Hint-SQL：基于自动线索生成的 Text-to-SQL 提示方法

谭钊[1)] 刘喜平[1)] 舒晴[2)] 万齐智[1)] 刘德喜[1)] 万常选[1)] 廖国琼[3),4)]

[1)](江西财经大学计算机与人工智能学院 南昌 330032)

[2)](江西财经大学信息管理与数学学院 南昌 330032)

[3)](江西财经大学虚拟现实(VR)现代产业学院 南昌 330032)

[4)](江西旅游商贸职业学院, 南昌 330100)

摘 要 Text-to-SQL 旨在将自然语言问题翻译为可被数据库系统执行的 SQL 语句，从而为数据查询提供便利。随着大语言模型（LLMs）技术的发展，基于 LLMs 的 Text-to-SQL 提示方法成为该领域的主流解决方案。近年来，研究者在 LLMs 的提示词中加入线索（Hint）来传递具体的 Text-to-SQL 建议，以引导 LLMs 生成 SQL。然而，现有线索多由研究者根据 Text-to-SQL 任务的特点人为撰写，其内容过于宽泛，难以根据具体的任务需求做出调整，无法适配所有 Text-to-SQL 任务。本文提出基于自动线索生成的 Text-to-SQL 提示方法 Hint-SQL，它能够根据当前 Text-to-SQL 任务自动地生成合适的语义、操作和结构线索，从而引导 LLMs 生成语义一致、结构正确的 SQL。为了生成任务定制化线索，我们构建了线索生成智能体（HAgent）。HAgent 基于两阶段微调框架，由开源 LLMs 微调而来，该框架自动合成微调所需数据，无需人工标注，为监督微调和偏好学习优化提供支持。Hint-SQL 既可以单独使用，也可以用来增强现有方法。大规模实验结果显示，Hint-SQL 独立使用时可以媲美主流方法，也可以显著增强现有方法性能，在 BIRD 数据集上，Hint-SQL 将当前最好方法的准确率提升到了 71.58%，提升幅度达到 4.37%。本研究揭示了线索在 Text-to-SQL 任务中的重要作用，为 Text-to-SQL 的后续研究提供了参考。

关键词 自然语言处理；Text-to-SQL；大语言模型；提示工程；线索

中图法分类号 TP18

收稿日期：2025-04-11；在线发布日期：2025-10-23. 本课题得到国家自然科学基金面上项目（62272205、62272206）、地区科学基金项目（62462034、62562033），江西省自然科学基金重点项目（20232ACB202008）、面上项目（20242BAB25119），江西省研究生创新专项资金项目（YC2023-B185）资助。**谭钊**，博士研究生，计算机学会（CCF）会员，主要研究领域为机器学习、自然语言处理。**刘喜平（通信作者）**，博士，教授，计算机学会（CCF）高级会员，主要研究领域为机器学习、自然语言处理。**舒晴**，博士研究生，计算机学会（CCF）会员，主要研究领域为机器学习、自然语言处理。**万齐智**，博士，讲师，计算机学会（CCF）会员，主要研究领域为自然语言处理、信息抽取、深度学习。**刘德喜**，博士，教授，计算机学会（CCF）高级会员，主要研究领域为社会媒体处理、信息检索、自然语言处理。**万常选**，博士，教授，计算机学会（CCF）高级会员，主要研究领域为情感分析、数据挖掘。**廖国琼**，博士，教授，计算机学会（CCF）高级会员，主要研究领域为数据库、区块链技术。

Hint-SQL: Text-to-SQL Prompting with Automatically-Generated Hints

TAN Zhao[1)] LIU Xi-Ping[1)] SHU Qing[2)] WAN Qi-Zhi[1)] LIU De-Xi[1)]
WAN Chang-Xuan[1)] LIAO Guo-Qiong[3),4)]

[1)](School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics, Nanchang 330032)

[2)](School of Information Management and Mathematics, Jiangxi University of Finance and Economics, Nanchang 330032)

[3)](School of Virtual Reality Modern Industry, Jiangxi University of Finance and Economics, Nanchang 330032, China)

[4)](Jiangxi Tourism and Commerce Vocational College, Nanchang 330100, China)

Abstract Text-to-SQL translation, the process of converting natural language questions into executable SQL statements, is pivotal for democratizing data access. It significantly lowers the technical barrier for data retrieval and analysis, empowering non-expert users across diverse domains like business intelligence and data science. With the advent of Large Language Models (LLMs), prompting-based methods have become the predominant approach in this field. A common Text-to-SQL prompt typically includes the database schema, user question, task instructions, and few-shot demonstrations. Recently, researchers have begun incorporating "hints" into these prompts to provide specific guidance, aiming to make the SQL generation process more robust and accurate. However, a significant limitation of existing approaches is their reliance on manually crafted, static hints. These pre-defined hints, often created by summarizing common error patterns, tend to be overly general. They lack the flexibility to adapt to the unique requirements of each specific query, rendering them sub-optimal or even misleading for many Text-to-SQL tasks. Moreover, the coverage of such manual hints is inherently limited, as they are designed for specific, anticipated cases and cannot adapt to the full spectrum of unforeseen task requirements. To address these deficiencies, this paper proposes Hint-SQL, a novel Text-to-SQL prompting methodology centered on automatically-generated, task-specific hints. Hint-SQL introduces a comprehensive hint structure designed to guide the LLM through the critical stages of query formulation: semantic understanding, operational planning, and structural mapping. This is achieved through a progressive, multi-faceted guidance system. First, Semantic Hints disambiguate the user's question by resolving vague expressions and mapping them to precise database schema elements. Building on this, Operational Hints outline a high-level, step-by-step plan for the query logic. Finally, Structural Hints complete the guidance by mapping these operational steps to concrete SQL keywords and syntax. This integrated approach ensures the final SQL is both semantically consistent and structurally correct. To realize this dynamic hint generation, we developed the Hint-generation Agent (HAgent), a specialized agent built by fine-tuning open-source LLMs. We propose a novel two-stage fine-tuning framework to train HAgent, which uniquely requires no manual data annotation. The first stage, Supervised Fine-Tuning (SFT), endows the agent with the fundamental ability to generate hints by learning the mapping between database schemas, questions, and corresponding hints from synthetically generated data. The second stage, Preference Learning Optimization, further refines the agent's capabilities. This stage addresses the challenge of capturing fine-grained details, where minor inaccuracies can lead to incorrect SQL. By training the model to distinguish between positive (correct) and negative (incorrect) hint examples, this stage significantly enhances the precision and reliability of the generated hints. Extensive experiments on five benchmark datasets and ten different LLMs validate our approach. When used as a standalone method, Hint-SQL achieves performance comparable to state-of-the-art techniques. More importantly, when integrated with existing methods, it acts as a powerful enhancer. On the challenging BIRD dataset, Hint-SQL boosted the execution accuracy of the current leading method to 71.58%, an absolute improvement of 4.37%. This study not only demonstrates the profound impact of tailored, dynamic hints in Text-to-SQL tasks but also offers a robust and automated framework for their generation, providing valuable insights for future research in this domain.

Key words Natural Language Processing; Text-to-SQL; Large Language Models; Prompt Engineering; Hint

1 引言

Text-to-SQL 任务旨在将自然语言问题翻译为可被数据库系统执行的 SQL 语句，使终端用户能够通过自然语言表达查询需求，而无需掌握数据库语言的专业知识。该方法显著降低了数据检索与分析的技术门槛，为商业智能、客户服务以及数据科学等多个领域的用户群体提供了重要支持。

随着大语言模型（Large Language Models，LLMs）的发展，基于 LLMs 的 Text-to-SQL 提示方法展现出了强大的潜力和优势[1–4]，成为 Text-to-SQL 研究的主流解决方案。这些方法通过自然语言提示词(prompt)引导 LLMs 执行 Text-to-SQL 任务。一个典型 Text-to-SQL 提示词包括：示例（demonstrations）、任务数据库模式、用户问题、任务指令（instruction）。除了这些内容外，近年来，研究者发现在提示词中加入额外建议，能有效提升 SQL 生成的质量。例如，C3-SQL[5]通过总结 SQL 的错误分布，发现 LLMs 的固有偏见造成了一类错误，因此在提示中建议“以防产生额外的结果，避免使用 LEFT JOIN、IN 或者 OR 关键词……”。为了避免与指令混淆，类似信息称为线索（hint）。C3-SQL[5]的线索作用于所有的 Text-to-SQL 任务，对于某些特定任务不一定适用，甚至可能产生误导。DEA-SQL[6]进一步细分线索的使用条件，针对不同类型的任务使用对应的线索，但是其中的线索依然是人为撰写。这类方法旨在使用线索传递具体的 SQL 生成建议，显式地引导 LLMs 执行或者避免某些特定的操作。然而，存在以下两个不足：（1）线索内容由人工预设，难以针对具体任务需求做出调整，无法满足所有任务的需求；（2）线索的作用范围过于狭窄，仅关注 SQL 操作层面的执行建议。针对现有方法的不足，本文提出基于自动线索生成的 Text-to-SQL 提示方法 Hint-SQL，通过线索生成智能体（Hints-generation Agent, HAgent）为每个 Text-to-SQL 任务自动生成定制化线索，该方法旨在克服现有线索的局限性，显著提升 SQL 生成的准确率。

图 1 显示了本文方法与现有方法的对比。现有方法人工预设线索，如 DEA-SQL 的线索提示 LLMs 使用“INTERSECT”等关键词。然而，在该任务中，“INTERSECT”并非必要，强行使用反而造成了错误。相比之下，HAgent 生成的定制化线索能够准确地匹配任务需求。此外，现有线索主要在 SQL 操作层面提供执行建议（下文称为“操作线索”），现有操作线索仅关注部分特定的操作，例如图 1 中 C3-SQL 线索的“COUNT”和“IN”等关键词，未能覆盖完整的 SQL 查询流程。为此，本文扩展操作线索的内容，例如，图 1 中的操作线索通过“filtering”、“group”和“order”等词汇清晰地描述了一条完整 SQL 查询流程。除了操作线索，Hint-SQL 进一步提出了语义线索和结构线索。语义线索消除用户问题中的模糊表达，如图 1 中所示，将用户问题中的“Which area”改写为“asking for the county”通过明确用户的查询目标，为操作线索提供准确的前置信息。结构线索则是将操作线索中的“group”等词汇映射到具体的 SQL 关键词。Hint-SQL 通过语义线索、操作线索和结构线索，从语义理解、操作规划到结构映射三个关键环节，为 Text-to-SQL 任务提供具体建议。

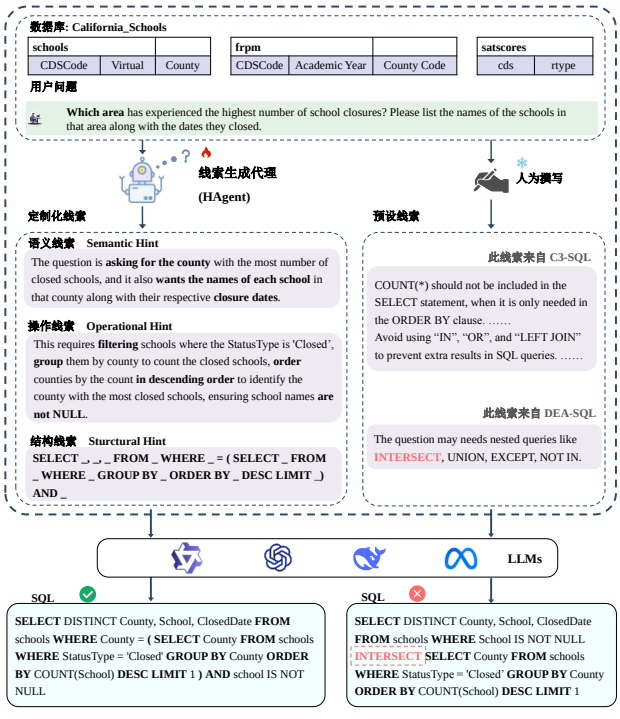

图 1 Hint-SQL 与现有方法的对比（省去了示例、任务指令等其它提示词内容）

为了自适应地生成线索，我们提出了两阶段微调框架，通过微调开源 LLMs 构建了线索生成智能体 HAgent。该框架包含监督微调和偏好学习两个阶

段，监督微调赋予模型基本的线索生成能力，而偏好学习则进一步提升了模型对细粒度信息的辨别与处理精度。在 Text-to-SQL 任务中，对细粒度信息的准确把握至关重要，例如，“grouping by the county”和“grouping by the city”之间的细微差别，就可能导致生成完全不同的 SQL 语句。传统的监督微调往往难以捕捉此类细节，而偏好学习阶段通过对比正例与负例样本，能够有效识别并抑制错误的细节信息[7]，提升线索的准确性。为了支持偏好学习，本文提出了一种基于线索生成与验证的算法，自动合成偏好学习所需的正负例数据。HAgent 采用递进式策略生成线索：首先生成语义线索，用于明确用户查询目标；然后基于语义线索生成操作线索，规划实现查询目标的操作步骤；最后结合前两者生成结构线索，以确定最终的 SQL 结构。该递进式策略旨在分解任务难度，在降低线索生成复杂度的同时，确保各阶段线索内容的一致性。如图 1 中所示，语义线索中的“county”字段为操作线索中的“group them by county”提供了模式实体的锚点，而操作线索中的“group them by county”进一步为结构线索中的“GROUP BY”提供了明确的指导。

本文的主要贡献包括以下三个方面：

（1）本文提出了一种新的 Text-to-SQL 提示方法 Hint-SQL，针对每个 Text-to-SQL 任务生成定制化的语义、操作和结构线索，为 LLMs 提供具体的建议，从而增强 LLMs 的 SQL 生成能力。

