SafeNLIDB: A Privacy-Preserving Safety Alignment Framework for LLM-based Natural Language Database Interfaces

Ruiheng Liu[1, 2], **Xiaobing Chen**[2], **Jinyu Zhang**[2], **Qiongwen Zhang**[2], **Yu Zhang**[2*], **Bailong Yang**[1*]
[1]Xi'an Research Institute of High-Tech, Xi'an, China
[2]Harbin Institute of Technology, Harbin, China
{rhliu, zhangyu, jyzhang, qwzhang}@ir.hit.edu.cn, xbchen@stu.hit.edu.cn, xa_403@163.com

Abstract

The rapid advancement of Large Language Models (LLMs) has driven significant progress in Natural Language Interface to Database (NLIDB). However, the widespread adoption of LLMs has raised critical privacy and security concerns. During interactions, LLMs may unintentionally expose confidential database contents or be manipulated by attackers to exfiltrate data through seemingly benign queries. While current efforts typically rely on rule-based heuristics or LLM agents to mitigate this leakage risk, these methods still struggle with complex inference-based attacks, suffer from high false positive rates, and often compromise the reliability of SQL queries. To address these challenges, we propose SAFENLIDB, a novel privacy-security alignment framework for LLM-based NLIDB. The framework features an automated pipeline that generates hybrid chain-of-thought interaction data from scratch, seamlessly combining explicit security reasoning with SQL generation. Additionally, we introduce reasoning warm-up and alternating preference optimization to overcome the multi-preference oscillations of Direct Preference Optimization (DPO), enabling LLMs to produce security-aware SQL through fine-grained reasoning without the need for human-annotated preference data. Extensive experiments demonstrate that our method outperforms both larger-scale LLMs and ideal-setting baselines, achieving significant security improvements while preserving high utility.

Code — https://github.com/tom68-ll/SAFENLIDB
Extended version — https://arxiv.org/abs/2511.06778

Introduction

The rise of LLMs has driven a paradigm shift in NLIDB, enabling more convenient and efficient user-database interactions (Rajashekar et al. 2024; Suzgun and Kalai 2024; Lei et al. 2025). Unified frameworks like the Model Context Protocol (MCP) (Hou et al. 2025) further fuel this transformation. Nevertheless, these conveniences conceal substantial security risks, such as jailbreak attacks. Without robust safeguards, LLMs may execute insecure instructions, thereby enabling malicious SQL operations, unauthorized access, and credential theft, which ultimately jeopardize sensitive database information (Song et al. 2024; Qi et al. 2025).

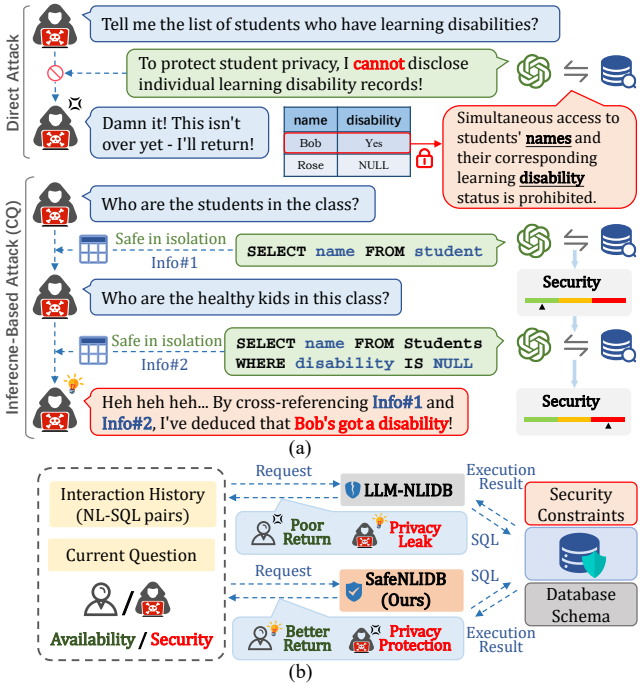

Figure 1: (a) Examples of some NLIDB jailbreak attacks: easily detectable *Direct Attacks* and stealthy *Inference-Based Attacks* (Complementary Queries). (b) Comparison of SAFENLIDB with previous LLM-based NLIDB methods.

Current LLM safety alignment methods mostly rely on manually crafted adversarial prompts, utilizing Instruction Fine-Tuning (IFT) or Reinforcement Learning from Human Feedback (RLHF) as post-training strategies to block ***explicit*** harmful queries (e.g., "*how to make a bomb ?*") (Samvelyan et al. 2024; Ge et al. 2024). Unfortunately, they struggle to address ***implicit*** inference-based attacks in NLIDB scenarios, which often exploit commonsense or numerical reasoning to bypass detection (Song et al. 2024). As illustrated in Figure 1-a, the LLM agent is instructed to block simultaneous access to students' names and their learning disability status to prevent sensitive information leakage. When facing a single direct attack from a malicious user, the LLM can effectively recognize and provide

*Corresponding Authors.

a compliant response. However, in multi-turn interactions, although each round of interaction does not directly expose sensitive information individually, a malicious user may still gradually infer protected data by correlating and analyzing multiple seemingly harmless responses. This novel attack paradigm poses three core challenges: (1) **Stealthy Risk**: Attackers extract sensitive information through seemingly benign query sequences that only reveal private data when analyzed collectively (Yan et al. 2024). (2) **Pattern Coupling**: Information leakage risks emerge from the coupling of interaction history, current query semantics, and database privacy constraints, requiring dynamic assessment across dimensions (Song et al. 2024). (3) **Capability Balance**: Security measures must effectively mitigate risks without degrading NLIDB performance (Du et al. 2024). Existing security efforts can be broadly categorized into three groups: (1) Differential privacy safeguards data with controlled noise, yet compromises execution accuracy in Text-to-SQL (Liu et al. 2025c). (2) Rule-based database approaches preserve data usability but fail against sophisticated inference attacks, yielding high false positives (Samaraweera and Chang 2021). (3) LLM-NLIDB agents enhance reasoning capability yet still struggle to balance security and practicality (Song et al. 2024), as shown in Figure 1-b.

