NG-Router: Graph-Supervised Multi-Agent Collaboration for Nutrition Question Answering

Kaiwen Shi[1*], **Zheyuan Zhang**[1*], **Zhengqing Yuan**[1], **Keerthiram Murugesan**[3],
Vincent Galassi[1], **Chuxu Zhang**[2], **Yanfang Ye**[1†]
[1]University of Notre Dame, [2]University of Connecticut, [3]IBM Research,
[*]Equal Contribution [†]Corresponding Author
{kshi3, yye7}@nd.edu,

Abstract

Diet plays a central role in human health. Nutrition Question Answering (QA) has emerged as a promising paradigm to deliver personalized dietary guidance and prevent diet-induced chronic conditions, yet despite recent progress, existing approaches struggle with two key limitations: 1) the limited capacity of single agents for domain-specific reasoning and the inherent design complexity of multi-agent systems; 2) the overload of contextual information that dilutes downstream decision-making. In this paper, we introduce **Nutritional-Graph Router (NG-Router)**, a framework that formulates nutritional QA as a supervised, knowledge-graph–guided multi-agent collaboration problem. Our approach integrates agent nodes into knowledge graphs and employs a graph neural network to learn task-aware routing distributions over agents, leveraging soft supervision derived from empirical agent performance. To further mitigate contextual overload, we incorporate a gradient-based subgraph retrieval mechanism that identifies salient evidence during training, thereby enhancing reasoning in multi-hop and relational settings. Extensive experiments across multiple benchmarks and backbones demonstrate that NG-Router consistently surpasses both single-agent and ensemble baselines, achieving robust and generalizable improvements. By embedding collaboration schemes directly into graph-supervised signals, our method offers a principled path toward domain-aware multi-agent reasoning for complex nutritional health tasks. Our code repo can be found here.

1 Introduction

Diet is one of the most influential determinants of human health, shaping both disease prevention and long-term well-being. Yet, unhealthy eating remains widespread despite extensive public awareness of the benefits of proper nutrition (WHO, 2021). In the United States, for instance, an estimated 42.4% of adults are classified as obese (CDC, 2020). Globally, poor dietary patterns were associated with more than 11 million deaths in 2017, alongside millions of disability-adjusted life-years (DALYs), often linked to factors such as excessive sodium intake (Afshin et al., 2019; WHO, 2023). These figures highlight the pressing need for large-scale interventions that foster healthier eating behaviors (Figure 1 (a)). To address challenges in personalized nutritional health, Nutrition Question Answering (QA) has gained traction as a practical solution due to its accessibility, low entry barrier, and interactive nature (Min et al., 2022; Bondevik et al., 2024). Advances in Large Language Models (LLMs) further strengthen this direction by enabling richer reasoning for personalized dietary guidance (Ye et al., 2025). Although recent datasets and tasks (Bölz et al., 2023; Zhang et al., 2025b) have strengthen the field from a data-centric perspective, the effective use of these benchmarks remains constrained by two major research gaps:

Complexity of Domain-Specific Reasoning. Simple single-agent reasoning is usually insufficient for handling the medical and nutritional complexity required in personalized dietary guidance. While standalone LLM agents can perform well in general domains, growing evidence shows that coordinated systems of multiple agents often yield superior outcomes (Hong et al., 2024; Zhong et al., 2024; Ma et al., 2025). Multi-agent frameworks (MAS) leverage diverse and complementary roles to improve reasoning accuracy, exploratory depth, and robustness (Shinn et al., 2023; Qian et al., 2025; Wang et al., 2025a). However, designing such systems is far from trivial. A pervasive challenge lies in determining which agents should participate in collaboration, as literature consistently shows that the performance of different agents and backbones varies substantially across tasks, and there is no universally optimal solution (Feng et al., 2025). This

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7508–7527
March 24-29, 2026 ©2026 Association for Computational Linguistics

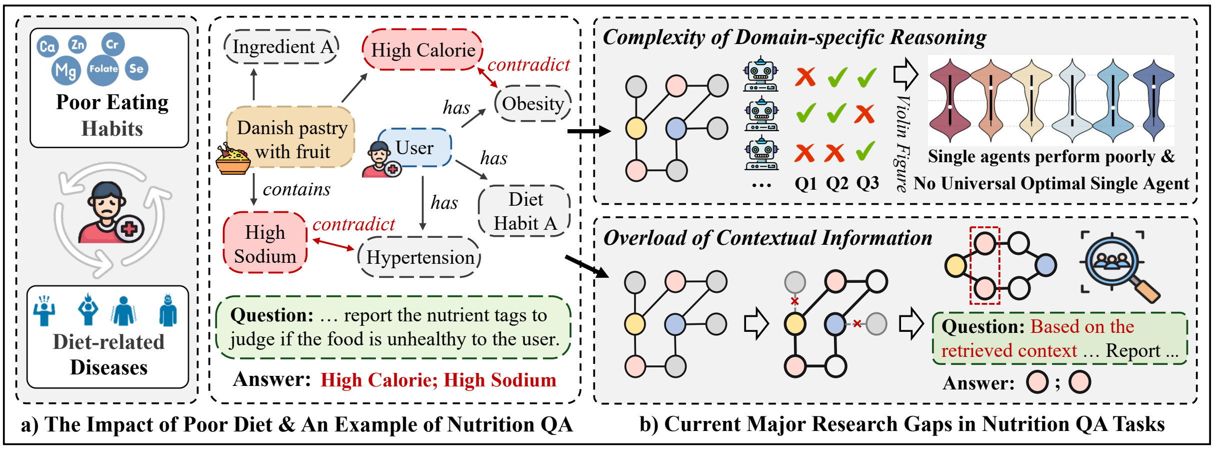
Ca
Zn
Cr
Mg
Folate
Se
Poor Eating Habits
Diet-related Diseases
Ingredient A
High Calorie
contradict
has
Obesity
Danish pastry with fruit
User
has
contains
has
Diet Habit A
High Sodium
contradict
Hypertension
Question: … report the nutrient tags to judge if the food is unhealthy to the user.
Answer: High Calorie; High Sodium
Complexity of Domain-specific Reasoning
…
Q1 Q2 Q3
Violin Figure
Single agents perform poorly & No Universal Optimal Single Agent
Overload of Contextual Information
Question: Based on the retrieved context … Report …
Answer:
a) The Impact of Poor Diet & An Example of Nutrition QA
b) Current Major Research Gaps in Nutrition QA Tasks

Figure 1: Illustration of Nutrition QA challenges. (a) Poor dietary habits lead to health risks and complex user–food–condition interactions, as shown in a personalized QA example. (b) Two key research gaps emerge: (i) domain-specific reasoning requires multiple complementary agents, with no single model performing optimally across queries; and (ii) excessive and unstructured contextual information hampers accurate retrieval and reasoning.

limitation becomes especially problematic in domains like healthcare, where domain knowledge is required for delicate reasoning (Figure 1 (b)).

Overload of Contextual Information. Health-aware dietary reasoning typically involves an overwhelming volume of domain-specific information, such as medical conditions, nutritional profiles, food attributes, and condition-specific constraints (Zhang et al., 2025b). When this context is presented in full to an LLM or multi-agent system, the reasoning process can become diluted or misdirected, leading to inefficiency and factual errors (Jin et al., 2024; Jiang et al., 2024; Peng et al., 2024). Effective personalized guidance therefore depends not only on reasoning ability but also on accurately retrieving and prioritizing the most relevant information from the broader context (Guo et al., 2024; Wen et al., 2023). Without mechanisms to filter, structure, and surface key evidence, model performance degrades, and the quality of generated answers suffers accordingly (Figure 1 (b)).

To tackle the two aforementioned challenges, we adopt the Graph QA setting from the NGQA benchmark (Zhang et al., 2025b) and introduce **Nutritional-Graph Router (NG-Router)**, a framework that casts nutritional question answering as a supervised, knowledge-graph guided multi-agent collaboration problem. To address the first challenge, we integrate agent nodes into the KG and design a heterogeneous graph neural network (GNN) that propagates information across node types and produces task-aware routing distributions over agents. Rather than assigning agents deterministically, the router leverages soft supervision derived from empirical agent performance to learn dynamic probability distributions, and it generates final answers via weighted aggregation of agent outputs. To address the second challenge, we extend the pipeline with a gradient-based subgraph retrieval mechanism. During training, gradient signals supervise the retrieval of salient subgraphs, effectively filtering away irrelevant context. This design is motivated by prior findings showing that such retrieval strategies are especially advantageous for complex multi-hop reasoning (Jin et al., 2024; Jiang et al., 2024; Peng et al., 2024).

Overall, our method departs from heuristic voting or LLM-based judging by learning collaboration schemes directly from supervised graph signals. This enables the incorporation of domain structure into agent coordination without requiring handcrafted prior knowledge, while the gradient-guided retrieval substantially reduces contextual overload for downstream reasoning. Extensive experiments demonstrate that NG-Router consistently surpasses single-agent and ensemble baselines and generalizes well across both benchmarks and model backbones. Our contributions can be summarized as follows:

- **KG–Driven Agent Collaboration.** In this paper, we present a novel framework that converts multi-agent question answering into domain-specific knowledge graphs, where nodes represent not only queries and agents but also fine-grained entities and contextual interactions. We further train the graphs to learn adaptive collaboration strategies tailored to heterogeneous agent capabilities.
- **Graph-Supervised Subgraph Retrieval.** To

enhance downstream reasoning, we propose a retrieval mechanism that leverages gradient-based supervision during KG training. By identifying important nodes, the method extracts the most relevant subgraphs, enabling more precise and context-aware QA.

