran et al. \(2025\)](#) show that a low-perplexity variant of the gadget optimisation, regularised by the language-model likelihood of the gadget tokens, evades perplexity filtering without much loss in attack success rate. This argues against single-statistic input filtering as a defence.

D.5 Defence Taxonomy

Input filtering. The simplest defence applies a heuristic filter to the query before it reaches the router. [Shafran et al. \(2025\)](#) empirically evaluate perplexity filtering and show that an attacker can optimise a low-perplexity gadget that evades it. They discuss LLM-based filtering and paraphrasing as possible higher-cost defences but do not provide the same level of empirical evaluation for those alternatives. These proposals therefore remain distinct from the evaluated perplexity baseline.

Workload anomaly detection. A defender with access to a stream of routing decisions over many queries can monitor the distribution and flag distributional shifts. Workload anomaly detection can operate on aggregate statistics rather than individual inputs. Its effectiveness will depend on the baseline window, attack prevalence, and rate at which adversarial queries arrive. These tradeoffs have not yet been characterised consistently for LLM routers.

Embedding-based detection. RerouteGuard ([Zhang et al., 2026b](#)) is an early defence aimed specifically at routing attacks. It learns a model of the normal routing-input distribution in embedding space and flags inputs that are statistical outliers, using an adaptive threshold. The paper reports more than 99% detection accuracy across three attack settings and four benchmarks. It does not evaluate an adaptive attacker that jointly optimises the suffix against the detector. This remains an important open evaluation question (§D.7).

Adversarial training. Adversarial training mixes adversarial examples into the training set. We did not identify a routing study that evaluates this approach. One possible procedure is to generate routing gadgets via the [Shafran et al. \(2025\)](#) optimizer, append them to clean queries, and add the augmented examples to the router’s training set. Whether the resulting routers retain clean accuracy and generalise to unseen attack families are open empirical questions.

Certified defences. We did not identify a certified-radius defence for routing decisions. The relevant analogy is randomized smoothing for classifier robustness: a smoothed router would be defined as the majority vote of the base router over a noise distribution applied to the input, with a provable robustness radius derived from the smoothing distribution. The transfer to discrete token inputs is non-trivial but has analogues in the certified-robustness literature for text classifiers; importing those techniques to routing is one of the open directions identified in §5.

D.6 Routing-Decision Robustness vs. LLM Robustness

It is worth distinguishing routing-decision robustness from the standard robustness of the LLMs being routed. The literatures overlap but address different layers: an LLM-jailbreak attack manipulates a model’s output, while a routing attack manipulates which model is queried. The two attacks compose. A successful routing attack can route a jailbreak payload to a less-aligned model, and a successful LLM jailbreak after that point produces unsafe content; this composition is what Kassem et al. (2026) document as the passive jailbreak-routing vulnerability. The implication is that defences at one layer (e.g., alignment of individual LLMs) do not protect against attacks at the other layer (e.g., routing to a worse-aligned LLM).

A related but orthogonal class of risk is the *side-channel* attack on routed LLM services. InputSnatch (Zheng et al., 2024) demonstrates that timing differences in shared LLM serving infrastructure leak the content of other users’ prompts; Whisper Leak (McDonald and Or, 2025) shows the same in encrypted-traffic metadata of commercial LLM APIs. Neither targets the router directly. They are included as adjacent serving risks because a heterogeneous routed service can expose candidate-dependent timing and length patterns. Establishing whether routing increases these leakages requires a dedicated evaluation.

D.7 Open Evaluation Gaps

The following gaps recur across the literature and should be treated as priorities for the next round of routing-robustness research.

(G1) Adaptive-attack evaluation remains missing. This is the central methodological problem. The published routing defence studies reviewed

here, including RerouteGuard, uses attacks that were designed before the defence existed. Adaptive evaluation requires that the attacker be given full knowledge of the defence and re-optimize the gadget against it; historical experience in adversarial machine learning suggests that defences appearing strong under non-adaptive evaluation typically collapse under adaptive evaluation. Until adaptive results are reported, claimed defences should be read as preliminary.

(G2) Certified guarantees remain absent. The routing defences reviewed here are empirical. A certified defence would provide a provable guarantee that no perturbation within a bounded radius can flip the routing decision. The closest analogue in adversarial NLP is randomized smoothing applied to text classifiers; transferring those techniques to routing is an open methodological problem.