（2）本文提出了两阶段微调框架来构建和微调 HAgent。该框架自动合成微调所需数据，无需人工标注，通过监督微调和偏好学习优化两个阶段生成准确的线索。

（3）本文在 5 个主流数据集和 10 个常见 LLMs 上进行了广泛实验。实验显示，Hint-SQL 独立使用时可以媲美主流方法；当它与其他方法结合时，可以显著增强现有方法的性能。在 Spider[8]数据集上，Hint-SQL 将当前最好方法的准确率提升了 1.35%，达到了 89.26%；在更具挑战性的 BIRD[2]数据集上，Hint-SQL 将当前最好方法的准确率提升到了 71.58%，提升幅度达到 4.37%。

2 相关工作

随着 LLMs 的发展，基于 LLMs 的 Text-to-SQL 方法展示出强大的性能[9–15]，成为 Text-to-SQL 领域的主流解决方案[16]。如图 2 所示，当前主要 Text-to-SQL 研究集中在 Text-to-SQL 流水线的三个阶段：前处理、SQL 生成以及后处理[4,17]。前处理阶段优化 LLMs 输入内容，SQL 生成阶段引导 LLMs 生成 SQL，后处理阶段进一步优化 SQL 结果。每个阶段存在多种策略，本文提出的 Hint-SQL 是一种 SQL 生成策略，旨在使用线索提供具体 SQL 生成建议协助 LLMs 生成 SQL 语句。

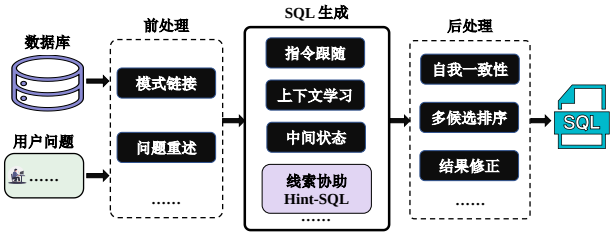

图 2 Text-to-SQL 三阶段流水线示意图

2.1 前处理与后处理策略

前处理阶段常使用模式链接(schema linking)[11,13,18]、问题重述[19,20]等策略。模式链接是 Text-to-SQL 领域的经典策略，旨在对数据库内容进行过滤，找出与当前任务高度相关的数据库模式元素。RSL-SQL[18]提出一种鲁棒的模式链接策略，显著的提升了 LLMs 的 Text-to-SQL 任务上的表现。问题重述则是通过对用户问题进行优化改写，使查询目标更加明确，从而提升 LLMs 对用户问题的理解[19,20]。本文提出的 Hint-SQL 在构造语义线索时，也采用了问题重述策略。

后处理常用的策略包括自我一致性(self-consistency)[5,14]、多候选排序[12,21]以及结果修正[11,13,22,23]等，其目的在于对生成的 SQL 进行优化修正，进一步提升 SQL 的正确性。

前处理和后处理在 Text-to-SQL 系统中表现出显著效果，被广泛使用。本文的重点不在前处理和后处理，而是专注 SQL 生成这一核心环节。但这并不意味着 Hint-SQL 无法与前处理或后处理策略兼容；相反，Hint-SQL 可以结合这些策略进一步提升性能（详见 4.3 小节）。Hint-SQL 的核心贡献在于对 SQL 生成策略的改进与优化。

2.2 SQL生成策略

常见的 SQL 生成方法包括以下三种策略：指令跟随、上下文学习以及中间状态。

指令跟随（instruction following）[24]是指 LLMs 能够理解并执行自然语言指令。研究者通过编写指令，要求 LLMs 完成特定行为。例如 Tan 等[25]通过指令引导 LLMs 逐子句生成完整的 SQL 语句。指令

还可以传达某些限制条件[6]。然而，目前大多数指令都是人为设定的，缺乏针对性。线索是一种特殊的指令，与传统指令用于传达任务要求不同，线索强调提供具体的 Text-to-SQL 建议。5.1 小节中详细讨论了 Hint-SQL 与指令跟随机制的联系。

上下文学习（in-context learning）[26]是指 LLMs 能够从提示词的示例中学习“输入-输出”的映射关系，又被称为少样本提示(few-shot prompting)。研究者通常关注如何选择高质量的示例[3,14,27]。其中，DAIL-SQL[14]通过语义和 SQL 结构的相似性，从示例检索库（训练集）中找出与当前任务高度相关的示例，是当前主流的 SQL 生成策略。Chang 等[28]研究了示例数量对结果的影响，发现 4 个示例即可逼近少样本提示的性能上限。尽管上下文学习是当前最有效的 SQL 生成策略，但依赖于示例的质量与数量，Hint-SQL 能有效缓解这一问题，这一点将在 5.2 小节中详细分析。

中间状态策略旨在提供 Text-to-SQL 过程的中间表达，比如用户问题分解[6,13,29]和 SQL 骨架(skeleton)[30–32]等，目的是降低直接生成 SQL 的复杂度。SQL 骨架由 SQL 关键字和占位符组成，是最接近 SQL 语句的中间表达。RESDSQL[33]与 FinSQL[34]采取“骨架-感知”生成策略，即先生成 SQL 骨架，再生成 SQL。SQL 骨架还可用于后处理阶段，用于判断 SQL 的正确性[35]。此外，通过对问题进行“模式掩码”处理，可以获得“问题骨架”，可用于检索高质量的少样本示例[36]。Hint-SQL 使用 SQL 骨架作为结构线索的载体。

2.3 Text-to-SQL线索

现有研究中的 Text-to-SQL 线索强调显式提供具体的 SQL 生成建议，其内容具有明确的指向性和可操作性。例如“当 ORDER BY 子句中需要 COUNT()时，COUNT()不需要出现在 SELECT 子句中”[5]。现有方法如 C3-SQL[5]、DEA-SQL[6]和 SQLfuse[37]通过统计和分析 Text-to-SQL 的常见错误，人为总结出一系列通用性原则，将其作为线索。以 DEA-SQL 为例，其通过统计错误发现，LLMs 不擅长解决复杂的嵌套任务和表连接问题。为此，DEA-SQL 提出 Classification & Hint 模块，将任务按嵌套和连接需求划分为 4 类，并为其匹配相应的线索。例如针对“nested”任务，线索内容为"The question may need nested queries like INTERSECT, UNION, EXCEPT, NOT IN"，提升了 LLMs 解决嵌套任务时的准确率。

然而，错误分布与具体的生成模型和数据集高度相关，不同模型与数据集的错误分布不尽相同，人为预设的线索难以保证对所有情况都有效。本文提出的 Hint-SQL 自动生成任务相关的定制化线索，具有更好的自适应性和性能表现。

2.4 Text-to-SQL Agent

Agent 作为能够独立执行任务的模块，在 Text-to-SQL 领域中得到了广泛应用。现有研究的重点主要集中于如何设计高效的多 Agent 协作机制，其中 MAC-SQL[13]与 MAG-SQL[38]构建的多 Agent 协作框架尤为典型。以 MAC-SQL 为例，其设计了三个核心 Agents：Selector、Decomposer 和 Refiner，分别负责预处理、SQL 生成和后处理任务，形成了完整的协作流程。除了直接应用于 Text-to-SQL 流水线，Agent 还在相关任务中展现了独特价值。例如，SQL-Factory[39]通过多 Agent 协作实现了高质量且大规模的 SQL 数据自动化合成；而 DB-GPT[40]则借助 Agent 对 SQL 进行分析，同时与用户交互，提升了系统的交互智能与分析能力。相比于上述方法，本文提出的 HAgent 是一种“即插即用”的独立模块，不依赖多 Agent 之间的协作，旨在通过生成可直接嵌入提示词的线索，来提升 SQL 生成的性能。

3 Hint-SQL 方法

Hint-SQL 整体流程如图 3 所示，包括两个主要阶段：线索生成和 SQL 生成。在线索生成阶段，智能体 HAgent 使用递进式的策略依次生成语义线索、操作线索和结构线索。每一阶段生成的线索为下一阶段的线索提供前置信息，旨在降低单步生成线索的难度以及确保各线索内容的一致性。在 SQL 生成阶段，语义线索、操作线索和结构线索分别传递“语义分析-操作规划-结构映射”这三个 Text-to-SQL 关键环节的辅助信息，三类线索共同辅助 LLMs 完成 Text-to-SQL 任务。下面将详细介绍 HAgent 的微调方法以及线索生成和 SQL 生成的实现过程。

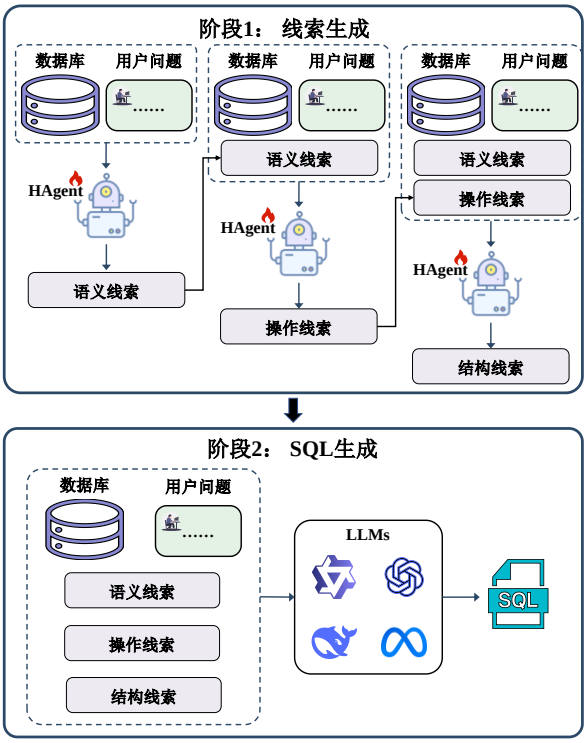

图 3 Hint-SQL 流水线示意图

3.1 HAgent微调方法

HAgent 是一个专注于线索生成的智能体，专门为每个 Text-to-SQL 任务自动生成定制化线索。如图 4 所示，HAgent 的构建过程包含两个阶段：监督微调与偏好学习。

监督微调旨在赋予模型基础的线索生成能力。线索生成要求模型精确捕捉数据库模式与用户问题之间的内在关联。为实现此目标，我们首先利用 LLMs 合成监督微调数据。每条数据样本均包含一个核心三元组：<数据库模式，问题，线索>。通过最小化交叉熵损失函数进行训练，模型能够学习在特定数据库模式约束下，问题与对应线索之间的映射关系，从而获得基础的线索生成能力。经过此阶段微调的模型被称为 HAgent-SFT。然而，仅通过监督学习，模型虽能模仿数据分布，但对线索的内在逻辑和细微偏差的辨别能力有限。

偏好学习的目标是进一步提升 HAgent-SFT 生成线索的准确性与可靠性。尽管 HAgent-SFT 已具备初步的线索生成能力，其生成的线索质量仍存在波动。因此，本文引入偏好学习纠正其错误的输出，从而优化 HAgent-SFT 的性能。具体而言，我们首先使用 HAgent-SFT 生成线索，随后对这些线索的正确性进行验证。基于验证结果，我们构建了一个偏好数据集，其中每个样本包含一个核心四元组<数据库模式，问题，正例线索，负例线索>。通过最大化正例线索与负例线索之间的奖励差异，HAgent-SFT 能够学习到线索正确性的评判标准，进而促使其生成质量更高的线索。

