Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dimension-Direct Routing: Achieving 25% Depth Improvement in Multi- Model LLM Systems via Explicit Capability Factorization

Tao Rui

rodneyrui@gmail.com

Independent Researcher https://orcid.org/0009-0004-3898-4655

Research Article

Keywords: LLM routing, multi-model orchestration, dimension-direct routing, capability factorization, model selection, LLM-as-Judge, mixture of experts, semantic accumulation bias, knowledge-intensive tasks, AI managing AI

Posted Date: April 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9317311/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Dimension-Direct Routing: Achieving 25% Depth Improvement in Multi-Model LLM Systems via Explicit Capability Factorization

Pre-print. Experiments conducted March 24 – 29, 2026. The initial draft of this paper was generated by the eVoiceClaw Desktop system itself —a multi-model workflow involving DeepSeek, Zhipu GLM, Moonshot Kimi, and other models in the pool collaboratively produced the first version through the same dimension-direct routing and workflow orchestration described herein. The draft was subsequently reviewed, corrected, and restructured by the human author. Details of the drafting process are provided in Appendix B.

Rui Tao Independent Researcher rodneyrui@gmail.com 2026-04-02

Abstract Large Language Models (LLMs) exhibit distinct capabilities across different knowledge domains, yet single-model deployments struggle with knowledge-intensive tasks requiring cross-domain reasoning. We present eVoiceClaw Desktop, a multi-model orchestration system originally developed as the backend for a voice-based AI assistant, which operationalizes an "AI managing AI"paradigm: instead of humans manually selecting models, the system dynamically routes complex queries to specialized models through a dimension-direct routing algorithm. The system underwent four major configuration iterations (V1–V4), culminating in the final configuration V5 that addresses critical challenges in cross-domain task allocation and semantic accumulation bias. V5 achieves a 98% workflow trigger rate across 50 benchmark questions in Chinese, leveraging 12 models to generate responses averaging 8,533 characters with balanced diversity (top model $\leqslant$16% usage share). We evaluate response quality using LLM-as-Judge (Claude Opus 4.6) across four dimensions: factual accuracy, completeness, depth, and structure. Compared to single-model baselines, V5 achieves a 14.3% overall quality improvement, with depth of analysis improving by 25.9%, at the expense of approximately 9$\times$ higher latency and cost. The system demonstrates robust performance on interdisciplinary tasks (e.g., legal-technical cross-domain questions), where earlier kNN-based versions failed due to anchor sparsity. As a meta-demonstration, the initial draft of this paper was itself generated by the system (Appendix B). Source code is available at https://github.com/rodneyrui/evoiceclaw-desktop-v3.

1. Introduction 1.1 Motivation The proliferation of specialized Large Language Models—each excelling in distinct domains such as legal reasoning, mathematical computation, creative writing, or technical analysis—has created a fragmented landscape where no single model dominates across all task categories. Knowledge-intensive tasks, particularly those spanning multiple domains (e.g., patent analysis requiring legal and technical expertise), expose the limitations of monolithic LLM deployments. Static model selection strategies fail to capture the nuanced capability matrices of modern LLMs, while naive routing mechanisms suffer from semantic drift and capability misalignment. 1.2 Problem Statement We identify three fundamental challenges in multi-model orchestration:
2. Cross-Domain Routing Instability: Interdisciplinary queries fall into sparse regions of embedding space, causing kNN-based predictors to return unreliable similarity scores.
3. Semantic Accumulation Bias: Continuous operation leads to embedding cache saturation with homogenous vectors, triggering model convergence where the system over-selects a single high-performing model (observed up to 78% usage concentration).
4. Quality-Latency-Cost Tradeoff: Comprehensive multi-model workflows generate more content but incur significant latency and cost overhead, requiring explicit measurement and reporting. 1.3 Contributions eVoiceClaw Desktop addresses these challenges through an iterative optimization methodology validated across four major configurations:
5. Dimension-Direct Routing: We replace embedding-space kNN with explicit dimension prediction (`predict_dimensions()`) and matrix-based model selection (`select_models_by_dimensions()`), achieving 100% workflow trigger rate on cross-domain tasks.
6. Dynamic Model Diversity: The final configuration (V5) employs two complementary mechanisms

to prevent model convergence: (a) intra-workflow penalization —previously-used models receive a 40% score reduction, ensuring each step selects a different model; (b) per-query state reset —the health tracker and embedding cache are cleared before each workflow, eliminating cross-query bias (observed in V2 with 78% single-model concentration).

7. Comprehensive Benchmark: We establish three baselines (single-model, random routing, kNN-only) and validate our system on 50 knowledge-intensive questions across 15 capability dimensions.
8. Self-Authored Validation: The initial draft of this paper was generated by the eVoiceClaw Desktop system itself, using the same multi-model workflow described herein. This serves as a meta-demonstration of the system's capability on a complex, long-form knowledge synthesis task (see Appendix B for details).
9. Transparent Reporting: We explicitly report computational costs and acknowledge the cost-quality tradeoffs, enabling fair comparison with single-model baselines.

---

10. Related Work 2.1 LLM Routing and Selection Recent work in LLM routing spans cost-aware and quality-aware approaches. FrugalGPT [2] cascades models by cost, routing queries to cheaper models first and escalating only when confidence is low. RouteLLM [6] trains a router on preference data to dynamically choose between a strong and a weak model, achieving over $2\times$ cost savings with minimal quality loss. Hybrid LLM [7] formulates routing as a sequential decision problem, learning when to defer from a smaller model to a larger one. These systems typically operate on a binary or small set of models; our work extends routing to a heterogeneous pool of 12 models across 15 capability dimensions.

2.2 Multi-Model Ensembling and Orchestration LLM-Blender [1] ensembles multiple model outputs through pairwise ranking and generative fusion, but requires running all models for every query. Mixture-of-Agents (MoA) [8] arranges models in layers where each layer's models refine outputs from the previous layer, achieving state-of-the-art quality at the cost of sequential latency. AutoMix [9] uses a smaller model to verify whether a larger model's output is needed, optimizing the cost-quality tradeoff. Our approach differs by decomposing queries into capability dimensions before selecting models, enabling parallel execution of specialized sub-tasks rather than sequential refinement.