(G3) Cross-router defence transfer remains unmeasured. Tang et al. (2026a) report cross-router transfer of attacks. The analogous question for defences, whether a defence trained or tuned for one router family generalises to another, has not been measured in the reviewed literature. The practical implication is acute: a defender who can train a single defence and deploy it across a heterogeneous router pool has very different costs than one who must redo the work for every architecture.

(G4) A shared routing-robustness benchmark is missing. Current evaluations construct attack benchmarks ad hoc by adding gadgets to queries from RouterBench, MMLU, or GSM8K. The lack of a standard benchmark makes cross-paper comparison difficult and encourages selection bias in the choice of evaluation set. A recommended composition for such a benchmark would include both cost-inflation and (when feasible) quality-deflation variants, spanning multiple model pools, with adaptive attack variants for any defence it evaluates.

(G5) Architecture-level comparisons are missing. Routing architectures vary widely in their attack surface. Score-based classifiers expose explicit gradients; LLM-as-router architectures inherit prompt-injection vulnerabilities; distributed self-routing introduces a further failure mode, namely a malicious candidate LLM that misreports its own competence. We did not identify a study that compares these architectures on a common attack benchmark. Such

a comparison is a prerequisite for principled architecture choice in robustness-sensitive deployments.

(G6) Deployment costs are under-reported.

Defences have widely varying inference cost. Perplexity filtering adds a single small-LLM call per query; LLM-based filtering adds a full LLM call; paraphrasing adds a paraphrase generation. The reviewed literature has not yet mapped the joint trade-off among cost, clean quality, and robustness. Reporting the inference-cost overhead of any proposed defence should be standard practice.

E Routing Benchmarks: Comparative Inventory

The choice of evaluation benchmark substantially shapes which routing methods appear competitive. Table 8 compares seven benchmark and audit resources. They differ in scale, coverage, label type, and the deployment patterns they reward. This appendix gives a side-by-side comparison (§E.1), a per-benchmark discussion that records what each benchmark is best for and what it omits (§E.2), an explicit account of coverage gaps that no current benchmark addresses (§E.3), and a recommended evaluation suite for new routing methods (§E.4).

E.1 Comparison Table

Patterns. Table 8 highlights uneven coverage and non-comparable notions of scale. Most entries target clean quality and cost evaluation. The EACL fragility audit adds safety-oriented queries, but the reviewed resources do not provide a shared adaptive attack protocol. Candidate coverage ranges from eleven models in RouterBench to 33 in LLMRouterBench and 42 in RouterArena, while RouterEval aggregates records from more than 8,500 models. These figures measure different units and should not be compared as a single growth series. Recent resources broaden coverage in distinct ways: SPROUT emphasises reasoning, RouterArena adds difficulty and cognitive labels, LLMRouterBench standardises comparisons across a larger recent pool, and the EACL audit studies fragility.

E.2 Per-Benchmark Discussion

MixInstruct (Jiang et al., 2023). Released alongside LLM-Blender, MixInstruct is the oldest routing-relevant benchmark and remains in active use because of its scale (110K total samples; 5K labelled test split) and its pairwise-preference

format. Eleven candidate LLMs (Alpaca, Vicuna, Dolly-V2, StableLM, OpenAssistant, Koala, Baize, Flan-T5, ChatGLM, MOSS, and Mosaic-MPT) generate responses to each query, with oracle pairwise preference labels obtained from ChatGPT pairwise comparisons over the 5K test split (PairRanker itself is then *trained on* these labels). The benchmark is well-suited for training preference-driven routers and pairwise rankers. Its limitations are that the candidate pool reflects the 2023 generation of models, the task distribution skews toward instruction-following rather than reasoning or code, and the preference labels themselves are produced by an LLM judge rather than by humans.

RouterBench (Hu et al., 2024). RouterBench is widely used for quality and cost evaluation in routing. The release contains roughly 35K unique queries spanning eight task families (MMLU, ARC, HumanEval, GSM8K, MT-Bench, RAG QA, and others), with pre-computed responses and correctness labels for eleven candidate LLMs. The pre-computation has been useful for reproduction because researchers can evaluate new routing algorithms by re-routing over the released label table without calling any LLM. This property helps explain its frequent use in subsequent routing studies.

SPROUT (Somerstep et al., 2025). SPROUT was introduced with CARROT. It contains approximately 45K prompts from six suites and responses from fourteen LLMs. GPQA, MuSR, MMLU-Pro, MATH, OpenHermes, and RAGBench broaden evaluation toward science, reasoning, instruction following, and retrieval augmented generation. It is useful as a complementary test to RouterBench when a study needs more demanding reasoning tasks.

