BiCSRouter: Bi-Level Cross-System Routing for Utility-Aware LLM Inference

Keyu Mao[1]* **Eiki Murata**[2,3] **Ukyo Honda**[3]
[1]Institute of Science Tokyo [2]AI Shift [3]CyberAgent
mao.k.aa@m.titech.ac.jp {murata_eiki, honda_ukyo}@cyberagent.co.jp

Abstract

Selecting an appropriate LLM configuration for a given query is critical, yet existing routing frameworks operate within a single computational paradigm. To address this gap, we formalize the Cross-System Routing Problem, a hierarchical decision-making task that decomposes routing into intra-regime configuration selection and inter-regime system selection. Building on this, we propose BiCSRouter, a bi-level cross-system routing framework that integrates two orthogonal regimes: intensive reasoning via single-agent systems and extensive collaboration via multi-agent systems. BiCSRouter performs policy learning within each system and employs a lightweight inter-regime router that selects the optimal regime based on predicted performance and cost. Experiments on the MBPP and MATH benchmarks demonstrate that BiCSRouter outperforms 15 representative baselines across three types. On MBPP, compared to the performance ceiling of GPT-5, BiCSRouter achieves a 46% reduction in cost with only a 2% drop in accuracy. Finally, we show that BiCSRouter can extend to additional regimes, highlighting its generality as a cross-system routing framework.

1 Introduction

The advent of Large Language Models (LLMs) has revolutionized artificial intelligence, demonstrating remarkable capabilities across diverse domains (Brown et al., 2020), including coding (Chen et al., 2021), translation (Qin et al., 2025), and mathematical reasoning (Wei et al., 2022). As the complexity and diversity of tasks have continued to grow in recent years, selecting the appropriate LLM for a given query has become a critical challenge.

To address this, the concept of LLM routing has emerged. Early research mainly concentrates on model routing, leveraging simple components to predict the query difficulty and then selecting between a large and a small model (Ding et al., 2024; Ong et al., 2025). Subsequent research shifts its concern to a multi-choice selection framework (Feng et al., 2025; Chen et al., 2024c; Zhang et al., 2025b), which further explores the performance-cost trade-off across a diverse pool of candidate models.

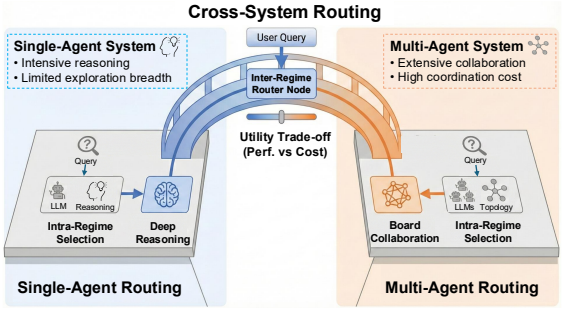

Figure 1: Cross-system routing between single-agent and multi-agent systems. Unlike prior routing methods that operate within a fixed system, BiCSRouter explicitly models heterogeneous computational regimes and selects the optimal system based on predicted utility.

Despite these remarkable achievements, existing routing frameworks still face significant limitations. First, current methods typically restrict themselves to a single regime, ignoring the orthogonality of computational regimes. Single-agent systems excel in **Intensive Reasoning**, which maintains answer consistency and provides depth for logic-heavy tasks (Wang et al., 2023, 2024), while multi-agent systems thrive in **Extensive Collaboration**, which offers diverse exploration for broad, creative tasks (Du et al., 2023; Shinn et al., 2023). Existing intra-regime routers fail to utilize both advantages simultaneously.

Motivated by the above observations, we argue that a robust routing system should transcend model selection and perform **Cross-System Routing**. However, training a monolithic policy to master different regimes is non-trivial due to their divergent configuration spaces. We posit that ef-

*This work was done during an internship at AI Shift.

Findings of the Association for Computational Linguistics: ACL 2026, pages 18979–18993

fective routing requires a hierarchical approach: optimizing the configuration *within* each regime (Intra-Regime) before selecting the optimal system *between* regimes (Inter-Regime). Recent work by Gao et al. (2025) represents the first attempt to integrate two systems, demonstrating performance improvements via simple routing. However, their method employs fixed configurations for each regime and ignores intra-regime optimization, failing to grasp characteristics that differentiate regime choice from model selection.

In this paper, we propose BiCSRouter (**Bi**-Level **C**ross-**S**ystem **Router**), a novel cross-system framework designed to bridge the gap between agent systems with different regimes. We define a Cross-System Router as a meta-controller that aggregates optimal configurations from distinct regimes and selects the most utility-maximizing system based on query-specific requirements. To achieve cost-effective routing, BiCSRouter operates in two orthogonal regimes: a single-agent regime that prioritizes the reasoning depth of the flagship model, and a multi-agent regime that prioritizes collaboration among efficient models. For precise cost-aware routing, we decouple the predictions of performance and cost. We build a configuration-based prediction model for performance and a cost predictor that simulates the execution workflow to dynamically estimate resource consumption. This allows our BiCSRouter to achieve an excellent performance-cost balance and conduct cost-efficient inference via structural cost prediction.

Our contributions can be summarized as follows:

- We formulate the problem of Cross-System Routing, strengthening the importance of both intra-regime routing and inter-regime routing.
- We propose BiCSRouter, a hierarchical framework that unifies intra-regime configuration selection and inter-regime selection. By integrating a decoupled performance predictor with a cost predictor, BiCSRouter achieves efficient routing with balanced and controllable inference costs.
- Extensive experiments on public benchmarks demonstrate that BiCSRouter significantly outperforms state-of-the-art regime-specific routers while maintaining a favorable performance-cost trade-off. Furthermore, we show that BiCSRouter exhibits robust out-of-distribution generalization and promising multi-system extensibility, serving as a flexible bridge between heterogeneous systems.

2 Related Work

LLM Reasoning Paradigms LLMs continue to evolve and demonstrate remarkable capabilities in complex tasks like logical reasoning (Zhang et al., 2025d) and code generation (Chen et al., 2024a). To leverage the intrinsic potential of single-agents, the research community first enhances inference-time reasoning within individual LLMs, primarily through algorithmic prompting and structured reasoning paradigms such as Chain of Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) and Tree of Thoughts (ToT) (Yao et al., 2023), strengthening the reasoning depth of LLMs. Although these methods enhance the performance of LLMs in several domains, their inference chains are restricted and lack diversity. (Du et al., 2023) and (Li et al., 2023) first broadened the research focus to multi-agent systems, following which more static architectures (Yin et al., 2023; Zhao et al., 2024a) were proposed. Subsequent research, such as GPTSwarm (Zhuge et al., 2024) and AgentPrune (Zhang et al., 2025a), concentrates on dynamic optimization of inter-agent topologies. Nevertheless, recent studies have revealed critical failure modes of multi-agent systems including sycophantic behaviors (Fanous et al., 2025) and conformity during debate (Wynn et al., 2025). More recent studies delve back into single-agent systems (Wang et al., 2024), revealing that advanced single-agent LLMs have mitigated many limitations that originally motivated multi-agent systems. All these works remain paradigm-specific, failing to adaptively leverage the complementary strengths of both systems.