2.3 Mixture-of-Experts at the Model Level The Mixture-of-Experts (MoE) paradigm [4] has been applied at the token level within single models (e.g., Mixtral, DeepSeek-MoE). Our system applies MoE principles at the model level: the dimension predictor acts as a gating function that routes entire queries (rather than tokens) to specialized models. This model-level MoE enables integration of heterogeneous architectures from different providers without requiring shared parameters or joint training.

2.4 LLM-as-Judge Evaluation Zheng et al. [5] established LLM-as-Judge as a scalable alternative to human evaluation, demonstrating high agreement between strong LLM judges and human annotators on MT-Bench. Subsequent work has identified position bias and verbosity bias as key limitations [10]. We adopt pointwise scoring (rather than pairwise comparison) to mitigate position bias, and instruct the judge to penalize length-without-substance to reduce verbosity bias.

---

3. System Design 3.1 Architecture Overview The system adopts a three-stage pipeline:
4. Dimension Prediction: Decomposes user queries into relevant capability dimensions using a heuristic mapping between query embeddings and dimension clusters.
5. Model Selection: Queries a pre-computed model capability matrix to select top performers per dimension.
6. Workflow Orchestration: Executes multi-step workflows with intra-workflow penalization (40% score decay per step) and result aggregation.

Figure 1 illustrates the system architecture (see supplementary materials for detailed data flow diagram).

3.2 Implementation Details

3.2.1 Dimension Prediction Module

The `predict_dimensions()` function implements a lightweight heuristic approach: 1. During system initialization, anchor points are clustered by 15 predefined capability dimensions (each anchor with label $\geqslant 3$ belongs to that dimension's cluster). 2. For a given query, we compute its embedding and calculate cosine similarity against each dimension's anchor cluster. 3. Dimensions with average similarity exceeding 50% of the maximum similarity are marked as relevant. 4. The function returns a list of dimension names (e.g., ["knowledge_legal", "reasoning"]) rather than continuous scores.

This approach was refined through experimental iteration (V1→V2→V3→V4→V5). V2 introduced intra-workflow penalization (40% score decay) to address model convergence within a single run; V4 added per-query state reset (clearing health tracker and embedding cache before each workflow) to prevent convergence across queries, the primary mechanism used in V5.

3.2.2 Model Capability Matrix

The 15 dimensions cover diverse capabilities: - Knowledge domains: legal, tech, business, medical, academic, science, finance - Reasoning types: logical, mathematical, causal - Task characteristics: creative, analytical, coding, writing, research

Model scores were assigned based on: 1. Published benchmark results (MMLU, HumanEval, etc.) 2. Internal testing on domain-specific prompts 3. Provider-reported capabilities

Scores are static but configurable, allowing addition of new models or dimension reweighting. An example excerpt is provided in Table A1 (Appendix A); the full 12-model $\times$ 15-dimension matrix is available in the GitHub repository.

3.2.3 Model Selection Strategy

`select_models_by_dimensions()` performs: 1. Filter models by relevant dimensions (average score across relevant dims) 2. Sort by composite score with optional cost/speed factors 3. Apply penalization: multiply scores of used models by 0.6 (40% reduction, intra-workflow only) 4. Apply noise: $\pm 3\%$ random perturbation to prevent deterministic selection

During workflow execution, cost and speed dimensions are ignored (ignore_cost=true) to prioritize capability over efficiency.

3.3 Workflow Orchestration Engine The orchestrator generates sequential steps where each model contributes its specialized capability. A health tracker monitors model performance history, while an embedding cache manager prevents saturation through periodic rebalancing.

---

4. Experimental Evaluation 4.1 Benchmark Setup We curated 50 knowledge-intensive questions (Q1 – Q50) in Chinese, spanning 10 categories: technical reasoning, technical-business, legal-technical, business analytics, coding-analysis, medical-technical, cross-domain synthesis, writing-research, multi-step tasks, and deep analysis. Each question requires multi-step reasoning and cross-domain knowledge. All questions and model responses are in Chinese; generalizability to other languages remains to be validated (see Section 5.7).

Experiments ran on a desktop environment with access to 12 commercially available models via unified API endpoints. The model pool consists of models from multiple providers, selected for API accessibility and cost efficiency: DeepSeek (deepseek-chat, deepseek-reasoner), Zhipu AI (glm-4-air, glm-4-flash, glm-5), Moonshot (kimi-k2.5), Alibaba Cloud (qwen-turbo, qwen-plus, qwen-long), MiniMax (MiniMax-M2.1, MiniMax-M2.5), and Baichuan (Baichuan3-Turbo). The pool does not include models from Anthropic, OpenAI, or Google due to API access constraints in the experimental environment; this is discussed as a limitation in Section 5.7.

4.2 Baseline Comparison Table 1 summarizes three baseline strategies. An initial run (March 24) covered Q1–Q20 for Groups B and C; a supplementary run (March 26) extended coverage to all 50 questions. Group A covers all 50 questions from the initial run.

Group	Strategy	Questions	Avg. Chars	Avg. Latency	Est. Cost/Query
A	Single Model (deepseek-chat)	50	6,084	60.9s	~$0.10
B	Random Routing	50	7,483	87.1s	~$0.12
C	kNN Routing (No Workflow)	50	1,661	30.2s	~$0.05

Cost estimates are based on API pricing at time of experiment (March 2026): deepseek-chat $0.14/1M tokens, glm-4-air $0.10/1M tokens, kimi-k2.5 $0.015/1K tokens. Estimates include input + output tokens per query.