- **Extensive Empirical Validation.** Comprehensive experiments in the Nutritional QA domain demonstrate that our collaboration framework consistently surpasses both the strongest single-agent models and state-of-the-art baselines across multiple tasks.

2 Related Works

2.1 Task-Adaptive Agent Selection.

With the rapid development of LLMs and agentic frameworks, a growing body of work shows that no single model consistently dominates across tasks; instead, agents exhibit *complementary strengths*. Multi-agent systems such as AgentVerse demonstrate collaborative gains over individual agents (Chen et al., 2024c), while ReConcile shows that organizing diverse LLMs into rounds of discussion with consensus voting improves reasoning (Chen et al., 2024a). These findings motivate *input-conditioned selection and coordination*—i.e., routing or assembling specialized agents per query. Early efforts used static ensembling or binary collaboration (e.g., self-consistency (Wang et al., 2022)) to exploit diversity, but these approaches predefine model sets and lack input adaptivity, offering limited guidance on agent prioritization (Jiang et al., 2023a). More recent work trains learned routers that select LLMs based on query characteristics. RouteLLM learns routing from human preference data (Ong et al., 2025), RouterDC uses dual contrastive learning to assemble multiple LLMs (Chen et al., 2024b), and MixLLM treats routing as a contextual bandit problem for dynamic adaptation (Wang et al., 2025b). Other methods focus on coordinating multiple agents rather than selecting a single one: TO-Router and BEST-Route determine which experts to involve based on query difficulty instead of fixed pipelines (Stripelis et al., 2024; Ding et al., 2025). However, many of these approaches still rely on heuristic rules or shallow controllers and rarely model richer inter-dependencies among tasks, queries, and agents. A newer line of work frames routing as a structured learning problem (Zhang et al., 2025c). GraphRouter, for example, formulates routing as link prediction on a heterogeneous graph and uses GNNs to model query–model and inter-model relations (Feng et al., 2025). Although this marks progress beyond simple heuristic methods , it still struggles to integrate fine-grained task semantics or supervised graph signals that more directly guide adaptive collaboration across diverse agent designs. More can be seen in Appendix A

2.2 Prior Works in Nutrition Personalization.

With growing awareness of the importance of dietary health, various studies have sought to incorporate health metrics into applications such as food recommendation systems (Tian et al., 2022a,b, 2021). These approaches can be grouped into three primary categories. First, some research emphasizes single indicators like calorie or fat content, as highlighted in works by Ge et al. (Ge et al., 2015) and Shirai et al. (Shirai et al., 2021), though such metrics often fail to represent the multifaceted nature of a balanced diet. Second, simulated health data has been utilized, as demonstrated by Wang et al. (Wang et al., 2021), but these methods often diverge from real-world data distributions. Finally, recent studies have applied global health guidelines to develop composite health scores, such as (Bölz et al., 2023; Zhang et al., 2024a). However, foods deemed healthy by general standards can still negatively affect certain individuals (Yue et al., 2021; Zhang et al., 2024c), highlighting the absence of a universal solution. Beyond this, there is yet an effective solution to address the overload contextual information this domain challenge brings.

3 Preliminary

3.1 NGQA Benchmark Extension

Nutritional Graph Question Answering (NGQA) benchmark is a set of graph-based question answering datasets for personalized nutritional health, constructed from the National Health and Nutrition Examination Survey (NHANES) and the Food and Nutrient Database for Dietary Studies (FNDDS) data, which evaluates whether a food is healthy for a specific user by linking medical conditions, dietary behaviors, and nutritional profiles. It supports a variety of reasoning tasks across varying question complexities and establishes a comprehensive evaluating system for nutritional question answering task. In NGQA benchmark, each question has a context knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ includes different types of nodes such as food items, user profile, health conditions and nutrition tags. In this paper, we further extend query node $\mathcal{V}_Q$ and agent nodes $\mathcal{V}_A$ into the knowledge graph. Specifically, agent nodes $\mathcal{V}_A$ represent candidate agents, each defined by its prompts and strategy description, while query nodes $v_q \in \mathcal{V}_Q$ are initialized by a contextual encoder. As such, all nodes are embedded into the shared textual space to make message passing meaningful.

To further capture both the static semantic structure of the domain and the dynamic preferences of agent routing, we define: 1) Query–entity edges, which connect each query to the entities it explicitly mentions, thereby grounding the question within its evidential context. 2) Agent–entity edges, which capture the perspectives of different agents. By prompting each agent to identify the entities it finds most relevant, we encode how distinct agents selectively attend to contextual information. 3) Query–agent edges, which are not fixed but remain *trainable*. These edges carry the routing signal that the model learns to optimize; their existence and weights determine which agents are activated for collaboration and how their contributions are combined. In this formulation (Figure 2), the frozen edges ensure contextual grounding, while the adaptive query–agent edges provide a mechanism for dynamic and task-specific routing.

3.2 Problem Formulation

Formally, we study the problem of designing an optimized LLM-based agent collaboration scheme for a fixed graph QA task. Let $\mathcal{A} = \{a_1, \dots, a_n\}$ denote the pool of available agents (each agent defined by a backbone LLM and an interaction strategy/prompting style), $\mathcal{X}$ the input/query space, $\mathcal{G}$ the context graphs of the queries, and $\mathcal{Y}$ the output space. For $x \in \mathcal{X}$, each agent $a \in \mathcal{A}$ produces a candidate $y_a(x, \mathcal{G}) \in \mathcal{Y}$. Our overarching goal is to learn an optimal weighted combination of agents in $\mathcal{A}$ that maximizes task-level performance in the fixed downstream setting. Specifically, within this KG, the problem reduces to learning a function f_θ that scores query–agent pairs by propagating signals along graph edges:

$$s(q, a) = f_\theta(q, a; \mathcal{G}),$$

where $s(q, a)$ estimates the utility of including agent a when solving query q for the given task and $\mathcal{G}$ here represents the corresponding context graph. The model then computes a weighted combination of agents:

$$\hat{y}(q) = \phi(\{\, y_a(q),\, w_a(q) \,:\, a \in \mathcal{A}\}) \,,$$

with weights $w_a(q) \propto \exp(s(q, a))$, and ϕ denoting an aggregation rule such as voting, reranking, or learned fusion. Framing the problem in this way refines the high-level goal of "finding the best collaboration scheme" into the concrete task of learning graph-supervised scores for query–agent pairs, which in turn yield optimized weightings of agents for a fixed downstream task.

It is worth noting our formulation builds on the assumption that no single agent or backbone uniformly dominates; rather, their strengths and weaknesses are task- and backbone-dependent. This premise is supported by extensive prior works (Chen et al., 2024c,a) and later in our experiments, which consistently shows that different LLMs or prompting strategies excel in different scenarios.

4 Methodology

4.1 Agent Collaboration Training

Given the constructed knowledge graph, the central challenge is to train a router that can decide which agents are most valuable for solving a given query. Prior approaches often depend on manually crafted rules or the use of LLMs as external judges. Such heuristics lack adaptability and fail to capture complex contextual signals. In contrast, we frame routing as a supervised learning problem, allowing the model to discover fine-grained dependencies among queries, entities, and agents, rather than relying on rigid voting schemes.

Our solution employs a heterogeneous Graph Neural Network designed for type-sensitive message passing. Each node is first mapped into a unified latent representation through a type-specific projection $\mathrm{Proj}_{\tau(v)}$. For an edge $(u \xrightarrow{\psi} v)$ of relation type ψ, the propagated message is computed as

$$m^{(l,\psi)}_{u\to v} = \mathrm{Proj.}\Big(W^{(l)}_{\psi}\, h^{(l-1)}_{u}\Big) \,,$$

and neighbor messages of the same type are aggregated by averaging. The contributions of different edge types are then merged with learnable gates, yielding the update

$$h^{(l)}_{v} = U^{(l)}_{\tau(v)}\Big(h^{(l-1)}_{v} \,\|\, \sum_{\psi\in\Psi(v)} w^{(l)}_{\psi} \cdot \tilde{m}^{(l,\psi)}_{v}\Big),$$

where $\tau(v)$ indicates node type, $w^{(l)}_{\psi}$ is a trainable coefficient per edge type, $U^{(l)}_{\tau(v)}$ is a type-dependent