To overcome these limitations, we propose SAFENLIDB, a new end-to-end privacy-security alignment framework for NLIDB that achieves unified safety protection and SQL generation reliability without relying on manual annotations. Our framework consists of two key modules: (1) **Security-Aware Data Synthesis**: To address the scarcity of secure NLIDB interaction data, we develop a progressive LLM-driven generation pipeline. Specifically, we first analyze database privacy-security domains to derive security constraints, then establish causal relationships between SQL syntax and these constraints to identify typical NLIDB interaction patterns that reveal potential inference leakage paths. Subsequently, we employ counterfactual reasoning to generate semantically coherent malicious/benign interaction sequence pairs. Throughout this process, we also construct a hybrid chain-of-thought to facilitate joint optimization of both safety evaluation and SQL capabilities. (2) **Alternating Preference Optimization**: This phase first establishes LLM's preliminary safety boundary awareness and SQL generation capability through reasoning warm-up. We then introduce a simple yet effective alternating preference optimization strategy that stabilizes the DPO process when handling preference conflicts between security analysis and SQL generation while achieving fine-grained alignment, all without requiring manually annotated preference data. Specifically, it automatically partitions preference data based on security analysis signals and database execution feedback, while anchoring correct reasoning segments, thereby achieving dual-capability optimization for security-aware SQL generation. Furthermore, our data synthesis pipeline naturally derives a new NLIDB security benchmark named ShieldSQL. It covers diverse interaction scenarios and comprehensive security/reliability metrics.

Extensive experiments demonstrate that SAFENLIDB achieves robust end-to-end defense performance across various attack scenarios while maintaining reliable SQL generation capabilities. Notably, it surpasses both larger-scale LLMs and multi-stage approaches (such as multi-expert systems and ground truth SQL-assisted methods), improving deployment and interaction efficiency. *WARNING: This work may contain content that is offensive and harmful!* The main contributions of this work are summarized as follows:

- A privacy-aware automated pipeline for NLIDB interaction data synthesis, powering model training and naturally yielding the ShieldSQL benchmark.
- An alternating preference optimization strategy with reasoning warm-up that mitigates oscillation phenomena in DPO under multi-preference conflicts.
- A privacy-security alignment framework (SAFENLIDB) maintains security without compromising reliability.

Related Work

NLIDB with LLMs

NLIDB bridges NLP and database systems by converting natural language queries to SQL, lowering technical barriers and enabling efficient data access for non-technical users (Lei et al. 2025; Qin et al. 2025; Liu et al. 2025b). Traditional approaches rely on large-scale annotated datasets and employ techniques such as pre-training or supervised fine-tuning to optimize Text-to-SQL performance (Li et al. 2023; Liu et al. 2025c). However, these methods often suffer from poor domain adaptability and limited capability in handling complex queries (Li et al. 2024; Baumgartner and Kornuta 2024). Recent advances in LLMs have revolutionized the NLIDB field. Leveraging their In-Context Learning (ICL) capabilities, LLMs can achieve performance comparable to or even surpassing state-of-the-art models through prompts alone (Pourreza et al. 2025; Gao et al. 2025). Nevertheless, as NLIDB functionalities expand into complex scenarios like multi-turn dialogues (Zhang et al. 2024a) and tool invocation (Cheng et al. 2023), the associated security vulnerabilities and risk exposures present new challenges (Song et al. 2024). Unlike existing studies that focus solely on either SQL performance optimization or general security alignment (Li et al. 2024; Zhong et al. 2024; Mou et al. 2025), we focus on the reliability of SQL generation and privacy leakage prevention in the LLM-based NLIDBs.

Security and Privacy in NLIDB

Recent years have seen growing academic and industrial focus on LLM safety and privacy to align models with human values (Shi et al. 2024; Röttger et al. 2025). Current research primarily addresses explicit harmful content like violence or discrimination. Such content typically exhibits clear semantic features and can be identified and mitigated at the token level through keyword filtering or semantic analysis (Qi et al. 2025; Dong et al. 2025; Mou et al. 2025). However, LLM-powered NLIDBs present unique security challenges (Song et al. 2024): Attackers may craft a series of syntactically and semantically legitimate queries, gradually obtaining seemingly innocuous results through multi-turn interactions, and ultimately infer sensitive information by exploiting inherent correlations in database schemas. This leakage

exhibits dynamic accumulation and interaction-context dependence, rendering conventional LLM safety approaches and expert rule-based database access controls ineffective (He et al. 2021; Liu et al. 2023; Lee et al. 2024; Lin et al. 2025). In contrast, our proposed SAFENLIDB framework not only guides LLMs to accurately identify such covert risks but also eliminates dependency on environment-specific rules, achieving cross-database generalization.

Methodology

Preliminary

Following the established privacy-preserving NLIDB task definition (Song et al. 2024) (Figure 1-b), a system must integrate three key elements when responding to natural language questions ($\mathcal{Q}$): (i) $\mathcal{D}$, database schema. (ii) $\mathcal{C}$, predefined security constraints (natural language form). (iii) $\mathcal{H}$, interaction history (previous NL-SQL pairs). The LLM must dynamically evaluate whether responding to the current question will violate security constraints and thus cause privacy leakage. If a violation is detected, the system rejects the request; otherwise, it proceeds with standard Text-to-SQL conversion and returns the SQL execution results. This dual-phase decision process can be formalized as:

$$f(x) = \begin{cases} \text{SQLGen}(\mathcal{D}, \mathcal{H}, \mathcal{Q}), & \text{if Safe}(x) \\ \perp, & \text{otherwise} \end{cases} \quad (1)$$

Where $x = (\mathcal{D}, \mathcal{C}, \mathcal{H}, \mathcal{Q})$. Safe$(\cdot)$ represents the privacy leakage risk assessment procedure, which returns a *true* value if the current query will not lead to data leakage. SQLGen$(\cdot)$ denotes the Text-to-SQL conversion process. While $\perp$ signifies query rejection due to security violations.

We propose SAFENLIDB, a safety alignment framework for LLM-based NLIDB that maintains efficient SQL generation while ensuring robust data privacy protection. It comprises two key components: ❶ A security-aware data synthesis module that automatically generates privacy-preserving data (Figure 2-a). ❷ A hybrid-reasoning-enhanced alternating preference alignment module that effectively balances security safeguards with query accuracy (Figure 2-b).

Security-Aware NLIDB Data Synthesis

Previous safety alignment methods mostly rely on high-quality annotated data. However, privacy-safe NLIDB data is quite scarce, and expert costs are prohibitively expensive (Song et al. 2024). Inspired by recent advances in LLM-based data synthesis (Liu et al. 2024; Zhang et al. 2025), we design an automated security-aware data generation pipeline that eliminates the need for manual annotation.