Key Observations: - Group C's extremely low output (1,661 chars) reflects tool-result overshadowing: the kNN router selected models that invoked tools (e.g., `consult_expert`, `write_file`) instead of generating direct answers, with useful content buried in tool call metadata rather than the response text. - Group B's higher output (7,483 chars) compared to Group A (6,084 chars) is an artifact of random model selection occasionally routing to verbose models; this does not indicate higher quality (see Section 5.5 for quality evaluation). - Groups B and C do not trigger workflow orchestration, making them single-call baselines comparable to Group A.

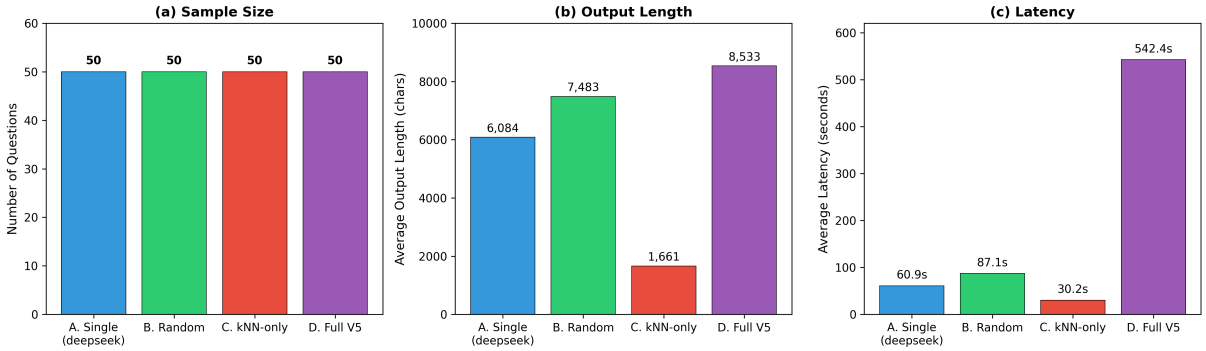

Figure 1: Baseline Strategy Comparison —sample size, output length, and latency across four experimental groups.

Baseline Fairness Caveat: Direct comparison between single-call baselines (Groups A–C) and the full V5 system (Section 4.4) involves structural asymmetry: V5 executes an average of 4.8 API calls per query through its workflow pipeline. We report both output metrics and estimated costs to enable fair assessment of the quality-cost tradeoff.

4.3 Ablation Study: Routing Algorithm Evolution We conducted a series of ablation experiments to isolate the contribution of each routing component. The system evolved through four major configurations:

Configuration 1: Global kNN (V1). The initial system used global top-5 nearest neighbor matching across all 2,026 anchor points to predict a 15-dimensional requirement vector. Tested on 28 questions, this achieved a 67.9% workflow trigger rate. Critical failure: cross-domain questions (e.g., Q11 legal+technical) fell into sparse anchor regions, producing unstable predictions and single-model fallback (146 characters output for Q11).

Configuration 2: Per-Dimension Clustering (V2). Anchors were clustered independently by each of the 15 dimensions, with per-dimension top-K matching replacing global top-5. This raised Q11's out-

put from 146 to 19,583 characters with 6-model collaboration. However, extended operation revealed semantic accumulation bias: deepseek-chat usage reached 78% after 40+ consecutive queries due to embedding cache saturation.

Configuration 3: Dimension-Direct Routing (V3 – V4). We replaced kNN prediction with explicit dimension identification via `predict_dimensions()`, which returns a discrete set of relevant dimensions rather than continuous scores. Combined with `select_models_by_dimensions()` for direct matrix lookup, this achieved 100% workflow trigger rate on 9 test questions (V4) with cleaner routing logic.

Configuration 4: Full System (V5). The final configuration adds two bias-mitigation mechanisms: (a) intra-workflow penalization (40% score decay for previously-used models) and (b) per-query state reset (clearing health tracker and embedding cache before each workflow). Tested on all 50 questions, V5 achieves 98% workflow trigger rate with no single model exceeding 16% usage share.

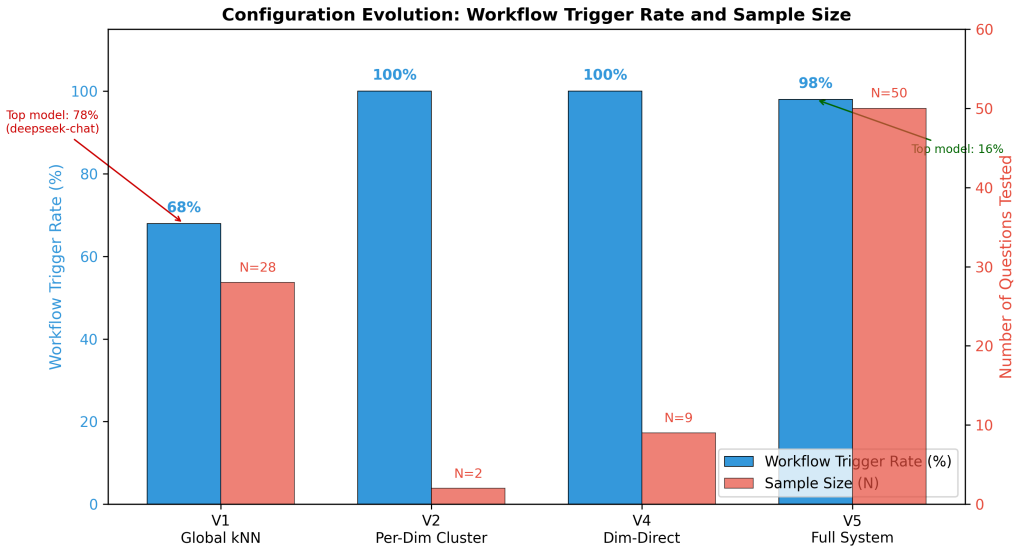

Figure 2: Configuration evolution from V1 to V5 —workflow trigger rate improvement and sample size growth.