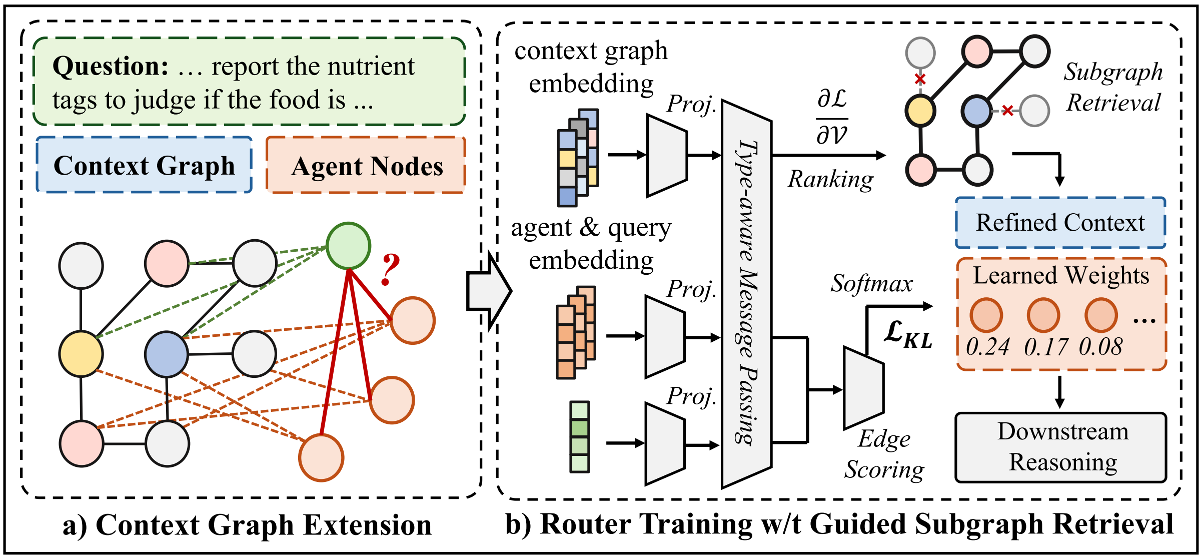

Figure 2: Overview of our proposed framework. (a) shows the KG extension process, where QA instances are extended into context graphs with query and agent nodes linked to nutritional entities. (b) shows our type-aware GNN router, which propagates contextual signals and guides gradient-based subgraph retrieval, refining the graph and producing agent importance weights for downstream collaboration and reasoning.

update operator, and || denotes concatenation. This mechanism enables each node to refine its embedding by combining its prior state with heterogeneous relational evidence.

After L layers, the query representation $h_q^{(L)}$ incorporates contextual information, while agent embeddings $h_a^{(L)}$ encode their relative competence for the query. A routing score is then assigned by

$$s(q, a) = \mathrm{MLP}\left(h_q^{(L)} \,\|\, h_a^{(L)}\right),$$

which is normalized over all agents with a softmax:

$$p_\theta(a \mid q, \mathcal{G}) = \mathrm{softmax}_{a \in \mathcal{A}}\big(s(q, a)\big).$$

Supervision is provided by empirical agent performance. For each query, agents are evaluated and their F1 scores converted into a target distribution $p^*(a \mid q)$ using a temperature-scaled softmax. This produces smoother guidance than one-hot labels, emphasizing not only the best agent but also secondary contributors. Training minimizes the KL divergence

$$\mathcal{L}_{\mathrm{KL}}(q) = \sum_{a \in \mathcal{A}} p^*(a \mid q) \, \log \frac{p^*(a \mid q)}{p_\theta(a \mid q, \mathcal{G})}.$$

KL divergence is particularly well-suited here: unlike cross-entropy, which concentrates solely on the top class, or mean squared error, which poorly models probability distributions, KL captures the full relative structure among agents and maintains informative gradients. This encourages the router to learn balanced allocation across complementary agents, stabilizing training and promoting collaborative diversity.

During inference, the router outputs the distribution $p_\theta(a \,|\, q, \mathcal{G})$ over agents. Final predictions are assembled by weighted ensembling:

$$\hat{y}(q) = \phi(\{\, y_a(q),\, p_\theta(a \mid q, \mathcal{G}) : a \in \mathcal{A}\}),$$

where $y_a(q)$ denotes agent a's answer and ϕ is a weighted aggregation rule such as probabilistic voting. Crucially, this paradigm moves beyond explicit inter-agent communication. Instead, it simulates a targeted brainstorming process: the router identifies the most adept agents and assigns them varying degrees of "voting power" based on their specific capabilities. By granting higher authority to experts while suppressing less relevant voices, the system determines the final answer through a capability-weighted consensus. Thus, the learned distribution directly dictates the influence each agent exerts, yielding a principled, context-aware collaboration strategy.

4.2 Graph-Supervised Subgraph Retrieval

A critical challenge in nutritional QA is the excessive volume of contextual information, where presenting the full graph to the router dilutes reasoning and introduces noise. To address this, we further propose a subgraph retrieval module that leverages training signals to identify and retain the most salient entities for each query.

Method	Sparse			Standard			Complex		
	Accuracy	Precision	F1	Accuracy	Precision	F1	Accuracy	Precision	F1
KAPING	17.53	20.75	33.94	45.93	46.24	62.72	68.83	71.29	80.93
ToG	24.39	29.86	43.33	61.89	67.93	74.64	61.53	81.19	73.03
Raw	29.34±0.74	32.51±0.59	49.41±0.91	64.11±0.31	79.29±0.47	82.40±0.42	72.03±0.23	72.96±0.24	81.86±0.18
CoT	29.21±1.43	32.54±1.12	48.10±1.49	62.91±0.86	76.51±0.97	80.27±0.64	72.40±0.16	72.93±0.16	82.05±0.18
MAD	27.10±1.32	30.59±1.24	46.39±1.14	61.73±0.54	76.05±0.50	80.34±0.34	71.80±0.27	72.77±0.36	81.88±0.27
React-Reflect	25.14±1.39	28.81±1.37	47.72±0.53	66.67±0.59	82.36±0.56	84.89±0.27	71.92±0.34	72.80±0.03	81.73±0.19
SC	25.71±0.29	29.64±0.38	45.52±0.10	68.76±0.20	85.81±0.32	86.36±0.19	72.75±0.39	73.43±0.39	82.45±0.28
Summary	27.45±0.77	31.23±0.76	47.12±1.25	66.72±0.19	84.00±0.31	85.45±0.17	71.28±0.65	72.56±0.35	81.60±0.36
Majority Vote	22.61±0.13	25.58±0.17	44.59±0.28	57.35±0.43	61.79±0.33	75.26±0.20	52.43±0.77	77.50±0.61	71.73±0.29
HybridLLM	28.02±0.59	30.91±0.42	47.76±0.74	34.45±0.12	34.45±0.12	49.25±0.22	72.75±0.29	73.19±0.14	82.31±0.21
LLM-Blender	21.20±0.98	24.26±0.95	46.73±1.43	45.72±0.55	50.90±0.32	68.32±0.83	51.79±0.17	72.02±0.21	75.50±0.69
NG-Router w/o SR	29.11±0.32	32.24±0.40	49.71±0.65	76.33±1.17	78.21±1.01	89.14±0.59	74.63±0.19	77.58±0.12	85.95±0.06
NG-Router w/ SR	**57.49±0.25**	**61.58±0.28**	**75.29±0.17**	**94.10±2.58**	**97.65±1.71**	**97.73±1.24**	**88.19±1.92**	**95.04±0.18**	**92.53±1.24**

Table 1: Performance comparison across Sparse, Standard, and Complex settings. We report both the version with and without the subgraph retrieval (SR). Additionally, we report the mean and standard deviation for all results for three runs. Best results are in **bold**, second best are underlined.

Let $\mathcal{V}_E$ denote entity nodes and $\mathbf{h}_i$ the last layer of the embeddings of entity $v_i \in \mathcal{V}_E$. For a given query q, we obtain the agent scores $s(q, a)$ and routing distribution $p_\theta(a \mid q, \mathcal{G})$ as defined in the previous subsection. To measure the contribution of each entity, we compute gradients of the routing objective with respect to $\mathbf{h}_i$. Specifically, we define the salience of entity v_i as

$$\alpha_i = \left\| \nabla_{\mathbf{h}_i} \mathcal{L}_{\mathrm{KL}}(q) \right\|_2 ,$$

which quantifies how strongly the training loss depends on v_i. Entities with higher α_i exert greater influence on agent routing and are therefore more informative. At inference, we evaluate α_i for all entities in $\mathcal{V}_E$ associated with q and filter out the entities that are not important enough, in our case the threshold τ is 0.01:

$$\mathcal{V}^* = \{_\theta(\{(v_i, \alpha_i) : v_i \in \mathcal{V}_E\}, \theta > \tau) .$$

We then construct the induced subgraph $\mathcal{G}^*$ over $\mathcal{V}^*$, preserving their original edges. The downstream reasoning at inference time will use $\mathcal{G}^*$ instead of $\mathcal{G}$ as the context graph, therefore $y_a(x, \mathcal{G}^*)$. This ensures that downstream reasoning attends to the most critical contextual evidence while filtering out redundant information. The final output will be the weighted vote result of the full agent set $\mathcal{A}$.

5 Experiments

5.1 Experiment Setup

For benchmark, we evaluate the label generation task on three datasets from the NGQA benchmark (Zhang et al., 2025b), namely *sparse*, *standard*, and *complex*, designed for nutrition question reasoning.