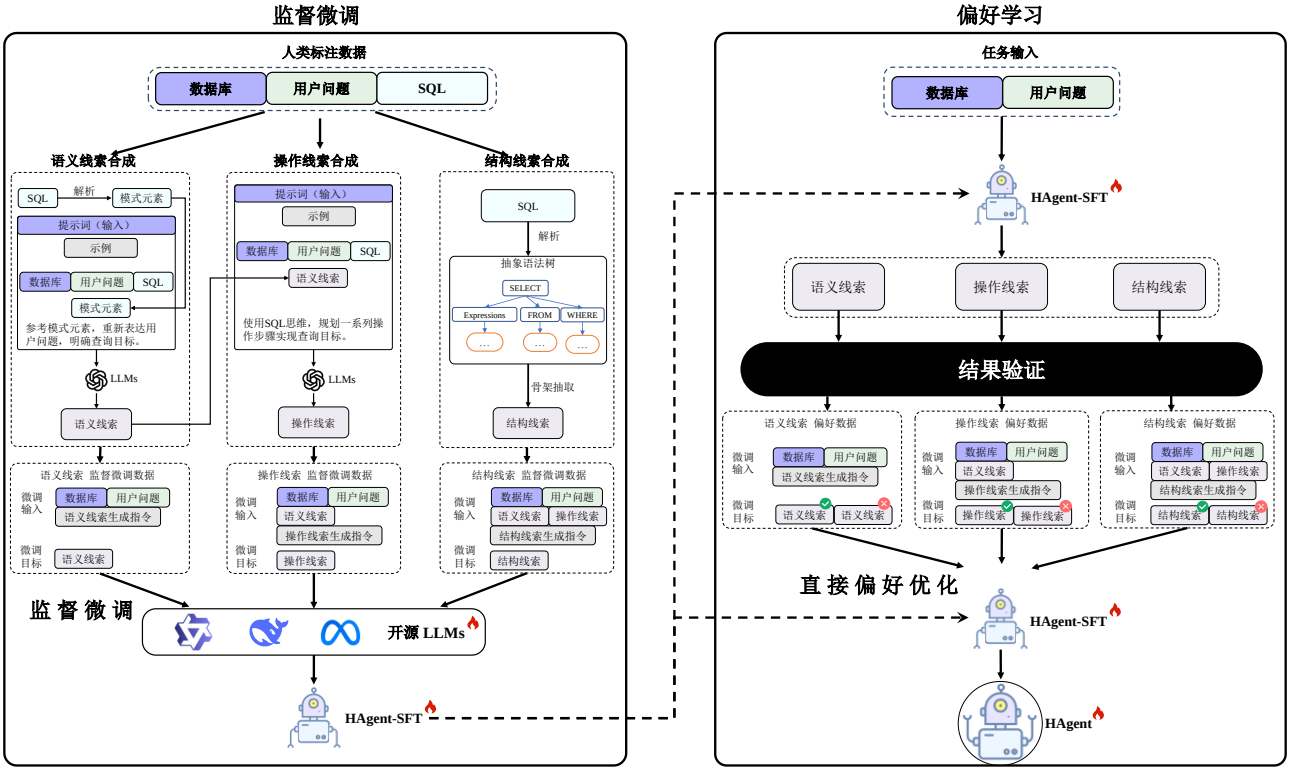

图 4 HAgent 两阶段微调框架

3.1.1 监督微调

本文基于 Spider 和 BIRD 数据集中的人工标注数据，构建用于监督微调的线索数据集。语义线索的合成依赖 LLMs。首先，解析 SQL 语句，提取 SELECT 子句中的模式元素（即表名或列名），然后使用这些模式元素构建提示词。提示词内容主要包括示例、数据库、用户问题、SQL 语句及模式元素，旨在引导 LLMs 使用模式元素分析用户问题的查询目标，消除问题中的模糊表达。类似地，操作线索的合成也依赖于 LLMs。其提示词内容包括示例、数据库、用户问题、SQL 语句及语义线索，旨在引导 LLMs 使用 SQL 特有思维规划具体的操作步骤。附录 A.1 和 A.2 中提供了用于合成语义线索和操作线索的提示词样例。相比之下，结构线索的合成较为简单，只需解析 SQL 的抽象语法树（AST），抽取骨架即可。这种分步、递进的合成策略，为不同类型的线索量身定制了高质量的训练数据。

基于合成线索，构建监督微调数据集$\mathcal{D}_{sft}$，其构成如下：

$$\mathcal{D}_{sft}=\left\{\begin{array}{l}\mathcal{D}_{sft}^{a}=\left\{\begin{array}{c}(x_i,y_i)\mid \\ x_i=(S_i,Q_i,I^a),\\ y_i=A_i^{syn},i=1,2,\ldots,N\end{array}\right\},\\ \mathcal{D}_{sft}^{b}=\left\{\begin{array}{c}(x_i,y_i)\mid \\ x_i=(S_i,Q_i,A_i^{syn},I^b),\\ y_i=B_i^{syn},i=1,2,\ldots,N\end{array}\right\},\\ \mathcal{D}_{sft}^{c}=\left\{\begin{array}{c}(x_i,y_i)\mid \\ x_i=(S_i,Q_i,A_i^{syn},B_i^{syn},I^c),\\ y_i=C_i^{syn},i=1,2,\ldots,N\end{array}\right\}.\end{array}\right\}\#(1)$$

其中，$\mathcal{D}_{sft}^{a}$、$\mathcal{D}_{sft}^{b}$和$\mathcal{D}_{sft}^{c}$分别代表语义线索、操作线索和结构线索的监督微调数据集，其各自的数据组成如图 4 中所示。S代表数据库模式，Q代表自然语言问题。$\mathrm{A}^{\mathrm{syn}}$、$B^{syn}$和$C^{syn}$分别代表合成的语义、操作和结构线索。$I^a$、$I^b$和$I^c$分别是语义线索、操作线索和结构线索的生成指令。$x$代表微调时的数据输入，$y$代表微调时的数据输出。

给定由输入x和目标输出y组成的$\mathcal{D}_{sft}$，监督微调开源 LLMs 的过程可以表示为最小化以下的负对数似然损失函数：

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{sft}}\left[\sum_{t=1}^{T}log\,p_\theta(y_t\mid y_{1:t-1},x)\right]\#(2)$$

其中，θ表示模型的参数，$p_\theta(y_t\mid y_{1:t-1},x)$代表输入为$x$时，目标输出$y$的条件概率分布。$T$是$y$的序列总长度，$t$是自回归解码的步骤。

3.1.2 偏好学习

偏好学习（preference learning）旨在从偏好数据中学习偏好关系，鼓励模型区分正确和错误的输出响应，从错误响应中学习，进而提升模型对正确响应的偏好。本阶段使用偏好学习进一步微调 HAgent-SFT，以提升模型对线索细粒度信息的控制能力。

偏好数据集的构建依赖 HAgent-SFT，具体流程如图 4 所示：首先使用 HAgent-SFT 生成语义线索、操作线索和结构线索，随后验证线索的正确性。由于语义线索和操作线索采用自然语言表述，难以直接从文本层面评估其有效性，因此本文首先单独使用语义线索或操作线索引导 LLMs 生成 SQL，根据 SQL 正确性反推线索的有效性。以语义线索为例，分别使用合成的语义线索A^{syn}与由 HAgent-SFT 生成的语义线索A^{sft}引导 LLMs 完成 Text-to-SQL 任务，分别得到结果SQL^{syn}和SQL^{sft}。当SQL^{syn}和SQL^{sft}一个正确且另一个错误时，标记对应的语义线索为正例或者负例。操作线索的处理与语义线索相同。结构线索的偏好数据构建则略有不同，由于结构线索使用 SQL 骨架作为载体，能通过字符匹配直接找出生成的结构线索C^{sft}中的初始负例样本。考虑到 SQL 结构的多样性，即便字符层面被判负的C^{sft}可能是同样合理的。因此，在初始负例样本的基础上，需进一步过滤掉被误判的样本。具体而言，首先引导 LLMs 使用初始负例的C^{sft}生成SQL^{sft}，当SQL^{sft}执行结果正确，且从SQL^{sft}中提取出的骨架包含了C^{sft}的所有元素时，则认为该样本是被误判的样本，并将其移出结构线索的偏好数据集。附录 B 展示了构建偏好数据集的算法内容。

偏好数据集$\mathcal{D}_{pl}$的构成如下：

$$\mathcal{D}_{pl}=\left\{\begin{array}{l}\mathcal{D}_{pl}^{a}=\left\{\begin{array}{c}(x_i,y_i,y_i^r)\mid \\ x_i=(S_i,Q_i,I^a),\\ y_i=A_i,y_i^r=A_i^r,\\ i=1,2,\ldots,N_a\end{array}\right\},\\ \mathcal{D}_{pl}^{b}=\left\{\begin{array}{c}(x_i,y_i,y_i^r)\mid \\ x_i=(S_i,Q_i,A_i,I^b),\\ y_i=B_i,y_i^r=B_i^r,\\ i=1,2,\ldots,N_b\end{array}\right\},\\ \mathcal{D}_{pl}^{c}=\left\{\begin{array}{c}(x_i,y_i,y_i^r)\mid \\ x_i=(S_i,Q_i,A_i,B_i,I^c),\\ y_i=C_i,y_i^r=C_i^r,\\ i=1,2,\ldots,N_c\end{array}\right\}.\end{array}\right\}\#(3)$$

其中，$\mathcal{D}_{pl}^{a}$、$\mathcal{D}_{pl}^{b}$和$\mathcal{D}_{pl}^{c}$分别代表语义线索、操作线

索和结构线索的偏好数据集，其各自的数据组成如图 4 中所示。A、B和C指的是正确的语义、操作和结构线索，它们的有效性经过了验证，不会误导 LLMs 产生错误结果。A^r、B^r和C^r则是错误的语义、操作和结构线索。I^a、I^b和I^c分别是语义线索、操作线索和结构线索的生成指令，与监督微调数据集的生成指令相同。x代表微调时的数据输入，y代表偏好学习的正面响应，y^r则代表偏好学习的负面响应。

本文使用直接偏好优化算法（direct preference optimization）[41]进一步微调 HAgent-SFT。给定由输入x、正面输出响应y以及负面输出响应y^r组成的偏好数据，直接偏好优化的目标是最大化以下目标函数：

$$\mathop{\mathbb{E}}_{(x,y,y^r)\sim\mathcal{D}_{pl}} log\sigma\begin{pmatrix}\beta\, log\frac{p_\theta(y|x)}{p_{\text{ref}}(y|x)}-\\ \beta\, log\frac{p_\theta(y^r|x)}{p_{\text{ref}}(y^r|x)}\end{pmatrix}\#(4)$$

其中，p_θ表示目标模型（即 HAgent）预测的概率分布，p_{ref}表示参考模型（即 HAgent-SFT）的概率分布，β是一个调节目标模型相对于参考模型偏离程度的参数。

直接偏好优化算法基于偏好数据直接微调模型，当目标模型对正面响应y的概率$p_\theta(y\mid x)$高于参考模型对正面响应y的概率$p_{\text{ref}}(y\mid x)$时，$log\big(p_\theta(y\mid x)/p_{\text{ref}}(y\mid x)\big)$则会增大，导致目标函数的数值提升；反之，若对目标模型对负面响应y^r更大，此时$log\big(p_\theta(y^r\mid x)/p_{\text{ref}}(y^r\mid x)\big)$增大，导致目标函数的数值降低。通过这种方式，偏好学习强化了模型对正面响应的偏好，同时抑制错误输出的概率，提升了 HAgent 生成线索的质量，5.6 小节中展示了偏好学习提升线索质量的具体样例。

3.2 线索生成

语义线索、操作线索和结构线索在逻辑上呈递进关系：语义线索用于明确查询目标，操作线索规划实现查询目标的具体步骤，而结构线索则将操作步骤映射到 SQL 结构。因此，HAgent 采用递进式策略生成线索，其生成过程可以形式化表达为：

$$\begin{aligned}A &= HAgent(\mathrm{S},Q,I^a),\\ B &= HAgent(\mathrm{S},\mathrm{Q},A,I^b),\\ C &= HAgent(\mathrm{S},\mathrm{Q},A,B,I^c).\end{aligned}\ \#(5)$$

其中，A、B和C分别代表语义线索、操作线索和结构线索。语义线索作为操作线索的前置信息，语义线索和操作线索作为结构线索的前置信息。这种递进式生成方式确保了逻辑上的一致性，同时降低了单步生成操作线索和结构线索的难度。本文通过设计提示词引导 HAgent 生成不同类型的线索，附录 A.3 中展示了生成线索所使用的提示词样例。

3.3 SQL生成

Hint-SQL 通过语义线索、操作线索和结构线索的协同作用，共同为 LLMs 提供具体的建议，引导 LLMs 生成 SQL。多线索协作旨在使线索信息更加丰富和多样化，从而更鲁棒地支持 LLMs 生成高质量的 SQL。其生成过程可以通过以下条件概率形式化表达：

$$P_{LLMs}(y\mid x)=P_{LLMs}\left(y\mid\begin{pmatrix}\{S_i,Q_i,A_i,B_i,C_i,I,,y_i\}_{i\le m},\\ (S,Q,A,B,C,I)\end{pmatrix}\right)\#(6)$$

其中，x代表给予 LLMs 的完整提示词，它由m个示例以及当前任务输入构成。当前任务输入主要包括：数据库模式S、用户问题Q、三类线索（A,B,C）以及任务指令I。当示例数量$m=0$时，即为零样本提示，当$m>0$时，则为少样本提示。附录 A.4 中详细展示了 SQL 生成提示词样例。

4 实验结果

4.1 实验设置

数据集。本文实验使用 5 个主流的 Text-to-SQL 数据集：Spider[8]、BIRD[2]、SYN[42]、REALISTIC[43]和 DK[44]。Spider 是一个跨领域数据集，包含 7000 条训练集数据和 1034 条开发集数据，涵盖 200 个不同的数据库和 138 个领域。BIRD 是当前最具挑战的数据集之一，侧重于海量真实的数据库内容，强调了自然语言问题和数据库内容之间的知识推理，包含 9428 条训练集数据和 1534 条开发集数据。SYN、REALISTC、DK 都是基于 Spider 的鲁棒性数据集，用于测试 Text-to-SQL 方法的鲁棒性。其中，SYN 通过字符串匹配，将问题中的模式名称替换为其同义词，以实现问题表达的多样化；REALISTIC 通过替换问题中提及的模式项，使问题表述更贴近真实世界场景，从而提升任务的现实性和适用性；DK 要求 Text-to-SQL 方法具备领域知识推理能力，从而支持更复杂的语义理解和推理过程。

评测指标。对于 Spider 及其鲁棒性数据集，本文采用 Spider 官方提供的评估脚本，并使用执行准确率（Execution Accuracy, EX）和测试套件准确率

（Test Suite Accuracy, TS）两种指标。EX 衡量 SQL 的执行结果是否与标准 SQL 的执行结果完全相同。TS 是一个更严格的指标，基于原数据库扩展了多个测试数据库，评估 SQL 是否能在所有测试数据库中通过 EX 测试。对于 BIRD 数据集，本文采用其官方提供的评估脚本，专注于 EX 评估。

对比方法。本文选择了 8 种 Text-to-SQL 提示方法作为对比方法。为了公平比较结果，本文使用 GPT-4o[45]模型复现了这些方法。下面简要介绍这些方法：

（1）Zero-shot 是一个基础提示方法，用于呈现 LLMs 本身的 Text-to-SQL 能力；

（2）C3-SQL[5]是一个经典的零样本提示方法，通过设计独特的指令，引导 LLMs 完成一系列 Text-to-SQL 子任务；

（3）DART-SQL[20]通过问题重述和结果修正策略，对前处理和后处理阶段进行优化，在现有方法（如 C3-SQL）的基础上进一步提升性能；

（4）DIN-SQL[11]是一个经典的分解策略，将 Text-to-SQL 分解成 4 个子任务，通过结合少样本提示和思维链提示[46]来引导 LLMs 依次完成子任务；