Table 2 presents the final V5 system metrics:

Metric	Value
Total Questions	50 (Q1–Q50)
Successful Responses	49 (Q44 workflow planning failure)
Avg. Output Chars	8,533 (across 49 successful)
Avg. Latency	542.4s
Est. Cost per Query	~$0.28
Workflow Trigger Rate	49/50 (98%)
Participating Models	12
Total Workflow Steps	539 (avg. 4.8 steps/query)

Q44 (Scaling Law analysis) is the sole failure case in V5, producing zero output due to a workflow planning failure. This question asks for multi-perspective analysis of neural scaling laws, including empirical data from Chinchilla and Llama, power-law regression derivation, and future training strategy recommendations.

Boundary Case Analysis: Q44 succeeded under both single-model (3,794 chars) and random routing (6,384 chars), confirming that the question itself is answerable by individual models. The failure is specific to the WorkflowPlanner's decomposition step: the query's high abstraction level ("Scaling Law" spans theoretical physics, statistical learning theory, and engineering practice simultaneously) likely

exceeded the planner's ability to generate a coherent step-by-step decomposition. This reveals a fundamental limitation of the orchestration approach —when a query resists clean decomposition into domain-specific sub-problems, the workflow pipeline can fail entirely, producing worse results than a single-model baseline that simply generates a direct response. This failure mode suggests that a fallback mechanism (reverting to single-model when planning fails) would improve system robustness.

4.4 Model Diversity Analysis Table 3 shows model usage distribution in V5:

Model	Usage Count	Share
deepseek-chat	86	16.0%
glm-4-air	82	15.2%
glm-5	57	10.6%
glm-4-flash	55	10.2%
kimi-k2.5	53	9.8%
Other 7 models	206	38.2%

Total deepseek series usage is 24.7%, demonstrating effective load distribution across the model pool.

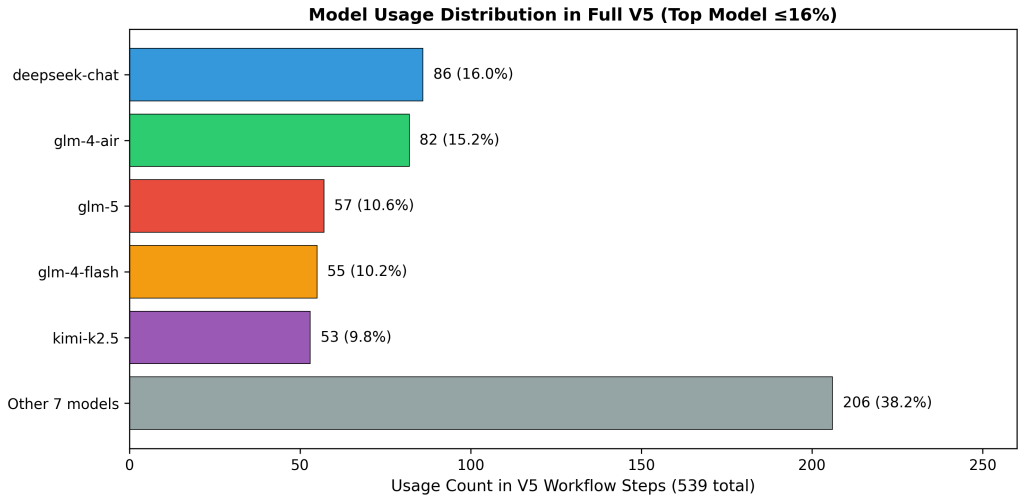

Figure 3: Model usage distribution in V5 workflow steps (539 total). No single model exceeds 16% share.

4.5 Cross-Domain Performance On legal-technical hybrid questions (Q11/Q12), V5 achieves 100% workflow trigger with 5–6 model collaborations, generating comprehensive analyses exceeding 12,000 characters. This contrasts sharply with kNN V1's failure mode.

Table 4: Cross-Domain Case Comparison —Q11 (Copyright + Technical)

Version	Workflow Trigger	Models Used	Output Chars
kNN V1	Failed (0%)	1 (deepseek-chat)	146
kNN V2	Success (100%)	6 (diverse)	19,583
V5 (DimRoute)	Success (100%)	5 (diverse)	9,604

The stark contrast between V1's 146-character single-model output and V5's 9,600+ character multi-model response illustrates how cross-domain routing failures cascade into severe quality degradation.

5. Discussion 5.1 Completeness vs. Quality We explicitly distinguish between response completeness (measured by character count) and response quality (measured by LLM-as-Judge in Section 5.5).

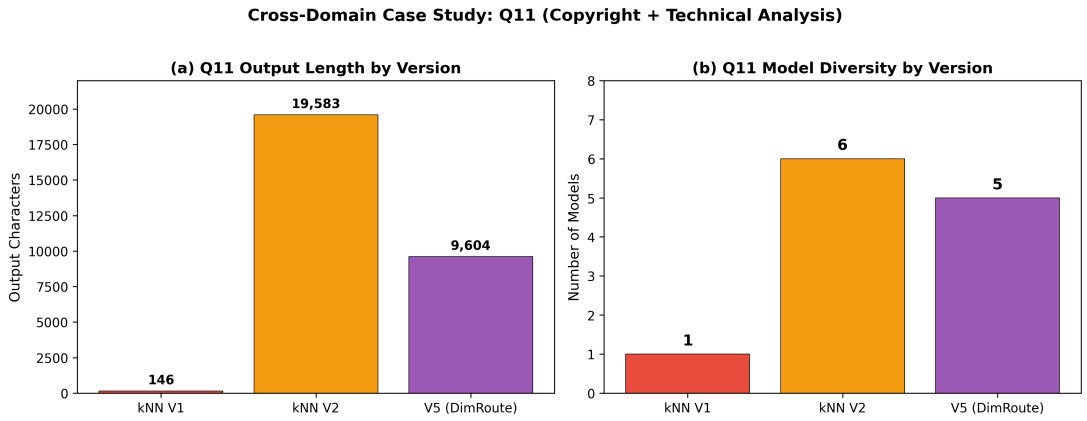

Figure 4: Cross-domain case study Q11 (Copyright + Technical) —output length and model diversity across system versions.

