
ConstrainedSQL: Training LLMs for Text2SQL via Constrained Reinforcement Learning

Wei Qin Chen[♣] Nhan Huu Pham[♣] Michael Robert Glass[♣] Long Hai Vu[♣]
Gaetano Rossiello[♣] Dharmashankar Subramanian[♣] Santiago Paternain[♣]

[♣] Rensselaer Polytechnic Institute
[♣] IBM Research

Abstract

Reinforcement learning (RL) has demonstrated significant promise in enhancing the reasoning capabilities of Text2SQL LLMs, especially with advanced algorithms such as GRPO and DAPO. However, the performance of these methods is highly sensitive to the design of reward functions. Inappropriate rewards can lead to “reward hacking”, where models exploit loopholes in the reward structure to achieve high scores without genuinely solving the task. This work considers a constrained RL framework for Text2SQL that incorporates natural and interpretable reward and constraint signals, while dynamically balancing trade-offs among them during the training. We establish the theoretical guarantees of our constrained RL framework and our numerical experiments on the well-known Text2SQL datasets substantiate the improvement of our approach over the state-of-the-art RL-trained LLMs.

1 Introduction

Text2SQL aims to convert natural language questions to structured SQL statements, thus enabling the interactions with the database without any expertise and bridging the gap between the human language and database queries [1]. Although large language models (LLMs) with supervised fine-tuning (SFT) have shown great progress, they are often limited to the database schema structure and data scale. Recently, the success of large reasoning models such as DeepSeek-R1 [2] and OpenAI o1 [3] has demonstrated the potential of reinforcement learning (RL) in training the reasoning capabilities of LLMs. In particular, RL enables dynamic adjustment of strategies through interaction with the environment, a capability that is crucial for complex reasoning tasks [2]. However, despite the advances in RL algorithms e.g., GRPO [4] and DAPO [5], the policies are highly sensitive to the reward design. This is not surprising, since the best imaginable algorithm can only perform as well as the task it is asked to solve (the reward it must maximize). In particular, current state-of-the-art (SOTA) RL-trained Text2SQL LLMs such as SQL-R1 [6] and Reasoning-SQL [7] rely on manually designed rewards, which may be neither efficient nor reliable. Notice that inappropriate rewards can lead to “reward hacking”, where models exploit loopholes in the reward structure to achieve high scores without genuinely solving the task. Moreover, carefully and manually designed rewards can be costly and often generalize poorly to new tasks.

In this context, we propose ConstrainedSQL that trains Text2SQL LLMs via constrained RL [8, 9]. In particular, we design several natural and interpretable reward and constraint metrics for Text2SQL, and dynamically balance the trade-off among them during the RL training. Furthermore, we establish the theoretical guarantee of our method in terms of a bounded primal-dual gap that is dominated by a parameterization gap. We conduct extensive experiments on standard Text2SQL benchmarks including Spider (Dev), Spider (Test), and BIRD (Dev), comparing our approach with both SOTA RL-trained and SFT-based Text2SQL models. The results demonstrate that our method outperforms

SOTA RL-trained models and achieves performance comparable to SFT-based models, despite the latter using roughly $50\times$ more training samples.

2 ConstrainedSQL

2.1 Dataset

In this work, we extract our dataset from SynSQL-2.5M [10], which is the first million-scale synthetic Text2SQL dataset and contains more than 2.5 million high-quality samples. SFT often demands a substantial amount of samples, which is time consuming and resource demanding and is typically limited to producing only simple to moderate SQL queries. This occurs because SFT often tends to memorize only the samples encountered in its training dataset [11]. To explore the capability of LLM to generate complex SQL queries under limited samples, this work considers a dataset of $(x_i, y_i^*)_{i=1}^{5000}$, where x_i denotes the prompt (encompass the natural language question) and y_i^* its corresponding ground-truth response (ground-truth SQL). Notably, these 5000 samples are randomly drawn from SynSQL-2.5M among those that are deemed *Complex* (other categories in the dataset are *Simple*, *Moderate*, etc.). Having established the dataset, our work focuses on fine-tuning the Text2SQL LLM using RL rather than SFT, as RL enables dynamic adjustment of the strategy through interactions with the environment and can offer superior performance in complex reasoning tasks. The procedure of RL training is introduced in the following subsection.