（5）DEA-SQL[6]提出工作流范式，同样将 Text-to-SQL 分解成 4 个子任务，并依次完成它们；

（6）DAIL-SQL[14]是一个经典的少样本提示方法，检索与当前任务最相似的示例，使用这些示例引导 LLMs 完成 Text-to-SQL 任务；

（7）MAC-SQL[13]是一个多智能体框架，涉及模式链接、问题分解和 SQL 优化等策略；

（8）RSL-SQL[18]提出了一种鲁棒的模式链接策略，强调了模式链接的重要性。

实现细节。本文提出两阶段微调框架，自动合成微调数据用于支持 HAgent 的两阶段微调。基于 Spider 和 BIRD 训练集，本文使用 GPT-4o 模型和 Deepseek-Coder-V2-Lite-Instruct（后文简称 Deepseek-Coder）[47]两个模型来合成本文所需的监督微调数据与偏好数据集。微调数据类型与数量分布如表 1 中所示。

从表 1 中可以看出，偏好学习的数据样本数量明显少于监督微调的数据样本，尤其是在语义线索和操作线索部分。这主要是因为语义线索和操作线索属于自然语言指令，难以直接判断其有效性。而偏好学习数据需要同时包含正例样本（即真实值）和负例样本，因此只能使用线索辅助 LLMs 生成 SQL，通过判断 SQL 正确性反推线索的真实性。然而，语义线索在单独使用时作用较弱，往往不足以将一个错误的生成结果修正为正确，因此难以通过这种方式匹配到足够数量的正例；操作线索的作用相对更强，但仍存在一定限制，导致数据量有限。相比之下，结构线索可直接从标注 SQL 中提取真实值（即 SQL 骨架），因此数据量相对较多。

表 1 微调数据类型与数量分布。

	语义线索	操作线索	结构线索
监督微调	16428	16428	16428
偏好学习	821	3854	6453

HAgent 的微调过程在 8×RTX4090 计算集群上进行。监督微调阶段，使用 AdamW 优化器，学习率为 5e-5，使用余弦预热调度器，训练周期为 5。在表 1 所示的数据规模下，监督微调过程耗时约 500 分钟。偏好学习阶段，使用直接偏好优化算法，使用 Adam 优化器，学习率为 5e-6，β参数设置为 0.2，训练周期为 3。在表 1 所示的数据规模下，偏好学习过程耗时约 110 分钟。本文主要实验所用的 HAgent 是基于 Deepseek-Coder 微调得到的。

SQL 生成依赖 LLMs，本文总共涉及到 10 个开源 LLMs：LLaMA3.1-8B-Instruct[48]（LLaMA3.1-8B）、LLaMA3.1-70B-Instruct[48]（LLaMA3.1-70B）、Qwen2.5-7B-Instruct[49]（Qwen2.5-7B）、Qwen2.5-72B-Instruct[49]（Qwen2.5-72B）、Deepseek-Coder-V2-Lite-Instruct[47]（DeepSeek-Coder）、GPT-4o-mini-2024-07-18[45]（GPT-mini）、GPT-4o-2024-08-06[45]（GPT-4o）、Claude-3-5-haiku-20241022[50]（Claude-3.5）、Gemini-1.5-flash[51]（Gemini-1.5），DeepSeek-V3[42]。为确保所有的实验结果可复现，本文设置 LLMs 的生成温度为 0.1，频率惩罚为 0，核采样频率为 1，以保证 LLMs 输出的稳定性和确定性。

4.2 主要实验结果

表 2 展示了 Hint-SQL 的主要实验结果。Hint-SQL 专注于 SQL 生成策略，不涉及前处理和后处理过程。

在 Spider 和 BIRD 数据集上，Hint-SQL 单独使用时的表现与当前最佳生成策略 DAIL-SQL 相当，进一步使用少样本示例后，其准确率超越 DAIL-SQL，分别提升了 0.97%和 3.46%。此外，Hint-SQL 不仅可以作为独立方法使用，还能与其他方法结合以增强其性能，例如结合 RSL-SQL 后，

其最终执行准确率在 Spider 和 BIRD 数据集上分别提高了 1.35%和 4.37%。

表 2 Spider 和 BIRD 数据集上的结果(使用 GPT-4o 模型)。

方法	示例数量 (SQL 生成)	Text-to-SQL 三阶段所用策略			Spider-dev		BIRD-dev
		前处理策略	SQL 生成策略	后处理策略	EX	TS	EX
Zero-shot	0	-	-	-	78.43	68.86	56.19
C3-SQL	0	模式链接	指令跟随	结果修正	82.98	74.85	56.84
DART-SQL	0	问题重述	指令跟随	结果修正	83.37	74.95	57.24
DIN-SQL	11	模式链接	上下文学习	结果修正	84.91	76.40	57.89
DEA-SQL	3	模式链接	上下文学习	结果修正	85.98	77.18	58.54
DAIL-SQL	5	-	上下文学习	-	86.17	78.63	59.06
MAC-SQL	2	模式链接	中间状态	结果修正	86.37	78.82	60.37
RSL-SQL	3	模式链接	上下文学习	结果修正	87.91	80.01	67.21
Hint-SQL	0	-	线索协助	-	85.11	77.56	61.02
Hint-SQL	5	-	线索协助	-	87.14	79.50	62.52
Hint-SQL + RSL-SQL	3	模式链接	线索协助	结果修正	**89.26**	**81.43**	**71.58**

综合表 2 结果可见，Hint-SQL 作为一种专注于 SQL 生成策略的新方法，其表现超越了当前最好的 SQL 生成策略。更重要的是，Hint-SQL 展现出卓越的增强能力，能够有效提升现有方法的表现。证明了 Hint-SQL 不仅自身是一种有效的 SQL 生成策略，还具备与现有方法高效协同的能力。

4.3 增强实验结果

本小节进一步分析 Hint-SQL 作为增强模块对其他方法的性能提升效果。图 5 展示了将 Hint-SQL 用作增量式方法的过程：首先，通过 HAgent 为每个 Text-to-SQL 任务生成定制化线索，然后将这些线索插入至 SQL 生成提示词中。该方法不仅简洁易用，而且能大幅提高 SQL 生成的准确率。

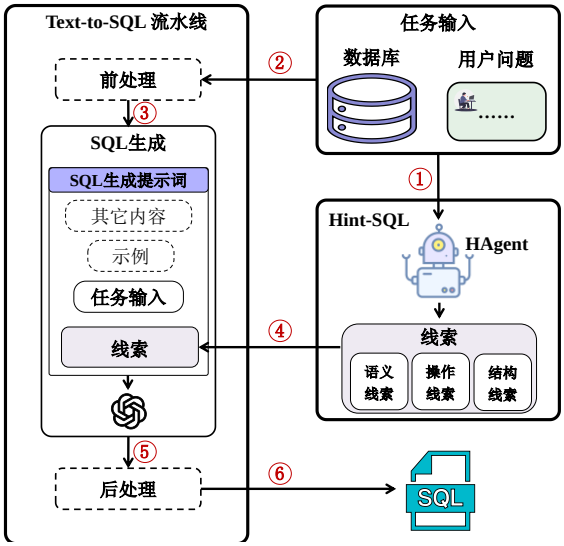

图 5 Hint-SQL 的增量式应用框架

本文在 6 个现有方法上评估 Hint-SQL 的增强表现，图 6 和图 7 分别展示了在 Spider 和 BIRD 数据集上的实验结果。实验结果表明，在 Spider 和 BIRD 数据集上，Hint-SQL 将这些方法的执行准确率平均分别提升了 2.74%和 4.63%。其中，在最具挑战性的 BIRD 数据集上提升显著，DIN-SQL、MAC-SQL 和 RSL-SQL 的准确率提升分别达到了 5.21%、5.47%和 4.37%。这些方法均设计了巧妙的前处理与后处理策略，而 Hint-SQL 则专注于 SQL 生成策略，与它们形成优势互补，从而显著地提升了这些方法的整体性能。提升效果揭示出，SQL 生成环节正是当前 Text-to-SQL 方法的性能瓶颈。现有方法常通过复杂的前处理或后处理来规避或修补生成阶段的缺陷，却未能直接提升模型的核心逻辑推理能力。Hint-SQL 的定制化线索则为 LLMs 提供了“领域专家”般的结构化思维框架。通过明确的语义、操作与结构指导，使模型能更精确地构建 SQL。

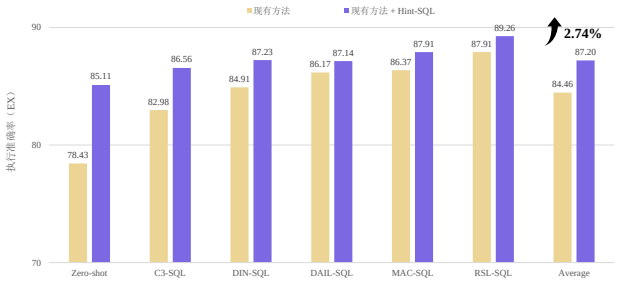

图 6 Spider 上的增强结果（使用 GPT-4o 模型）

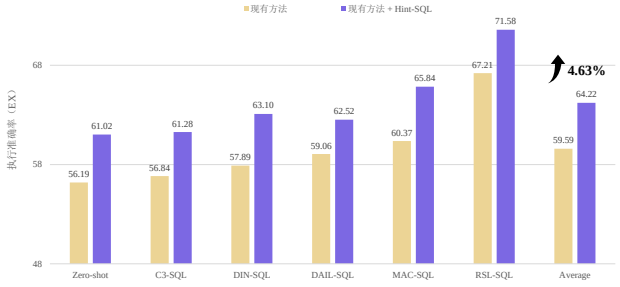

图 7　BIRD 上的增强结果（使用 GPT-4o 模型）

4.4　不同难度任务的实验结果

根据所涉表格数量、查询逻辑复杂度等因素，Text-to-SQL 任务可被划分为不同难度等级。表 3 和表 4 分别展示了 Hint-SQL 方法在 Spider 和 BIRD 数据集上，针对不同难度任务的实验结果。

表 3　Spider 数据集上不同难度任务的执行准确率。

方法	简单	中等	困难	极难	全部
Zero-shot + GPT-4o	90.32	84.01	71.02	53.61	78.43
C3-SQL + GPT-4o	92.74	87.16	81.25	59.04	82.98
DAIL-SQL + GPT-4o	94.76	**90.54**	83.52	63.25	86.17
MAC-SQL + GPT-4o	**95.56**	89.64	84.09	66.27	86.37
Hint-SQL + GPT-4o	95.16	90.09	**86.93**	**67.47**	**87.14**

表 4　BIRD 数据集上不同难度任务的执行准确率。

方法	简单	中等	困难	全部
Zero-shot + GPT-4o	64.86	43.97	40.00	56.19
C3-SQL + GPT-4o	63.35	48.71	41.38	56.84
DAIL-SQL + GPT-4o	66.92	48.28	43.45	59.06
MAC-SQL + GPT-4o	66.81	**53.66**	40.69	60.37
Hint-SQL + GPT-4o	**69.95**	51.51	**50.34**	**62.52**

结果表明，Hint-SQL 在处理高难度 Text-to-SQL 任务时表现出更强的优势。具体来说，表 3 结果显示，在 Spider 数据集中的困难和极难任务上，Hint-SQL 的执行准确率分别超越 MAC-SQL 2.84%和 1.2%。表 4 中的结果进一步展示了 Hint-SQL 在更具挑战性的 BIRD 数据集上的表现，Hint-SQL 在其最高难度的任务中实现了 6.98%的执行准确率提升，展现了处理复杂 Text-to-SQL 任务的能力。

与简单任务相比，困难任务查询逻辑更加复杂，例如问题“从 2000 年到 2005 年，1985 年以前出生的驾驶员和圈数超过 50 的驾驶员的比例是多少？”，该问题涉及复杂的计算与推理。以上述问题为例，Hint-SQL 的定制化操作线索为：“完成这个目标，需要关联赛事、成绩、车手三表，筛选 2000-2005 年的数据，统计满足年龄和单圈数要求的车手人数，并除以总参赛人数以计算百分比”。操作线索以一种符合 SQL 逻辑的思维方式，对任务所需的操作步骤进行推理，对于解决困难任务具有显著效果。

4.5　鲁棒性数据集结果

表 5 展示了 Hint-SQL 在鲁棒性数据集 SYN、REALISTIC 和 DK 上的实验结果。SYN 和 REALISTIC 通过在问题中进行同义词替换，来检测 Text-to-SQL 方法在语义模糊挑战下的性能。DK 则是检测 Text-to-SQL 在领域知识缺乏情况下的推理能力。表 5 的结果表明，Hint-SQL 在 SYN、REALISTIC 和 DK 数据集上均取得了最好的表现，相比于 MAC-SQL，平均表现提升 2.55%。

表 5　鲁棒性数据集的实验结果（使用 GPT-4o 模型）。

方法	SYN		REALISTIC		DK	平均
	EX	TS	EX	TS	EX	
C3-SQL	70.79	62.47	73.23	64.76	65.23	67.30
DAIL-SQL	72.24	64.12	79.53	74.02	71.21	72.22
MAC-SQL	75.24	67.41	81.10	76.77	73.64	74.83
Hint-SQL	**78.14**	**69.44**	**84.91**	**78.92**	**75.51**	**77.38**