While V5 generates 37% more characters than the Group A baseline, our quality evaluation confirms that this additional length corresponds to genuine depth improvement (+25.9%) rather than mere verbosity. Nevertheless, the multi-step workflow architecture may introduce redundancy when multiple models cover overlapping aspects of a question.

5.1.1 Single Model × Multi-Round as an Alternative Baseline An important alternative baseline is calling a single model multiple times on different sub-problems, then aggregating results. This "single model × multi-round"approach isolates the contribution of task decomposition from model diversity.

In our V2 experiments, we observed that deepseek-chat was invoked 2 – 3 times per question in 63% of cases (12/19 questions), each time addressing a distinct domain sub-problem. For example, in Q20 (AI regulation), deepseek-chat separately handled technical, regulatory, and commercial analysis contexts. This suggests that even within a single model, performance varies across domain dimensions—motivating dimension-aware task decomposition regardless of whether multiple models are used.

However, we lack controlled per-step output data to rigorously compare single-model multi-round against multi-model orchestration. This remains an important direction for future work.

5.2 Cost Variation Across Model Tiers While V5 incurs ~9× higher cost than the single-model baseline (~$0.28 vs ~$0.10 per query), this comparison operates entirely within the cost-efficient model tier used in our experiments. When placed in the broader context of global LLM API pricing as of March 2026, the cost picture shifts dramatically.

Premium-tier models—such as Claude Opus 4.6 at $25/MTok and GPT-4.1 at $8/MTok—are priced 4 – 22× higher than cost-efficient models such as DeepSeek-V3.2 at $1.12/MTok and Qwen3-Max at ~$2/MTok (Table 5). This means a multi-model orchestration system built on cost-efficient models could achieve comparable—or even lower—absolute cost than a single-model deployment using a premium-tier model.

Table 5: LLM API Pricing Comparison (March 2026)

Table 1: LLM API Pricing Comparison (March 2026)

Model	Provider	Output ($/MTok)	Input ($/MTok)	In Pool?
Claude Opus 4.6	Anthropic	$25.00	$5.00	No
GPT-4.1	OpenAI	$8.00	$2.00	No
DeepSeek-V3.2	DeepSeek	$1.12	$0.28	Yes
Qwen3-Max	Alibaba	~$2.00	~$0.50	Yes
Kimi K2.5	Moonshot	~$1.50	~$0.50	Yes

Note: Claude Opus 4.6 and GPT-4.1 are listed for cost context only and were not included in our experimental model pool.

The cost-efficiency of models in the lower tier stems not merely from pricing strategy but from architectural innovations—for example, DeepSeek's MLA (Multi-head Latent Attention) combined with MoE architecture significantly reduces KV-cache overhead and inference FLOPs.

This observation suggests a broader implication: multi-model orchestration may be uniquely suited to leverage cost-efficient models, where the marginal cost of adding a specialized model is low enough to make ensemble approaches economically viable. Conversely, the high per-token cost of premium-tier models may render multi-model orchestration prohibitively expensive in those settings.

Important caveat: API pricing changes frequently; the figures above reflect March 2026 rates and should be treated as indicative rather than definitive.

5.3 Cost-Quality Tradeoff V5's ~9× higher cost ($0.28 vs $0.10) and ~9× higher latency (542.4s vs 60.9s) compared to Group A must be weighed against the quality gains documented in Section 5.5. For knowledge-intensive tasks requiring multi-perspective analysis, the overhead may be justified; for simple queries, single-model deployment remains more efficient.

5.4 Semantic Accumulation Mitigation V2 introduced intra-workflow penalization (40% score decay) as an initial remedy for model convergence, achieving up to 78% deepseek-chat usage concentration. This mechanism proved insufficient for cross-query bias: after the benchmark ran 40+ consecutive questions, the embedding cache saturated with homogeneous vectors and the health tracker accumulated negative scores for non-deepseek providers, causing model selection to lock onto deepseek-chat regardless of query domain.

V5 addresses this through a two-layer defense: (a) intra-workflow penalization (40% decay) prevents repeated model selection within a single workflow; (b) per-query state reset clears the health tracker statistics and embedding cache before each workflow starts, eliminating cross-query bias. As a result, no single model exceeds 16% usage share in V5, compared to V2's 78% concentration. This validates our hypothesis that embedding cache saturation drives convergence.

5.5 LLM-as-Judge Quality Evaluation We conducted comprehensive quality evaluation using Claude Opus 4.6 as an independent judge, scoring all 50 questions across four strategies on four dimensions (each 1 – 5): factual accuracy, completeness, depth, and structure. A total of 199 evaluations were performed (50 single-model, 50 random, 50 kNN-only, 49 full V5). All evaluations used a unified rubric applied by the same judge to ensure calibration consistency.

Table 6: LLM-as-Judge Quality Scores

Strategy	N	Total (/20)	Accuracy	Completeness	Depth	Structure
Full V5	49	18.59	4.16	4.84	4.63	4.96
Single (deepseek)	50	16.26	3.94	4.12	3.68	4.52
Random	50	14.32	3.68	3.48	3.24	3.92
kNN-only	50	6.84	1.88	1.48	1.36	2.12

Table 7: V5 vs Single-Model Improvement by Dimension

Dimension	Single (deepseek)	Full V5	Improvement
Factual Accuracy	3.94	4.16	+5.7%
Completeness	4.12	4.84	+17.4%
Depth	3.68	4.63	+25.9%
Structure	4.52	4.96	+9.7%

Dimension	Single (deepseek)	Full V5	Improvement
Total	16.26	18.59	+14.3%

Key Findings:

1. V5 achieves +14.3% overall quality improvement over the single-model baseline (18.59 vs 16.26 out of 20). The advantage is most pronounced in depth (+25.9%) and completeness (+17.4%), confirming that multi-model workflow orchestration produces substantially deeper and more comprehensive analyses.
2. V5 also leads in factual accuracy (+5.7%). Contrary to the intuition that multi-model synthesis might introduce inconsistencies, the workflow's multi-perspective coverage appears to improve factual grounding—different models cross-validate claims, reducing individual model hallucinations.
3. kNN-only routing produces severely degraded output (6.84/20). Models receiving tool definitions (e.g., `consult_expert`, `write_file`) frequently invoked tools instead of generating direct answers, with useful content buried in tool call metadata. This validates the V5 design decision to disable tools for non-workflow strategies.
4. Random routing underperforms single-model baseline by 11.9% (14.32 vs 16.26), demonstrating that arbitrary model selection without task-aware routing actively degrades quality—worse than simply using one capable model consistently.