LLM Routing LLM routing strategies have been proposed to efficiently balance model performance and computational cost. Early research turned to single-agent routing, whose focus has shifted from simple binary routing (Ding et al., 2024; Ong et al., 2025; Chen et al., 2024b) to multi-choice selection frameworks (Feng et al., 2025; Chen et al., 2024c; Zhang et al., 2025b). Recently, multi-agent routing was introduced to address inter-agent topological constraints. DyLan (Liu et al., 2024b) first proposed the mechanism to dynamically select agent teams and MasRouter (Yue et al., 2025) systematically defines multi-agent system routing, which allocates roles and collaboration modes to agents in multi-agent systems. Moreover, BAMAS (Yang

et al., 2025) builds multi-agent systems with budget awareness. In parallel, Router-R1 (Zhang et al., 2025b) explores multi-round aggregation within a flat model pool, providing an additional perspective on adaptive routing. All these routing strategies are constrained within a single paradigm, neglecting inter-regime differences. Gao et al. (2025) first attempt to integrate both systems, but they lack the consideration of routing within each system, and construct a traditional binary router regardless of system regime and detailed configurations.

Token Length Prediction Predicting the generated token length of LLMs is beneficial to control computational cost. Early research adopted machine learning structures like random forest (Cheng et al., 2024) and deep learning models based on encoder-only structures (Qiu et al., 2024; Stojkovic et al., 2025) or decoder-only structures (Zheng et al., 2023). More recently, OmniRouter (Mei et al., 2026) applies a bucket-based classification to estimate the range of token length. Despite their progress in single-LLM scenarios, these methods lack structural consideration within systems and cannot be transferred to multi-agent scenarios.

3 Problem Formulation

In this section, we formalize the proposed Cross-System Routing Problem (CSRP). CSRP formulates routing as a bi-level decision-making task, consisting of intra-regime routing and inter-regime routing. In CSRP, we consider multiple computational regimes, where each regime t corresponds to a system characterized by a distinct configuration search space Ω_t. A configuration c_t specifies a concrete system instantiation, including the selected models, the allocation of roles or reasoning strategies, and the underlying topology.

3.1 Intra-regime Routing

For a given regime t, a configuration c_t is sampled from the corresponding subspace:

$$c_t \sim \pi_t(\cdot|q;\Omega_t). \tag{1}$$

Here, π_t denotes a regime-specific policy that samples a configuration from the constrained subspace.

For each regime t, we aim to find a policy $\pi_t^*(\cdot|q;\Omega_t)$ that selects the optimal configuration c_t^* from subspace Ω_t. The objective is to maximize the expected local utility U as a function of performance P and cost C:

$$U_t(y_t, c_t; g) = P(y_t; g) - \lambda C(y_t, c_t), \tag{2}$$

$$c_t^* = \underset{c_t \in \Omega_t}{\operatorname{argmax}} \, \mathbb{E}_{y_t \sim \mathcal{G}_t(q,c_t)} \{U_t(y_t, c_t; g)\} \tag{3}$$

where q denotes the input query, y_t denotes the model output generated under regime t with configuration c_t, g denotes the ground-truth answer used for evaluation, and $\mathcal{G}_t(q, c_t)$ denotes the conditional generation distribution induced by executing configuration c_t under regime t for query q.

3.2 Inter-regime Routing

Given the regime-specific configurations $\{c_t^*\}_{t=1}^T$ obtained from intra-regime policy learning, the goal of the inter-regime router is to select the execution regime that maximizes the expected utility for each query q. Formally, the router chooses:

$$c_{\text{opt}}^* = \underset{c \in \{c_t^*\}_{t=1}^T}{\operatorname{argmax}} \, U_t(y_t, c; g) \tag{4}$$

where c_{opt}^* denotes the configuration for execution.

4 Methodology

Figure 2 demonstrates an overview of our proposed framework, BiCSRouter. In BiCSRouter, we consider a system-level decision-making problem between two orthogonal computational regimes: Intensive Reasoning (ir) and Extensive Collaboration (ec) with search subspaces Ω_{ir} and Ω_{ec}. We apply topological constraints to these two subspaces:

- **Ensemble Constraint (Ω_{ec}):** Defined as $\mathcal{M}_{ec}^N \times \mathcal{T}_{ec}$, where $\mathcal{M}_{ec}$ denotes the model pool, $\mathcal{T}_{ec}$ denotes the multi-agent topology space, and N denotes the number of agents. This subspace allows multi-agent topologies through an inference graph $\mathcal{G}_{ec} = (V, E)$. Each node $v_i \in V$ represents an agent instance instantiated by a tuple (m_i, r_i), where $m_i \in \mathcal{M}_{ec}$ is a selected model and r_i denotes a role selected from the role set $\mathcal{R}$. Edges in E are determined by the topology $\tau \in \mathcal{T}_{ec}$.
- **Singleton Constraint (Ω_{ir}):** Defined as $\mathcal{M}_{ir} \times \mathcal{T}_{ir}$, where $\mathcal{T}_{ir}$ represents the search space of reasoning strategies. The inference graph $\mathcal{G}_{ir}$ of this subspace is restricted to a single-node structure, and only sequential reasoning is allowed for the topology to maximize the inference depth of a single-agent.

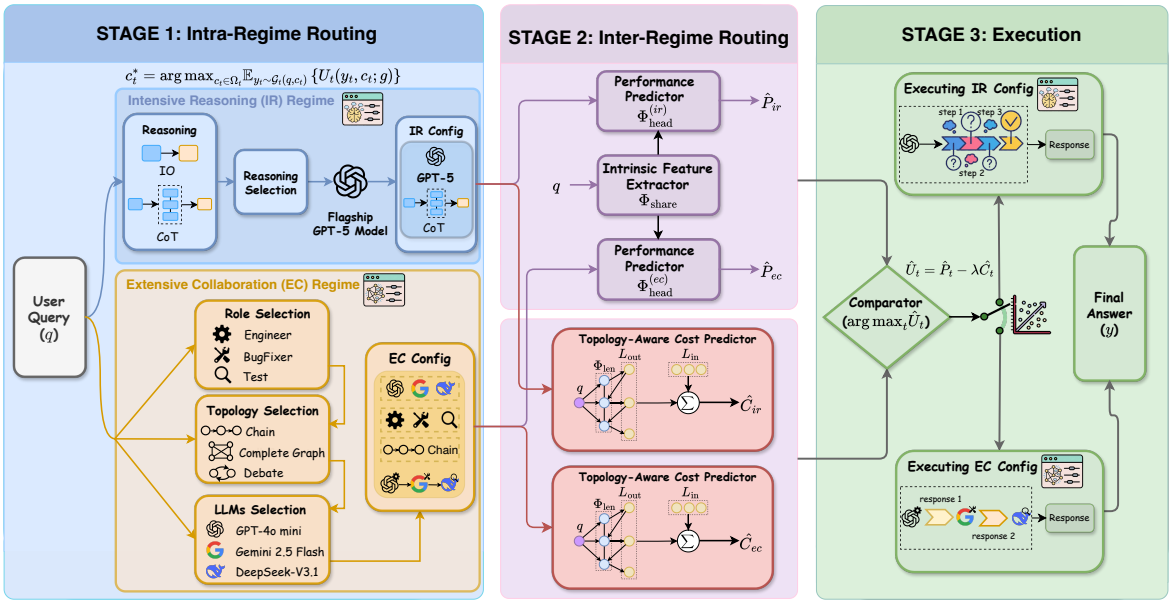

Figure 2: The overall framework of BiCSRouter, which consists of an intra-regime policy learning module for configuration sampling and an inter-regime utility estimator that selects the regime with the highest expected utility.