在 SYN 和 REALISTIC 数据集上的实验结果表明，Hint-SQL 能够有效应对语义模糊的挑战。这得益于 Hint-SQL 的语义线索，该线索专为消除问题中的模糊表达而设计，通过明确查询目标来消除问题中的语义模糊，帮助 LLMs 更好地理解问题与数据库的关联。

5　实验分析

5.1　零样本场景结果分析

本节分析 Hint-SQL 在零样本场景中的表现。零样本场景下，LLMs 仅通过自然语言描述（如线索和指令）获取信息，高度依赖 LLMs 的指令跟随能力完成 Text-to-SQL 任务。该场景下的实验结果能够直观体现定制化线索对 Text-to-SQL 任务的提升效果。

5.1.1　动态线索有效性分析

现有研究如 C3-SQL 和 DEA-SQL 通过总结 Text-to-SQL 的常见错误，设计了相应的静态线索。然而，这些线索是静态的，无法根据具体任务动态调整；此外，当模型或数据集发生变化时，难以适应所有场景。为解决这一问题，本文提出 Hint-SQL，

能够自动生成动态的线索。

为直观比较动态线索与静态线索的效果，本小节设计了相关实验。表 6 展示了实验结果。在该实验中，Zero-shot 方法作为对照组，用于反映模型本身的能力；C3-SQL-Hint 仅使用 C3-SQL 中的静态线索生成 SQL，不涉及额外的前处理和后处理；DEA-SQL-Hint 同样仅使用 DEA-SQL 提出的静态线索生成 SQL。零样本场景下，上述对照方法与 Hint-SQL 的唯一区别在于所使用的线索不同。

表 6 零样本场景下动态线索与静态线索对比。

方法	Spider		BIRD
	EX	TS	EX
Zero-shot + GPT-4o	78.43	68.86	56.19
静态线索			
C3-SQL-Hint + GPT-4o	**80.01**	**70.60**	55.28
DEA-SQL-Hint + GPT-4o	79.11	69.92	**56.39**
动态线索			
Hint-SQL + GPT-4o	**85.11**	**77.56**	**61.02**
语义线索 + GPT-4o	78.82	69.53	56.71
操作线索 + GPT-4o	82.01	72.82	58.28
结构线索 + GPT-4o	82.98	73.50	57.24

实验结果表明，动态线索的效果显著优于静态线索。C3-SQL 的线索来源于对 Spider 数据集错误结果的总结，然而这些线索并不适用于 BIRD 数据集，导致 C3-SQL 线索在 BIRD 数据集上的表现甚至低于 Zero-shot 方法。DEA-SQL 的线索主要针对表连接和嵌套操作提出建议，但过于宽泛且缺乏针对性，对 LLMs 的帮助较小。相比之下，Hint-SQL 使用为每个任务动态生成的自适应线索，其结果显著优于使用静态线索的方法。

进一步分析不同类型线索的作用发现，对于 Spider 数据集，结构线索更加重要；而对于 BIRD 数据集，操作线索更加重要。这是因为，Spider 数据集相对简单，涉及复杂查询过程的任务较少，而 BIRD 刚好相反。操作线索在处理复杂任务时更加有效，因此在 BIRD 数据集上的重要性更高。

5.1.2 线索的模型适用性分析

Hint-SQL 方法的效果依赖于 LLMs 的指令跟随能力，而此能力在不同模型间存在显著差异。此外，考虑到部分 LLMs 在预训练阶段可能已接触过 Spider 或 BIRD 等基准测试集，这可能导致其基线性能虚高。为全面评估 Hint-SQL 的有效性与泛化能力，本节在多种 LLMs 上进行了广泛实验

图 8 和图 9 展示了 Hint-SQL 在不同尺寸 LLM 上的零样本性能。结果显示，在 Spider 数据集上，该方法为小、大尺寸模型分别带来 6.96%和 4.82%的平均增益；在 BIRD 数据集上，增益则分别高达 17.79%和 8.11%，证明了其广泛的适用性。

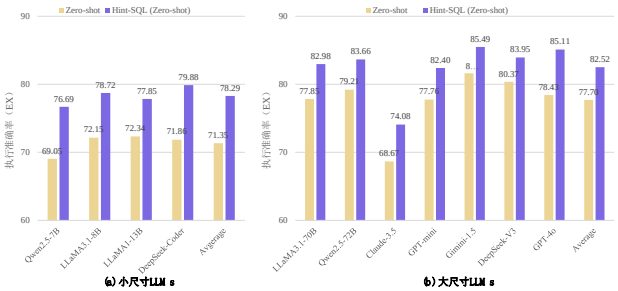

图 8 零样本场景下 Hint-SQL 的结果（Spider 数据集）

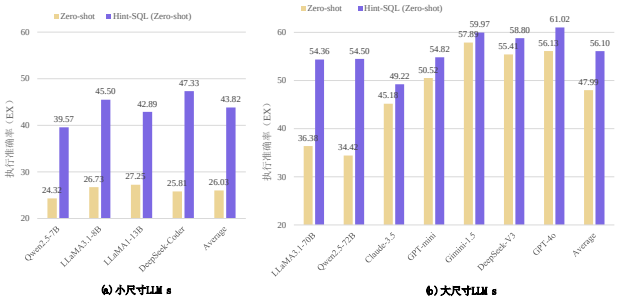

图 9 零样本场景下 Hint-SQL 的结果（BIRD 数据集）

为了尽可能地排除数据污染的干扰，我们选取了四款基线较低的小尺寸模型（Qwen2.5-7B、LLaMA3.1-8B、LLaMA1-13B、DeepSeek-Coder）进行了实验，它们在 Spider 和 BIRD 数据集上有着较低的基线准确率，意味着它们受数据污染的影响较低。它们的表现可以更加客观地展现 Hint-SQL 的有效性。图 8 和图 9 的结果表明，在 Spider 和 BIRD 数据集上，Hint-SQL 对这些模型的平均增幅分别为 6.96%与 17.79%。在 BIRD 数据集上的表现尤为显著，由于小尺寸模型在 BIRD 上的基线准确率普遍更低，这为 Hint-SQL 方法带来了更大的性能提升空间。

值得注意的是，图 9 中，我们在 BIRD 数据集上使用 LLaMA1-13B 模型进行了实验。该模型的发布时间早于 BIRD 数据集的发布时间，能够确保该模型在训练阶段未接触过 BIRD 数据集。其实验结果更加客观地验证了 Hint-SQL 的有效性。

5.2 少样本场景结果分析

本节分析 Hint-SQL 在少样本场景中的表现。少样本场景中，示例与定制化线索共同作用，进一步提升 LLMs 的 Text-to-SQL 的性能。本节从示例的数量和质量两个方面分析少样本场景下

Hint-SQL 的性能。

5.2.1 示例数量分析

图 10 和图 11 分别展示了 Spider 和 BIRD 数据集上，示例数量对 Hint-SQL 表现的影响。此实验选择 DAIL-SQL 作为对照方法，在示例数量相同的条件下，DAIL-SQL 与 Hint-SQL 使用完全相同的示例。图中结果可以看出，Hint-SQL 对示例数量并不敏感，仅使用 1 个示例便可大幅度提升表现，逼近该方法的性能上限，而 DAIL-SQL 往往需要更多的示例才能达到同等水平。这一趋势在 DeepSeek-V3、GPT-4o 和 GPT-mini 模型上均得到了验证。

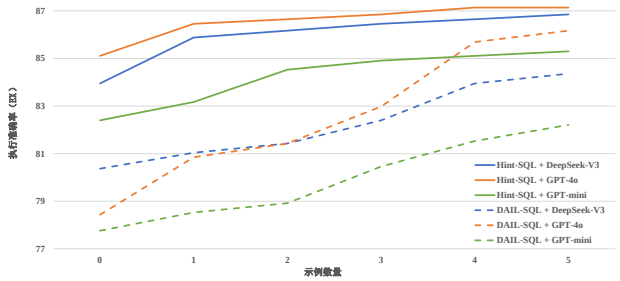

图 10 示例数量对结果的影响（Spider 数据集）

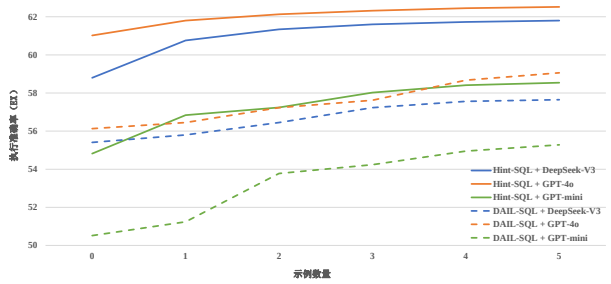

图 11 示例数量对结果的影响（BIRD 数据集）

本文认为，示例和线索的目的相同，都是向 LLMs 提供信息。然而，两者的方式存在差异：示例通过“输入-输出”映射传递隐式信息，旨在引导 LLMs 模仿示例的行为；而线索则传递显式信息，直接向 LLMs 提供具体建议，引导其按照建议完成任务。示例中的隐式信息往往不够直接，在数量受限的情况下，LLMs 能“模仿”的对象有限。Hint-SQL 对示例数量的依赖较小，因为其定制化的语义、操作和结构线索包含足够的信息。因此，仅使用少量示例，Hint-SQL 的表现便能显著提升。

从另一个角度看，这表明线索提供的显式指导在效率上要高于示例提供的隐式类比。当模型已经接收到关于“做什么”和“如何做”的清晰指令时，它就不再需要从大量范例中进行复杂的归纳和模仿。这种特性在现实应用中极具价值，因为它意味着可以显著降低对高质量、大规模示例库的依赖，从而降低部署成本和提问延迟，使方法更具实用性。

5.2.2 示例质量分析

示例的质量同样重要，当前方法通常基于相似性或者多样性原则，从训练集中检索高质量示例[3,14,27]。然而，在真实场景中，往往缺乏一个像训练集这样与目标任务高度匹配的示例检索库，因此难以保证示例的质量。

本小节设置了 4 种示例检索策略，以得到不同质量的示例，从而评估示例质量对 Hint-SQL 的影响。这 4 种策略分别是：（1）相似检索，即选择与当前任务最相似的示例[14]，是当前主流的示例检索策略，示例质量较高；（2）随机检索，即随机选择示例，是一种朴素的示例选择策略；（3）难度检索，即选择最高难度的示例，示例质量较差[2]；（4）交叉检索，即交换训练集作为示例检索库（检索方法遵循相似性），用于模拟真实场景中示例检索库与目标任务不匹配的情况。

图 12 和图 13 分别展示了 Spider 和 BIRD 数据集上的结果。此实验选择朴素的少样本提示方法（Few-shot）作为对照方法，使用 GPT-4o 作为生成模型。相同检索策略和数量的情况下，Few-shot 与 Hint-SQL 所用示例一致。图 12 结果显示，在 Spider 数据集上，Hint-SQL 的最终结果的波动范围在 1.07%之内；图 13 结果显示，在 BIRD 数据集上，Hint-SQL 最终结果的波动范围在 1.63%之内。而 Few-shot 方法的波动范围则分别是 6%和 3.45%。实验结果表明，Hint-SQL 对示例的质量并不敏感，即便面对质量较差的示例，Hint-SQL 也能有稳定的表现。这种稳定性进一步证明，Hint-SQL 中的定制化线索扮演了“稳定器”和“校准器”的角色。当示例质量下降、提供错误或无关的引导信息时，线索所构建的强大先验知识能够帮助模型抵御干扰，使其始终聚焦于正确的解题路径上。这确保了方法在面对不可预知的、嘈杂的真实世界输入时，依然能保持高水平的性能。

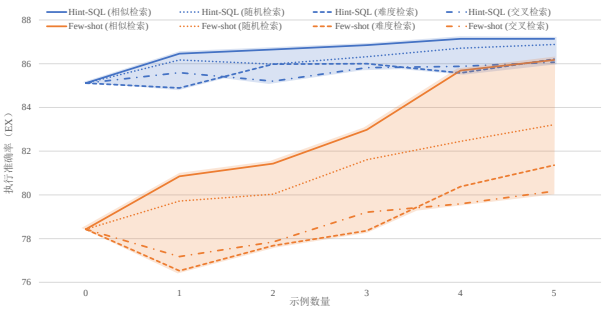

图 12 示例质量对结果的影响（Spider 数据集）

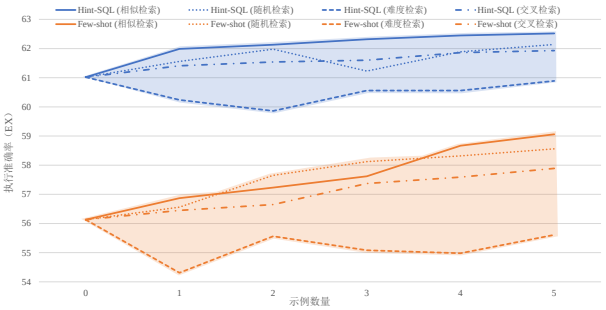

图 13 示例质量对结果的影响（BIRD 数据集）

示例质量代表的是传递信息的有效性，高质量的示例提供有价值的信息，而低质量的示例则包含误导性的信息。例如对于一个简单任务，提供高难度的示例，LLMs 可能会模仿示例中的复杂操作来处理当前任务，导致将简单问题复杂化，从而引发错误。

值得注意的是，检索高质量的示例需要合适的示例检索库，然而在实际场景中，这一条件往往难以满足。为此，本文设置“交叉检索”模拟这一情况，以评估当示例检索库不匹配时对结果的影响。如图 12 所示，在 Spider 数据集上，若使用 BIRD 数据集作为示例检索库，Few-shot 的表现显著下降，与难度检索的效果相当。这是因为 BIRD 数据集的整体难度比 Spider 数据集更高，此时交叉检索与难度检索具有相似的效果。相反，图 13 中使用 Spider 作为 BIRD 的示例检索库，并没有造成显著的表现下降，因为 Spider 任务相对简单，此时的交叉检索与随机检索的效果相似。