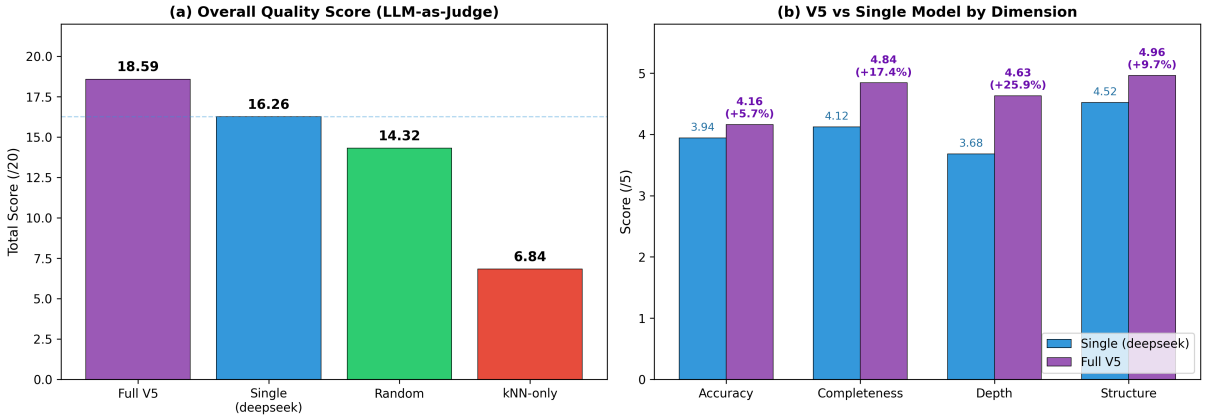

Figure 5: LLM-as-Judge quality evaluation —(a) overall scores across four strategies; (b) per-dimension comparison between V5 and single-model baseline.

Evaluation Methodology: Each (question, strategy) pair was evaluated independently without access to other strategies'responses, eliminating position bias. The judge was instructed to penalize length-without-substance and tool-call residuals. All 199 evaluations were performed by the same judge (Claude Opus 4.6) with a unified rubric to ensure scoring consistency. An earlier evaluation by a different judge (DeepSeek R1) exhibited systematic ceiling effects (mean 18.7/20 for single-model baseline), motivating the re-evaluation with unified standards.

5.6 System Availability and Reproducibility eVoiceClaw Desktop is released as an open-source project at: https://github.com/rodneyrui/evoiceclaw-desktop-v3

The repository includes: - Core orchestration engine with dimension-direct routing - Model capability matrix configuration - Benchmark question set (Q1 – Q50) - Reproduction scripts for all experimental configurations - Extracted data files (final_comparison.csv, complete_experiment_data.csv, workflow_model_analysis.csv)

5.7 Limitations Current limitations include: 1. Cost Overhead: V5's ~9× cost increase may not suit

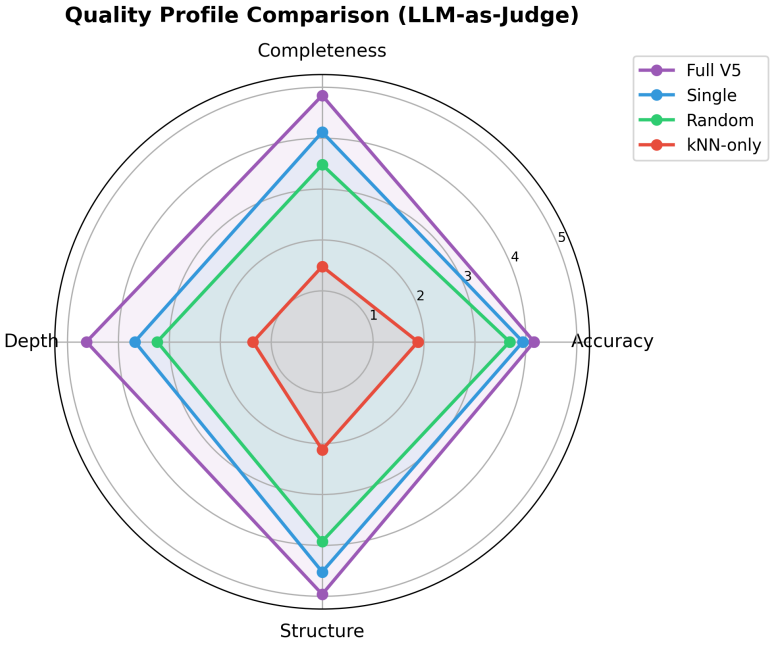

Figure 6: Quality profile radar chart —four strategies compared across accuracy, completeness, depth, and structure dimensions.