The intra-regime learning module samples the optimal configuration from each regime's subspace. Then, the inter-regime module predicts the utility of both regimes and chooses the regime with higher expected utility. These two modules are treated as separate components during training and updated using different strategies.

4.1 Intra-Regime Structural Policy Learning

To navigate the discrete and combinatorial configuration subspace Ω_t, we decompose the sampling policy $\pi_t(\cdot|q;\Omega_t)$ into a hierarchical generative process. We apply a modular neural architecture that sequentially determines agent scale, reasoning strategy, role composition, and model selection under constraints of each regime.

Given a single text query q, we first leverage a text-encoder like MiniLM (Wang et al., 2020) to extract the text query's semantic information $\mathbf{f}_q$:

$$\mathbf{f}_q = \text{Encoder}_{\text{text}}(q). \tag{5}$$

The interaction topology τ is formalized as a semantic matching problem in a latent manifold. The input feature $\mathbf{f}_q$ and the available topology set or reasoning strategy set $\mathcal{T}_t$ are projected into a latent space via a Variational Autoencoder (VAE) architecture, yielding $\mathbf{z}_q$ and $\mathbf{Z}_{\mathcal{T}_t}$, respectively. We then learn the conditional probability:

$$P(\tau \mid \mathbf{f}_q) = \text{Softmax}\left(\frac{\mathbf{z}_q \cdot \mathbf{Z}_{\mathcal{T}_t}^T}{\gamma_1}\right) \tag{6}$$

where $\mathbf{z}_q$ and $\mathbf{Z}_{\mathcal{T}_t}$ denote latent representations, and γ_1 is a temperature parameter controlling the sharpness of the distribution.

For the Extensive Collaboration regime, roles are allocated via an autoregressive process:

$$\mathbf{h}_{\text{ctx}}^{(i)} = \text{Linear}\left([\mathbf{f}_q; \mathbf{h}_{\text{role}}^{(i)}]\right), \tag{7}$$

$$r_i \sim \text{Categorical}\left(\text{Softmax}\left(\frac{\mathbf{h}_{\text{ctx}}^{(i)} \cdot \mathbf{Z}_{\mathcal{T}_{ec}}^T}{\gamma_2}\right)\right) \tag{8}$$

where $\mathbf{h}_{\text{role}}^{i}$ denotes the history embedding that summarizes prior selected roles, $\mathbf{Z}_{\mathcal{T}_{ec}}$ denotes the embedding matrix of all roles, $[\cdot;\cdot]$ denotes vector concatenation, and $\text{Categorical}(\cdot)$ denotes sampling from a categorical distribution parameterized by the given probability vector. The history embedding is updated autoregressively as $\mathbf{h}_{\text{role}}^{(i)} = \text{LayerNorm}(\mathbf{h}_{\text{role}}^{(i-1)} + \mathbf{e}_{r_i})$ with the initialization $\mathbf{h}_{\text{role}}^{(0)} = \mathbf{0}$, where $\mathbf{e}_{r_i}$ denotes the embedding of the present selected role embedding obtained from the embedding matrix $\mathbf{Z}_{\mathcal{T}_{ec}}$.

We then match LLMs based on all contextual information: query, interaction topology, and allocated roles. We create a comprehensive context vector $\mathbf{c}_{\text{full}}^{ec}$ by encoding concatenated contextual information, and sample LLMs by:

$$\mathbf{m}^{ec} \sim \text{Categorical}\left(\text{Softmax}\left(\frac{\mathbf{c}_{\text{full}}^{ec} \cdot \mathbf{Z}_{\mathcal{M}_{ec}}^T}{\gamma_3}\right)\right) \tag{9}$$

where $\mathbf{Z}_{\mathcal{M}_{ec}}$ represents the embedding matrix of candidate LLMs for the Extensive Collaboration regime, and γ_3 is a temperature hyperparameter.

To optimize the intra-regime policies, we employ the REINFORCE algorithm, defining the reward $R(c_t)$ as a trade-off between performance and cost:

$$R(c_t) = P(c_t) - \lambda \cdot C(c_t) \tag{10}$$

where $P(\cdot)$ is the actual performance and $C(\cdot)$ denotes the actual execution cost, and λ is a trade-off coefficient. To reduce variance, we utilize a dynamic baseline b_t tracking the moving average of rewards. The objective is to minimize the reward loss:

$$\mathcal{L}_{re} = -\sum_{t\in\{ec,ir\}} (R(c_t) - b_t) \log \pi_t(c_t|\mathbf{f}_q) \tag{11}$$

where, π_t represents the joint probability of the generated configuration sequence.

4.2 Decoupled Utility Estimation for Routing

In the routing task, we propose a dual-head performance estimator to simultaneously predict the expected utility of both the Intensive Reasoning (ir) and Extensive Collaboration (ec) regimes. The utility U_t is decoupled into two components: performance probability P_t and execution cost C_t.

4.2.1 Polarized Performance Prediction

We use a dual-head performance predictor structure to predict performance $P(y_t|q, c_t)$. The predictor first extracts the intrinsic query difficulty feature $\mathbf{h}_{\text{shared}}$ via a shared encoder Φ_{share}:

$$\mathbf{h}_{\text{shared}} = \Phi_{\text{share}}(\mathbf{f}_q). \tag{12}$$

Then we predict logits ℓ_t of each regime through two independent branches. For each regime $t \in \{ec, ir\}$, we construct a composite feature vector by concatenating the shared difficulty representation with the regime-specific configuration embeddings. The logit ℓ_t is computed via a regime-specific Multi-Layer Perceptron (MLP) head:

$$\ell_t = \Phi_{\text{head}}^{(t)}\left([\mathbf{h}_{\text{shared}}\,;\,\bar{\mathbf{e}}_{\mathcal{M}}^{(t)}\,;\,\mathbf{e}_{\tau}^{(t)}\,;\,N^{(t)}]\right). \tag{13}$$

Here, N denotes the number of instantiated agents. The aggregated embedding $\bar{\mathbf{e}}_{\mathcal{M}}$ is computed as the mean of the embeddings of all selected LLMs. Both model embeddings $\mathbf{e}_m$ and topology embeddings $\mathbf{e}_\tau$ are obtained by encoding their textual descriptions using the text encoder $\text{Encoder}_{\text{text}}(\cdot)$. The predicted performance probability $\hat{P}_t$ is then computed as $\hat{P}_t = \sigma(\ell_t)$, where $\sigma(\cdot)$ denotes the sigmoid function.

To optimize the routing decision boundary while calibrating the probability distribution, we propose a hybrid curriculum objective that integrates point estimation and pair ranking learning. Our training objective comprises three key components:

1. **Binary Cross-Entropy (BCE):** Used for fundamental probability calibration, ensuring that predictions approximate the exact outcomes.

$$\mathcal{L}_{\text{BCE}} = \sum_{t\in\{ec,ir\}} \text{BCE}(\hat{P}_t, P_t) \tag{14}$$

2. **Uplift Ranking Loss:** Inspired by previous research (Burges et al., 2005; Chen et al., 2009), we utilize a Ranking Loss to directly penalize incorrect ordering:

$$\mathcal{L}_{\text{rank}} = |\Delta P| \left[m - \Delta\hat{P}\,\text{sign}(\Delta P)\right]_+ \tag{15}$$

where $\Delta\hat{P}$ and ΔP represent the differences in predicted probabilities and factual outcomes, respectively, and m is the margin.