2.2 Training of ConstrainedSQL

Denote by \mathcal{X} and \mathcal{Y} the prompt space and the LLM response space. An LLM policy $\pi_\theta : \mathcal{X} \rightarrow \Delta(\mathcal{Y})$ represents a mapping from the prompt space to the set of all distributions over the response space $\Delta(\mathcal{Y})$, where θ denotes the parameters in the LLM. The reward $r : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ evaluates the quality of the response generated by the LLM policy, which can be instantiated either as a reward model trained on human preference datasets or as a human-designed reward function. Consider a reference LLM policy π_{ref} (e.g., a SFT model). The training of the LLM policy via RL is formulated as

$$\max_{\theta \in \Theta} \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r_0(x, y)] - \beta D_{\text{KL}}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x))], \quad (1)$$

where $D_{\text{KL}}(\cdot \| \cdot)$ denotes the KL-divergence and β represents the regularization constant. Then, the objective of RL is to optimize the LLM policy parameters to maximize the expected reward while ensuring the LLM policy does not deviate excessively from the reference LLM. In the context of Text2SQL, the reward structure often adopts a composite reward with several sub-rewards such as format reward, execution reward, etc [6]. To this end, manually designing each reward and carefully tuning the corresponding weights between them becomes highly challenging. Moreover, this can lead to reward hacking. To address this issue, we adopt a constrained RL framework for training the LLM

$$\begin{aligned} P^* &= \max_{\theta \in \Theta} \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)] - \beta D_{\text{KL}}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x))] \\ \text{s.t. } &\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [c_i(x, y)]] \geq b_i, \quad i = 1, \dots, m \end{aligned} \quad (2)$$

where we consider a single reward function r , and c_i and b_i denote the i -th constraint and the corresponding threshold. The reward r and constraints c_i should be designed to align with the problem’s objective and requirements. For brevity, let us define $g_i(x, y) = c_i(x, y) - b_i$. Section 4 presents a specific instantiation of problem (2) in the Text2SQL setting, where we consider natural and interpretable reward and constraints without the need for carefully designing each component. Before proceeding, we introduce in the next section the theoretical guarantees of problem (2).

3 Theoretical Guarantees

A common approach to solve the constrained optimization problem (2) is to solve its unconstrained approximation using the Lagrangian

$$L(\pi_\theta, \lambda) := \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y) + \lambda^\top g(x, y)] - \beta D_{\text{KL}}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x))], \quad (3)$$

where $g = [g_1, g_2, \dots, g_m]^\top$ and $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]^\top$ denote the constraints and the corresponding Lagrange multipliers that trade-off the objective and the corresponding constraints. Indeed, for

Algorithm 1 Primal Dual Algorithm

Input: Initial Lagrange multipliers $\lambda^0 = [\lambda_1^0, \lambda_2^0, \dots, \lambda_m^0]^\top$, constraint thresholds $b = [b_1, b_2, \dots, b_m]^\top$, primal learning rate η_θ , dual learning rate η_λ , reference model π_{ref}

Let $\pi_{\theta^0} = \pi_{\text{ref}}$

1: **for** $k = 0, 1, \dots$ **do**

2: Update the primal variable (LLM policy parameters) using GRPO

$$\theta^{k+1} = \theta^k + \eta_\theta \nabla_\theta^{\text{GRPO}} L(\pi_{\theta^k}, \lambda^k)$$