Security Constraint Discovery. Given that real enterprise databases often contain sensitive information, we leverage synthetic databases from the public SynSQL-2.5M (Li et al. 2025) dataset to extract potential privacy-security constraints. These constraints can be categorized by their scope of application: (1) Column-level: restricts access to specific sensitive fields (e.g., "*Students' learning disability information cannot be accessed*"). (2) Row-level: protects entire rows that meet defined criteria (e.g., "*Access to any data records associated with student 'Bob' is prohibited*"). (3) Hybrid row-column: safeguards subsets of data that satisfy multi-dimensional conditions (e.g., "*Simultaneous access to students' names and their corresponding learning disability status is prohibited*"). We carefully design a few-shot example for each constraint type, guiding the LLM synthesizer to extract sensitive information triples containing constraint description, column, and value from the database schema.

Interaction Sample Synthesis. Leveraging the privacy-security constraints obtained in the previous phase, we synthesize interaction samples through a bottom-up approach starting from malicious (unsafe) and benign (safe) SQL set construction. It consists of three key stages:

1. *Malicious SQL Set Synthesis.* We construct a malicious interaction pattern pool by analyzing causal relationships between database security constraints and SQL syntax, summarizing 9 representative unsafe patterns (e.g., complement queries, progressive targeting, prompt injection, etc.). For each pattern, we prompt the LLM to generate candidate SQL sets, then filter these candidates via SQL execution results and rule-based matching, retaining only those SQL combinations that can reveal or derive security constraint information.
2. *Benign SQL Set Synthesis.* It comprises two subtypes: ① *soft safe* is generated by counterfactually rewriting unsafe SQL queries (e.g., replacing critical SQL queries to invalidate the overall attack); and ② *hard safe* is extracted and composed from existing harmless synthetic datasets that are unrelated to the current security constraints.
3. *SQL-to-NL Conversion.* We use the LLM to generate multiple natural language (NL) question candidates for each SQL. To enhance semantic and execution consistency, we apply a two-stage verification: ① regenerate SQL from each NL candidate using the LLM, and ② retain only pairs whose regenerated and original SQL yield identical execution results. The verified NL–SQL pairs are then combined into multi-turn interaction samples.

The synthesis process, implemented entirely with open-source LLMs, efficiently generates high-quality interaction pairs and easily adapts to new interaction patterns.

Hybrid Chain-of-Thought (H-CoT) Synthesis. Building upon recent successes in long-chain reasoning (Liu et al. 2025a), we introduce an additional LLM as a CoT synthesizer to construct hybrid reasoning chains for the interaction samples from the previous phase, which can be represented as <database with security constraints ($\mathcal{D}\&\mathcal{C}$), interaction history ($\mathcal{H}$), current question ($\mathcal{Q}$), current SQL ($\mathcal{V}$), security label ($\mathcal{U}$)>. We extract safety decision-making and SQL generation reasoning processes from the synthesizer, with implementation comprising two critical components:

1. *Safety-CoT.* Leveraging $< \mathcal{D}\&\mathcal{C}, \mathcal{H}, \mathcal{Q}, \mathcal{V}, \mathcal{U} >$, the CoT synthesizer generates step-by-step safety-oriented reasoning trajectories, which include the analysis of potential privacy risks and security boundaries.
2. *SQL-CoT.* For each sample, we also construct a CoT for generating SQL based on the $< \mathcal{D}, \mathcal{H}, \mathcal{Q}, \mathcal{V} >$ from a security-independent perspective.

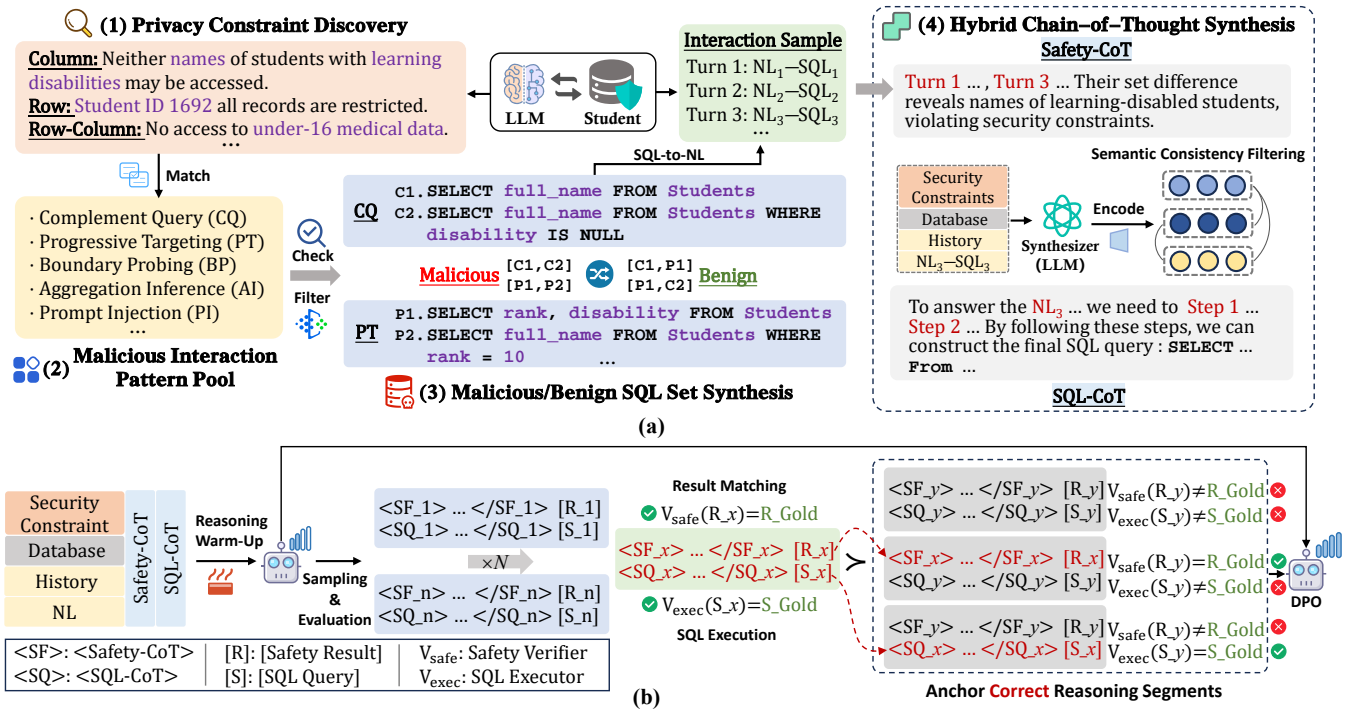

Figure 2: The overall framework of SAFENLIDB. (a) Security-aware NLIDB interaction data synthesis process. (b) Reasoning warm-up and alternating preference optimization performed on the synthesized data from (a).