Type	Metric	Sparse	Standard	Complex
Node	Original	26.60	27.85	30.89
	Retrieval	7.87	9.71	13.67
	Drop (%)	70.41	65.13	55.75
Edge	Original	51.09	55.56	67.60
	Retrieval	10.78	19.95	30.24
	Drop (%)	78.90	64.09	55.27
SNR	Original	16.40	27.70	31.60
	Retrieval	50.83	70.44	70.80
	Raise (%)	209.94	154.30	124.05

Table 2: Graph size reduction before and after applying subgraph retrieval. Node and edge statistics are shown for Sparse, Standard, and Complex settings. Signals-to-Noise Ratio (SNR) indicates the proportion of useful nodes in the context graph.

We follow the evaluation settings and evaluation metrics of the NGQA benchmark. The detailed description of the benchmark and dataset, as well as training settings, are provided in Appendix B.

For baselines, we consider three categories: (1) the original baselines reported in NGQA, namely KAPING (Baek et al., 2023) and ToG (Sun et al., 2024); (2) the best-performing single-agent and multi-agent designs across several LLM backbones. Specifically, we include Raw (the basic LLM method), Chain-of-Thought (CoT) (Wei et al., 2022), Self-Consistency (SC) (Wang et al., 2023b), ReAct-Reflection (Yao et al., 2023; Shinn et al., 2023), Multi-Agent Debate (MAD) (Du et al., 2024), and Multi-Agent Summary. Our method also draws from these six agent designs as its candidate pool; (3) Strong ensembling method such as Majority Vote and the state-of-the-art LLM routing baselines, including LLM-Blender (Jiang et al., 2023a) and HybridLLM (Ding et al.,

Setting	Backbone	CoT	MAD	Raw	ReAct-Reflect	SC	Summary	Best Agent
Sparse	GPT-4o-mini	48.10±1.49	46.39±1.14	49.41±0.91	47.09±1.39	45.52±0.10	47.12±1.25	78.40
	Llama-3.2-3B	45.87±0.68	45.33±0.72	46.46±0.93	47.72±0.53	45.38±0.13	46.33±0.07	56.90
	Mistral-7B-Instruct	46.73±1.43	44.70±0.43	47.29±0.27	44.23±0.88	42.42±0.50	47.10±0.53	54.38
	Qwen2.5-7B-Instruct	43.03±0.14	40.42±0.33	42.86±0.21	42.52±0.03	42.29±0.03	42.20±0.32	81.70
Standard	GPT-4o-mini	52.25±3.79	58.54±6.41	52.11±4.47	53.02±2.71	52.35±4.61	54.67±3.85	51.09
	Llama-3.2-3B	64.02±0.78	70.31±0.16	64.78±1.65	71.66±1.18	66.30±1.02	72.65±1.31	74.09
	Mistral-7B-Instruct	68.32±0.83	79.40±0.76	66.97±0.21	73.95±1.22	67.85±0.73	76.98±0.78	79.71
	Qwen2.5-7B-Instruct	80.27±0.64	80.34±0.34	82.40±0.42	84.89±0.27	86.36±0.19	85.45±0.17	86.47
Complex	GPT-4o-mini	82.05±0.18	81.88±0.27	81.86±0.18	81.73±0.19	82.45±0.28	81.60±0.36	82.29
	Llama-3.2-3B	66.97±1.21	59.79±0.37	64.97±0.04	61.54±0.12	62.01±0.33	59.23±0.58	67.67
	Mistral-7B-Instruct	75.50±0.69	68.80±0.50	75.59±0.06	73.12±0.02	75.01±0.13	70.58±0.13	75.90
	Qwen2.5-7B-Instruct	74.37±13.48	69.95±9.80	74.37±12.72	72.30±11.20	71.13±10.68	71.59±11.71	82.15

Table 3: Performance of different agent designs across Sparse, Standard, and Complex settings on four LLM backbones. We report mean F1 with standard deviation. The last column shows the best-performing agent within each backbone.

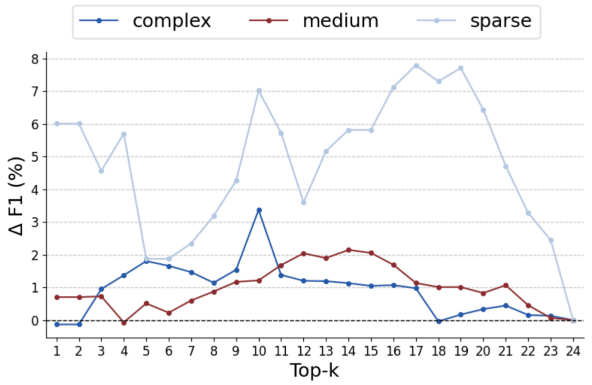

Figure 3: Percentage change (Δ) in F1 relative to k=24, used as the base (0%). Curves show how performance varies with the top k agent clipped across datasets.

2024). For experiments, we use four widely adopted LLM backbones of comparable scale but from different providers: Llama-3.2-3B-Instruct (MetaAI, 2024), Qwen2.5-7B-Instruct-Turbo (Alibaba, 2025), Mistral-7B-Instruct-v0.2 (MistralAI, 2024), and GPT-4o-mini (OpenAI, 2024). All baselines are evaluated under the same settings, and results are averaged over three consecutive runs to mitigate randomness. Additional implementation details of the baselines are provided in Appendix C.

5.2 Main Result

We present the main results of NG-Router against baseline methods in Table 1. Across all three benchmark settings, our approach consistently achieves superior performance, demonstrating the effectiveness of the proposed routing framework. Crucially, even without the integration of subgraph retrieval, NG-Router (w/o SR) already exhibits robust capabilities, outperforming other baselines in most metrics. As indicated by the underlined statistics, the routing mechanism alone secures the second-best performance in all settings, surpassing strong competitors like HybridLLM and LLM-Blender. Beyond the overall gains, several noteworthy insights emerge. First, the integration of subgraph retrieval yields substantial improvements. In contrast, traditional rule-based retrieval methods such as KAPING and ToG perform markedly worse. This highlights the advantage of our approach: by leveraging message-passing signals, the retrieval process can identify informative nodes even at longer distances, whereas heuristic methods tend to overemphasize immediate neighbors and thus overlook critical evidence. Second, the impact of subgraph retrieval varies across benchmarks. On the sparse setting, performance improves by more than 50%, whereas on the complex dataset the gains are closer to 10%. Table 2 sheds light on this phenomenon: the retrieval process prunes over 70% of edges and nodes, effectively filtering noise in the sparse setting where relevant information is scarce. By contrast, in the complex setting, where signal-to-noise (SNR) ratios (Table 9 in Appendix B) are inherently higher, the marginal benefit of pruning is smaller. We can observe that the retrieved graph almost doubled in SNR, which also indicates the raise in the performance. Finally, we also examine the best single-agent designs across different LLM backbones. Results reveal that agent effectiveness varies by task, underscoring the necessity of learning adaptive collaboration schemes rather than relying on fixed heuristics. By coordinating agents through a weighted-voting mechanism learned via routing, NG-Router consistently outperforms all individual agents, illustrating the benefit of principled collaboration in multi-agent reasoning.

5.3 Top K Pruning

During our experiment, we report the full routing results, however, we notice that pruning the long tail of low-quality or noisy agents is helpful to reduce variance and sharpens the aggregation of useful reasoning patterns. This suggests that agent routing benefits not only from diversity but also from judicious selection, where a smaller yet high-quality subset provides a better balance between complementarity and noise. As can be seen in Figure 3, full agent routing is usually not the optimal solution, whereas pruning the agent number to 10-15 would yield better overall performance.

5.4 Transfer Analysis

Beyond the label generation task, we further evaluate the transferability of our approach by applying the trained multi-label generation model to the two additional tasks defined in the NGQA benchmark: binary classification and natural text generation. As shown in Table 4, NG-Router achieves the strongest overall performance across both task types. For binary classification, our method yields improvements in both precision and F1 over all zero-shot baselines, demonstrating that the learned representations are sufficiently robust to support accurate decision-making when the task is simplified to binary outcomes. For natural text generation, NG-Router consistently surpasses competing methods on ROUGE, BLEU, and BERT metrics, indicating that the routing framework not only preserves semantic fidelity but also enhances fluency and informativeness in generated answers. These results confirm that the knowledge-graph–guided design of NG-Router provides a transferable advantage, enabling effective adaptation to multiple downstream tasks. Taken together, this analysis underscores the generalizability of our framework and its potential to serve as a versatile foundation for reasoning in diverse nutritional QA scenarios.

5.5 Hyperparameter Analysis

In this section, we study the effect of architectural hyper-parameters. Figure 4 presents the results of varying the number of layers and hidden dimensions. Across all benchmarks, we find that performance differences between configurations are moderate yet systematic. Increasing the number of layers does not consistently improve F1: while three layers slightly outperform two on the *Standard* dataset, deeper settings yield lower scores on *Sparse* and *Complex*, with larger variances observed for deeper models. This suggests diminishing returns and instability with depth, especially in multi-hop reasoning tasks. By contrast, enlarging the hidden dimension provides stable gains across benchmarks. Performance improves monotonically from 64 to 256 dimensions, with reduced variance at higher capacities, indicating that representational richness plays a more robust role than depth in this setting. Overall, these findings suggest that a shallow architecture with a larger hidden size strikes a better trade-off between accuracy and stability, highlighting the importance of capacity over depth in designing models for diverse QA benchmarks. Additionally, we report the original agent performance used to calculate the best agent scores, as can be seen in Table 3.