3. **Entropy Regularization:** To prevent mode collapse and encourage the router to explore when the difference between regimes is trivial, we introduce an entropy regularization term. We first project performance margin into routing probability ρ_r:

$$\rho_r = \sigma(\hat{P}_{ir} - \hat{P}_{ec}) \tag{16}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The regularization loss is defined as the negative binary entropy of this distribution:

$$\mathcal{L}_{\text{e}} = \rho_r \log(\rho_r) + (1-\rho_r)\log(1-\rho_r). \tag{17}$$

To stabilize training, we employ a curriculum schedule to dynamically adjust the weights α_k. In the early stages, the model focuses on BCE loss for robust representation learning. Then as training progresses, the weight shifts towards Ranking Loss to focus on Decision Boundary Optimization:

$$\mathcal{L}_{\text{perf}} = \alpha_k\mathcal{L}_{\text{rank}} + (1-\alpha_k)\mathcal{L}_{\text{BCE}} + \lambda_{\text{e}}\mathcal{L}_{\text{e}}, \tag{18}$$

$$\alpha_k = \alpha_{\text{init}} + (\alpha_{\text{final}} - \alpha_{\text{init}}) \cdot \frac{k}{K} \tag{19}$$

where k is the index of the current training batch and K is the total number of training batches. α_{init} and α_{final} denote the initial and final values of α_k.

Table 1: **Main Results on MBPP and MATH Benchmarks.** We compare BiCSRouter with three groups of baselines. The *Single-Agent* group serves as the theoretical bounds. **Bold** values indicate the best results regardless of reference bounds. Results marked with $*$ are reported in MasRouter (Yue et al., 2025). Columns **Intra** and **Inter** indicate whether a method performs intra-regime routing or inter-regime routing.

		MBPP		MATH		Routing Capability	
Method	**Backbone**	**Pass@1 (%)**	**Cost**	**Acc. (%)**	**Cost**	**Intra**	**Inter**
I. Single-Agent Baselines (Reference Bounds)							
Chain-of-Thought (CoT)	GPT-4o mini	71.40	0.82	63.97	0.17	✗	✗
	GPT-5	91.40	2.87	91.71	1.78	✗	✗
II. Static Multi-Agent Architectures							
LLM Debate*	GPT-4o mini	73.60	–	64.68	–	✗	✗
MacNet (Complete)*	GPT-4o mini	75.20	–	67.63	–	✗	✗
AFlow*	GPT-4o mini	82.20	–	73.35	–	✗	✗
III. Adaptive Routing Methods							
RaterLLM	LLM Pool	80.20	1.22	69.94	1.10	✗	✓
HybridLLM	LLM Pool	69.60	0.06	69.94	0.20	✓	✗
GraphRouter	LLM Pool	76.60	0.13	79.00	0.32	✓	✗
Eagle	LLM Pool	80.80	3.39	83.82	3.32	✓	✗
RouterDC	LLM Pool	82.00	0.54	80.92	1.63	✓	✗
AutoMix	LLM Pool	83.60	1.70	75.53	0.85	✓	✗
BAMAS (low)	LLM Pool	75.00	1.16	66.34	2.57	✓	✗
BAMAS (high)	LLM Pool	75.40	1.77	69.14	3.95	✓	✗
RouteLLM	LLM Pool	87.30	0.99	64.55	3.61	✓	✗
MasRouter	LLM Pool	84.00	1.63	89.98	1.88	✓	✗
MixLLM	LLM Pool	85.00	3.31	84.59	1.58	✓	✗
Router-R1	LLM Pool	85.40	3.78	88.63	3.70	✓	✗
BiCSRouter (Ours)	LLM Pool	**89.40**	1.54	**90.37**	1.33	✓	✓

4.2.2 Topological Cost Modeling

Accurately estimating the cost of multi-agent systems is challenging due to the variable length of model responses and topological dependencies. We therefore propose a topology-aware token predictor that simulates the topological flow of the inference graph, where we model the total cost as the summation of costs incurred by each agent in the topological order, combining learnable token prediction with deterministic cost calculation.

Learnable Output Prediction The output token length $L_{\text{out},i}^{(t)}$ is predicted by an MLP $\Phi_{\text{len}}^{(t)}$ conditioned on the agent's configuration and the cumulative previous output length $L_{\text{prev}}^{(t)}$:

$$s_i^{(t)} = \Phi_{\text{len}}^{(t)}\left(\mathbf{e}_{m_i}^{(t)}, \mathbf{e}_{r_i}^{(t)}, \mathbf{f}_q, \mathbf{e}_{\tau_i}^{(t)}, L_{\text{prev}}\right), \quad (20)$$

$$\hat{L}_{\text{out},i}^{(t)} = L_{\min} + (L_{\max} - L_{\min})\sigma\left(s_i^{(t)}\right) \quad (21)$$

where $\sigma(\cdot)$ denotes the sigmoid function. $L_{\max}$ and $L_{\min}$ are two hyperparameters to control the prediction range.

Deterministic Input Calculation The input token length $L_{\text{in},i}^{(t)}$ is exactly calculated as:

$$L_{\text{in},i}^{(t)} = L_{\text{sys}} + L_q + L_{\text{ctx},i}^{(t)} \quad (22)$$

where L_{sys} and L_q represent the length of the system prompt and query q respectively. The accumulated context is determined by the agent interaction topology and we use the predicted value to simulate the process:

$$L_{\text{ctx},i}^{(t)} = \sum_{j \in \mathcal{N}_{\text{in}}(i)} \hat{L}_{\text{out},j}^{(t)} \quad (23)$$

where $\mathcal{N}_{\text{in}}$ denotes nodes whose outputs are routed to agent i under specific topology.

Given regime t, the predicted cost is derived as:

$$\hat{C}_t = \sum_{i=1}^{|\mathcal{G}_t|} \sum_{\text{d} \in \{\text{in},\text{out}\}} \hat{L}_{\text{d},i}^{(t)} \cdot \text{CPT}_{\text{d},i}^{(t)} \quad (24)$$

where $\text{CPT}_{\text{in},i}^{(t)}$ and $\text{CPT}_{\text{out},i}^{(t)}$ represent the exact input and output cost per token of the LLM assigned to the i-th agent. $|\mathcal{G}_t|$ denotes the execution turns under inference graph $\mathcal{G}_t$. The cost prediction module is then supervised via mean squared error:

$$\mathcal{L}_{\text{cost}} = \sum_{t \in \{ir,ec\}} \text{MSE}\left(\hat{C}_t, C_t\right). \quad (25)$$

4.3 Training Objective

Following the RCT-based training framework (Hu et al., 2024), we execute the sampled configurations

for both regimes to collect factual results. We train the entire framework end-to-end by jointly optimizing the policy generation and utility estimation. The total objective function is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{re}}\mathcal{L}_{\text{re}} + \lambda_{\text{perf}}\mathcal{L}_{\text{perf}} + \lambda_{\text{cost}}\mathcal{L}_{\text{cost}}. \quad (26)$$

Here, hyperparameters λ_{re}, λ_{perf}, and λ_{cost} balance policy improvement and estimation accuracy.