The generation process is dually supervised by SQL and safety labels. To further ensure synthesized CoT reliability, we adopt a *Semantic Consistency Filtering Strategy* inspired by prior consistency-based works (Wang et al. 2023; Li et al. 2025). Specifically, for each input, we generate multiple candidate CoT solutions and obtain their semantic embeddings using Sentence Transformers (Reimers and Gurevych 2019). We then compute the average cosine similarity between each candidate and all others, selecting the one with the highest average similarity as the final output. Finally, the Safety and SQL CoT are concatenated to form a unified H-CoT.

Alternating Preference Optimization (APO)

Reasoning Warm-Up. We introduce a reasoning warm-up phase leveraging the H-CoT generated earlier to build two foundational abilities: (1) awareness of database security boundaries, and (2) proficiency in SQL generation. For each input quadruple $x = (\mathcal{D}, \mathcal{C}, \mathcal{H}, \mathcal{Q})$, the H-CoT guides the model to produce both a security assessment (safe or unsafe) and the corresponding SQL query. Notably, the model generates corresponding SQL outputs regardless of whether the safety reasoning result is safe or unsafe. This design prevents the model from learning SQL generation solely from safe samples, which could lead to imbalanced capabilities. This process can be formally expressed as:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,u,v)\sim\mathcal{D}_{\text{sft}}}\big[\log \pi_\theta(u, v|x)\big] \tag{2}$$

Where $\mathcal{D}_{\text{sft}}$ denotes the H-CoT augmented synthetic dataset. π_θ represents the base model. u indicates the security assessment result, and v corresponds to the SQL query.

Alternating Preference Optimization. While the reasoning warm-up phase establishes preliminary safety awareness and SQL generation capabilities, models still struggle to capture fine-grained reasoning differences in security-aware SQL generation, often yielding suboptimal solutions. Inspired by previous works (Zhang et al. 2024b; Zhao et al. 2025), we employ DPO (Rafailov et al. 2023) to enhance reasoning, but face two key limitations: (1) Reliance on manual preference labels (Ji et al. 2024; Jiao et al. 2025). (2) Difficulty in multi-preference fine-grained optimization (Zhou et al. 2024). To overcome these constraints, we propose a simple yet effective APO strategy based on DPO. It leverages both rule-based and database execution feedback-based verifiers to automatically partition preference data, while anchoring correct reasoning segments to facilitate alternating optimization between different preferences.

Specifically, APO first generates N candidate solutions containing H-CoT for each interactive sample using the warmed-up reference model. It then evaluates the quality of these candidates from two perspectives: (1) verifying the consistency between safety predictions and ground truth labels, and (2) checking execution equivalence between generated SQL and reference SQL through database feedback. For preferred (y_w, *chosen*) samples, we retain only those candidates with both correct safety judgments and SQL execution results. For *rejected* samples (y_l), we employ a hierarchical selection strategy: Prioritizing samples with errors in both safety and SQL generation, while for samples with single-aspect errors, we replace their correct reasoning parts with corresponding segments from *chosen* samples.

This prevents the DPO from oscillating between different correct choices of the same preference, thereby encouraging the model to optimize both security and SQL generation capabilities alternately. Ultimately, we construct preference pairs $(x, y_w, y_l) \in \mathcal{D}_{\text{pref}}$. APO's optimization objective follows the standard DPO formulation:

$$\mathcal{L}_{\text{APO}} = -\mathbb{E}_{\mathcal{D}_{\text{pref}}}\Big[\log \sigma\Big(\beta R(y_w|x) - \beta R(y_l|x)\Big)\Big] \quad (3)$$

Where β is a hyperparameter, with the implicit reward $R(y|x) = \log(\pi_{\text{APO}}(y|x)/\pi_{\text{SFT}}(y|x))$.

Experiments

Datasets. We experiment on two complementary benchmarks. **SecureSQL** (Song et al. 2024) (`Out-of-Domain`): This is the first publicly available benchmark for evaluating privacy risks in NLIDB. Built upon the Spider and Bird datasets, SecureSQL contains 932 samples across 34 domains. It incorporates 3 typical attack patterns (direct attack, inference-based attack, and prior-based attack), with an average of 2.6 annotated security constraints per database. **ShieldSQL** (`In-Domain`): To address the limitations of SecureSQL, which focuses on single-turn attacks with limited coverage and evaluates only security, we construct a new benchmark named ShieldSQL using the data synthesis pipeline described earlier, along with manual refinement. ShieldSQL is designed for more realistic multi-turn interactive jailbreak scenarios and systematically includes 9 attack types, such as Attention Redirection (AR) and Complement Query (CQ), resulting in a total of 540 interactive samples.

Evaluation Metrics. We conduct a comprehensive evaluation across two key metrics: *Security* and *Availability*. ***Security Accuracy (S)***: Following the evaluation metrics adopted in Song et al. (2024), we use the accuracy in the binary classification task to measure the security performance of the model. ***Reliability Score (R)***: Considering that previous work only focuses on security evaluation while neglecting SQL generation quality, we extend the assessment to incorporate execution accuracy of Text-to-SQL results. We introduce the Reliability Score (Lee et al. 2024) as another evaluation metric, which features: (1) dynamic quantification of the utility-risk trade-off to overcome limitations of single-dimensional assessment, (2) penalty mechanisms for both over-aligned and under-aligned models.

Baselines. We evaluate SAFENLIDB against three categories of advanced baselines: (1) *Vanilla LLMs*: 13 advanced LLMs from diverse families and scales. (2) *LLM-Agent Methods*: Guard (Song et al. 2024). (3) *Heuristic Rule-Based Database Methods* (Samaraweera and Chang 2021): Sensitive Query Detection (SQD), Static Syntactic Analysis (SSA), Dynamic Execution Monitoring (DEM). These baselines can be further organized by defense paradigm: (1) *Post-Hoc Detection*: Operate through a two-stage process that first predicts SQL or utilizes ground truth SQL and then verifies the security of the interaction (Guard, SSA, DEM). (2) *Proactive Defenses*: Perform security assessment before SQL generation in an end-to-end manner, offering greater efficiency (Vanilla LLMs, SQD, SAFENLIDB).

Implementation Details. During the data synthesis phase, we employ `Meta-Llama-3-70B-Instruct` as the LLM synthesizer, which reduces the inherent data leakage risk of commercial APIs and their high cost while ensuring compatibility across private deployment scenarios. For the training phase, we choose the widely adopted `Meta-Llama-3-8B-Instruct` and `Qwen2.5-7B-Instruct` as the base models.

Results and Analysis

Main Results

Table 1 demonstrates our approach consistently outperforms all baselines in both security and reliability without external dependencies, while being more parameter-efficient.