本文提出的 Hint-SQL 对示例质量表现出较强的鲁棒性。因为 Hint-SQL 并不完全依赖示例传递信息，当示例质量较低时，定制化线索将作为一种“保险机制”，为 LLMs 提供可靠的备选信息来源，从而保障 Text-to-SQL 任务的稳定执行。

综上分析，少样本提示通过示例激活 LLMs 的上下文学习能力，是一种有效的 SQL 生成策略。然而，少样本提示的效果受到示例数量和质量的影响。相比之下，Hint-SQL 通过线索激活 LLMs 的指令跟随能力，从而增强 LLMs 的 Text-to-SQL 能力。这两种 SQL 生成策略具有互补性：少样本场景下的 Hint-SQL 通过结合示例与线索完成 Text-to-SQL 任务，不仅降低了对示例数量和质量的依赖，并且进一步提升了性能。

5.3 经济与时间成本分析

LLMs 的使用成本通常与其处理的词元(token)数量成正比。表 7 展示了 Hint-SQL 与其他方法在词元消耗上的对比。对比方法中，C3-SQL、DIN-SQL 和 MAC-SQL 涉及 Text-to-SQL 前处理与后处理步骤，需要多次引导 LLMs，因此其词元消耗较高；DAIL-SQL 则是通过检索相似性示例直接引导 LLMs 生成 SQL，词元消耗相对较少。从表中可以看出，Hint-SQL 以较少的词元消耗实现了极具竞争力的性能，展现出了良好的经济性。在不使用示例的情况下，Hint-SQL 在 BIRD 数据集上的表现甚至超越了 MAC-SQL，而其词元消耗仅为 MAC-SQL 的 14%；在使用 5 个示例的情况下，Hint-SQL 的表现进一步提升，但词元消耗依然显著低于其他表现相近的方法。

表 7 Hint-SQL 经济成本统计（使用 GPT-4o 模型）。

方法	示例数量	词元数量/任务量		EX	
		Spider	BIRD	Spider	BIRD
C3-SQL	0	2604	3457	82.98	56.84
DIN-SQL	11	9571	16308	84.91	57.89
DAIL-SQL	5	1936	3785	86.17	59.06
MAC-SQL	2	4372	5894	86.37	60.37
Hint-SQL	0	**422**	**813**	85.11	61.02
	5	2439	4390	**87.14**	**62.52**

Hint-SQL 在降低 LLMs 的词元开销的同时，其推理延迟也保持在较低水平。表 8 展示了 HAgent 在 BIRD 数据集上的时间成本，不同大小的 HAgent 生成线索的时间消耗均较小，证明了 Hint-SQL 的额外推理延迟可控，不会显著影响整体效率。

表 8 Hint-SQL 时间成本分析

模型	总时长(s)/任务数
HAgent(14B)	1.35
HAgent(8B)	0.93
HAgent(7B)	0.79

综合来看，Hint-SQL 在性能与成本之间取得了平衡。它在提升准确率这一核心指标的同时，有效控制了计算资源的消耗。这种特性，使其不仅在学

术研究上具有先进性，更在面向实际应用的工业界部署中，展现出巨大的潜力和可行性。

5.4 消融实验

本小节从两个方面开展消融实验：一是分析不同微调阶段的有效性，二是评估三类线索各自对最终结果的贡献。

表 9 展示了不同微调方法的实验结果。通过偏好学习（第二阶段微调），Hint-SQL 的性能在 Spider 和 BIRD 数据集上的执行准确率分别提升了 1.26% 和 1.57%。

表 9 两阶段微调消融实验（少样本场景）。

方法	Spider		BIRD
	EX	TS	EX
监督微调（第一阶段）			
Hint-SQL + GPT-4o	85.88	76.89	60.95
偏好学习（第二阶段）			
Hint-SQL + GPT-4o	87.14	79.50	62.52

表 10 展示了语义、操作与结构三类线索对 SQL 生成的影响。实验表明，不同线索的重要性因任务特性而异：在 Spider 数据集中，结构线索的作用最为显著，因为它提供的 SQL 骨架与最终查询的结构高度吻合，表现出显著作用；而在 BIRD 数据集中，操作线索通过明确 SQL 执行步骤，对涉及领域知识推理和复杂查询逻辑的任务展现出更大作用。

表 10 三类线索消融实验（少样本场景）。

方法	Spider		BIRD
	EX	TS	EX
所有线索 + GPT-4o	87.14	79.50	62.52
w/o 语义线索	86.56	78.92	61.80
w/o 操作线索	85.78	77.47	58.54
w/o 结构线索	82.79	73.89	60.37

5.5 两阶段微调可迁移性分析

为验证 HAgent 两阶段微调框架的有效性和可迁移性，本文在 Qwen2.5-7B 和 Llama3.1-8B 模型上分别实施了相同的微调流程，分别得到 Hint-SQL^{Q} 和 Hint-SQL^{L} 方法。表 11 和表 12 分别展示了它们在两阶段微调过程中的表现。Hint-SQL^{Q} 在经过偏好学习后，在 Spider 和 BIRD 数据集上的执行准确率分别提升了 1.36%和 1.17%；Hint-SQL^{L} 则是 1.16%和 1.5%。该实验结果证明了我们提出的两阶段微调框架能够有效迁移至不同的基础模型，具备良好的通用性。

表 11 基于 Qwen2.5-7B 的两阶段微调结果。

方法	Spider		BIRD
	EX	TS	EX
监督微调（第一阶段）			
Hint-SQL^{Q} + GPT-4o	82.01	73.63	58.67
偏好学习（第二阶段）			
Hint-SQL^{Q} + GPT-4o	83.37	75.21	59.84

表 12 基于 Llama3.1-8B 的两阶段微调结果。

方法	Spider		BIRD
	EX	TS	EX
监督微调（第一阶段）			
Hint-SQL^{L} + GPT-4o	83.95	74.47	59.58
偏好学习（第二阶段）			
Hint-SQL^{L} + GPT-4o	85.11	77.76	61.08

5.6 跨域线索生成能力分析

在 Text-to-SQL 任务中，不同领域的数据库在模式结构、语义上存在差异。当模型面对一个未见过的领域的数据库时，其性能通常会下降。然而，在实际应用中，跨域场景非常常见，因此 Text-to-SQL 方法的跨域能力显得尤为重要。

为了验证 Hint-SQL 的跨域泛化能力，本小节设置了两个实验：（1）仅使用 Spider 数据集训练 HAgent；（2）仅使用 BIRD 数据集训练 HAgent。随后分别评估它们在未经训练的数据集上的表现。表 13 展示了训练数据的线索分布情况。

表 13 训练数据的线索分布。

数据集	语义线索	操作线索	结构线索
监督微调（第一阶段）			
Spider	7000	7000	7000
BIRD	9428	9428	9428
总数	16428	16428	16428
偏好学习（第二阶段）			
Spider	428	1,321	2,828
BIRD	393	2,533	3,625
总数	821	3854	6453

表 14 展示了 Hint-SQL 在跨域场景下的实验结果。实验结果表明，当模型在 Spider 数据集上训练并在 BIRD 数据集上测试时，其表现为 58.80%，仅比域内表现下降 2.22 个百分点；反之，在 BIRD 上训练并在 Spider 上测试时，性能为 83.56%，下降了 1.55 个百分点。实验结果表明 Hint-SQL 具有良好的跨域泛化能力。我们认为这种优势主要源于以下两个因素：首先，Hint-SQL 生成的线索更侧重于查询逻辑的表示，而非特定数据库的细节特征，这

降低了模型对特定领域数据库模式结构的依赖；其次，Hint-SQL 使用经过大规模预训练的 LLMs 来生成 SQL，有效利用了 LLMs 中蕴含的丰富领域知识，从而显著提升了跨域场景下的表现。

表 14 Hint-SQL 跨域结果（0-shot 场景，GPT-4o 模型）。

训练集	测试集	EX
Spider + BIRD	Spider	85.11
	BIRD	61.02
Spider	Spider	84.62
	BIRD	58.80
BIRD	Spider	83.56
	BIRD	60.04

这证明了 Hint-SQL 学习到的并非是针对特定领域数据库的刻板模式知识，而是更高层次、更抽象的 Text-to-SQL 逻辑推理能力。

5.7 样例展示

表 15 展示 Hint-SQL 在处理 Text-to-SQL 任务时产生的真实数据样例，完整呈现了语义线索、操作线索、结构线索和最终 SQL。可以看出语义线索、操作线索与结构线索的显著区别：语义线索重述问题，明确查询目标；操作线索规划 SQL 执行步骤，实现查询目标；结构线索将操作步骤映射到 SQL 骨架。三类线索协同作用，共同引导 LLMs 处理 Text-to-SQL 任务。尤其在复杂逻辑任务中（如表 15 的第 3 个样例，涉及三层嵌套查询），操作线索能够精准地对查询逻辑进行建模，从而显著提升了模型处理复杂任务的能力。

表 15 中还展示了监督微调和偏好学习阶段的线索差异。在第一个样例中，偏好学习阶段的语义线索补充了“FRPM”的详细解释，操作线索和结构线索补充了排序的逻辑；在第二个样例中，偏好学习阶段的语义线索补充了学生的年龄范围，操作线索将过滤条件从“student age”修正为“free meal count”，结构线索补全了 SQL 骨架。在第三个样例中，偏好学习补充了操作线索的过滤条件，并修正结构线索的关键词错误。这些样例强调了偏好学习对线索细粒度信息的优化作用。通过偏好学习，模型学会了识别并补全那些在监督微调阶段容易被忽略的关键细节，例如实体别名的澄清、排序逻辑的补充、边界条件的精确化等。这种迭代优化过程，是 HAgent 能够生成高质量、高精度线索的关键。

表 15 样例展示。

	监督微调阶段	偏好学习阶段
用户问题	What is the unabbreviated mailing street address of the school with the highest FRPM count for K-12 students?	
语义线索	The question is looking for the full mailing street address of the school with the largest number of students eligible for FRPM benefits.	The question is looking for the full mailing street address of the school that has the highest count of students eligible for FRPM (Free and Reduced-Price Meal) benefits for K-12 students.
操作线索	This requires joining two tables (frpm and schools), sorting the results by the FRPM count, and selecting the top result.	This requires joining two tables (frpm and schools), sorting the results by the FRPM count in descending order, and selecting the top result.
SQL 骨架	**SELECT** _ **FROM** _ **JOIN** _ **ON** _ = _ **ORDER BY** _ **LIMIT** _	**SELECT** _ **FROM** _ **JOIN** _ **ON** _ = _ **ORDER BY** _ **DESC LIMIT** _
SQL	**SELECT** T2.MailStreet **FROM** frpm AS T1 **JOIN** schools AS T2 **ON** T1.CDSCode = T2.CDSCode **ORDER BY** T1.`FRPM Count (K-12)` **DESC LIMIT** 1	
用户问题	If there are any, what are the websites address of the schools with a free meal count of 1,900-2,000 to students aged 5-17? Include the name of the school.	
语义线索	The question is looking for the websites and names of schools where the count of students receiving specific benefits within a certain age range falls between 1,900 and 2,000.	The question is looking for the websites and names of schools that have a free meal count for students aged 5-17 within the range of 1,900 to 2,000.
操作线索	This requires joining two tables: one containing school information and the other containing free meal counts, and then filtering the results based on the student age.	This requires joining two tables: one containing school information and the other containing free meal counts, and then filtering the results based on the free meal count.
SQL 骨架	**SELECT** _, _ **FROM** _ **JOIN** _ **ON** _ = _ **WHERE** _ **BETWEEN** _ **AND** _	**SELECT** _, _ **FROM** _ **JOIN** _ **ON** _ = _ **WHERE** _ **BETWEEN** _ **AND** _ **AND** _

SQL	**SELECT** T2.Website, T1.`School Name` **FROM** frpm AS T1 **JOIN** schools AS T2 **ON** T1.CDSCode = T2.CDSCode **WHERE** T1.`Free Meal Count (Ages 5-17)` **BETWEEN** 1900 **AND** 2000 **AND** T2.Website IS NOT NULL	
用户问题	In which Formula_1 race did Lewis Hamilton rank the highest?	
语义线索	The question is looking for the name(s) of the Formula 1 race(s) where Lewis Hamilton achieved his highest ranking.	The question is looking for the name(s) of the Formula 1 race(s) where Lewis Hamilton achieved his highest ranking.
操作线索	This requires first identifying the driver's unique identifier by filtering for their full name, then retrieving all race records associated with that driver, and finally fetching the corresponding race names based on the race IDs.	This requires first identifying the driver's unique identifier by filtering for their full name, then retrieving all race records where that driver achieved the highest ranking, and finally fetching the corresponding race names based on the race IDs.
结构线索	**SELECT** _ **FROM** _ **WHERE** _ **=** (**SELECT** _ **FROM** _ **WHERE** _ = (**SELECT** _ **FROM** _ **WHERE** _ = _ **AND** _ = _))	**SELECT** _ **FROM** _ **WHERE** _ **IN** (**SELECT** _ **FROM** _ **WHERE** _ = _ **AND** _ = (**SELECT** _ **FROM** _ **WHERE** _ = _ **AND** _ = _))
SQL	**SELECT** name **FROM** races **WHERE** raceId **IN** (**SELECT** raceId **FROM** results **WHERE** rank = 1 **AND** driverId = (**SELECT** driverId **FROM** drivers **WHERE** forename = 'Lewis' **AND** surname = 'Hamilton'))	

6 结论

本文提出一种基于自动线索生成的 Text-to-SQL 提示方法 Hint-SQL，通过 HAgent 为每个 Text-to-SQL 任务生成定制化线索，向 LLMs 提供具体的 SQL 生成建议，从而提升其在 Text-to-SQL 任务上的性能。实验结果表明，Hint-SQL 作为一种新颖的 SQL 生成策略，相较传统的上下文学习策略（DAIL-SQL），在 BIRD 数据集上的执行准确率高出 1.96%；此外，Hint-SQL 与上下文学习策略能够形成优势互补：二者结合不仅能进一步提升性能，还显著降低了传统少样本方法对示例数量和质量的敏感度。Hint-SQL 的一个核心优势在于其作为一种即插即用的 SQL 生成策略，能够无缝集成并增强现有的 Text-to-SQL 提示方法。通过与 Hint-SQL 的结合，6 种主流方法的性能均得到显著提升，在最具挑战性的 BIRD 数据集上的执行准确率平均提高了 4.63%。其中包括当前性能领先的 RSL-SQL 方法，通过与 Hint-SQL 结合，其在 BIRD 数据集上的执行准确率再度提升了 4.37%。

本文首次对 Text-to-SQL 线索进行了系统性研究，揭示了线索在 Text-to-SQL 任务中的重要作用，为该领域的后续研究提供了参考。未来工作将进一步探索线索的其他用法，例如通过线索来提示 LLMs 融入恰当的领域知识等。

参 考 文 献

[1] Liu A, Hu X, Wen L, et al. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability. arXiv preprint arXiv:2303.13547, 2023.