5 Experiments

5.1 Experimental Setup

Datasets For dataset selection, we use MBPP (Austin et al., 2021) for coding, and MATH (Hendrycks et al., 2021b) for mathematical reasoning. Following Yue et al. (2025), we adopt a 519-problem subset of MATH as the test set. For out-of-distribution (OOD) evaluation, we test on GSM8K (Cobbe et al., 2021) (356 sampled problems) and HumanEval (Chen et al., 2021). To show generality of BiCSRouter on natural language tasks, we further report experimental results on MMLU (Hendrycks et al., 2021a) and OpenBookQA (Mihaylov et al., 2018) in the Appendix E.

Baselines We compare against three baseline types: (1) **Single agents**: two representative LLMs from our pool using CoT reasoning (Wei et al., 2022); (2) **Static multi-agent systems**: LLM-Debate (Du et al., 2023), Mac-Net (Qian et al., 2025), and AFlow (Zhang et al., 2025c), using a representative LLM from our Extensive Collaboration regime as the backbone; (3) **Adaptive routing**: HybridLLM (Ding et al., 2024), GraphRouter (Feng et al., 2025), Eagle (Zhao et al., 2024b), RouterDC (Chen et al., 2024c), AutoMix (Aggarwal et al., 2024), RouteLLM (Ong et al., 2025), MixLLM (Wang et al., 2025), Router-R1 (Zhang et al., 2025b) with LLaMA-3.2-3B-Instruct as the policy model, BAMAS (Yang et al., 2025) with per-query budgets of $0.0015 (low) and $0.005 (high), MasRouter (Yue et al., 2025), and the rating-based routing method proposed by Gao et al. (2025) with Gemini 2.5 Flash (Comanici et al., 2025) as the rater, which we refer to as RaterLLM.

5.2 Performance of BiCSRouter

In this section, we compare BiCSRouter with selected baselines on two benchmarks. Overall, BiCSRouter consistently achieves strong performance across both datasets, demonstrating an effective balance between performance and inference cost.

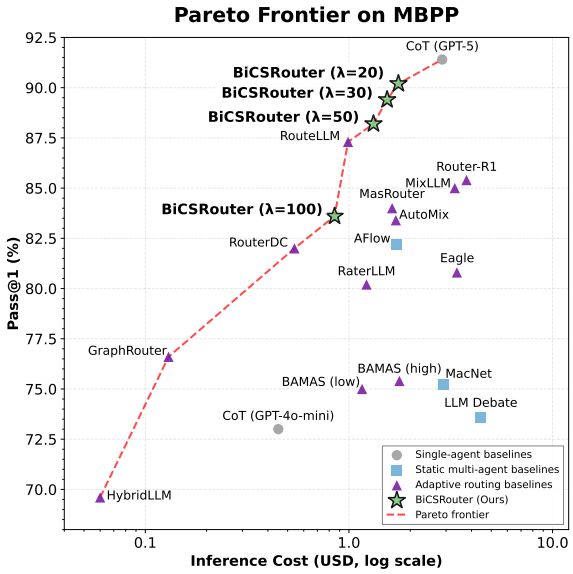

Figure 3: Performance–cost Pareto frontier on MBPP.

Regarding performance, Table 1 shows that BiCSRouter constructs a robust cross-system architecture by jointly leveraging intra-regime and inter-regime routing. BiCSRouter consistently outperforms both static multi-agent baselines and adaptive routing baselines on both datasets. On the MBPP dataset, BiCSRouter surpasses adaptive routing baselines by an average of 8.99% in Pass@1. It also outperforms MasRouter by 5.4% and RaterLLM by 9.2%, which handle only intra-regime and inter-regime routing respectively.

With respect to the performance–cost trade-off, Figure 3 shows that BiCSRouter achieves the best Pareto-optimal performance on the MBPP dataset among all practical baselines, second only to the reference-bound GPT-5 CoT. Compared to the reference-bound setting, BiCSRouter reduces inference cost by 46% with only a 2% decrease in accuracy on MBPP, and by 25% with only a 1.34% accuracy decrease on MATH. These results indicate that BiCSRouter effectively exploits two orthogonal computational regimes and achieves state-of-the-art balanced performance–cost efficiency.

5.3 Ablation Study

We conduct ablation experiments on BiCSRouter under four different settings: (1) w/o inter-regime, where inter-regime routing is disabled and regime selection is replaced by random routing under the same regime selection ratio; (2) w/o intra-regime, where intra-regime routing is disabled by removing $\mathcal{L}_{\text{re}}$ during training; (3) w/o cost predictor, where only the performance predictor is kept for

Table 2: **Ablation Study on MBPP and MATH.**

Variant	MBPP		MATH	
	Pass@1	Cost	Acc.	Cost
BiCSRouter	89.40	1.54	90.37	1.33
Routing Architecture				
w/o Inter-Regime	87.00	1.55	88.63	1.51
w/o Intra-Regime	85.60	1.57	86.51	1.14
w/o Cost Predictor	90.40	1.86	92.49	1.72
Regimes Analysis				
Fixed Regime (EC)	83.00	0.49	88.44	1.40
Fixed Regime (IR)	91.40	2.87	91.71	1.78

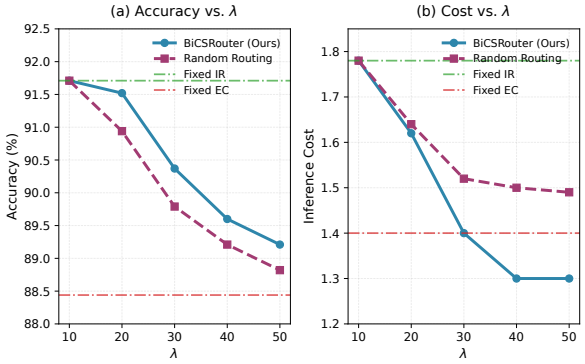

Figure 4: Sensitivity of BiCSRouter to the cost-penalty parameter λ on MATH. Cost denotes the inference cost.

inter-regime routing; (4) Fixed Regime, where only one regime is enabled with intra-regime learning preserved. As shown in Table 2, removing inter-regime routing leads to a noticeable performance drop accompanied by a slight increase in cost, indicating that the inter-regime module effectively captures regime heterogeneity and allocates queries more appropriately. Similarly, disabling intra-regime routing also degrades performance, with moderate cost fluctuations, suggesting that the intra-regime module plays an important role in organizing model behavior within each regime. Besides, removing the cost head improves accuracy but substantially increases inference cost. For the fixed-regime ablations, the performance of BiCSRouter on both datasets lies between the two fixed-regime baselines and is closer to the stronger one. Specifically, BiCSRouter achieves a superior performance–cost trade-off, particularly on the MATH dataset. Both fixed-regime models incur higher costs than BiCSRouter, demonstrating the effectiveness and efficiency of BiCSRouter.

5.4 Sensitivity Analysis

We analyze the sensitivity of BiCSRouter with respect to the cost-penalty parameter λ. We evaluate

Table 3: **Multi-system extension of BiCSRouter.** BiCS denotes the original setting with two regimes. Results are reported on the MATH dataset.

Method	#Regimes	Accuracy (%)	Cost
BiCS(IR + EC)	2	90.37	1.33
BiCS + LR	3	86.71	1.45
BiCS + AC	3	88.63	1.40
BiCS + LR + AC	4	89.60	1.43

model performance on the MATH dataset using five values of $\lambda \in \{10, 20, 30, 40, 50\}$. For each setting, we compare BiCSRouter against a random-routing baseline that preserves the same regime selection ratio. Additionally, we include the two fixed-regime results as reference lines in the figure. As shown in Figure 4, BiCSRouter exhibits a clear tendency toward more cost-efficient routing decisions as λ increases. BiCSRouter consistently outperforms random routing in both performance and cost-effectiveness, demonstrating the robustness of the inter-regime routing mechanism. These results indicate BiCSRouter reliably balances performance and cost across cost-penalty values.