3: Update the dual variable (Lagrange multipliers) using dual gradient descent

$$\lambda^{k+1} = [\lambda^k - \eta_\lambda (c_{\pi_{\theta^{k+1}}} - b)]_+$$

4: **end for**

large values of λ_i solutions will prioritize the i -th constraint, whereas for small values of λ_i the solutions will focus on the objective and other constraints. To this end, each constraint threshold b_i will correspond to a specific optimal Lagrange multiplier λ_i^* . The dual function associated with (3) is defined as $D(\lambda) := \max_{\theta \in \Theta} L(\pi_\theta, \lambda)$. Subsequently, the dual problem for (2) is given by

$$D^* = \min_{\lambda \geq 0} D(\lambda). \quad (4)$$

It is worth highlighting that problem (4) is always convex despite the non-convexity of problem (2) [12], thus enabling the employment of gradient-based approaches. However, the primal and dual problems become equivalent solely in certain scenarios such as convex problems, as illustrated below

$$\begin{aligned} \max_{\pi \in \Pi} \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)] - \beta D_{\text{KL}}(\pi(\cdot | x) \| \pi_{\text{ref}}(\cdot | x))] \\ \text{s.t. } \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [g_i(x, y)]] \geq 0, \quad i = 1, \dots, m. \end{aligned} \quad (5)$$

The previous problem is convex as the expectation is linear and the KL divergence is convex in the distribution $\pi(\cdot | x)$ over responses. Yet, (2) is generally a non-convex optimization problem, primarily due to factors such as the nonlinearities introduced by the Transformer blocks [13] in the LLM policy. Thus, it is significant to investigate the primal-dual gap between problems (2) and (4). To proceed, we rely on the following assumptions.

Assumption 1 (Boundness). *There exists a constant B such that, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $i \in \{1, 2, \dots, m\}$, $\max(|r(x, y)|, |g_i(x, y)|) \leq B$.*

Assumption 2 (Parametrization Gap). *There exists a constant ν such that for any policy $\pi \in \Pi$ and any $x \in \mathcal{X}$, there exists $\theta \in \Theta$ such that $\max(\|\pi_\theta(\cdot | x) - \pi(\cdot | x)\|_1, |D_{\text{KL}}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) - D_{\text{KL}}(\pi(\cdot | x) \| \pi_{\text{ref}}(\cdot | x))|) \leq \nu$.*

Assumption 3 (Strict Feasibility). *There exists $\theta \in \Theta$ and $\xi > 0$ such that $\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta(\cdot | x)} [g_i(x, y)]] \geq B\nu + \xi$ for all $i = 1, \dots, m$.*

Assumption 1 is common in the literature [14, 15, 16, 17]. In particular, in the case of finite state-action spaces such as language and SQL query space, it is always possible to design the reward to avoid the possibility of being unbounded. Assumption 2 states that the parameterization space Θ in (2) covers the policy space Π in (5) up to an error ν . Assumption 3 is also a standard assumption in the primal-dual theory [12], positing the existence of a strictly feasible parameter.

Theorem 1 (Primal-dual gap). *Let Assumptions 1, 2 and 3 hold. Denote by $\tilde{D}(\lambda)$ the dual function of (5) and define $\lambda_\nu^* = \text{argmin}_{\lambda \geq 0} \tilde{D}(\lambda) - B\nu\|\lambda\|_1$. Then, it holds that*

$$0 \leq D^* - P^* \leq (\beta + B + B\|\lambda_\nu^*\|_1) \nu. \quad (6)$$

Proof. Refer to Appendix A. □

Notably, Theorem 1 implies that the primal-dual gap between (2) and (4) is dominated by the parameterization gap ν . In fact, a zero parameterization gap guarantees a zero primal-dual gap despite the non-convexity of (2) [8, 18].

4 Experiments

In this section, we demonstrate the numerical performance of our ConstrainedSQL method. Our implementation (see Algorithm 1) is built on the well-known VERL library [19] and the GRPO algorithm [4]. The example of the prompt and the response of our ConstrainedSQL is provided in Appendix B, where the prompt includes the natural language question and the response encompasses a reasoning process within the `<think>...</think>` tag and the answer and corresponding SQL within the `<answer>...</answer>` tag.