[2] Li J, Hui B, Qu G, et al. Can llm already serve as a dtabase interface? a big bench for large-scale database grounded Text-to-SQLs//Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA, 2023: 183-210.

[3] Nan L, Zhao Y, Zou W, et al. Enhancing text-to-sql capabilities of large language models: a study on prompt design strategies//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore, 2023: 14935-14956.

[4] Li B, Luo Y, Chai C, et al. The dawn of natural language to sql: are we fully ready//Proceedings of the 50th International Conference on Very Large Databases. Guangzhou, China, 2024: 3318-3331.

[5] Dong X, Zhang C, Ge Y, et al. C3: zero-shot text-to-sql with chatgpt. arXiv preprint arXiv: 2307.07306.

[6] Xie Y, Jin X, Xie T, et al. Decomposition for enhancing attention: improving llm-based text-to-sql through workflow paradigm//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand, 2024: 10796-10816.

[7] Yang J, Hui B, Yang M, et al. Synthesizing text-to-sql data from weak and strong llms//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand, 2024: 7864-7875.

[8] Yu T, Zhang R, Yang K, et al. Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, 2018: 3911-3921.

[9] Zhang H, Cao R, Chen L, et al. Act-sql: in-context learning for text-to-sql with automatically-generated chain-of-thought//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore, 2023: 3501-3532.

[10] Li B, Zhang J, Fan J, et al. Alpha-sql: zero-shot text-to-sql using monte carlo tree search. arXiv preprint arXiv:2502.17248, 2025.

[11] Pourreza M, Rafiei D. Din-sql: decomposed in-context learning of text-to-sql with self-correction//Proceedings of the 37th Annual Conference on Neural Information Processing Systems. New Orleans, USA, 2023: 36339-36348.

[12] Gao Y, Liu Y, Li X, et al. XiYan-SQL: A multi-generator ensemble framework for text-to-sql. arXiv preprint arXiv: 2411.08599.

[13] Wang B, Ren C, Yang J, et al. Mac-sql: a multi-agent collaborative framework for text-to-sql//Proceedings of the 13th International Conference on Computational Linguistics. Singapore, 2025: 540-557.

[14] Gao D, Wang H, Li Y, et al. Text-to-sql empowered by large language models: a benchmark evaluation//Proceedings of the 50th International Conference on Very Large Databases, Guangzhou, China, 2024: 1132-1145.

[15] Chen K, Chen Y, Koudas N, et al. Reliable text-to-sql with adaptive abstention//Proceedings of the ACM on Management of Data, 2025, 3(1): 69:1-69:30.

[16] Liu X, Shen S, Li B, et al. A survey of nl2sql with large language models: where are we, and where are we going. arXiv preprint arXiv: 2408.05109, 2024.

[17] Maamari K, Abubaker F, Jaroslawicz D, et al. The death of schema linking? text-to-sql in the age of well-reasoned language models. arXiv preprint arXiv:2408.07702, 2024.

[18] Cao Z, Zheng Y, Fan Z, et al. Rsl-sql: robust schema linking in text-to-sql generation. arXiv e-prints, 2024: arXiv: 2411.00073.

[19] Chai L, Xiao D, Yan Z, et al. Qurg: question rewriting guided context-dependent text-to-sql semantic parsing//Proceedings of the 20th Pacific Rim International Conference on Artificial Intelligence. Singapore, 2023: 275-286.

[20] Mao W, Wang R, Guo J, et al. Enhancing text-to-sql parsing through question rewriting and execution-guided refinement//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand, 2024: 2009-2024.

[21] Talaei S, Pourreza M, Chang Y C, et al. Chess: contextual harnessing for efficient sql synthesis. arXiv preprint arXiv:2405.16755, 2024.

[22] Chen X, Lin M, Schaerli N, et al. Teaching large language models to self-debug//Proceedings of the Association for Computational Linguistics ACL 2023. Toronto, Canada, 2023: 456-534.

[23] Askari A, Pöditz C, Tang X. Magic: generating self-correction guideline for in-context text-to-sql//Proceedings of The 39th Annual AAAI Conference on Artificial Intelligence. Philadelphia, USA, 2025: 23433-23441.

[24] Lou R, Zhang K, Yin W. Large language model instruction following: a survey of progresses and challenges. Computational Linguistics, 2024, 50(3): 1053-1095.

[25] Tan Z, Liu X, Shu Q, et al. Enhancing text-to-sql capabilities of large language models through tailored promptings//Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia, 2024: 6091-6109.

[26] Dong Q, Li L, Dai D, et al. A survey on in-context learning//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore, 2024: 1107-1128.

[27] Chang S, Fosler L E. Selective demonstrations for cross-domain text-to-sql//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore, 2023: 14174-14189.

[28] Chang S, Fosler-Lussier E. How to prompt llms for text-to-sql: a study in zero-shot, single-domain, and cross-domain settings. arXiv preprint arXiv:2305.11853, 2023.

[29] Fan Y, He Z, Ren T, et al. Metasql: a generate-then-rank framework for natural language to sql translation//Proceedings of the 40th International Conference on Data Engineering. Utrecht, The Netherlands, 2024: 1765-1778.

[30] Fan J, Gu Z, Zhang S, et al. Combining small language models and large language models for zero-shot nl2sql//Proceedings of the 50th International Conference on Very Large Databases. Guangzhou, China, 2024, 17(11): 2750-2763.

[31] Qu G, Li J, Li B, et al. Before generation, align it! a novel and effective strategy for mitigating hallucinations in text-to-sql generation//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand, 2024: 5456-5471.

[32] Gu Z, Fan J, Tang N, et al. Few-shot text-to-sql translation using structure and content prompt learning. Proceedings of the ACM on Management of Data, 2023, 1(2): 1-28.

[33] Li H, Zhang J, Li C, et al. Resdsql: decoupling schema linking and skeleton parsing for text-to-sql//Proceedings of the 37th AAAI Conference on Artificial Intelligence. Washington, USA, 2023: 13067-13075.

[34] Zhang C, Mao Y, Fan Y, et al. Finsql: model-agnostic llms-based text-to-sql framework for financial analysis//Proceedings of the 2024 International Conference on Management of Data. Lima, Peru, 2024: 93-105.

[35] Wang D, Dou L, Zhang X, et al. Dac: decomposed automation correction for text-to-sql. arXiv preprint arXiv:2408.08779, 2024.

[36] Guo C, Tian Z, Tang J, et al. Prompting gpt-3.5 for text-to-sql with de-semanticization and skeleton retrieval//Proceedings of the 20th Pacific Rim International Conference on Artificial Intelligence. Singapore, 2023: 262-274.

[37] Zhang T, Chen C, Liao C, et al. Sqlfuse: enhancing text-to-sql performance through comprehensive llm synergy. arXiv preprint arXiv:2407.14568, 2024.

[38] Xie W, Wu G, Zhou B. Mag-sql: multi-agent generative approach with soft schema linking and iterative sub-sql refinement for text-to-sql. arXiv preprint arXiv:2408.07930, 2024.

[39] Li J, Wu T, Mao Y, et al. Sql-factory: a multi-agent framework for high-quality and large-scale sql generation. arXiv preprint arXiv:2504.14837, 2025.

[40] Xue S, Jiang C, Shi W, et al. Db-gpt: empowering database interactions with private large language models. arXiv preprint arXiv:2312.17449, 2023.

[41] Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: your language model is secretly a reward model\\Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA, 2023: 53728-53741.

[42] Gan Y, Chen X, Huang Q, et al. Towards robustness of text-to-sql models against synonym substitution//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand, 2021: 2505-2515.

[43] Deng X, Hassan A, Meek C, et al. Structure-grounded pretraining for text-to-sql//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics. Mexico City, Mexico, 2021: 1337-1350.

[44] Gan Y, Chen X, Purver M. Exploring underexplored limitations of cross-domain text-to-sql generalization//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana, Dominican Republic, 2021: 8926-8931.

[45] Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[46] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 2022, 35: 24824-24837.

[47] Liu A, Feng B, Wang B, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434, 2024.

[48] Grattafiori A, Dubey A, Jauhri A, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

[49] Bai J, Bai S, Chu Y, et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

[50] Singh S, Lodwal H, Malwat H, et al. Unlocking model insights: a dataset for automated model card generation. arXiv preprint arXiv:2309.12616, 2023.

[51] Team G, Georgiev P, Lei V I, et al. Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530, 2024.

[52] Liu A, Feng B, Xue B, et al. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437, 2024.

附录A. 提示词样例

本附录展示了本文所使用的提示词样例，包括语义线索的合成提示词、操作线索的合成提示词、线索生成提示词以及SQL生成提示词。下面依次介绍各提示词的结构与内容。

A.1 语义线索的合成提示词

语义线索的合成提示词用于引导LLMs分析用户问题的查询意图，旨在消除用户问题中的歧义，并精确地阐明查询意图。图 14 展示了该提示词的结构，包括任务描述、若干示例、当前任务的数据库、用户问题、真实SQL以及模式元素。任务描述用于向LLMs说明合成语义线索的要求，示例则是“输入-输出”参考，任务数据库传达了所有数据集模式、用户问题则是用户查询需求，真实SQL是数据集中标注的答案，模式元素则是提取了SQL中SELECT子句中的数据库模式词。该提示词引导LLMs合成语义线索数据，用于支持HAgent的监督微调。

合成 语义线索 的提示词

任务描述

Your task is to analyze the semantic hint of a Text-to-SQL task. You will be provided with a database schema, a question, SQL query, and intent-related schema elements. Referring to intent-related schema elements, rephrase the question to clarify the query objective.

示例

Database Schema:
CREATE TABLE "department" (
"Department_ID" int;
(... 其他的表和列...)
Question: List the name, born state and age of the heads of departments ordered by age.
SQL: SELECT name, born_state, age FROM head ORDER BY age
Intent-related schema elements：name, born_state, age .
Semantic hint: The question is asking for a list of the names, birth states, and ages of the heads of departments, sorted by their age in ascending order.
(...其他示例...)

数据库

Database Schema:
CREATE TABLE "stadium" (
"Stadium_ID" int,
(… 其他的表和列…)

用户问题

Question: Show the name and the release year of the song by the youngest singer.

SQL

SQL: SELECT song_name, song_release_year FROM singer ORDER BY age LIMIT 1

模式元素

Intent-related schema elements： song_name, song_release_year

图 14　合成语义线索的提示词样例

A.2 操作线索的合成提示词

操作线索的合成提示词用于引导LLMs使用SQL特有的思维，规划操作步骤以实现查询目标。图 15 展示了该提示

词的样例，包括任务描述、若干示例、当前任务的数据库、用户问题、真实SQL以及语义线索。其中，语义线索是前置条件，由LLMs预先合成而来。该提示词引导LLMs合成操作线索数据，用于支持HAgent的监督微调。

合成 操作线索 的提示词

任务描述 Your task is to analyze the operational hint of a Text-to-SQL task question. You will be provided with a database schema, a question, SQL query, and the semantic hint. Using SQL-specific thinking to plan a series of operational steps to achieve the query objective in t the semantic hint.

示例
Database Schema:
CREATE TABLE "frpm" (
"CDSCode" text;
(... 其他的表和列...)
Question: Please list the lowest three eligible free rates for students aged 5-17 in continuation schools.
SQL: SELECT `Free Meal Count (Ages 5-17)` / `Enrollment (Ages 5-17)` FROM frpm WHERE `Educational Option Type` = 'Continuation School' AND `Free Meal Count (Ages 5-17)` / `Enrollment (Ages 5-17)` IS NOT NULL ORDER BY `Free Meal Count (Ages 5-17)` / `Enrollment (Ages 5-17)` ASC LIMIT 3,
Semantic hint: The question is asking for the lowest three free meal eligibility rates for students aged 5-17 in continuation schools.
Operational hint: This requires focusing on continuation schools as specified by the Educational Option Type, calculating the free meal eligibility rate using the formula 'Free Meal Count (Ages 5-17)/Enrollment (Ages 5-17)', sorting the results from lowest to highest, and selecting the first three entries.
(...其他示例...)