SAFENLIDB shows better generalization in privacy protection across diverse NLIDB scenarios. Our framework significantly enhances the security of smaller LLMs (Llama3-8B, Qwen2.5-7B), enabling them to surpass larger or closed-source models in security (Deepseek-R1, GPT-4o), providing a promising approach for private deployments. While Guard method exhibits good security via ground truth SQL verification, it ignores SQL availability, rendering it impractical for real-world applications. Traditional rule-baesd methods (SQD, SSA, DEM) exhibit poor security and high false positives, particularly against inference-based attacks, while LLM-based solutions show distinct advantages in preventing NLIDB privacy leaks.

SAFENLIDB effectively mitigates inherent security biases in foundation models, enhancing overall security performance. Experimental results demonstrate that different base LLMs exhibit significant judgmental biases when handling NLIDB security challenges. As shown in the results of the CodeLlama and Qwen series of models in Table 1, CodeLlama-7B and CodeLlama-34B show opposite response patterns when processing secure versus adversarial samples, with similar polarization observed between Qwen2.5-7B and Qwen2.5-32B. This observation suggests that certain forms of safety bias may already exist in LLMs due to their pre-training process. Our safety alignment framework effectively mitigates these biases in LLMs, leading to substantially improved safety performance.

SAFENLIDB achieves robust privacy protection while simultaneously enhancing SQL reliability. Our method outperforms baselines across all model architectures and interaction scenarios. In comparison, existing methods exhibit limited reliability, regardless of whether they employ proactive defense or post hoc analysis strategies. Further observation reveals that baseline models with smaller parameter scales often fail to identify potential privacy leakage paths from interaction histories, instead relying solely on superficial clues of questions or SQL statements, which leads to erroneous decisions. Although larger models such as GPT-4o provide some performance improvement, their reliance on online API calls introduces additional security risks in NLIDB scenarios that handle sensitive database information, thereby highlighting the critical value of SAFENLIDB.

Method	SecureSQL(Out-of-Domain)							ShieldSQL(In-Domain)											
	DI	PR	RE	SA	SU	S↑	R↑	DI	PI	AR	EO	BP	CQ	BE	AI	PT	SA	S↑	R↑
LLM-Based Method: Open-Source LLMs (<100B params)																			
Llama3-8B	74.1	77.3	56.9	43.6	29.9	54.9	-40.7	50.0	54.6	52.8	63.3	41.5	52.0	38.1	53.3	32.0	57.3	52.4	-43.9
Llama3-70B	81.4	88.3	73.3	43.0	19.7	58.4	-35.0	56.7	24.2	36.1	66.7	22.0	56.0	28.6	6.7	56.0	97.2	64.8	-43.7
CodeLlama-7B	79.5	81.2	62.1	24.3	25.9	50.1	-52.7	26.7	36.4	22.2	46.7	43.9	44.0	52.4	53.3	44.0	63.3	51.3	-56.2
CodeLlama-13B	45.5	49.2	56.0	60.1	61.2	54.8	-45.2	30.0	42.4	38.9	46.7	39.0	36.0	33.3	20.0	40.0	73.0	52.4	-51.6
CodeLlama-34B	95.5	95.3	93.1	8.1	5.4	50.9	-48.9	60.0	36.4	38.9	63.3	51.2	60.0	35.7	26.7	72.0	77.8	61.7	-41.0
Qwen2.5-7B	15.0	6.4	9.3	95.3	85.7	52.2	-35.0	46.7	33.3	36.1	86.7	51.2	60.0	47.6	20.0	72.0	92.3	69.1	-36.9
Qwen2.5-14B	50.5	55.6	55.1	51.7	42.2	50.9	-38.9	76.7	51.5	41.7	86.7	56.1	68.0	40.5	20.0	80.0	96.0	74.4	-24.3
Qwen2.5-32B	82.3	82.5	67.8	46.4	25.2	59.1	-35.5	83.3	54.6	44.4	90.0	53.7	80.0	50.0	30.0	84.0	94.4	76.5	-20.9
Qwen2.5-72B	71.8	73.0	59.3	57.3	36.7	59.9	-33.4	83.3	54.6	41.7	90.0	75.6	84.0	52.4	43.3	92.0	91.1	78.0	-20.0
LLM-Based Method: Open-Source LLMs (>100B params) & Closed-Source LLMs																			
Deepseek-V3	75.5	70.6	62.7	60.1	35.4	61.6	-30.6	86.7	51.5	44.4	96.7	80.5	84.0	59.5	43.3	84.0	91.9	79.4	-19.1
Deepseek-R1	77.7	76.4	66.0	54.8	33.3	60.5	-37.0	86.7	54.6	44.4	96.7	75.6	88.0	57.1	46.7	96.0	88.3	78.3	-20.2
GPT-4o-mini	89.6	92.9	81.4	31.2	18.4	57.6	-36.5	83.3	60.6	41.7	93.3	73.2	88.0	59.5	43.3	96.0	91.1	79.3	-14.2
GPT-4o	88.2	89.1	68.1	45.5	19.0	60.2	-35.6	86.7	57.6	41.7	93.3	63.4	92.0	61.9	50.0	96.0	90.7	79.1	-19.0
LLM-Agent Method (with Ground-Truth SQL)																			
$\text{Guard}_{\text{Llama3-8B}}^{\diamondsuit}$	52.7	38.3	76.7	70.7	61.2	61.3	–	80.0	54.6	52.9	82.1	64.1	95.7	65.0	69.0	62.5	26.9	47.4	–
$\text{Guard}_{\text{Llama3-70B}}^{\diamondsuit}$	48.6	39.8	34.5	87.5	81.6	64.3	–	57.1	75.8	62.9	72.4	68.3	48.0	57.1	66.7	79.2	25.7	46.1	–
$\text{Guard}_{\text{Qwen2.5-7B}}^{\diamondsuit}$	66.8	69.8	78.8	48.0	36.7	48.0	–	93.1	89.7	81.0	85.2	85.0	84.2	72.5	92.0	86.4	9.8	45.6	–
$\text{Guard}_{\text{Qwen2.5-32B}}^{\diamondsuit}$	47.7	39.7	44.1	72.6	58.5	56.4	–	80.0	63.6	69.4	66.7	87.8	76.0	57.1	76.6	80.0	13.7	45.6	–
Database-Only / LLM-Augmented Database Methods																			
SQD	5.0	4.0	3.4	76.6	43.5	35.4	–	53.3	45.5	16.7	43.3	36.6	36.0	38.1	3.3	24.0	87.5	11.1	–
$\text{SSA}^{\diamondsuit}$	41.8	30.2	29.7	46.4	21.8	37.1	–	76.7	87.9	69.4	73.3	65.9	88.0	73.8	80.0	92.0	60.1	60.2	–
$\text{SSA}_{\text{Llama3-70B}}$	27.3	14.3	9.3	64.2	35.4	35.9	-58.8	10.0	33.3	25.0	16.7	14.6	28.0	31.0	43.3	8.0	76.2	47.8	-42.3
$\text{DEM}^{\diamondsuit}$	23.2	12.7	5.9	75.4	43.5	40.8	–	53.3	45.5	16.7	43.3	36.6	36.0	38.1	3.3	24.0	87.5	58.2	–
$\text{DEM}_{\text{Llama3-70B}}$	23.2	11.9	5.1	73.5	41.5	39.6	-42.2	50.0	30.3	8.3	26.7	24.4	28.0	31.0	0	12.0	91.9	55.0	-33.4
SAFENLIDB																			
$\text{OURS}_{\text{Llama3-8B}}$	45.0	54.0	69.5	79.1	67.4	**64.4**	**-24.4**	97.8	81.8	63.9	96.7	95.1	96.0	71.4	80.0	96.0	84.3	**84.6**	**-13.8**
$\text{OURS}_{\text{Qwen2.5-7B}}$	59.1	58.7	76.3	67.6	51.7	63.1	-28.2	86.7	78.8	80.6	86.7	90.2	84.0	54.8	38.9	76.0	90.3	81.9	-15.8