Reward/Constraints. In this work, we adopt natural and interpretable reward and constraint designs for problem (2) with each represented by an indicator function. Specifically, $r = 1$ if the generated SQL yields the correct query result (matching the ground-truth) and $r = 0$ otherwise; $g_{\text{format}} = 1$ if the response from the LLM policy shows the correct format (see the example of response format in Appendix B) and $g_{\text{format}} = 0$ otherwise; $g_{\text{execution}} = 1$ if the generated SQL is executable and $g_{\text{execution}} = 0$ otherwise; $g_{\text{length}} = 1$ if the response is longer than a pre-designed threshold (300 in this work) and $g_{\text{length}} = 0$ otherwise; $g_{\text{answer}} = 1$ if the proportion of the answer in the generated response is within a pre-designed threshold (25% – 75% in this work) and $g_{\text{answer}} = 0$ otherwise; $g_{\text{SQL}} = 1$ if the generated SQL is longer than a pre-designed threshold (25% – in this work) of the answer and $g_{\text{SQL}} = 0$ otherwise. All corresponding constraint thresholds are set to be 95%. Then with the indicator-function designs for the reward and constraints, problem (2) aims to maximize the probability of generated SQL matching the ground-truth while each of the constraints is satisfied (e.g., the response format being correct) with probabilities at least 95%.

Table 1: Performance comparison (Execution Accuracy) of 3B models

Dataset	SQL-R1 (reproduced)	SQL-R1 (reported)	ConstrainedSQL
Spider (Dev)	79.3	78.1	82.0
Spider (Test)	79.7	78.9	81.3
BIRD (Dev)	51.3	54.6	55.0

Baselines. We consider SQL-R1 [6] as our main baseline as it is the SOTA RL-based Text2SQL approach. In addition, we compare with other SFT-based methods such as OmniSQL [10] that often consumes a larger number of samples. To ensure a fair comparison, all methods are built on Qwen2.5-Coder series models [20].

Benchmarks. The ConstrainedSQL, along with all other baselines, is evaluated on two widely-used Text2SQL benchmarks: Spider [21] and BIRD [22]. The Spider dataset consists of 10,181 natural language questions paired with 5,693 complex SQL queries across 200 databases and 138 domains. The BIRD dataset contains 12,751 Text2SQL pairs covering 95 databases across 37 specialized domains. We adopt the standard Execution Accuracy metric for both Spider and BIRD benchmarks, which measures the proportion of questions that yield outcomes consistent with their corresponding ground-truth queries across all query requests.

Table 2: Performance comparison (Execution Accuracy) of 7B models

Dataset	SQL-R1 (reproduced)	SQL-R1 (reported)	ConstrainedSQL
Spider (Dev)	83.3	84.5	85.2
Spider (Test)	83.2	86.1	86.7
BIRD (Dev)	60.9	63.1	61.9

Results. Our main results are presented in Tables 1 and 2. We compare our ConstrainedSQL approach with two versions of SQL-R1 method: reproduced (our reproduced results using the official implementation of SQL-R1) and reported (the results reported in the original SQL-R1 paper). Table 1 summarizes the experimental results for the 3B models, showing that our ConstrainedSQL method consistently outperforms the two SQL-R1 baselines across all benchmarks. The results for the 7B models are reported in Table 2. ConstrainedSQL continues to outperform both SQL-R1 baselines on the Spider (Dev) and Spider (Test) datasets. On the BIRD (Dev) dataset, ConstrainedSQL surpasses the SQL-R1 (reproduced) baseline and delivers performance comparable to SQL-R1 (reported).

Table 3: Execution Accuracy (%) of different Text2SQL methods on Spider and BIRD benchmark.

Text2SQL Method	Base Model	Spider (Dev)	Spider (Test)	BIRD (Dev)
Alpha-SQL [23]	Qwen2.5-Coder-7B	84.0	-	66.8
SQL-o1 [24]	Qwen2.5-Coder-7B	84.7	85.1	66.7
OmniSQL [10]	Qwen2.5-Coder-7B	85.5	88.9	66.1
DeepRetrieval [25]	Qwen2.5-Coder-7B	-	76.1	56.0
SQL-R1	Qwen2.5-Coder-7B	84.5	86.1	63.1
ConstrainedSQL (Ours)	Qwen2.5-Coder-7B	85.2	86.7	61.9

Meanwhile, it is worth noting that the SQL-R1 baselines rely on carefully crafted reward designs [6], whereas our approach simply adopts the natural and interpretable reward and constraint designs.