数据库
Database Schema:
CREATE TABLE "account" (
"account_id" text;
(… 其他的表和列…)

用户问题 Question: How many accounts who choose issuance after transaction are staying in East Bohemia region?

SQL SQL: SELECT COUNT(T2.account_id) FROM district AS T1 JOIN account AS T2 ON T1.district_id = T2.district_id WHERE T1.A3 = 'east Bohemia' AND T2.frequency = 'POPLATEK PO OBRATU',

语义线索 Semantic hint: The question is looking to count the number of accounts that have chosen issuance after a transaction and are located in the East Bohemia region.

图 15　合成操作线索的提示词样例

A.3 线索生成提示词

图 16 依次展示了生成语义、操作和结构线索的提示词样例。这些提示词的共同部分包括数据库模式与用户问题，而各自的任务指令（I^a, I^b, I^c）则有所不同。操作线索的提示词中包含前置生成的语义线索，结构线索的提示词包含前置生成的语义线索以及操作线索。本文使用这些提示词递进式地引导HAgent自动为每个Text-to-SQL任务生成定制化的语义、操作和结构线索。这些线索将会被用于辅助LLMs生成最终的SQL语句。

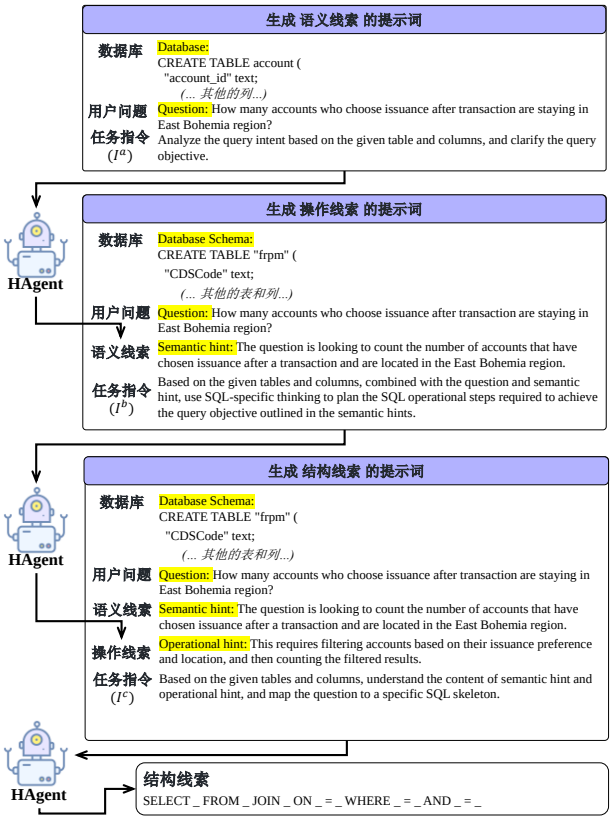

图 16　线索生成的提示词样例

A.4 SQL生成提示词

图17和图18分别展示了零样本场景和少样本场景下的SQL生成提示词样例。该提示词旨在使用语义、操作和结构线索引导LLMs生成SQL。其主要内容包括数据库、用户问题、语义线索、操作线索、结构线索和任务指令。本文将领域知识作为数据库内容的补充信息。

SQL生成 的提示词（零样本场景）

数据库
Database:
CREATE TABLE account (
"account_id" text;
数据库模式 *(... 其他的列...)*
/* 3 example rows:
SELECT * FROM employee LIMIT 3;
account_id | district_id | frequency | date
数据库内容 1 | 18 | POPLATEK MESICNE | 1995-03-24
2 | 1 | POPLATEK MESICNE | 1993-02-26
3 | 5 | POPLATEK MESICNE | 1997-07-07
*/
(... 其他的表...)
领域知识 -- External Knowledge: 'POPLATEK PO OBRATU' represents for 'issuance after transaction'.

用户问题 Question: How many accounts who choose issuance after transaction are staying in East Bohemia region?

语义线索 Semantic hint:The question is looking to count the number of accounts that have chosen issuance after a transaction and are located in the East Bohemia region.

操作线索 Operational hint: This requires filtering accounts based on their issuance preference and location, and then counting the filtered results.

结构线索 Sturctural hint :SELECT _ FROM _ JOIN _ ON _ = _ WHERE _ = _ AND _ = _

任务指令 (I) Semantic hint clarify the query's objective, operational hint indicate the logical steps of the query, and structural hint form the SQL skeleton. Based on the provided tables and columns, please write a correct SQLite query to answer the question.

SQL
SELECT COUNT(T2.account_id) FROM district AS T1 JOIN account AS T2 ON T1.district_id = T2.district_id WHERE T1.A3 = 'east Bohemia' AND T2.frequency = 'POPLATEK PO OBRATU'

图 17　零样本场景下的SQL生成提示词样例

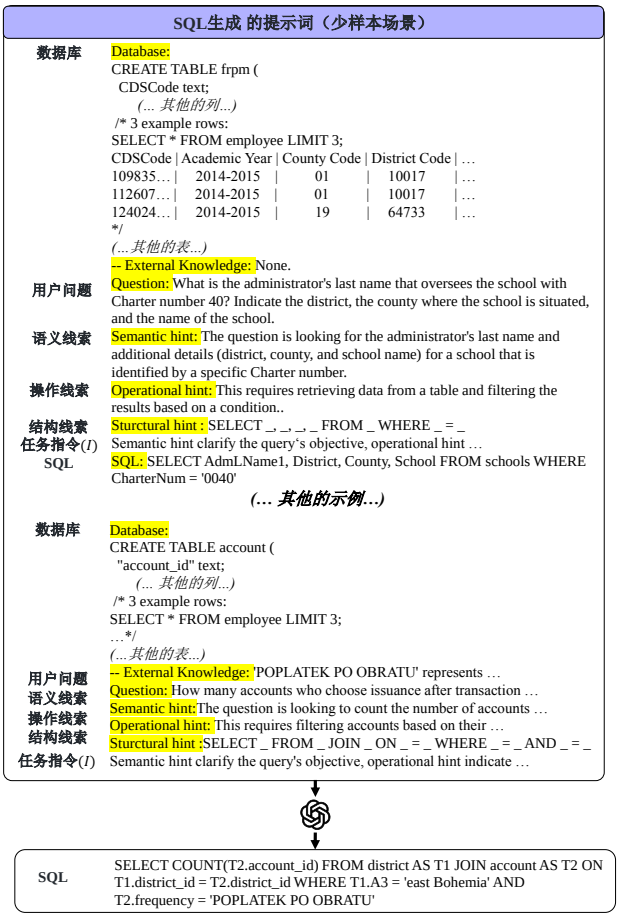

图 18　少样本场景下的SQL生成的样例

附录B. 偏好数据集构建算法

算法 1 展示了偏好数据集的构建过程。首先使用微调监督后的线索生成智能体HAgent-SFT生成三类线索，然后对线索有效性进行验证，得到对应的正面响应输出与负面响应输出，从而构建偏好数据集。

算法 1. 偏好数据集构建过程

输入：监督微调数据集$\mathcal{D}_{sft}$；监督微调后的线索生成智能体 *HAgent-SFT*；大语言模型 *LLMs*

输出：偏好数据集$\mathcal{D}_{pl}=\{\mathcal{D}_{pl}^{a},\mathcal{D}_{pl}^{b},\mathcal{D}_{pl}^{c}\}$

1. 初始化偏好数据集$\mathcal{D}_{pl}^{a}=\emptyset$，$\mathcal{D}_{pl}^{k}=\emptyset$，$\mathcal{D}_{pl}^{c}=\emptyset$;
2. **FOR** $\left(S_i,Q_i,A_i^{\text{syn}},B_i^{syn},C_i^{syn},I^a,I^b,I^c\right)\in\mathcal{D}_{sft}^{n}$ **DO**
3. 生成线索：

$$A_i^{sft}=HAgent-SFT(S_i,Q_i,I^a)$$
$$B_i^{sft}=HAgent-SFT\left(S_i,Q_i,A_i^{\text{syn}},I^b\right)$$
$$C_i^{sft}=HAgent-SFT\left(S_i,Q_i,A_i^{\text{syn}},B_i^{syn},I^c\right)$$

4. 验证语义线索，使用A_i^{sft}和A_i^{syn}引导 *LLMs* 完成 Text-to-SQL 任务：

$$SQL_i^{A-sft}=LLMs\left(S_i,Q_i,A_i^{sft}\right)$$
$$SQL_i^{A-syn}=LLMs\left(S_i,Q_i,A_i^{syn}\right)$$

5. 判断结果：

IF SQL_i^{A-sft}正确 **AND** SQL_i^{A-syn}错误 **THEN**

$$A_i=SQL_i^{A-sft},A_i^r=SQL_i^{A-syn}$$

IF SQL_i^{A-sft}错误 **AND** SQL_i^{A-syn}正确 **THEN**

$$A_i=SQL_i^{A-syn},A_i^r=SQL_i^{A-sft}$$

6. 更新数据集:

$$\mathcal{D}_{pl}^{a}=\mathcal{D}_{pl}^{a}\cup\{(S_i,Q_i,I^a,A_i,A_i^r)\}$$

7. 验证操作线索，使用B_i^{sft}和B_i^{syn}引导 *LLMs* 完成 Text-to-SQL：

$$SQL_i^{B-sft}=LLMs\left(S_i,Q_i,B_i^{sft}\right)$$
$$SQL_i^{B-syn}=LLMs\left(S_i,Q_i,B_i^{syn}\right)$$

8. 判断结果：

IF SQL_i^{B-sft}正确 **AND** SQL_i^{B-syn}错误 **THEN**

$$B_i=SQL_i^{B-sft},B_i^r=SQL_i^{B-syn}$$

IF SQL_i^{B-sft}错误 **AND** SQL_i^{B-syn}正确 **THEN**

$$B_i=SQL_i^{B-syn},B_i^r=SQL_i^{B-sft}$$

9. 更新数据集:

$$\mathcal{D}_{pl}^{b}=\mathcal{D}_{pl}^{b}\cup\{(S_i,Q_i,I^b,B_i,B_i^r)\}$$

10. 验证结构线索，**IF** $C_i^{sft}\neq C_i^{syn}$ **THEN**

$$C_i=C_i^{syn},C_i^r=C_i^{sft}$$

11. 验证是否为误判：

$$SQL_i^{C-sft}=LLMs(S_i,Q_i,C_i^r)$$

IF $\text{SQL}_\text{i}^{\text{C-sft}}$执行正确 **AND** $\text{SQL}_\text{i}^{\text{C-sft}}$的骨架与$\text{C}_\text{i}^{r}$一致 **THEN**

CONTINUE

ELSE

12. 更新数据集

$$\mathcal{D}_{pl}^{c}=\mathcal{D}_{pl}^{c}\cup\{(S_i,Q_i,I^c,C_i,C_i^r)\}$$

13. **END FOR**
14. 返回$\mathcal{D}_{pl}=\{\mathcal{D}_{pl}^{a},\mathcal{D}_{pl}^{b},\mathcal{D}_{pl}^{c}\}$。

TAN Zhao, Ph.D. candidate. His main research areas are machine learning and natuxxral language processing.

LIU Xi-Ping, Ph.D., professor. His main research areas are machine learning and natural language processing.

SHU Qing, Ph.D. candidate. Her main research areas are machine learning and natural language processing.

WAN Qi-Zhi, Ph.D., lecturer. His main research areas are natural language processing, information extraction, and deep learning.

LIU De-Xi, Ph.D., professor. His main research areas are social media processing, information retrieval, and natural language processing.

WAN Chang-Xuan, Ph.D., professor. His main research areas are sentiment analysis and data mining.

LIAO Guo-Qiong, Ph.D., professor. His main research areas are database and blockchain technology.

Background

Text-to-SQL is a cross-disciplinary problem at the intersection of natural language processing and databases. Its goal is to convert natural language queries into executable SQL statements, enabling non-expert users to access data with ease. Currently, prompting large language models (LLMs) has become the mainstream approach for tackling Text-to-SQL tasks. A typical prompt includes demonstrations, the task database, user queries, and task instructions. Recently, researchers have started incorporating hints into prompts that provide specific Text-to-SQL suggestions. However, most existing hints are derived from statistical analyses of common errors in Text-to-SQL tasks, rendering them generalized and less effective for diverse and specific queries.

This paper systematically explores the role of hints in Text-to-SQL and introduces Hint-SQL, a novel SQL generation method via task-specific hints. Hint-SQL employs a Hint-generation Agent (HAgent) to produce tailored semantic, operational, and structural hints, providing LLMs with precise and actionable guidance. By integrating Hint-SQL, the effectiveness of six existing Text-to-SQL prompting methods is significantly improved, with execution accuracy on the Spider and BIRD datasets increasing by an average of 2.74% and 4.63%, respectively. Furthermore, Hint-SQL enhances the performance of the state-of-the-art method, RSL-SQL, boosting its execution accuracy on the Spider and BIRD datasets by 1.35% and 4.37%, respectively. This study is the first to systematically investigate the importance of hints in Text-to-SQL, offering valuable insights and laying the foundation for future research in this area.

This work is supported by the General Program of the National Natural Science Foundation of China (No. 62272205, 62272206), the Regional Science Fund Program of the National Natural Science Foundation of China (No. 62462034, 62562033), the Key Program of the Natural Science Foundation of Jiangxi Province (No. 20232ACB202008), the General Program of the Natural Science Foundation of Jiangxi Province (No. 20242BAB25119), and the Jiangxi Provincial Graduate Innovation Special Fund (No. YC2023-B185).