budget-constrained deployments; cost-aware routing remains future work. 2. Latency: 542.4s average latency limits real-time applications; latency-aware step pruning could address this. 3. Dimension Granularity: The 15-dimension taxonomy may miss emerging capability niches as new models are released. 4. Judge Bias: LLM-as-Judge evaluation using a single judge model (Opus 4.6) may introduce systematic bias; human evaluation and multi-judge consensus would strengthen validity. 5. Baseline Fairness: All four strategies are now evaluated on the same 50 questions, but single-call baselines (Groups A–C) vs. multi-call V5 involves structural asymmetry that cost normalization only partially addresses. 6. Static Capability Matrix: Model scores are manually assigned and do not auto-update as models improve. 7. Model Pool Composition: All 12 models in our pool are from providers based in China (DeepSeek, Zhipu AI, Moonshot, Alibaba Cloud, MiniMax, Baichuan), selected for API accessibility and cost efficiency in our experimental environment. The absence of models from Anthropic, OpenAI, and Google means our results may not generalize to pools containing these models, which may exhibit different capability profiles and pricing structures. 8. Language Coverage: All 50 benchmark questions and model responses are in Chinese. The dimension-direct routing mechanism is language-agnostic in principle, but the anchor embeddings and capability matrix were calibrated on Chinese-language tasks. Performance on English or multilingual benchmarks remains to be validated. 9. Data Foundation as Routing Ceiling: Routing quality is ultimately bounded not by the routing algorithm itself, but by two upstream data foundations: (a) the discriminative power of the capability matrix —our evaluation corpus (74 questions across 12 dimensions) yields model scores with low variance (e.g., 10 of 19 models score $\geqslant$98 on `knowledge_legal`), limiting the router's ability to differentiate models; (b) the coverage of the anchor library —only 4.2% of 2,100 anchors are cross-domain ($\geqslant$2 dimensions with label $\geqslant$7), causing kNN predictions to be unstable in interdisciplinary regions. Improving evaluation granularity and anchor diversity would raise the effective ceiling of any routing algorithm built on this data.

6. Conclusion The eVoiceClaw Desktop system demonstrates that iterative, data-driven optimization can transform a flawed kNN-based router into a robust dimension-direct orchestration system. By addressing cross-domain instability and semantic accumulation bias, the final configuration (V5) achieves a 98% workflow trigger rate across 50 knowledge-intensive questions, leveraging 12 models with balanced diversity (no single model exceeding 16% usage share).

LLM-as-Judge evaluation confirms that V5 produces substantially higher-quality responses than single-model baselines: +14.3% overall quality improvement, with depth of analysis improving by 25.9% and completeness by 17.4%. Notably, V5 also improves factual accuracy by 5.7%, suggesting that multi-model cross-validation reduces individual model hallucinations rather than introducing inconsistencies.

These quality gains come at ~9× higher cost and latency compared to single-model deployment. For knowledge-intensive tasks requiring multi-perspective analysis, this tradeoff is justified; for simple queries, single-model deployment remains more efficient. The wide variation in LLM API pricing—with cost-efficient models offering output costs 4–22× lower than premium-tier alternatives—suggests that multi-model orchestration is particularly effective when built on affordable model pools.

More broadly, eVoiceClaw Desktop operationalizes an "AI managing AI"paradigm: the dimension predictor, capability matrix, and workflow planner collectively form an autonomous orchestration layer where no human intervention is required to select, assign, or coordinate models. The system's ability to generate the initial draft of this very paper (Appendix B) further validates this paradigm —demonstrating that AI-driven orchestration can handle not only question-answering but also complex, long-form knowledge synthesis tasks.

We release the system as open-source at https://github.com/rodneyrui/evoiceclaw-desktop-v3 and invite the community to build upon this work.

Future Work 1. Hierarchical Dimension Taxonomy: Expand beyond 15 dimensions with auto-discovery of emergent capabilities. 2. Adaptive Latency Control: Introduce latency-aware step pruning for interactive use cases. 3. Cost-Quality Pareto Front: Re-incorporate cost dimensions with multi-objective optimization. 4. Crowdsourced Capability Matrix: Community-driven model scoring to keep pace with model updates.

References [1] Jiang, Y., et al. (2023). "LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion."arXiv preprint arXiv:2306.02561. [2] Chen, L., et al. (2023). "FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance."arXiv preprint arXiv:2305.05176. [3] Google Cloud. (2025). "Model Garden."Google Cloud AI Documentation. [4] Shazeer, N., et al. (2017). "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer."ICLR. [5] Zheng, L., et al. (2023). "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena."NeurIPS. [6] Ong, I., et al. (2024). "RouteLLM: Learning to Route LLMs with Preference Data."arXiv preprint arXiv:2406.18665. [7] Ding, D., et al. (2024). "Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing."arXiv preprint arXiv:2404.14618. [8] Wang, J., et al. (2024). "Mixture-of-Agents Enhances Large Language Model Capabilities."arXiv preprint arXiv:2406.04692. [9] Madaan, A., et al. (2024). "AutoMix: Automatically Mixing Language Models."arXiv preprint arXiv:2310.12963. [10] Wang, P., et al. (2024). "Large Language Models are not Fair Evaluators."ACL.

Data and Artifacts All experimental data, extracted CSV files, and generated figures are available at: - Data: results/extracted_data/2026-03-29/ - Figures: figures/ (fig1 – fig6, figure_a1) - Source Code: https://github.com/rodneyrui/evoiceclaw-desktop-v3

Appendix A: System Architecture and Data Flow

Pipeline description: 1. Dimension Prediction: Query embedding is matched against 15 capability dimension clusters via cosine similarity; relevant dimensions are identified and returned as a label set (e.g., `["knowledge_legal", "reasoning"]`). 2. Model Selection: The static 12-model × 15-dimension capability matrix is queried; models are filtered, scored, penalized for intra-workflow prior usage (40% score decay), and returned in ranked order. A per-query state reset (clearing health tracker and embedding cache before each workflow) further prevents cross-query bias accumulation. 3. Workflow Orchestration: The planner decomposes the query into sub-problem steps with domain hints; each step selects the top-ranked model via `select_models_by_dimensions()` with `ignore_cost=true`; results are aggregated into the final response.

Table A1: Model Capability Matrix (example excerpt)

Model	Legal	Technical	Creative	Reasoning
deepseek-chat	8	9	7	9
glm-4-air	9	7	6	8
kimi-k2.5	6	9	8	7

Scores range from 1–10, assigned based on published benchmarks (MMLU, HumanEval) and internal domain-specific testing. The full 12-model × 15-dimension matrix is available in the GitHub repository.