In Table 3, we also compare ConstrainedSQL with other SFT-based Text2SQL methods such as OmniSQL [10]. Notably, ConstrainedSQL achieves comparable performance to these SFT-based methods, despite their use of up to 2.5 million training samples, while we rely on only 5,000 samples. Figure 1 depicts the evolution of each metric (both reward and constraints) during the training of ConstrainedSQL, demonstrating substantial improvements and clear convergence for all metrics.

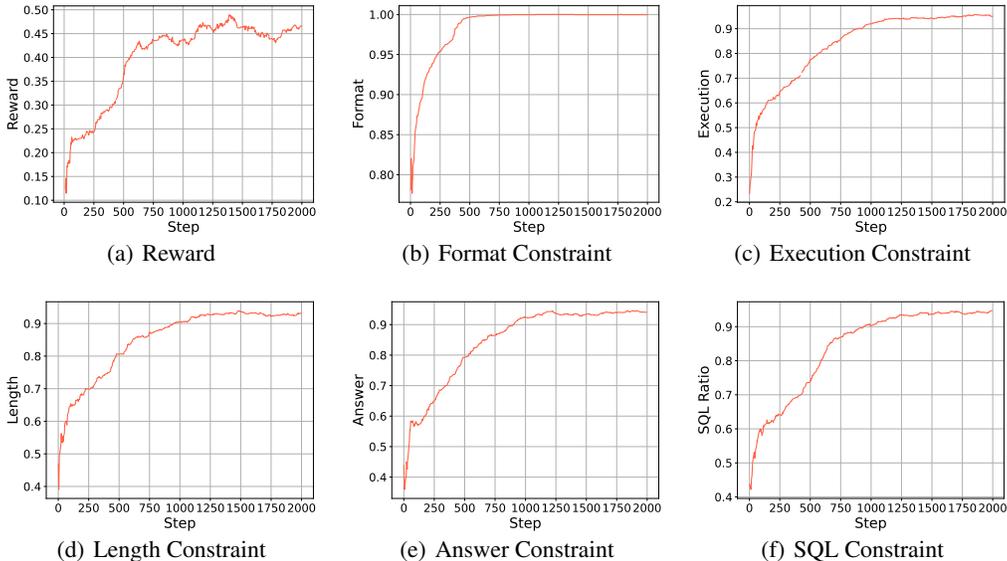


Figure 1: Training curves of the reward and constraints in ConstrainedSQL.

5 Conclusion

In this work, we introduce ConstrainedSQL that leverages natural and interpretable reward and constraint signals for training Text2SQL LLMs. By dynamically balancing trade-offs among multiple metrics, our approach tackles the issues such as reward hacking. We provide theoretical guarantees for our framework and demonstrate through extensive experiments on the standard Text2SQL benchmarks (Spider and BIRD) that it outperforms the current SOTA RL-trained LLMs. These results highlight the potential of constrained RL to improve both the reliability and interpretability of LLMs’ post-training.

Acknowledgments

This work was supported by IBM through the IBM-Rensselaer Future of Computing Research Collaboration.

References

- [1] Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuxin Zhang, Ju Fan, Guoliang Li, Nan Tang, and Yuyu Luo. A survey of nl2sql with large language models: Where are we, and where are we going? *arXiv preprint arXiv:2408.05109*, 2024.
- [2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [3] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [4] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [5] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- [6] Peixian Ma, Xialie Zhuang, Chengjin Xu, Xuhui Jiang, Ran Chen, and Jian Guo. Sql-r1: Training natural language to sql reasoning model by reinforcement learning. *arXiv preprint arXiv:2504.08600*, 2025.
- [7] Mohammadreza Pourreza, Shayan Talei, Ruoxi Sun, Xingchen Wan, Hailong Li, Azalia Mirhoseini, Amin Saberi, Sercan Arik, et al. Reasoning-sql: Reinforcement learning with sql tailored partial rewards for reasoning-enhanced text-to-sql. *arXiv preprint arXiv:2503.23157*, 2025.
- [8] Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.
- [9] Weiqin Chen, Dharmashankar Subramanian, and Santiago Paternain. Probabilistic constraint for safety-critical reinforcement learning. *IEEE Transactions on Automatic Control*, 69(10):6789–6804, 2024.
- [10] Haoyang Li, Shang Wu, Xiaokang Zhang, Xinmei Huang, Jing Zhang, Fuxin Jiang, Shuai Wang, Tiejing Zhang, Jianjun Chen, Rui Shi, et al. Omnisql: Synthesizing high-quality text-to-sql data at scale. *arXiv preprint arXiv:2503.02240*, 2025.
- [11] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.
- [12] Stephen P Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [14] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017.
- [15] Chen-Yu Wei, Mehdi Jafarnia Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, pages 10170–10180. PMLR, 2020.
- [16] Zihan Zhang, Simon Du, and Xiangyang Ji. Near optimal reward-free reinforcement learning. In *International Conference on Machine Learning*, pages 12402–12412. PMLR, 2021.