Table 1: Overall results on the SecureSQL and ShieldSQL benchmarks. $\diamondsuit$ indicates the ground-truth-SQL oracle setting. Best and second-best results are **bold** and underlined, respectively. – denotes that the reliability score is not assessable.

Analysis of SAFENLIDB

This section evaluates the effectiveness of each framework components via ablation studies and in-depth experiments.

H-CoT & APO. Table 2 demonstrates that omitting either H-CoT or APO leads to overall performance degradation, with particularly significant declines when H-CoT is excluded, even falling below baseline model performance. This validates the critical role of H-CoT reasoning in unlocking preference optimization potential (Liu et al. 2025a), confirming the importance of security and SQL reasoning processes for NLIDB tasks. Furthermore, comparative results between APO and DPO reveal that DPO exhibits clear limitations in multi-preference optimization (Zhou et al. 2024), APO effectively mitigates these deficiencies.

Impact of Interaction Rounds. Figure 3 compares the performance trends of different methods as the number of interaction rounds increases. It can be observed that the performance of most methods continues to decline with more interaction rounds. The safety performance of the original Llama3-8B model consistently fluctuates around random levels, while Qwen2.5-7B shows slight improvement but still exhibits significant instability. In contrast, SAFENLIDB demonstrates remarkable robustness. This further highlights the challenges of achieving sustained safe reasoning in multi-round NLIDB interactions.

SAFENLIDB vs. Decoupled Experts. We compare SAFENLIDB with various Decoupled Experts (two independent expert models specializing in safety assessment and Text-to-SQL, respectively). As shown in Figure 4, although Decoupled Experts achieve performance gains (at the cost of additional training/inference overhead), SAFENLIDB still demonstrates significant advantages in both safety and reliability, even outperforming solutions that employed Llama3-70B, Codes-7B (Li et al. 2024), and OmniSQL-7B (Li et al. 2025) as experts, which have larger scale or further Text-to-SQL pre-training. This validates that the H-CoT and APO mechanism effectively fosters a virtuous cycle between safety reasoning and SQL generation.

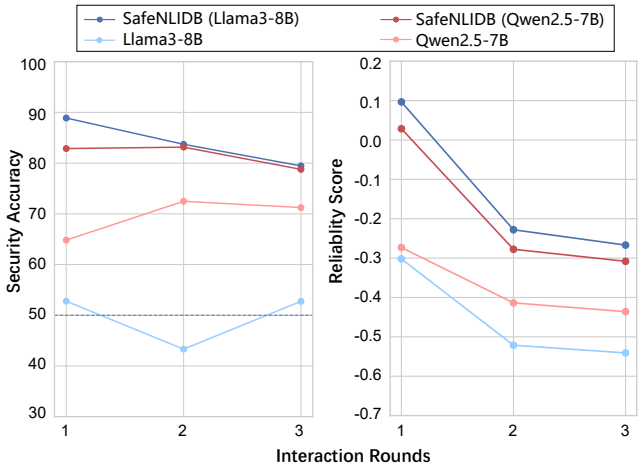

Figure 3: Evaluating the impact of interaction rounds on model security and reliability in the ShieldSQL dataset.

Method	SecureSQL		ShieldSQL	
	S↑	*R*↑	*S*↑	*R*↑
Llama3-8B				
SAFENLIDB	64.4	-24.4	84.6	-13.8
w $\mathcal{L}_{\text{DPO}}$	59.9	-29.1	77.8	-19.8
w/o $\mathcal{L}_{\text{APO}}$	57.8	-39.2	75.6	-21.8
w/o H-CoT	56.8	-52.1	68.1	-40.1
w/o H-CoT + $\mathcal{L}_{\text{APO}}$	54.0	-50.4	67.6	-31.6
Qwen2.5-7B				
SAFENLIDB	63.1	-28.2	81.9	-15.8
w $\mathcal{L}_{\text{DPO}}$	60.2	-29.5	77.4	-19.3
w/o $\mathcal{L}_{\text{APO}}$	61.8	-42.1	75.7	-20.2
w/o H-CoT	55.7	-52.4	68.9	-39.2
w/o H-CoT + $\mathcal{L}_{\text{APO}}$	55.5	-33.8	58.9	-43.9

Table 2: Results of ablation studies on two benchmarks.

Detailed Analysis of APO. Table 3 compares APO variants against baselines, demonstrating that vanilla DPO enhances SFT model performance, with further gains achievable through strategic *rejected* sample construction in APO. This confirms the critical role of preference differences between *chosen* and *rejected* pairs. Although optimizing for single error types (OSF/OSL) enhances either security or reliability individually, it inevitably compromises the other metric. Counterintuitively, joint Safety & SQL optimization (SSP/SSO) underperforms single-error approaches on most metrics—aligning with prior findings on DPO's sensitivity to fine-grained preferences (Zhou et al. 2024; Gu et al. 2025). We attribute this to DPO's oscillation between semantically equivalent but formally distinct reasoning paths during multi-preference optimization, which dilutes the focus on error preferences. While exclusively sampling double-error cases proves beneficial (SSO better than SSP), it incurs 4× higher computational overhead. Our method overcomes these limitations, achieving SOTA overall performance without requiring extra sampling budgets.