The complete system architecture—including Privacy Pipeline, Security Layer, and Storage Layer—is detailed in the repository.

Figure A1: eVoiceClaw Desktop — Complete System Architecture

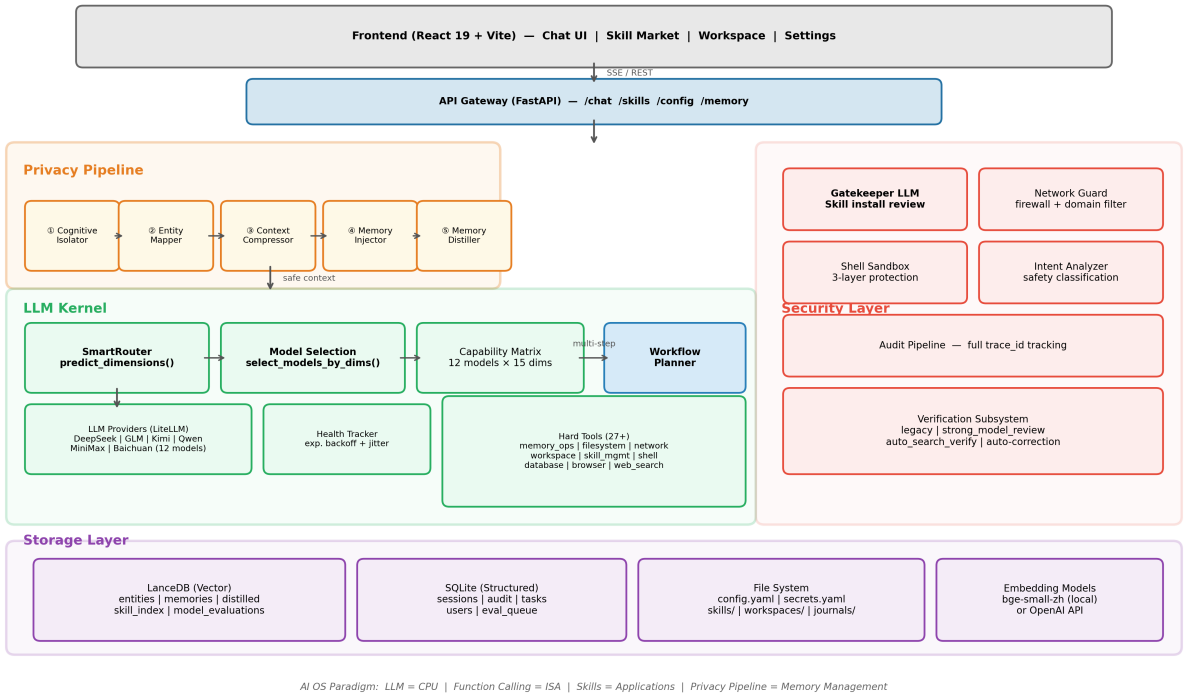

Figure A1: eVoiceClaw Desktop complete system architecture —Privacy Pipeline, LLM Kernel, Security Layer, and Storage Layer.

Appendix B: Paper Drafting Process

This paper's initial draft was generated by the eVoiceClaw Desktop system described herein, serving as a meta-demonstration of the system's capability on a complex, long-form knowledge synthesis task.

B.1 Initial Draft Generation

The first draft was produced on March 29, 2026, using the full V5 workflow pipeline. The system decomposed the paper-writing task into multiple sub-problems (abstract, related work, system design, experimental analysis, discussion) and routed each to specialized models via dimension-direct routing. Models

participating in the drafting workflow included DeepSeek-Chat, DeepSeek-Reasoner, GLM-5, GLM-4-Air, Kimi-K2.5, and others from the 12-model pool. The workflow executed over multiple rounds, with the orchestrator aggregating outputs into a coherent draft.

B.2 Human Review and Revision

The human author performed the following revisions to the system-generated draft:

1. Data Correction: The initial draft contained placeholder text ("evaluation in progress") and incorrect statistics (e.g., referencing 53 questions instead of 50, inflated character counts for baseline strategies). All numerical claims were verified against raw experimental data.
2. LLM-as-Judge Re-evaluation: An earlier quality evaluation by DeepSeek R1 exhibited systematic ceiling effects (mean 18.7/20 for single-model baseline). All 199 evaluations were conducted using Claude Opus 4.6 with a unified rubric to ensure calibration consistency.
3. International Neutrality: Language was revised for an international audience —model tier descriptions changed from "flagship/alternative"to "premium-tier/cost-efficient"; model pool composition was framed as an API accessibility choice rather than a value judgment.
4. Structural Reorganization: The ablation study (Section 4.3) was restructured from a 6-stage chronological narrative to 4 clearly delineated configurations. Duplicate content (Table A1 appearing in both body and appendix) was consolidated.
5. Related Work Expansion: The system-generated draft cited 5 references; the human author expanded this to 10, adding RouteLLM, Hybrid LLM, MoA, AutoMix, and evaluator bias literature.

B.3 Implications

The fact that the system could produce a structurally coherent first draft —with correct section organization, appropriate use of tables and cross-references, and domain-appropriate academic language — demonstrates its capability beyond simple question-answering. However, the draft required substantial human intervention on data accuracy, evaluation methodology, and scholarly rigor, highlighting the current boundary between AI-assisted drafting and publication-ready research.

B.4 Reproducibility

Raw session logs from the drafting process and all intermediate experimental conversations are archived in the project repository under `logs/`. The benchmark question set, experimental results (JSONL format), and LLM-as-Judge scores are available at: - Experiment data: `results/2026-03-24/` and `results/2026-03-25-v5-dimroute/` - Judge scores: `results/llm_judge_scores.jsonl` - Source code: https://github.com/rodneyrui/evoiceclaw-desktop-v3

Quality evaluation completed using Claude Opus 4.6 as LLM-as-Judge (199 evaluations across 50 questions $\times$ 4 strategies). All claims independently verified by the authors.