- [17] Weiqin Chen, Sandipan Mishra, and Santiago Paternain. Domain adaptation for offline reinforcement learning with limited samples. *arXiv preprint arXiv:2408.12136*, 2024.
- [18] Botong Zhang, Shuo Li, Ignacio Hounie, Osbert Bastani, Dongsheng Ding, and Alejandro Ribeiro. Alignment of large language models with constrained learning. *arXiv preprint arXiv:2505.19387*, 2025.
- [19] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.
- [20] Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.
- [21] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*, 2018.
- [22] Jinyang Li, Binyuan Hui, Ge Qu, Jiayi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357, 2023.
- [23] Masahiro Matsui, Takuto Sugisaki, Kensaku Okada, and Noboru Koshizuka. Alphasql: Open source software tool for automatic dependency resolution, parallelization and validation for sql and data. In *2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*, pages 38–45. IEEE, 2022.
- [24] Shuai Lyu, Haoran Luo, Ripeng Li, Zhonghong Ou, Jiangfeng Sun, Yang Qin, Xiaoran Shang, Meina Song, and Yifan Zhu. Sql-o1: A self-reward heuristic dynamic search method for text-to-sql. *arXiv preprint arXiv:2502.11741*, 2025.
- [25] Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025.

A Proof of Theorem 1

To prove Theorem 1, we rely on the following technical lemma.

Lemma 1. *Let Assumptions 1 and 2 hold. For any policy $\pi \in \Pi$, there exists $\theta \in \Theta$ such that the following inequalities hold*

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]]| \leq B\nu, \quad (7)$$

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [g_i(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [g_i(x, y)]]| \leq B\nu, \quad i = 1, \dots, m \quad (8)$$

Proof. We start by proving (7). For any policy $\pi \in \Pi$, there exists $\theta \in \Theta$ such that

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]]| \quad (9)$$

$$= |\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)] - \mathbb{E}_{y \sim \pi} [r(x, y)]]| \quad (10)$$

$$= \left| \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{y \in \mathcal{Y}} \pi_\theta(y | x) r(x, y) - \sum_{y \in \mathcal{Y}} \pi(y | x) r(x, y) \right] \right| \quad (11)$$

$$= \left| \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{y \in \mathcal{Y}} (\pi_\theta(y | x) - \pi(y | x)) r(x, y) \right] \right|. \quad (12)$$

Given that the absolute value function is convex, it follows from the Jensen's inequality that

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]]| \quad (13)$$

$$\leq \mathbb{E}_{x \sim \mathcal{D}} \left[\left| \sum_{y \in \mathcal{Y}} (\pi_\theta(y | x) - \pi(y | x)) r(x, y) \right| \right]. \quad (14)$$

By triangle inequality, we further obtain

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]]| \quad (15)$$

$$\leq \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{y \in \mathcal{Y}} |(\pi_\theta(y | x) - \pi(y | x))| \cdot |r(x, y)| \right] \quad (16)$$

$$\leq \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{y \in \mathcal{Y}} |(\pi_\theta(y | x) - \pi(y | x))| \cdot B \right], \quad (17)$$

where the last inequality follows from Assumption 1. Moreover, Assumption 2 implies that

$$\|\pi_\theta(\cdot | x) - \pi(\cdot | x)\|_1 \leq \nu. \quad (18)$$

Consequently, it holds by (18) and the definition of L_1 norm that

$$|\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi_\theta} [r(x, y)]] - \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]]| \leq \mathbb{E}_{x \sim \mathcal{D}} [\nu \cdot B] = B\nu. \quad (19)$$

The proof of (8) is identical to that of (7), as Assumption 1 indicates both $|r(x, y)| \leq B$ and $|g_i(x, y)| \leq B$, $i = 1, \dots, m$, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This completes the proof of Lemma 1. \square

Having introduced Lemma 1, we are in the stage of proving Theorem 1.