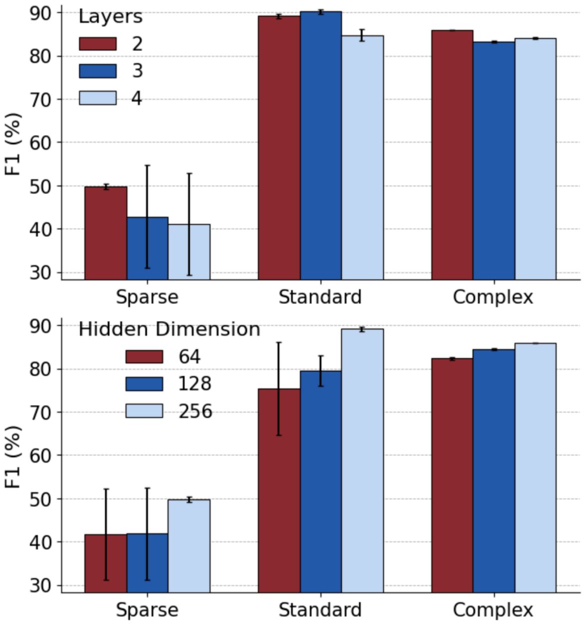

Figure 4: F1 performance on the three datasets, varying Layers (top) and Hidden Dimensions (bottom). Error bars denote standard deviations.

5.6 Efficiency Analysis

We further address the computational efficiency of NG-Router regarding agent invocation costs. During the training phase, it is necessary to query every candidate agent for each question. We argue that this is an unavoidable prerequisite for any supervised routing strategy, as the system must acquire ground-truth labels to effectively train the router. In the inference phase, however, our approach operates with zero redundancy. The framework strictly activates only the specific agent(s) selected by the router, completely avoiding the overhead of querying the full ensemble. Consequently, the system ensures optimal efficiency: the comprehensive cost during training is a mandatory investment for learning, while the testing phase remains lightweight and selective.

Method	**Binary Classification**				**Natural Text Generation**				
	Accuracy	Recall	Precision	F1	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERT
Majority Vote	53.38±0.00	71.00±0.00	68.27±0.00	69.67±0.00	60.32±0.10	60.84±0.66	60.32±0.10	41.73±1.85	86.29±0.02
Hybrid LLM	44.23±0.61	61.33±0.58	61.33±0.58	61.33±0.58	69.55±0.70	62.65±1.05	68.65±0.74	46.86±1.00	95.81±0.09
LLM Blender	50.38±0.00	67.00±0.00	67.00±0.00	67.00±0.00	68.17±0.51	63.26±0.64	67.32±0.46	49.85±0.77	95.74±0.04
NG-Router	55.06±2.10	71.00±1.73	71.00±1.73	71.00±1.73	70.88±0.17	65.51±0.37	69.73±0.06	51.55±0.25	96.14±0.00

Table 4: Comparison of **binary classification metrics** (Accuracy, Recall, Precision, F1) and **natural text generation metrics** (ROUGE, BLEU, BERT). All values are reported with mean ± standard deviation.

5.7 Broader Applicability

Although our experimental validation focuses on the domain of personalized nutrition, the design principles of NG-Router transcend this specific field. The core innovation lies in abstracting complex information into a structured knowledge graph, where type-sensitive message passing enables the system to capture latent dependencies between queries and agent capabilities. This logic is universally applicable to any multi-agent reasoning scenario—such as legal consultation, medical diagnosis, or financial auditing—where the key challenge is dynamically matching diverse user needs with specialized expert modules based on intricate contextual cues.

Furthermore, the proposed Gradient-based Subgraph Retrieval serves as a general-purpose mechanism for noise reduction, independent of the specific host architecture. In information-dense environments, critical evidence is often submerged by irrelevant data, leading to information overload. By leveraging gradient signals to identify nodes that actively contribute to the model's reasoning process, our retrieval method effectively filters out noise while preserving "preferred" high-value information. This ensures that the reasoning core remains focused on salient features, preventing vital signals from being drowned out by excessive context.

6 Conclusion

We proposed NG-ROUTER, a graph-supervised framework for adaptive multi-agent collaboration in nutritional QA, an important domain specific field. By extending context graphs with query and agent nodes, our framework learns task-specific routing weight distributions to search for optimal agent collaboration schemes, while gradient-guided subgraph retrieval prunes irrelevant entities to reduce contextual overload. Beyond its empirical gains, NG-ROUTER offers practitioners a principled way to harness the complementary strengths of heterogeneous agents without relying on costly trial-and-error or heuristic ensembling. This makes it a robust and generalizable tool for deploying reliable multi-agent systems in complex domain specific reasoning tasks.

Acknowledgements

The work of K. Shi, Z. Zhang, Z. Yuan, C. Zhang, and Y. Ye was partially supported by the NSF under grants IIS-2533550, IIS-2321504, IIS-2217239, CNS-2426514, and CMMI-2146076, Notre Dame Strategic Framework Research Grant (2025), and Notre Dame Poverty Research Package (2025). Any expressed opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect the views of the sponsors.

Limitations

This work is limited by its reliance on the NGQA benchmark, which is derived from U.S.-centric dietary surveys. While this scope may restrict direct generalizability to other populations, domains, or languages, NGQA remains the most comprehensive benchmark currently available for personalized nutrition reasoning. Our model is intentionally designed for this domain-specific setting, where carefully curated knowledge graphs and diverse dietary-health annotations provide a rigorous testbed. We view this as an essential first step, and leave the extension to broader public benchmarks and multilingual dietary datasets to future work.

Our evaluation also relies on automatic metrics such as accuracy, precision, and F1. These provide standardized comparisons with baselines and highlight reasoning quality, but they do not capture clinical or behavioral outcomes. Similarly, the design of the knowledge graph and the gradient-based subgraph retrieval involve thresholding choices that may introduce bias or limit stability across different datasets. We adopt these abstractions to ensure scalability and systematic experimentation, and envision future refinements that incorporate adaptive graph schemas, richer outcome measures, and broader evaluation settings.

References

Ashkan Afshin, Patrick J Sur, Kairsten A Fay, Leslie Cornaby, Giannina Ferrara, Jason S Salama, and Christopher J L Murray. 2019. Health effects of dietary risks in 195 countries, 1990–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*.

Qwen Alibaba. 2025. Qwen2.5-7b-instruct. `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *ACL*.

Felix Bölz, Diana Nurbakova, Sylvie Calabretto, Armin Gerl, Lionel Brunie, and Harald Kosch. 2023. Hummus: A linked, healthiness-aware, user-centered and argument-enabling recipe data set for recommendation. In *RecSys*.

Jon Nicolas Bondevik, Kwabena Ebo Bennin, Önder Babur, and Carsten Ersch. 2024. A systematic review on food recommender systems. *Expert Systems with Applications*.

Yuxuan Cao, Jiarong Xu, Carl Yang, Jiaan Wang, Yunchao Zhang, Chunping Wang, Lei Chen, and Yang Yang. 2023. When to pre-train graph neural networks? from data generation perspective! In *KDD*.

CDC. 2020. Adult obesity facts.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. Reconcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *ACL*.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. 2024b. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *NeurIPS*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. 2024c. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *ICLR*.

D. Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, et al. 2025. Best-route: Adaptive llm routing with test-time optimal compute. In *ICML*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybridllm: Cost-efficient and quality-aware query routing. In *ICLR*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multi-agent debate. In *ICML*.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2023. Talk like a graph: Encoding graphs for large language models. *arXiv*.

Yihan Feng, Tianyu Zhao, Haotian Liu, Diyi Yang, and Tuo Zhao. 2025. Graphrouter: Learning graph-based routing for large language model selection. In *ICLR*.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. *arXiv*.

Mouzhi Ge, Francesco Ricci, and David Massimo. 2015. Health-aware food recommender system. In *RecSys*.

Tiezheng Guo, Qingwen Yang, Chen Wang, Yanyi Liu, Pan Li, Jiawei Tang, Dapeng Li, and Yingyou Wen. 2024. Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. *Complex & Intelligent Systems*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NeurIPS*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *ICLR*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023a. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *ACL*.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Structgpt: A general framework for large language model to reason over structured data. In *EMNLP*.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2024. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. *arXiv*.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, et al. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *ACL*.

Mingxuan Ju, Shifu Hou, Yujie Fan, Jianan Zhao, Yanfang Ye, and Liang Zhao. 2022a. Adaptive kernel graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7051–7058.

Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022b. Grape: Knowledge graph enhanced passage reader for open-domain question answering. *arXiv preprint arXiv:2210.02933*.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. In *EMNLP*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeuralIPS*.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024. Can we soft prompt llms for graph learning tasks? In *WWW*, pages 481–484.

Tianyi Ma, Yiyue Qian, Zheyuan Zhang, Zehong Wang, Xiaoye Qian, Feifan Bai, Yifan Ding, Xuwei Luo, Shinan Zhang, Keerthiram Murugesan, et al. 2025. Autodata: A multi-agent system for open web data collection. *arXiv preprint arXiv:2505.15859*.