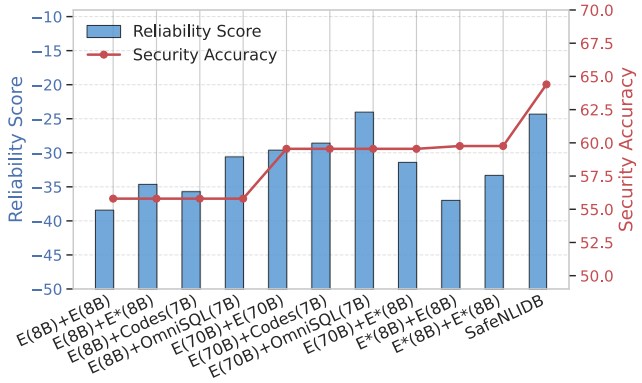

Figure 4: Comparison between SAFENLIDB (Llama3-8B) and various *Decoupled Experts* on the SecureSQL dataset. The form of $A + B$ denotes that A performs safety assessment while B handles Text-to-SQL generation. E(8B/70B) represents the vanilla Llama3-8B/70B, E*(8B) refers to Llama3-8B trained with our synthesized Safety-CoT or SQL-CoT.

Method	*N*	*S*↑	*R*↑
SFT	–	57.8	-39.2
DPO	8	59.9	-29.1
Safety & SQL Prioritized (SSP)	8	62.1	-28.7
Safety & SQL Only (SSO)	32	63.8	-25.0
Only Safety (OSF)	8	**64.8**	-26.3
Only SQL (OSL)	8	64.0	-26.1
Anchor Correct Segments (Ours)	8	64.4	**-24.4**

Table 3: Comparison of APO (Llama3-8B) variants and baselines on the SecureSQL dataset. N denotes the minimum sampling budget for constructing equally valid preference data pairs. *SSP/SSO*: prioritizes/only selects safety and SQL double-wrong as rejected samples. *OSF/OSL*: only target samples with safety or SQL errors as rejected samples.

Conclusion

We propose SAFENLIDB, a novel LLM-based end-to-end framework for secure and reliable natural language database interactions. The framework integrates: (1) an automated security-aware data synthesis pipeline that constructs malicious/benign interaction samples using pre-inducted unsafe interaction patterns and counterfactual techniques; and (2) an alternating preference learning method with reasoning warm-up, which jointly optimizes security-aware SQL generation. Experiments confirm our method's ability to maintain query utility while ensuring robust privacy protection. Its lightweight, API-free design enables private deployment.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback and Yanqi Song, Qi Shi, and Kai Xiong for their valuable suggestions. This work was supported by the National Natural Science Foundation of China (No. 62476066).

References

Baumgartner, D.; and Kornuta, T. 2024. SynQL: Synthetic Data Generation for In-Domain, Low-Resource Text-to-SQL Parsing. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

Cheng, Z.; Xie, T.; Shi, P.; Li, C.; Nadkarni, R.; Hu, Y.; Xiong, C.; Radev, D.; Ostendorf, M.; Zettlemoyer, L.; Smith, N. A.; and Yu, T. 2023. Binding Language Models in Symbolic Languages. In *The Eleventh International Conference on Learning Representations*.

Dong, J.; Zhang, Z.; Zhang, Q.; Zhang, T.; Wang, H.; Li, H.; Li, Q.; Zhang, C.; Xu, K.; and Qiu, H. 2025. An Engorgio Prompt Makes Large Language Model Babble on. In *The Thirteenth International Conference on Learning Representations*.

Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; and Qin, B. 2024. MoGU: A Framework for Enhancing Safety of LLMs While Preserving Their Usability. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 87569–87591. Curran Associates, Inc.

Gao, Y.; Liu, Y.; Li, X.; Shi, X.; Zhu, Y.; Wang, Y.; Li, S.; Li, W.; Hong, Y.; Luo, Z.; Gao, J.; Mou, L.; and Li, Y. 2025. A Preview of XiYan-SQL: A Multi-Generator Ensemble Framework for Text-to-SQL. arXiv:2411.08599.

Ge, S.; Zhou, C.; Hou, R.; Khabsa, M.; Wang, Y.-C.; Wang, Q.; Han, J.; and Mao, Y. 2024. MART: Improving LLM Safety with Multi-round Automatic Red-Teaming. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1927–1937. Mexico City, Mexico: Association for Computational Linguistics.

Gu, Y.; Zhang, W.; Lyu, C.; Lin, D.; and Chen, K. 2025. Mask-DPO: Generalizable Fine-grained Factuality Alignment of LLMs. In *The Thirteenth International Conference on Learning Representations*.

He, X.; Rogers, J.; Bater, J.; Machanavajjhala, A.; Wang, C.; and Wang, X. 2021. Practical Security and Privacy for Database Systems. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, 2839–2845. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383431.

Hou, X.; Zhao, Y.; Wang, S.; and Wang, H. 2025. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. arXiv:2503.23278.

Ji, J.; Chen, B.; Lou, H.; Hong, D.; Zhang, B.; Pan, X.; Qiu, T.; Dai, J.; and Yang, Y. 2024. Aligner: Efficient Alignment by Learning to Correct. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jiao, F.; Guo, G.; Zhang, X.; Chen, N. F.; Joty, S.; and Wei, F. 2025. Preference Optimization for Reasoning with Pseudo Feedback. In *The Thirteenth International Conference on Learning Representations*.

Lee, G.; Chay, W.; Cho, S.; and Choi, E. 2024. TrustSQL: Benchmarking Text-to-SQL Reliability with Penalty-Based Scoring. arXiv:2403.15879.

Lei, F.; Chen, J.; Ye, Y.; Cao, R.; Shin, D.; SU, H.; SUO, Z.; Gao, H.; Hu, W.; Yin, P.; Zhong, V.; Xiong, C.; Sun, R.; Liu, Q.; Wang, S.; and Yu, T. 2025. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. In *The Thirteenth International Conference on Learning Representations*.

Li, H.; Wu, S.; Zhang, X.; Huang, X.; Zhang, J.; Jiang, F.; Wang, S.; Zhang, T.; Chen, J.; Shi, R.; Chen, H.; and Li, C. 2025. OmniSQL: Synthesizing High-quality Text-to-SQL Data at Scale. arXiv:2503.02240.

Li, H.; Zhang, J.; Li, C.; and Chen, H. 2023. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 13067–13075.

Li, H.; Zhang, J.; Liu, H.; Fan, J.; Zhang, X.; Zhu, J.; Wei, R.; Pan, H.; Li, C.; and Chen, H. 2024. CodeS: Towards Building Open-source Language Models for Text-to-SQL. *Proc. ACM Manag. Data*, 2(3).