5.5 Multi-System Extensibility

In this section, we demonstrate that BiCSRouter can be naturally extended to the multi-system setting. We further introduce two additional regimes to evaluate the scalability of BiCSRouter: (1) an efficient single-agent regime with only Limited Reasoning (LR) capability, and (2) a cost-intensive multi-agent regime featuring Advanced Collaboration (AC) among flagship models. As shown in Table 3, we observe that arbitrarily increasing the number of regimes does not necessarily lead to performance gains. The LR regime was rarely selected in practice (see Appendix D) and instead interfered with the model's ability to discriminate between the original two regimes. The AC regime failed to improve the handling of marginal cases by invoking the flagship repeatedly. Overall, expanding the number of regimes notably increases the difficulty of learning heterogeneity across regimes and enlarges the policy search space, which hinders stable convergence during training. These findings validate the design choice of our bi-level framework.

5.6 Out-of-Distribution Behavior

We trained our models on the MATH and MBPP datasets and evaluated their out-of-distribution (OOD) generalization on GSM8K and HumanEval.

Table 4: **OOD generalization results on GSM8K and HumanEval.** GPT-5 CoT serves as a reference upper bound and is not included in comparisons.

	GSM8K (OOD)		HumanEval (OOD)	
Method	**Acc.**	**Cost**	**Pass@1**	**Cost**
BiCSRouter	**95.79**	0.42	**92.25**	0.34
MasRouter	94.62	0.56	89.84	0.36
RaterLLM	94.38	0.43	89.15	0.34
GPT-5 CoT†	96.07	0.45	96.12	0.48

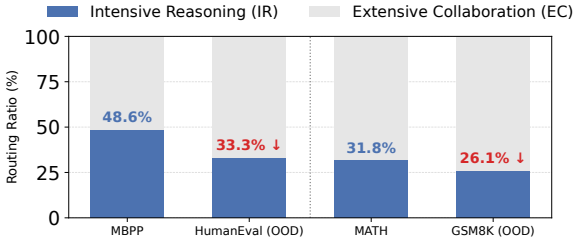

Figure 5: Regime selection of BiCSRouter under distribution shift.

We compare BiCSRouter against two representative adaptive routing baselines, MasRouter and RaterLLM, and use GPT-5 with Chain-of-Thought (CoT) as a reference-bound upper baseline.

As shown in Table 4, BiCSRouter consistently maintains a favorable performance–cost trade-off under OOD settings. Figure 5 further illustrates the distribution of regimes selected by BiCSRouter across different datasets. We observe that when transferring from the training datasets to relatively simpler OOD benchmarks, BiCSRouter adaptively reduces the selection ratio of Intensive Reasoning regimes, thereby achieving a more efficient balance between performance and computational cost.

6 Conclusion

In this paper, we formulate the Cross-System Routing Problem, highlighting the joint role of intra-regime and inter-regime routing in exploiting complementary strengths of heterogeneous computation regimes. Building on this, we propose BiCSRouter, a lightweight router integrating two orthogonal regimes: Intensive Reasoning and Extensive Collaboration. By optimizing intra-regime policies and learning inter-regime heterogeneity, BiCSRouter selects optimal model configurations for each query, achieving a favorable performance–cost trade-off. Our results show that structured regime selection is effective and scalable without unnecessary expansion of the search space. We hope BiCSRouter provides a principled foundation for system-level routing and inspires future research on efficient multi-system model selection.

Limitations

In our framework, we only consider a single powerful and expensive flagship model for constructing the Intensive Reasoning regime. However, prior studies suggest that smaller models can also achieve strong reasoning performance through enhanced reasoning strategies or sufficiently consistent and deep system topologies, potentially reducing costs without sacrificing accuracy. In addition, our experiments indicate that arbitrarily increasing the number of regimes does not lead to meaningful performance gains; instead, it amplifies inter-regime heterogeneity, enlarges the search space, and makes policy learning more difficult.

Ethical Considerations

While BiCSRouter improves performance–cost efficiency, it may inherit and propagate biases from the underlying LLMs through its routing decisions. In particular, the inter-regime router may systematically favor certain systems under specific query distributions, potentially amplifying biased behaviors. In addition, improved efficiency may lower the barrier to large-scale deployment, which could increase the risk of misuse in downstream applications. We encourage careful evaluation and responsible deployment.

Acknowledgements

We would like to thank Yuta Tomomatsu and Haruki Nagasawa for their valuable discussions and insights throughout the project.

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.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Liguo Chen, Qi Guo, Hongrui Jia, Zhengran Zeng, Xin Wang, Yijiang Xu, Jian Wu, Yidong Wang, Qing Gao, Jindong Wang, and 1 others. 2024a. A survey on evaluating large language models in code generation tasks. *arXiv preprint arXiv:2408.16498*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024b. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*. Featured Certification.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. 2024c. Routerdc: query-based router by dual contrastive learning for assembling large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. 2009. Ranking measures and loss functions in learning to rank. *Advances in neural information processing systems*, 22.

Ke Cheng, Wen Hu, Zhi Wang, Peng Du, Jianguo Li, and Sheng Zhang. 2024. Enabling efficient batch serving for lmaas via generation length prediction. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 853–864. IEEE.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Aaron Fanous, Jacob Goldberg, Ank Agarwal, Joanna Lin, Anson Zhou, Sonnet Xu, Vasiliki Bikia, Roxana Daneshjou, and Sanmi Koyejo. 2025. Syceval: Evaluating llm sycophancy. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 893–900.

Tao Feng, Yanzhen Shen, and Jiaxuan You. 2025. Graphrouter: A graph-based router for LLM selections. In *The Thirteenth International Conference on Learning Representations*.

Mingyan Gao, Yanzi Li, Banruo Liu, Yifan Yu, Phillip Wang, Ching-Yu Lin, and Fan Lai. 2025. Single-agent or multi-agent systems? why not both? *Preprint*, arXiv:2505.18286.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024b. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*.

Kai Mei, Wujiang Xu, Minghao Guo, Shuhang Lin, and Yongfeng Zhang. 2026. Omnirouter: Budget and performance controllable multi-llm routing. *SIGKDD Explor. Newsl.*, 27(2):107–116.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/.

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2025. Scaling large language model-based multi-agent collaboration. In *The Thirteenth International Conference on Learning Representations*.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1).

Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Saurabh Jha, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Tamer Başar, and Ravishankar K. Iyer. 2024. Efficient interactive llm serving with proxy model-based sequence length prediction. In *The 5th International Workshop on Cloud Intelligence / AIOps at ASPLOS 2024*, volume 5, pages 1–7, San Diego, CA, USA. Association for Computing Machinery.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Josep Torrellas, and Esha Choukse. 2025. Dynamollm: Designing llm inference clusters for performance and energy efficiency. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1348–1362. IEEE.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, Bangkok, Thailand. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2025. 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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. 2025. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *arXiv preprint arXiv:2509.05396*.

Liming Yang, Junyu Luo, Xuanzhe Liu, Yiling Lou, and Zhenpeng Chen. 2025. Bamas: Structuring budget-aware multi-agent systems. *Preprint*, arXiv:2511.21572.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuan-Jing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153.