Theorem 1 (Primal-dual gap). Let Assumptions 1, 2 and 3 hold. Denote by $\tilde{D}(\lambda)$ the dual function of (5) and define $\lambda_\nu^* = \operatorname{argmin}_{\lambda \geq 0} \tilde{D}(\lambda) - B\nu\|\lambda\|_1$. Then, it holds that

$$0 \leq D^* - P^* \leq (\beta + B + B\|\lambda_\nu^*\|_1) \nu.$$

Proof. Notice that the left hand side inequality in (6) follows directly from the standard weak duality theorem [12]. We then focus on proving the right hand side inequality in (6). To do so, we rely on the following perturbed problem of (5), whose optimal solution is denoted by π_ν^*

$$\begin{aligned} \tilde{P}^*(\nu) &= \max_{\pi \in \Pi} \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [r(x, y)]] - \beta D_{\text{KL}}(\pi(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \\ \text{s.t. } &\mathbb{E}_{x \sim \mathcal{D}} [\mathbb{E}_{y \sim \pi} [g_i(x, y)]] \geq B\nu, \quad i = 1, \dots, m \end{aligned} \quad (20)$$

Assumption 3 indicates the strict feasibility of (20) due to the fact that $\pi_\theta \in \Pi$ for any $\theta \in \Theta$. Then strong duality holds for (20) [12], i.e.,

$$\tilde{P}^*(\nu) = \min_{\lambda \geq 0} \max_{\pi \in \Pi} L_\nu(\pi, \lambda), \quad (21)$$

where $L_\nu(\pi, \lambda)$ represents the Lagrangian of the perturbed problem (20), and can be re-written as

$$L_\nu(\pi, \lambda) = L(\pi, \lambda) - B\nu \|\lambda\|_1. \quad (22)$$

Denote by λ_ν^* the dual optimizer of (21). Since D^* represents the minimum of the dual function $D(\lambda)$, $\forall \lambda \geq 0$, it holds that

$$D^* \leq D(\lambda_\nu^*) = \max_{\theta \in \Theta} L(\pi_\theta, \lambda_\nu^*), \quad (23)$$

where the last equation follows from the definition of the dual function. Since $\pi_\theta \in \Pi$ for any $\theta \in \Theta$, we can further obtain

$$D^* \leq \max_{\theta \in \Theta} L(\pi_\theta, \lambda_\nu^*) \leq \max_{\pi \in \Pi} L(\pi, \lambda_\nu^*). \quad (24)$$

Substituting (22) into the previous inequalities yields

$$D^* \leq \max_{\pi \in \Pi} L_\nu(\pi, \lambda_\nu^*) + B\nu \|\lambda_\nu^*\|_1. \quad (25)$$

By strong duality of (20), the previous inequality reduces to

$$D^* \leq \tilde{P}^*(\nu) + B\nu \|\lambda_\nu^*\|_1 \quad (26)$$

$$= \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] - \beta D_{\text{KL}}(\pi_\nu^*(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] + B\nu \|\lambda_\nu^*\|_1, \quad (27)$$

where the last equation follows from the definition of the primal optimum. Adding and subtracting a P^* in (27) further yields

$$D^* \leq P^* + \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] - \beta D_{\text{KL}}(\pi_\nu^*(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] - P^* + B\nu \|\lambda_\nu^*\|_1. \quad (28)$$

Notably, it follows from Lemma 1 that there exists a parameter $\theta_{\lambda_\nu^*}$ such that

$$\left| \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [r(x, y)] \right] - \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] \right] \right| \leq B\nu, \quad (29)$$

$$\left| \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [g_i(x, y)] \right] - \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [g_i(x, y)] \right] \right| \leq B\nu, \quad i = 1, \dots, m \quad (30)$$