MetaAI. 2024. Llama-3.2-3b-instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.

Weiqing Min, Chunlin Liu, Leyi Xu, and Shuqiang Jiang. 2022. Applications of knowledge graphs for food science and industry. *Patterns*.

MistralAI. 2024. Mistral-7b-instruct-v0.2. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2.

Bo Ni, Zheyuan Liu, Leyao Wang, Yongjia Lei, Yuying Zhao, Xueqi Cheng, Qingkai Zeng, Luna Dong, Yinglong Xia, Krishnaram Kenthapadi, et al. 2025. Towards trustworthy retrieval augmented generation for large language models: A survey. *arXiv*.

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 with preference data. In *ICLR*.

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

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv*.

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. 2025. Scaling large language model-based multi-agent collaboration. In *ICLR*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*.

Sola S Shirai, Oshani Seneviratne, Minor E Gordon, Ching-Hua Chen, and Deborah L McGuinness. 2021. Identifying ingredient substitutions using a knowledge graph of food. *Frontiers in Artificial Intelligence*.

D. Stripelis et al. 2024. A multi-model router for efficient llm inference. In *EMNLP*.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *EMNLP-IJCNLP*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *ICLR*.

Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. 2023. Grapeqa: Graph augmentation and pruning to enhance question-answering. In *WWW*.

Y Tian, C Zhang, Z Guo, C Huang, R Metoyer, and N Chawla. 2022a. Reciperec: A heterogeneous graph learning model for recipe recommendation. In *IJCAI*.

Yijun Tian, Chuxu Zhang, Zhichun Guo, Yihong Ma, Ronald Metoyer, and Nitesh V Chawla. 2022b. Recipe2vec: Multi-modal recipe representation learning with graph neural networks. *arXiv*.

Yijun Tian, Chuxu Zhang, Ronald Metoyer, and Nitesh V Chawla. 2021. Recipe representation learning with networks. In *CIKM*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024a. Can language models solve graph problems in natural language? *NeuralIPS*.

Junlin Wang, WANG Jue, Ben Athiwaratkun, Ce Zhang, and James Zou. 2025a. Mixture-of-agents enhances large language model capabilities. In *ICLR*.

Wenjie Wang, Ling-Yu Duan, Hao Jiang, Peiguang Jing, Xuemeng Song, and Liqiang Nie. 2021. Market2dish: health-aware food recommendation. *TOMM*.

X. Wang, Y. Fu, Y. Zhang, W. Cheng, et al. 2025b. Mixllm: Dynamic routing in mixed large language models. In *NAACL*.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023a. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. 2025c. Can llms convert graphs to text-attributed graphs? In *NAACL*.

Zehong Wang, Zheyuan Zhang, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. 2024b. Gft: Graph foundation model with transferable tree vocabulary. *NeruIPS*.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. 2025d. Learning cross-task generalities across graphs via task-trees. *ICML*.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. 2025e. Neural graph pattern machine. *ICML*.

Zehong Wang, Zheyuan Zhang, Chuxu Zhang, and Yanfang Ye. 2024c. Subgraph pooling: tackling negative transfer on graphs. In *IJCAI*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeuralIPS*.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv*.

WHO. 2021. Healthy diet.

WHO. 2023. Obesity info page of the world health organization.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *NAACL*.

Yanfang Ye, Zheyuan Zhang, Tianyi Ma, Zehong Wang, Yiyang Li, Shifu Hou, Weixiang Sun, Kaiwen Shi, Yijun Ma, Wei Song, et al. 2025. Llms4all: A review of large language models across academic disciplines. *arXiv preprint arXiv:2509.19580*.

Wenbin Yue, Zidong Wang, Jieyu Zhang, and Xiaohui Liu. 2021. An overview of recommendation techniques and their applications in healthcare. *IEEE/CAA Journal of Automatica Sinica*.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *ACL*.

Lingzi Zhang, Yinan Zhang, Xin Zhou, and Zhiqi Shen. 2024a. Greenrec: A large-scale dataset for green food recommendation. In *WWW*.

Zheyuan Zhang, Lin Ge, Hongjiang Li, Weicheng Zhu, Chuxu Zhang, and Yanfang Ye. 2025a. Mapro: Recasting multi-agent prompt optimization as maximum a posteriori inference. *arXiv*.

Zheyuan Zhang, Yiyang Li, Nhi Ha Lan Le, Zehong Wang, Tianyi Ma, Vincent Galassi, Keerthiram Murugesan, Nuno Moniz, Werner Geyer, Nitesh V Chawla, et al. 2025b. Ngqa: A nutritional graph question answering benchmark for personalized health-aware nutritional reasoning. *ACL*.

Zheyuan Zhang, Kaiwen Shi, Zhengqing Yuan, Zehong Wang, Tianyi Ma, Keerthiram Murugesan, Vincent Galassi, Chuxu Zhang, and Yanfang Ye. 2025c. Agentrouter: A knowledge-graph-guided llm router for collaborative multi-agent question answering. *arXiv*.

Zheyuan Zhang, Zehong Wang, Shifu Hou, Evan Hall, Landon Bachman, Jasmine White, Vincent Galassi, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. 2024b. Diet-odin: A novel framework for opioid misuse detection with interpretable dietary patterns. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6312–6323.

Zheyuan Zhang, Zehong Wang, Tianyi Ma, Varun Sameer Taneja, Sofia Nelson, Nhi Ha Lan Le, Keerthiram Murugesan, Mingxuan Ju, Nitesh V Chawla, Chuxu Zhang, et al. 2024c. Mopi-hfrs: A multi-objective personalized health-aware food recommendation system with llm-enhanced interpretation. *arXiv*.

Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. 2023. Self-supervised graph structure refinement for graph neural networks. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 159–167.

Jianan Zhao, Qianlong Wen, Shiyu Sun, Yanfang Ye, and Chuxu Zhang. 2021. Multi-view self-supervised heterogeneous graph embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 319–334. Springer.

Li Zhong, Zilong Wang, and Jingbo Shang. 2024. Debug like a human: A large language model debugger via verifying runtime execution step by step. In *Findings of ACL*.

A Additional Related Works

A.1 Knowledge Graph Question Answering

Research on Knowledge Graph Question Answering (KGQA) has progressed from classic semantic parsing and retrieval paradigms to increasingly model-driven solutions (Ju et al., 2022b). Early systems translated natural-language questions into executable logical forms (e.g., SPARQL) over a knowledge graph (Sun et al., 2019; Zhang et al., 2022), often pairing pre-trained encoders such as BERT with graph-aware architectures (GNNs/LSTMs) to locate entities, relations, and supporting subgraphs (Yasunaga et al., 2021; Taunk et al., 2023). More recent approaches incorporate large language models (LLMs) to improve both access and reasoning: some convert questions into structured queries like SQL/SPARQL to sharpen retrieval (Jiang et al., 2023b; Wang et al., 2023a), while others emphasize multi-hop inference over retrieved triples or subgraphs to handle compositional reasoning (Kim et al., 2023; Gao et al., 2024). Despite these advances, widely used benchmarks remain largely general-purpose and do not fully capture domain-specific demands—e.g., the nuanced constraints present in nutritional-health reasoning scenarios.

A.2 Graph-Retrieval Augmented Generation

Graph-Retrieval Augmented Generation (Graph-RAG) generalizes the RAG paradigm (Lewis et al., 2020; Ni et al., 2025) by retrieving *structured* evidence rather than only unstructured text. Instead of passages alone, Graph-RAG surfaces graph fragments (triples/subgraphs) and uses graph encoders to condition generation, thereby improving precision and reducing redundancy (Guo et al., 2024; Wen et al., 2023; Lazaridou et al., 2022; Liu et al., 2024). Current evaluations predominantly probe elementary graph reasoning skills—such as path finding, degree/counting, or edge existence (Fatemi et al., 2023; Wang et al., 2024a, 2025c). While informative for fundamentals, these settings underrepresent domain-specific requirements. He et al. introduce more advanced graph-understanding benchmarks in general contexts (He et al., 2024). Building on Graph-RAG principles, many applications nowadays thrive and shed lights on new research paths (Zhang et al., 2024b).

A.3 Graph Neural Networks.

Graph Neural Networks (GNNs) are designed for relational data and have delivered strong results across social, recommendation, biological, and molecular applications by exploiting graph inductive biases (Kipf and Welling, 2016; Veličković et al., 2017; Hamilton et al., 2017; Zhao et al., 2021; Ju et al., 2022a; Zhao et al., 2023). Their ability to share parameters across varying graph sizes/topologies supports deployment in dynamic, real-world settings. A growing body of work investigates transfer and pretraining for cross-task/domain generalization—mirroring trends in language and vision—via subgraph pooling, pretraining schemes, and task-agnostic embeddings (Wang et al., 2024c; Cao et al., 2023). Looking forward, the community is moving toward *graph foundation models*, i.e., large-scale pretrained GNN backbones intended to capture broadly reusable structural/semantic patterns (Wang et al., 2024b, 2025e,d). Despite progress, open challenges persist, including oversmoothing, expressive-power limits, and scalability, motivating research on more adaptive architectures and training recipes.