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, Vienna, Austria. Association for Computational Linguistics.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2025a. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *The Thirteenth International Conference on Learning Representations*.

Haozhen Zhang, Tao Feng, and Jiaxuan You. 2025b. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025c. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*.

Kechi Zhang, Ge Li, Jia Li, Huangzhao Zhang, Jingjing Xu, Hao Zhu, Lecheng Wang, Yihong Dong, Jing Mai, Bin Gu, and 1 others. 2025d. Computational thinking reasoning in large language models. *arXiv preprint arXiv:2506.02658*.

Qinlin Zhao, Jindong Wang, Yixuan Zhang, Yiqiao Jin, Kaijie Zhu, Hao Chen, and Xing Xie. 2024a. Competeai: understanding the competition dynamics of large language model-based agents. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zesen Zhao, Shuowei Jin, and Z Morley Mao. 2024b. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.

Zangwei Zheng, Xiaozhe Ren, Fuzhao Xue, Yang Luo, Xin Jiang, and Yang You. 2023. Response length perception and sequence scheduling: An llm-empowered llm inference pipeline. *Advances in Neural Information Processing Systems*, 36:65517–65530.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*.

A Implementation Details

In our experiments, we set the learning rate to 0.001 and the temperature to 1. The initial value of the loss weight α_{init} is set to 0.7, and the final value α_{final} is set to 0.3. The regime-specific baseline b_t is initialized to 0.5. For the trade-off parameters, we set λ_{e}, λ_{re}, λ_{perf}, and λ_{cost} to 0.01, 0.5, 10, and 3, respectively. We empirically set $L_{\min}$ and $L_{\max}$ to 50 and 500. For the Extensive Collaboration regime and all adaptive routing baselines, the number of agents was fixed to 4. Since the static multi-agent baselines are limited to a GPT-4o mini backbone, we report the 6-agent results from (Yue et al., 2025) with the same dataset split as reference and do not compare costs. We used a random seed of 42 to split the training and validation set in a 9:1 ratio, and trained the models for up to 5 epochs with early stopping. For the cost-penalty coefficient λ, we explored the range $\{10, 20, 30, 40, 50\}$. For the Intensive Reasoning regime, we selected IO (input and output) and CoT as candidate reasoning strategies. For the Extensive Collaboration regime, we considered IO, Chain, Complete Graph, and Debate as candidate topologies. The role pool was constructed by referencing the role prompts provided by Yue et al. (2025).

B LLM Pool

We set up two LLM pools for the Intensive Reasoning regime and the Extensive Collaboration regime. For the Intensive Reasoning regime, to reflect the advantages of the latest single-agent in terms of reasoning depth, we used GPT-5 (OpenAI, 2025) as the backbone. For the Extensive Collaboration regime, to leverage the efficient collaboration of the multi-agent system, we used GPT-4o mini (OpenAI, 2024), Gemini 2.5 Flash-lite (Comanici et al., 2025), and DeepSeek-V3.1 (Liu et al., 2024a) as the candidate LLM pool. For routers that require two LLMs (e.g., AutoMix and RouteLLM), we uniformly use GPT-4o mini as the small model and GPT-5 as the large model in our implementation to ensure fair comparisons.

C Inference Workflow

We present the complete inference workflow of BiCSRouter in Algorithm 1.

D Detailed Regime Selection Distribution

Complementing the discussion in Section 5.5 concerning Multi-System Extensibility, we provide a detailed breakdown of the regime selection ratios in Figure 6. While our framework progressively integrates the additional regimes into its decision space, the empirical distribution reveals significant inefficiencies in their utilization. As shown in the figure,

Algorithm 1 BiCSRouter Inference Workflow

Input: Query q, Subspaces Ω_{ir}, Ω_{ec}, Policies π_{ir}, π_{ec}. Neural Modules: Shared Encoder Φ_{share}, Prediction Heads Φ_{head}, Length Predictor Φ_{len}.
Output: Final Response y.

```
// Phase 1: Semantic Encoding
f_q ← TextEncoder(q)
// Phase 2: Intra-Regime Configuration Sampling
c_ir ~ π_ir(·|f_q; Ω_ir) ;      // Routing within ir regime
c_ec ~ π_ec(·|f_q; Ω_ec) ;      // Routing within ec regime
// Phase 3: Decoupled Utility Estimation
// 1. Dual-Head Performance Prediction
h_shared ← Φ_share(f_q) ;                          // Eq.(12)
(P̂_ir, P̂_ec) ← Φ_head(h_shared, c_ir, c_ec) ;      // Eq.(13)
// 2. Topology-Aware Cost Estimation
foreach t ∈ {ir, ec} do
    Initialize total cost Ĉ_t ← 0 ; // Simulate inference
    foreach node i in TopoSort(c_t) do
        L_ctx ← Σ_{j∈N_in(i)} L̂^(t)_out,j ;                    // Eq.(23)
        L^(t)_in,i ← L_sys + L_q + L_ctx ;                     // Eq.(22)
        L̂^(t)_out,i ← Φ^(t)_len(c_t, f_q, L^(t)_in,i) ;        // Eq.(20, 21)
        Ĉ_t ← Ĉ_t + (L^(t)_in CPT_in,i + L̂^(t)_out CPT_out,i) ;  // Eq.(24)
    U_t ← P̂_t − λ · Ĉ_t
// Phase 4: Routing and Execution
if U_ec > U_ir then
    t* ← ec;  G ← ConstructGraph(c_ec)
else
    t* ← ir;  G ← ConstructGraph(c_ir)
y ← ExecuteGraph(G, q) ;                           // Execution
return y
```

the Limited Reasoning regime is selected with negligible frequency (approximately $2.7\% - 3.9\%$), indicating that it lacks sufficient complementarity with the existing regimes to be prioritized by the router. Furthermore, the Advanced Collaboration regime tends to cannibalize the selection of the Intensive Reasoning regime rather than reducing the reliance on the costly Extensive Collaboration regime.

E Extended Evaluation on Natural Language Understanding Tasks

To further validate the scalability and generalization ability of BiCSRouter, we extend our evaluation to natural language understanding tasks beyond the reasoning-intensive benchmarks presented in the main paper.

While the primary experiments focus on coding and mathematical reasoning tasks (MBPP and MATH), we expand the evaluation along two dimensions. First, for cross-domain scale expansion, we evaluate on MMLU (Hendrycks et al., 2021a), a comprehensive benchmark spanning 57 academic

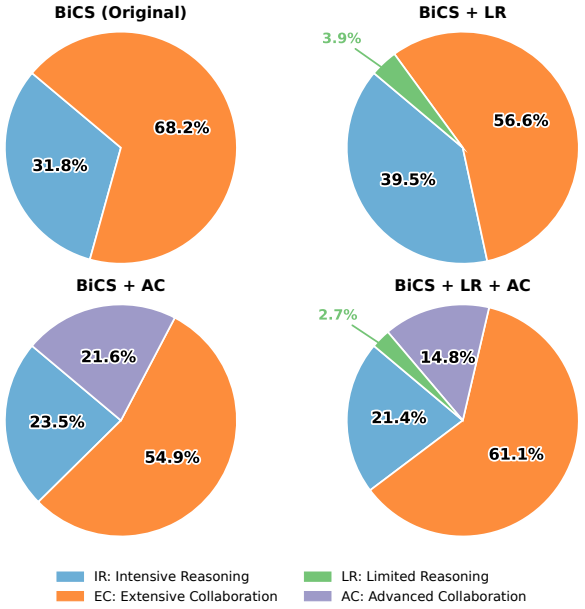

Figure 6: Regime Selection Ratios across Multi-System Settings.