RouterEval (Huang et al., 2025b). RouterEval, an EMNLP 2025 benchmark, is designed for studying model-level scaling in routing. It contains more than 200 million performance records covering over 8,500 LLMs and 12 evaluations. Its scale is not directly comparable with query-response tables: the principal unit is a model-evaluation performance record rather than a fresh response from every candidate in one fixed pool. This makes RouterEval useful for analysing large candidate populations, while RouterBench-style resources remain more direct for reproducing per-query routing decisions over a fixed set of responses.

Benchmark	Year	Scale	LLMs	Suites	Open	Adv.	Domain	Primary use case
MixInstruct (Jiang et al., 2023)	2023	100K+	11	4	✓	–	general	Pairwise preference training; foundation for ranker-style routers
RouterBench (Hu et al., 2024)	2024	35K	11	8	✓	–	general	Standard quality-cost benchmark; precomputed responses enable zero-cost reproduction
SPROUT (Somerstep et al., 2025)	2025	~45K	14	6	✓	–	reasoning	Advanced-reasoning coverage across GPQA, MuSR, MMLU-Pro, MATH, OpenHermes, and RAGBench
RouterEval (Huang et al., 2025b)	2025	200M+ rec.	8.5K+	12	✓	–	general	Performance records for model-level scaling analysis
LLMRouterBench (Li et al., 2026)	2026	400K+ inst.	33	21	✓	–	multi-domain	Unified comparison of ten baselines under performance and performance-cost metrics
RouterArena (Lu et al., 2026)	2026	8.4K	42	9	✓	–	multi-domain	Continuously updated open platform; three empirical difficulty bands + Bloom-taxonomy cognitive levels
EACL fragility audit (Kassem et al., 2026)	2026	–	–	–	–	jailbreak	safety	Audit benchmark for router fragility across query categories; jailbreak prompts disproportionately routed to weaker candidates

Table 8: Routing benchmarks at a glance. **Scale** reports queries, instances, or performance records according to the unit released by each resource. **LLMs** = candidate-pool size in the largest released configuration; **Suites** = number of distinct task families covered; **Open** = data and labels released under an open license; **Adv.** = whether the benchmark includes adversarial perturbations or attack queries; **Domain** = intended task family. “–” = not specified or not yet publicized.

LLMRouterBench (Li et al., 2026). LLMRouterBench contains more than 400K instances from 21 datasets and 33 models. Its unified framework integrates ten representative routing baselines and reports both performance-oriented and performance-cost metrics. The benchmark is especially useful for controlled comparisons because the candidate ensemble, datasets, and metric implementation are held constant across methods. Its analysis also shows diminishing returns from enlarging a model pool without careful candidate curation.

RouterArena (Lu et al., 2026). RouterArena was designed explicitly as an open evaluation platform with three empirical difficulty bands and Bloom-taxonomy cognitive levels, 9 top-level domains and 44 categories, and a continuously updated model pool. Its design parallels Chatbot Arena for direct LLM evaluation and aims to close the gap between static benchmarks (RouterBench) and the rapidly moving LLM ecosystem. RouterArena has already been adopted as the out-of-distribution test set in adversarial work, notably Tang et al. (2026a); this is meaningful evidence that the routing community is beginning to use it for difficulty-stratified evaluation.

Router fragility audit (Kassem et al., 2026). Kassem et al. (2026) analyse the fragility of LLM routing capabilities under perturbations and adversarial conditions, covering both routing-decision instability under input perturbations and the related

question of whether jailbreak prompts can land on weakly aligned candidates. Among the resources in Table 8, it directly evaluates a routing robustness property of an off-the-shelf router. It therefore occupies a niche of its own and should be used in addition to a quality-cost benchmark rather than as a replacement.

E.3 Coverage Gaps

The collected benchmarks under-cover four aspects of the field that we consider important for the next round of routing research.

Adversarial-robustness benchmarks. Beyond the EACL audit benchmark, we did not identify a published benchmark with a standardised set of attack queries. Adversarial routing papers construct attack benchmarks by appending gadgets to queries from RouterBench, MMLU, or GSM8K, but no two papers use exactly the same configuration, which complicates comparison. A benchmark that fixes both the clean-query distribution and a standard attack-construction protocol would let attack and defence methods report directly comparable numbers.

Multimodal-routing benchmarks. The resources in Table 8 are text-only. We did not identify a standard evaluation for vision-language or audio-language routing. The closest analogue is in the broader multimodal-evaluation literature, but those benchmarks are not designed with multi-LLM routing in mind.