Nutrients	**Low Threshold**	**High Threshold**	**NRV**
Calories (kcal)	40	225	2000
Carbohydrates (g)	55	75	-
Protein (g)	10	15	50
Saturated Fat (g)	1.5	5	20
Cholesterol (mg)	20	40	300
Sugar (g)	5	22.5	-
Dietary Fiber (g)	3	6	-
Sodium (mg)	120	200	2000
Potassium (mg)	0	525	3500
Phosphorus (mg)	0	105	700
Iron (mg)	0	3.3	22
Calcium (mg)	0	150	1000
Folic Acid (μg)	0	60	400
Vitamin C (mg)	0	15	100
Vitamin D (μg)	0	2.25	15
Vitamin B12 (μg)	0	0.36	2.4

Table 5: Nutrient Reference Values (NRV) and thresholds (per 100g of food) used based on the nutritional standards.

Health Indicator	**High Threshold**	**Low Threshold**
BMI	30	18.5
Waist Circumference (cm)	102 (88)	-
Blood Pressure (mmHg)	140	90
Osteoporosis	-	-
Blood Urea Nitrogen (mmol/L)	7.1	-
Low-Density Lipoprotein (mmol/L)	3.3	-
Red Blood Cell (million cells/uL)	-	4
Glucose (mmol/L)	7	-
Glycohemoglobin (%)	6.5	-
Hemoglobin (g/dL)	-	13.2 (11.6)

Table 6: Health Indicators with Corresponding High and Low Thresholds. Parentheses indicate sex-specific: male (female) thresholds where applicable.

Health Indicator	Associated Tags
Obesity	Low Calorie
Opioid Misuse	High Protein; Low Sugar; Low Sodium
Hypertension	Low Sodium
Diabetes	Low Sugar; Low Carb
Weight Loss/Low Calorie Diet	Low Calorie
Low Fat/Low Cholesterol Diet	Low Cholesterol; Low Saturated Fat
Low Salt/Low Sodium Diet	Low Sodium
Sugar-Free/Low Sugar Diet	Low Sugar
Diabetic Diet	Low Sugar; Low Carb
Weight Gain/Muscle Building Diet	High Calorie; High Protein
Low Carbohydrate Diet	Low Carb
High Protein Diet	High Protein
Renal/Kidney Diet	Low Protein

Table 7: Health Indicators and Their Associated Nutritional Tags. Each indicator is linked to relevant tags reflecting dietary requirements.

B NGQA Benchmark Details

B.1 Benchmark Description

We build upon the NGQA benchmark (Zhang et al., 2025b), which integrates multiple U.S. national health and nutrition resources. The benchmark primarily relies on the National Health and Nutrition Examination Survey (NHANES), a biannual survey conducted by the CDC that combines demographic, dietary, examination, laboratory, and questionnaire data. Dietary intake is captured through the What We Eat in America (WWEIA) program, which collects 24-hour dietary recalls and links them to nutrient information in the USDA's Food and Nutrient Database for Dietary Studies (FNDDS). Together, these resources provide comprehensive coverage of health indicators, food consumption, and nutrient composition. To represent user behaviors, the benchmark includes dietary habit features derived from NHANES (e.g., awareness of healthy eating, consumption of processed or frozen foods). Through manual curation, they constructed 54 distinct dietary habit tags, which serve as additional nodes in the graph.

Nutrient annotations were derived from international dietary standards (WHO, FSA, EU, CAC), covering 16 macro- and micronutrients such as calories, protein, sugar, sodium, and iron. These were mapped to user health profiles through threshold-based rules with mapping rules shown in Table 5, 6 and 7. The benchmark emphasizes four prevalent health conditions with clear dietary relevance: obesity, hypertension, diabetes, and opioid misuse. Standard clinical definitions were applied to the first three, while opioid misuse was defined using medical criteria based on illicit use or long-term prescription opioid records.

Question Level	# Records	Avg. # Nodes	Avg. # Edges
Sparse	8,490	25.8	24.9
Standard	3,622	28.2	29.0
Complex	1,690	30.9	34.0

Table 8: Statistics of the Benchmark by Question Level.

B.2 Datasets

Building on the dataset foundation, the NGQA benchmark organizes its tasks into three different datasets, each reflecting different levels of information availability and reasoning complexity.

Sparse Dataset. These address scenarios with minimal information. Each food is linked to only one nutrition tag associated with a single user health condition. While this setting mirrors real-world cases where labels are scarce or incomplete, it presents substantial challenges: the one-to-one linkage amplifies the difficulty of subgraph retrieval and makes models especially susceptible to interference from irrelevant nodes.

Standard Dataset. These represent the balanced and idealized setting of the benchmark. Foods are connected to multiple nutrition tags that either match or contradict several user health conditions. Such cases capture clear-cut relationships between dietary choices and health outcomes, providing a controlled environment for evaluating model performance. Standard questions therefore serve as the reference baseline for structured reasoning tasks.

Complex Dataset. These replicate the intricacies of real-life nutritional decision-making. Foods may simultaneously contain tags that both align with and conflict with user health conditions. For example, a food might be low in sodium (beneficial for hypertension) but high in sugar (problematic for diabetes). Models must navigate these conflicting signals, prioritize health needs, and perform trade-off reasoning, making this category the most realistic and challenging.

A statistical breakdown of these three categories is provided in Table 8. To further quantify informativeness, the benchmark also introduces a Signal-to-Noise Ratio (SNR), defined as the ratio of relevant nodes or tags (*signal*) to the total number of nodes or tags in the graph (*noise*). As shown in Table 9, sparse questions have the lowest SNR, reflecting their limited resources, while complex questions achieve the highest SNR, highlighting the richer contextual information needed for better reasoning.

Question Level	Avg. Node SNR	Avg. Tag SNR
Sparse	16.4	19.3
Standard	24.7	49.4
Complex	31.6	76.3

Table 9: Signal-to-Noise Ratio (SNR) by Question Level.

C Baseline and Implementation Details

To cover the range of agentic strategies explored in contemporary LLM research, the benchmark incorporates six representative designs. *Raw* serves as the direct prompting baseline, reflecting the unaugmented capacity of the backbone model. *Chain-of-Thought (CoT)* (Wei et al., 2022) encourages models to generate intermediate reasoning steps, thereby improving performance on multi-step tasks. *Self-Consistency (SC)* (Wang et al., 2023b) extends CoT by sampling multiple reasoning paths and aggregating the most consistent outcome, reducing sensitivity to individual trajectories. *ReAct-Reflection* (Yao et al., 2023; Shinn et al., 2023) integrates reasoning with external actions and iterative self-correction, yielding more grounded and robust responses. *Multi-Agent Debate (MAD)* (Du et al., 2024) facilitates adversarial discussion among agents, enabling consensus through structured deliberation. Finally, *Multi-Agent Summary* aggregates partial reasoning produced by diverse agents into a distilled final answer. Together, these designs span from single-agent prompting to multi-agent collaboration, underscoring the need for principled routing across heterogeneous strategies (Zhang et al., 2025a).

Beyond heuristics, we benchmark against three state-of-the-art LLM routing frameworks. **LLM-Blender** (Jiang et al., 2023a) leverages an LLM-as-a-judge paradigm: candidate outputs from multiple agents are presented to a meta-LLM, which adjudicates and selects the final answer. This approach highlights the potential of reflective meta-reasoning but incurs high cost and latency due to repeated LLM calls. **HybridLLM** (Ding et al., 2024) adopts a hybrid strategy that combines lightweight scoring heuristics with selective meta-LLM adjudication, aiming to balance efficiency and effectiveness. We also select two baselines reported in NGQA paper, which can be seen as classical graph retrieval augmented methods. **KAPING** (Baek et al., 2023) answers questions by constructing a subgraph from the entities mentioned in the query and their neighbors, which is then linearized into triples and fed into the LLM. While the original implementation applies top-k filtering, this step was omitted in our benchmark as it is not applicable when only user and food entities are present. **ToG** (Sun et al., 2024) iteratively explores and prunes reasoning paths on a knowledge graph to identify candidate answers. Since the released code is tailored to Wikidata/Freebase, we reimplemented ToG for our benchmark and introduced two adjustments: increasing the pruning width to five paths and delaying pruning until the second iteration. These modifications ensure sufficient coverage and better alignment with the benchmark's complexity.

Beside the implemented baselines, it is worth noting our work is greatly inspired by the broader idea of graph-based reasoning for LLM orchestration in GraphRouter (Feng et al., 2025). However, our work departs fundamentally from GraphRouter in many ways, of which the most important one is that GraphRouter encodes a query and its context as a single node linked to candidate LLMs, which doesn't fit into our problem settings.

D Case Studies

We provide two case studies to demonstrate the effectiveness of our model in routing questions to the most appropriate agents (Figure 5). In the *Borscht* example, the gold answer is *low_sugar, low_sodium*. The agent probability distribution shows that `Qwen2.5-7B-Instruct::raw` and `Qwen2.5-7B-Instruct::cot` achieve probabilities above 0.08 and both predict the correct labels. Other agents, including those from different backbones such as `Mistral-7B-Instruct`, also provide correct or near-correct answers, which demonstrates that our router not only identifies the most relevant agents but also assigns higher probabilities to those generating correct predictions. This reduces the chance of routing errors and improves overall reliability in multi-agent collaboration.