Table 5: Results on MMLU and OpenBookQA. **Bold** values denote the best performance, while underlined values indicate the second-best performance.

	MMLU		OpenBookQA	
Model	**Acc. (%)**	**Cost ($)**	**Acc. (%)**	**Cost ($)**
BAMAS (low)	85.83	5.51	94.00	4.33
BAMAS (high)	86.41	5.54	**96.00**	4.32
MasRouter	86.87	2.81	95.20	0.85
MixLLM	87.07	5.07	90.60	1.27
Eagle	**87.72**	6.53	95.00	1.21
BiCSRouter (Ours)	87.33	2.29	**96.00**	0.96

disciplines, including humanities, social sciences, and STEM fields. To ensure representative coverage while maintaining computational feasibility, we sample approximately 1,500 instances from the official validation split. Second, for task type expansion, we include OpenBookQA (Mihaylov et al., 2018), a widely used question-answering benchmark that emphasizes commonsense reasoning and the integration of external knowledge. Evaluation is conducted on the complete 500-example test set.

We compare BiCSRouter against several representative adaptive routing methods, including MasRouter, BAMAS, MixLLM, and Eagle. These approaches provide strong and diverse references for evaluating routing effectiveness across different task distributions.

Table 5 presents the performance and inference cost on MMLU and OpenBookQA. The results show that BiCSRouter achieves competitive accuracy across both benchmarks while maintaining

a clear advantage in inference cost compared to most baselines. Although it does not consistently achieve the highest accuracy, it demonstrates a favorable performance–cost trade-off. In particular, BiCSRouter achieves strong performance on OpenBookQA and competitive results on MMLU with substantially lower cost than several representative routing approaches. These findings indicate that the routing strategy remains stable and effective across diverse task distributions. Therefore, BiCSRouter is not limited to mathematical and code reasoning tasks but also generalizes well to broader natural language understanding scenarios, supporting the scalability and robustness of the proposed framework.

F Dataset Statistics

This section presents detailed statistics of the datasets used in our experiments, as summarized in Table 6. For MATH, we construct a subset by randomly sampling approximately 10% of the original test split using a fixed random seed of 42, while preserving the original difficulty-level distribution across categories. For GSM8K, we randomly sample 30% of the dataset using the same fixed random seed. For MMLU, we utilize the dev split for training, which contains 285 questions spanning 57 academic subjects, and evaluate on the validation split. For OpenBookQA, we select the first 300 samples from the official training set for training and evaluate on the standard test set.

Table 6: **Dataset Statistics.** We use subsets of MATH and MBPP for training and validation, while GSM8K and HumanEval are employed as out-of-distribution (OOD) evaluation benchmarks.

Dataset	Train	Val	Test
MATH (Subset)	688	77	519
MBPP	336	38	500
MMLU (Dev, Validation)	257	28	1,531
OpenBookQA	270	30	500
Out-of-Distribution Evaluation			
GSM8K (Subset)	–	–	356
HumanEval	–	–	129

G Intra-Regime Reinforcement Learning Details

To improve reproducibility, we provide implementation details of the intra-regime reinforcement learning optimization, including hyperparameters, variance reduction techniques, and convergence diagnostics.

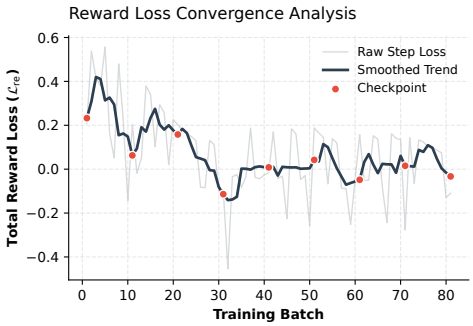

Figure 7: Reward loss over 80 training batches.

Hyperparameters. We employ the Adam optimizer with an initial learning rate of 1×10^{-4} and a batch size of 32. A StepLR scheduler is adopted, decaying the learning rate every two epochs with a decay factor of 0.8. No warmup or additional scheduling strategies are used.

Variance Reduction. To reduce gradient variance, we employ an exponential moving average baseline. The baseline b_t in Equation (11) is updated as:

$$b_{t+1} \leftarrow 0.9\, b_t + 0.1 \cdot \mathrm{avg}(R(c_t)), \qquad (27)$$

where $R(c_t)$ is the reward defined in Equation (10) as a weighted combination of performance and cost, and $\mathrm{avg}(R(c_t))$ denotes the average reward within the current batch.

Convergence Diagnostics. To verify training stability, we plot the reward loss over 80 batches during training on MBPP dataset as shown in Figure 7.

H Case Study

In this section, we examine the behavior of inter-regime routing on the MATH dataset under three representative scenarios. **(1) Reasoning Requirement.** The Intensive Reasoning regime produces a correct answer, whereas the Extensive Collaboration regime fails. In this case, the router should identify the inferior performance of the Extensive Collaboration regime and select the Intensive Reasoning regime. **(2) Cost Efficiency.** Both regimes generate correct answers. The router is expected to select the regime with lower computational cost based on cost prediction. **(3) Marginal Cases.** The Extensive Collaboration regime leverages collaboration among smaller models to produce an answer that outperforms the flagship model used in the

Table 7: **Case Study of Inter-Regime Router Decisions.** We compare the Intensive Reasoning (IR) and Extensive Collaboration (EC) regimes. The **Router Decision** column visualizes the actual selection process. The symbols ✓ and ✗ denote the optimal and suboptimal selections, respectively.

Problem Specification (Type & Level)	**IR (Acc | Cost)**	**EC (Acc | Cost)**	**Router Decision**
Case 1: Reasoning Requirement (IR Correct, EC Incorrect)			
Precalculus (L3): A 135° rotation around the origin is applied to $\sqrt{2} - 5\sqrt{2}i$. What is the resulting complex number?	**1.0** | \$0.0028	0.0 | \$0.0017	
Precalculus (L2): Solve $\cos 3x = 1$ for $0 \le x \le 2\pi$. Enter all the solutions, separated by commas.	**1.0** | \$0.0048	0.0 | \$0.0018	
Case 2: Cost Efficiency (Both Correct, Router selects cheaper)			
Prealgebra (L5): What is the reciprocal of $0.\overline{714285}$? Express your answer as a decimal.	1.0 | **\$0.0019**	1.0 | \$0.0020	
Precalculus (L5): Let θ be an acute angle such that $\sin 5\theta = \sin^5 \theta$. Compute $\tan 2\theta$.	1.0 | \$0.0028	1.0 | **\$0.0019**	
Case 3: Marginal Cases (EC Correct, IR Incorrect)			
Int. Algebra (L4): Find $a + b + c$ such that $x^4 + ax^3 + \ldots$ are both squares of polynomials.	0.0 | \$0.0044	**1.0** | **\$0.0030**	
Precalculus (L5): Solve equation $\arctan x + \arccos \frac{y}{\sqrt{1+y^2}} = \ldots$ for (a, b, c).	0.0 | \$0.0019	**1.0** | **\$0.0010**	

Intensive Reasoning regime. In this scenario, the router should favor the Extensive Collaboration regime.

Table 7 presents two representative examples for each scenario, illustrating the decision-making behavior of the proposed inter-regime routing mechanism.