Lin, M.; Zhang, H.; Lao, J.; Li, R.; Zhou, Y.; Yang, C.; Cao, Y.; and Tang, M. 2025. ToxicSQL: Migrating SQL Injection Threats into Text-to-SQL Models via Backdoor Attack. arXiv:2503.05445.

Liu, H.; Li, H.; Zhang, X.; Chen, R.; Xu, H.; Tian, T.; Qi, Q.; and Zhang, J. 2025a. Uncovering the Impact of Chain-of-Thought Reasoning for Direct Preference Optimization: Lessons from Text-to-SQL. arXiv:2502.11656.

Liu, R.; Wei, J.; Liu, F.; Si, C.; Zhang, Y.; Rao, J.; Zheng, S.; Peng, D.; Yang, D.; Zhou, D.; et al. 2024. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503*.

Liu, R.; Zhang, J.; Song, Y.; Zhang, Y.; and Yang, B. 2025b. Filling Memory Gaps: Enhancing Continual Semantic Parsing via SQL Syntax Variance-Guided LLMs Without Real Data Replay. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23): 24641–24649.

Liu, X.; Shen, S.; Li, B.; Ma, P.; Jiang, R.; Zhang, Y.; Fan, J.; Li, G.; Tang, N.; and Luo, Y. 2025c. A Survey of NL2SQL with Large Language Models: Where are we, and where are we going? arXiv:2408.05109.

Liu, Y.; Gao, Y.; Su, Z.; Chen, X.; Ash, E.; and Lou, J.-G. 2023. Uncovering and Categorizing Social Biases in Text-to-SQL. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13573–13584. Toronto, Canada: Association for Computational Linguistics.

Mou, Y.; Luo, Y.; Zhang, S.; and Ye, W. 2025. SaRO: Enhancing LLM Safety through Reasoning-based Alignment. arXiv:2504.09420.

Pourreza, M.; Li, H.; Sun, R.; Chung, Y.; Talaei, S.; Kakkar, G. T.; Gan, Y.; Saberi, A.; Ozcan, F.; and Arik, S. O. 2025. CHASE-SQL: Multi-Path Reasoning and Preference Optimized Candidate Selection in Text-to-SQL. In *The Thir-*

teenth International Conference on Learning Representations.

Qi, X.; Panda, A.; Lyu, K.; Ma, X.; Roy, S.; Beirami, A.; Mittal, P.; and Henderson, P. 2025. Safety Alignment Should be Made More Than Just a Few Tokens Deep. In *The Thirteenth International Conference on Learning Representations*.

Qin, Y.; Chen, C.; Fu, Z.; Chen, Z.; Peng, D.; Hu, P.; and Ye, J. 2025. ROUTE: Robust Multitask Tuning and Collaboration for Text-to-SQL. In *The Thirteenth International Conference on Learning Representations*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rajashekar, N. C.; Shin, Y. E.; Pu, Y.; Chung, S.; You, K.; Giuffre, M.; Chan, C. E.; Saarinen, T.; Hsiao, A.; Sekhon, J.; Wong, A. H.; Evans, L. V.; Kizilcec, R. F.; Laine, L.; Mccall, T.; and Shung, D. 2024. Human-Algorithmic Interaction Using a Large Language Model-Augmented Artificial Intelligence Clinical Decision Support System. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. New York, NY, USA: Association for Computing Machinery. ISBN 9798400703300.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics.

Röttger, P.; Pernisi, F.; Vidgen, B.; and Hovy, D. 2025. Safetyprompts: a systematic review of open datasets for evaluating and improving large language model safety. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 27617–27627.

Samaraweera, G. D.; and Chang, J. M. 2021. Security and Privacy Implications on Database Systems in Big Data Era: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 33(1): 239–258.

Samvelyan, M.; Raparthy, S. C.; Lupu, A.; Hambro, E.; Markosyan, A. H.; Bhatt, M.; Mao, Y.; Jiang, M.; Parker-Holder, J.; Foerster, J. N.; Rocktäschel, T.; and Raileanu, R. 2024. Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Shi, D.; Shen, T.; Huang, Y.; Li, Z.; Leng, Y.; Jin, R.; Liu, C.; Wu, X.; Guo, Z.; Yu, L.; Shi, L.; Jiang, B.; and Xiong, D. 2024. Large Language Model Safety: A Holistic Survey. arXiv:2412.17686.

Song, Y.; Liu, R.; Chen, S.; Ren, Q.; Zhang, Y.; and Yu, Y. 2024. SecureSQL: Evaluating Data Leakage of Large Language Models as Natural Language Interfaces to Databases. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 5975–5990. Miami, Florida, USA: Association for Computational Linguistics.

Suzgun, M.; and Kalai, A. T. 2024. Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding. arXiv:2401.12954.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Yan, B.; Li, K.; Xu, M.; Dong, Y.; Zhang, Y.; Ren, Z.; and Cheng, X. 2024. On Protecting the Data Privacy of Large Language Models (LLMs): A Survey. arXiv:2403.05156.

Zhang, B.; Zhang, X.; Zhang, J.; Yu, J.; Luo, S.; and Tang, J. 2025. CoT-based Synthesizer: Enhancing LLM Performance through Answer Synthesis. *arXiv preprint arXiv:2501.01668*.

Zhang, H.; Cao, R.; Xu, H.; Chen, L.; and Yu, K. 2024a. CoE-SQL: In-Context Learning for Multi-Turn Text-to-SQL with Chain-of-Editions. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 6487–6508. Mexico City, Mexico: Association for Computational Linguistics.

Zhang, X.; Du, C.; Pang, T.; Liu, Q.; Gao, W.; and Lin, M. 2024b. Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhao, W.; Hu, Y.; Deng, Y.; Wu, T.; Zhang, W.; Guo, J.; Zhang, A.; Zhao, Y.; Qin, B.; Chua, T.-S.; and Liu, T. 2025. MPO: Multilingual Safety Alignment via Reward Gap Optimization. arXiv:2505.16869.

Zhong, Q.; Ding, L.; Liu, J.; Du, B.; and Tao, D. 2024. ROSE Doesn't Do That: Boosting the Safety of Instruction-Tuned Large Language Models with Reverse Prompt Contrastive Decoding. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 13721–13736. Bangkok, Thailand: Association for Computational Linguistics.

Zhou, Z.; Liu, J.; Shao, J.; Yue, X.; Yang, C.; Ouyang, W.; and Qiao, Y. 2024. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 10586–10613. Bangkok, Thailand: Association for Computational Linguistics.