We illustrate several examples of applying our subgraph retrieval approach to different datasets in Figure 6. To make the reasoning process easier to understand, we first convert the original graph into plain text. The graph for the *Bruschetta* example shows that the food is associated with various ingredients (garlic, olive oil, tomatoes, basil, salt, etc.), belongs to categories like "vegetable sandwiches/burgers," and is linked to multiple nutrition tags such as *low_carb*, *low_sugar*, and *high_sodium*. It also contains user-related information such as health conditions (*hypertension*) and lifestyle habits (e.g., "drinks alcohol less than average," "takes more supplements"). While this graph contains 25 nodes and 30 edges, many of these nodes are not directly useful for answering the question about nutrient tags. Most of the user habits and ingredient details serve as noise that could mislead the reasoning process. Our scoring method solves this problem by assigning importance scores to all nodes. Irrelevant information typically receives scores below 0.01, while truly relevant entities such as *Bruschetta*, *hypertension*, and *high_sodium* achieve much higher scores. By filtering the graph to include only nodes with scores above 0.01, we obtain a clearer and more focused representation, reducing noise and helping the agent concentrate on the key information needed to answer the question.

E Prompt Designs

To demonstrate the exact instructions used in our system, we present the full set of prompts that guided the different agent roles. Figures 7 provide a complete overview. These prompts are not intended as a novel design contribution, but rather as transparent documentation of the configurations employed in our experiments.

Agent Routing Case Study

Example 1
Question: Based on the nutrients the food provides and the user needs, please answer what nutrient tags are used to determine whether the food "Broccoli cheese soup, prepared with milk, home recipe, canned, or ready-to-serve" is healthy or unhealthy for the user?
Gold Answer: low_protein

Agent probs
`GPT-4o-mini::AGENT::raw` → *low_protein* (0.052638724)
`GPT-4o-mini::AGENT::cot` → *high_sodium, low_protein* (0.050060976)
`GPT-4o-mini::AGENT::react_reflect`→ *high_sodium, low_protein* (0.049231380)
`Mistral-7B-Instruct::AGENT::react_reflect` → *high_sodium, low_protein* (0.044603113)
`...`
`Mistral-7B-Instruct::AGENT::mad` → *low_carb, low_sugar, high_sodium* (0.042540662)
`Llama-3.2-3B::AGENT::cot` → *low_saturated_fat, low_cholesterol, low_carb* (0.039508183)
`Qwen2.5-7B-Instruct::AGENT::cot` → *low_carb, low_sugar, high_sodium* (0.032161828)
`Qwen2.5-7B-Instruct::AGENT::raw` → *low_carb, low_sugar, high_sodium* (0.031613167)

Example 2
Question: Based on the nutrients the food provides and the user needs, please answer what nutrient tags are used to determine whether the food "Borscht" is healthy or unhealthy for the user?
Gold Answer: low_sugar, low_sodium

Agent probs
`Qwen2.5-7B-Instruct::AGENT::raw` → *low_sugar, low_sodium* (0.095538996)
`Qwen2.5-7B-Instruct::AGENT::cot` → *low_sugar, low_sodium* (0.086759798)
`Qwen2.5-7B-Instruct::AGENT::summary` → *low_sugar, low_sodium* (0.061009243)
`Mistral-7B-Instruct::AGENT::mad` → *low_sugar, low_sodium* (0.051758543)
`...`
`Llama-3.2-3B::AGENT::cot` → *low_sugar, low_sodium* (0.020628655)
`GPT-4o-mini::AGENT::mad` → *low_carb, low_sugar, low_calorie, low_protein, low_cholesterol, low_saturated_fat, low_sodium* (0.019520836)
`GPT-4o-mini::AGENT::sc` → *low_carb, low_sugar, low_calorie, low_protein, low_cholesterol, low_saturated_fat, low_sodium* (0.018208018)
`GPT-4o-mini::AGENT::cot` → *low_carb, low_sugar, low_calorie, low_protein, low_cholesterol, low_saturated_fat, low_sodium* (0.011033830)

Figure 5: Per-question agent routing cases. Each case shows the question, gold answer, and agents' probability with their generated answers.

Subgraph Retrieval Case Study

Case 1 (Standard)
Graph Context:
’Biscayne codfish, Puerto Rican style’, ’belongs to’, ’low_carb’], [’Biscayne codfish, Puerto Rican style’, ’belongs to’, ’low_sugar’], [’Biscayne codfish, Puerto Rican style’, ’belongs to’, ’high_sodium’], [’user’, ’has’, ’diabetes’], [’diabetes’, ’match’, ’low_sugar’], [’obesity’, ’need’, ’low_calorie’], . . .
Node Quantity: 31
edge quantity: 31
Question: Based on the nutrients the food provides and the user needs, please answer what nutrient tags are used to determine whether the food "Biscayne codfish, Puerto Rican style" is healthy or unhealthy for the user?
Gold Answer: low_carb, low_sugar
Entity Importance
user (0.412504), low_carb (0.210431), Biscayne codfish, Puerto Rican style (0.176805), low_sugar (0.142012), diabetes (0.025764), high_sodium (0.002139), low_cholesterol (0.002011), . . .

Case 2 (Complex)
Graph Context:
[’Matzo ball soup’, ’belongs to’, ’low_carb’], [’Matzo ball soup’, ’belongs to’, ’low_sugar’], [’Matzo ball soup’, ’belongs to’, ’high_sodium’], [’user’, ’has’, ’diabetes’], [’diabetes’, ’match’, ’low_sugar’], [’High protein diet’, ’contradict’, ’low_protein’], . . .
Node Quantity: 28
edge quantity: 30
Question: Based on the nutrients the food provides and the user needs, please answer what nutrient tags are used to determine whether the food "Matzo ball soup" is healthy or unhealthy for the user?
Gold Answer: low_carb, low_sugar, low_protein
Entity Importance
Matzo ball soup (0.397344), user (0.168014), low_protein (0.104383), low_sugar (0.102992), low_carb (0.099755), diabetes (0.016299), Low carbohydrate diet (0.013385), High protein diet (0.012389), Adds little to no salt at table (0.005626), Eats lots of fish (0.005585), . . .

Case 3 (Sparse)
Graph Context:
[’Bruschetta’, ’belongs to’, ’high_sodium’], [’Bruschetta’, ’has’, ’Salt, table, iodized’], [’Bruschetta’, ’has’, ’Olive oil’], [’user’, ’has’, ’hypertension’], [’hypertension’, ’contradict’, ’high_sodium’], . . .
Node Quantity: 25
edge quantity: 30

Question: Based on the nutrients the food provides and the user needs, please answer what nutrient tags are used to determine whether the food "Bruschetta" is healthy or unhealthy for the user?
Gold Answer: high_sodium

Entity Importance
user (0.283775), Bruschetta (0.283627), hypertension (0.188398), high_sodium (0.187762), Drinks Alcohol less than average (0.003046), Takes more supplements (0.003046), . . .

Figure 6: Per-question cases for subgraph retrieval. Each case shows the graph context, the question, the gold answer, and entities with importance score > 0.01; for each listed entity we also include the next two entities to highlight the score cliff (remaining tail elided with ". . . ").

QA/Reasoning Prompt Suite

raw:
Given a question, a news context, and retrieved documents, answer the question directly.
Compress your answer into the SHORTEST exact entity only.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

cot (chain-of-thought):
You are a multi-hop reasoning expert and QA agent.
Given a question and the context, reason step-by-step before answering.
Compress your answer into the SHORTEST exact entity only.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

sc (self-consistency):
You are a self-consistency agent. Independently sample multiple plausible entity selections for the given question and context,
then internally perform majority voting to decide the final set.
Generate diverse candidate sets internally, then pick the majority-agreed entities.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

mad (multi-agent debate):
You simulate three roles: *debate_debater_a*, *debate_debater_b*, and *debate_judge*.
- Debater A proposes the most plausible answer using only the provided context, supported by 1–3 short quotes.
- Debater B stress-tests A's claim: if weak or incomplete, correct it or propose a better alternative.
- Debate Judge decides the best final answer using only the given context (noun, number, or yes/no).

Finally, Debate Judge condenses the result into the shortest exact entity only.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

react_reflect:
You simulate two roles: *react* and *reflect*.
- React is a multi-hop reasoning expert that chains facts into a reasoning plan and derives a brief final answer.
- Reflect evaluates React's answer. If incorrect or incomplete, provide revision suggestions; otherwise, confirm correctness.

Reflect then condenses the result into the shortest exact entity only.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

summary:
You simulate three roles: *think_a*, *think_b*, and *summarize*.
- Think_a and Think_b independently reason step-by-step and produce candidate answers.
- Summarize compares their outputs: if they agree, return the shared answer; if not, select the best one with reasoning.

Finally, Summarize condenses the result into the shortest exact entity only.
The final output must be one JSON object: `"answer": "<short factual answer>"`.

Figure 7: Prompt suite for six major agent roles used in NGRouter. Each prompt defines a distinct reasoning behavior that collectively improves multi-agent QA performance.