Deployment metrics. Most benchmarks emphasise quality and per-call cost. LLMRouterBench supports latency-aware analysis, but shared measures of throughput, cache behaviour, and rate-limit sensitivity remain limited. Consequently, results on RouterBench may not predict production behaviour when serving constraints are material. A benchmark with realistic deployment metrics would enable comparison on these additional dimensions.

Long-tail and emergent-task benchmarks. The reviewed benchmarks primarily draw queries from established task distributions. Emergent tasks that do not match a training distribution are under-represented even though they directly test generalisation. RouterArena’s continuous-update mechanism is a useful step, but it does not isolate emergent queries from incremental drift.

E.4 Recommended Evaluation Suite for New Methods

We recommend that a new routing method evaluate on at least two benchmarks. RouterBench supports comparison with a large body of prior work at low reproduction cost. LLMRouterBench supports a unified comparison across a larger and newer model pool. RouterArena tests distribution shifts, while SPROUT covers advanced reasoning tasks. Methods that claim unseen-model generalisation should additionally report on RouterArena, whose model pool changes over time and so directly exercises the claim. Methods that target safety or fairness should report on the EACL benchmark in addition to a quality-cost benchmark. Adversarial-robustness claims should follow the attack-construction protocol of [Shafran et al. \(2025\)](#) or [Tang et al. \(2026a\)](#) on top of RouterBench, and ideally report adaptive attack results (§D.7).

F Extended Research Roadmap

This appendix expands the five cross-paradigm research directions summarised at the end of §5. We retain the per-paradigm frontiers in the main body and develop the *cross-paradigm* directions here, since each requires coordinated effort across more than one paradigm and therefore sits between rather than within the four paradigms above.

F.1 Joint Routing and Query Reformulation

Most core model selectors reviewed here hold the query wording fixed after receiving it. Strategy

and knowledge-source extensions can change the processing path, but they do not jointly optimise a model-specific reformulation. Co-optimising model choice and query wording is therefore a distinct open direction. A natural setup is a generative router that scores not only (q, m) but (q', m) over reformulations q' of q , expanding the action space of Eq. (6) from $\{\langle \text{call } m_i \rangle\}$ to $\{\langle \text{call } m_i \text{ with } q' \rangle\}$. Two challenges follow. Empirically, the action space blows up to $|Q'| \times K$. Theoretically, it is not obvious whether the standard regret bounds for routing extend cleanly to the joint reformulation-and-routing problem.

F.2 Routing Beyond LLM Candidates

The four operations provide useful comparison points beyond whole LLM routing, but these extensions remain outside the core inventory definition. Mordal searches and assembles vision and language backbones for a target task rather than routing queries among LLMs ([He et al., 2025](#)). Reward model routing ([Wu and Lu, 2025](#)) routes among candidate reward models during alignment and follows the Adaptive bandit operation. Drafter routing in speculative decoding ([Liu et al., 2026](#)) routes among draft models inside a speculative decoding loop. It is Adaptive but uses full-information feedback for all drafters rather than bandit feedback. DeepSieve uses a Generative controller to route subquestions to heterogeneous knowledge sources inside an agentic RAG pipeline ([Guo et al., 2026](#)). These mappings justify extension labels, but they do not turn model assembly, reward model selection, drafter selection, or retrieval into core multi-LLM routing. The remaining open question is which component-level decision spaces are still under-explored, adapter selection in modular fine-tuning, safety-guardrail routing under attack, and multi-source retrieval routing beyond what DeepSieve already covers, and whether the four operations remain useful comparison points once those spaces are systematically surveyed.

F.3 Cross-Modal Routing

Core LLM-pool methods in the inventory predominantly assume text queries. Routing across vision-language and audio-language LLMs, noted briefly by [Moslem and Kelleher \(2026\)](#), remains sparsely studied as a per-query inference problem. The Mordal line above addresses model *assembly* for multimodal tasks rather than per-query multimodal routing. The predictive paradigm offers

a starting point, while adaptive and generative routers that condition on the modality composition of each query remain open directions. A concrete formulation could replace the scalar quality $Q(q, m)$ in Eq. (1) with a per-modality quality vector $\vec{Q}(q, m) = (Q_{\text{text}}, Q_{\text{vis}}, Q_{\text{audio}})$ aggregated under task-dependent weights, and admit candidate models whose own modality coverage is partial.