Therefore,

$$\mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [g_i(x, y)] \right] \geq \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [g_i(x, y)] \right] - B\nu \geq 0, \quad i = 1, \dots, m \quad (31)$$

where the previous inequalities hold by the fact that π_ν^* is a feasible solution to problem (20). Accordingly, $\theta_{\lambda_\nu^*}$ is feasible to the original problem (2) whose optimal value is P^* , thus

$$P^* \geq \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [r(x, y)] - \beta D_{\text{KL}}(\pi_{\theta_{\lambda_\nu^*}}(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right]. \quad (32)$$

Substituting (32) into (28) yields

$$\begin{aligned} D^* &\leq P^* + \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] - \beta D_{\text{KL}}(\pi_\nu^*(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] - \\ &\quad \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [r(x, y)] - \beta D_{\text{KL}}(\pi_{\theta_{\lambda_\nu^*}}(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] + B\nu \|\lambda_\nu^*\|_1 \end{aligned} \quad (33)$$

$$\begin{aligned} &= P^* + \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] \right] - \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [r(x, y)] \right] + \\ &\quad \mathbb{E}_{x \sim \mathcal{D}} \left[\beta D_{\text{KL}}(\pi_{\theta_{\lambda_\nu^*}}(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) - \beta D_{\text{KL}}(\pi_\nu^*(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] + B\nu \|\lambda_\nu^*\|_1 \end{aligned} \quad (34)$$

$$\begin{aligned} &\leq P^* + \left| \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\nu^*} [r(x, y)] \right] - \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_{\theta_{\lambda_\nu^*}}} [r(x, y)] \right] \right| \\ &\quad + \left| \mathbb{E}_{x \sim \mathcal{D}} \left[\beta D_{\text{KL}}(\pi_{\theta_{\lambda_\nu^*}}(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) - \beta D_{\text{KL}}(\pi_\nu^*(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right] \right| + B\nu \|\lambda_\nu^*\|_1 \end{aligned} \quad (35)$$

It then holds by (29) and Assumption 2 that

$$D^* \leq P^* + B\nu + \beta\nu + B\nu\|\lambda_\nu^*\|_1. \quad (36)$$

i.e.,

$$D^* - P^* \leq (\beta + B + B\|\lambda_\nu^*\|_1)\nu. \quad (37)$$

This completes the proof of Theorem 1.

□

B Example of the Prompt and Response

The example of the prompt and response in ConstrainedSQL are presented in Figure 2 and Figure 3, respectively.

Prompt

You are a helpful SQL expert assistant.
The assistant first thinks about how to write the SQL query by analyzing the question, database schema and external knowledge, then provides the final SQL query. The reasoning process and SQL query are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags respectively. The answer must contain the SQL query within `““sql . . . ““` tags.

Database Schema: {schema}

External Knowledge: {external_knowledge}

For example:
`<think>`
...
`</think>`
`<answer>`
...
`</answer>`

Question: {What is the unabbreviated mailing street address of the school with the highest FRPM count for K-12 students?}

Figure 2: The example of the prompt in ConstrainedSQL.

Example Response

<think>

To answer the question, we need to find the school with the highest FRPM count for K-12 students and then retrieve its unabbreviated mailing street address. Here are the steps to generate the SQL query:

1. **Identify the relevant table**: The 'frpm' table contains the FRPM count information for K-12 students.
2. **Sort the data by FRPM count**: We need to sort the data in descending order based on the 'FRPM Count (K-12)' column to find the school with the highest count.
3. **Retrieve the required information**: We need to select the 'MailStreet' column from the 'schools' table, which contains the unabbreviated mailing street address.
4. **Join the tables**: Since the 'frpm' table does not contain the mailing street address, we need to join it with the 'schools' table using the 'CDSCode' column, which is the primary key in both tables.
5. **Limit the result**: We only need the school with the highest FRPM count, so we use the 'LIMIT 1' clause to restrict the result to one row.

</think>

<answer>

```
““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
““
```

</answer>

Figure 3: The example of the response in ConstrainedSQL.