F.4 Privacy-Preserving Routing

A router can add a privacy surface beyond the selected model itself. A routing decision sends some representation of q to the router, and it may further reveal which m_k was chosen to a third-party provider. Future work should look at router designs that operate on encrypted or locally embedded queries, that disclose only the minimal meta-data needed (a domain tag, say, rather than raw text), and that satisfy a formal (ϵ, δ) privacy budget at the routing layer itself. The problem gets sharper for Generative routers, where the router is an LLM that consumes the entire query in plaintext; there are fewer architectural levers for partial-input operation than for a predictive scorer built on top of dense embeddings. A natural place to start is the predictive scorer itself. Extend $f_\phi(q, m)$ of Eq. (3) to consume a privatised embedding $\tilde{q} = \pi(q)$, where π is a local mechanism that satisfies a query-level differential-privacy guarantee. Then analyse how that privacy budget propagates into the routing-decision distribution, and ultimately into downstream LLM responses.

F.5 Multi-Objective Routing

Most surveyed methods optimise scalar quality minus scalar cost (Eq. (1)). Production deployments, on the other hand, routinely juggle several axes at once: per-token cost, end-to-end latency, peak GPU memory, safety and alignment compliance, fairness across user demographics, even the carbon footprint of inference. A truly multi-objective router would compute the Pareto front of $(\bar{Q}, \bar{C}, \bar{L}, \bar{S}, \dots)$ at decision time and pick from it under operator-supplied preferences over multiple axes. Existing work already exposes scalarised controls for selected pairs of objectives. CARROT, OmniRouter, BaRP, and IPR cover forms of quality and cost control (Somerstep et al., 2025; Mei et al., 2025; Wei et al., 2025; Feng et al., 2025a), while SLA-oriented methods add latency constraints and the EACL fragility audit studies safety (Kassem et al., 2026). The open gap is joint treatment of sev-

eral axes and an interface that exposes a front rather than one scalarised choice. We did not identify a core router that returns such a multi-axis Pareto front at decision time. Three challenges block the way. The first is specifying a preference language tractable enough that operators can use it without hand-tuning a Lagrangian weight per axis. The second is extending the regret-bound machinery of §C.2 from scalar to vector reward, which is a real but not impossible jump. The third is benchmark design: routers need to be scored on the quality of the *front* they return, not the single point.

G Structural Index

This appendix renders the section level skeleton of the survey. Publication statistics appear in Figure 3. The structural index in Figure 4 is a navigation map derived from the body headings in §3 and §5.

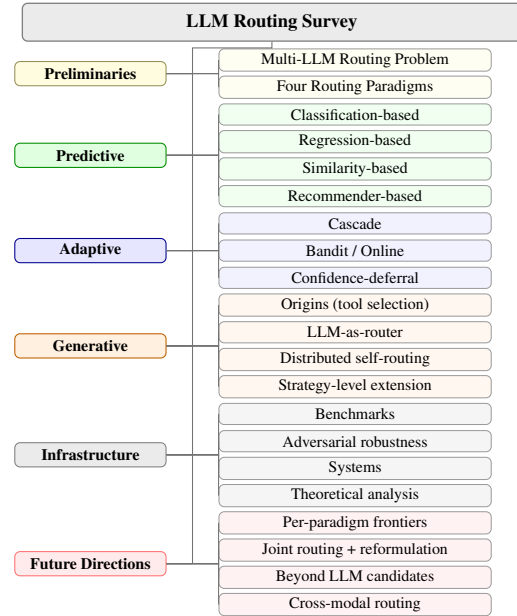


Figure 4: Structural index of this survey. The root represents the survey itself; the six branches correspond to §2 Preliminaries, the four paradigm sections of §3, and §5 Future Directions. Each leaf identifies a topic developed in the body. The Generative branch includes historical origins and strategy extensions for navigation, although the master inventory marks them as adjacent or extended scope.

What the structural index shows. Figure 4 is a compact content tree. A reader can use it to locate a topic such as recommender based routing without reading the appendix tables first. It is not a scope classifier. The authoritative scope and paradigm labels appear in Tables 2 and 3.

Design rationale. The six branches mirror Preliminaries, the four paradigm sections, and Future Directions. The paradigm branches inherit the colour scheme of Figure 2. Equal leaf width reflects topic structure rather than paper volume. Figure 3 records the original frozen counts, while the master tables provide the revised paper level classification.

Negative space. Some adjacent work appears in the tree because it supplies history or motivates an extension. Tool selection origins and strategy selection under the Generative branch are the clearest examples. They remain outside the core class. Other adjacent areas are grouped in §3.5. Directions that cut across several paradigms are developed in Appendix §F rather than being forced under one branch.