ANALYZING PREFERENCE OPTIMIZATION WITH SYNTHETIC CONTRASTIVE REASONING TRACES FOR MULTI-TABLE Q&A

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-table question answering requires models to retrieve relevant evidence, link schemas, and perform compositional reasoning across relational tables. Preference optimization over paired correct and incorrect reasoning traces is ideally suited for teaching LLMs such capabilities, but it is unclear which underlying mechanisms are most critical for this task: the training objective, or the properties of the contrastive reasoning traces. We study this question using synthetic contrastive reasoning traces for MMQA, consisting of validated positive traces and plausible negative traces generated by heterogeneous LLMs, and three open-weight LLMs (Qwen3-14B, Mistral-8B, and Llama-3.1-8B) evaluated on three table Q&A benchmarks, plus an out-of-distribution multi-table evaluation set built from BIRD. To investigate the role of optimization methods, we hold the preference pairs fixed and compare DPO, DPO with an NLL regularizer, and reference-free Contrastive Preference Optimization (CPO). We then further analyze the role of contrastive data diversity, quality, and source of the traces: whether heterogeneous generators for positive and negative traces strengthen the contrastive signal, how correct, faithful, and coherent the traces are under automated and human evaluation, and whether open-weight generators can replace proprietary ones. We find that CPO achieves absolute average improvements over Q&A supervised fine-tuning ranging from 10.7 to 19.3 percentage points, while DPO on the same pairs degrades two of the three models relative to positive-trace SFT and is only partially repaired by regularization. Removing the reference model improves every model over regularized DPO and is the larger of the two changes for all three; generating negatives with a different LLM than the positives gives a small but directionally consistent advantage; and an open-weight generator comes within 1–2 points of the proprietary one on every comparison. A case study and out-of-distribution transfer to three-table queries indicate that the negatives encode realistic multi-table failure modes, such as skipped constraint verification, that the trained models learn to reject.

1 INTRODUCTION

Large Language Models (LLMs) are increasingly used in settings where the knowledge needed to answer a question is not contained in the model parameters, but in external structured data. In many enterprise and scientific workflows, this information is stored in relational databases. A useful AI agent must therefore be able to retrieve evidence from tables, identify how schemas relate, and reason over joins, filters, comparisons, and aggregations. These requirements make multi-table question answering (Q&A) substantially harder than single-table reasoning. A model may need to identify the relevant tables from a long context, align primary and foreign keys across different schemas, and carry intermediate results through several reasoning hops. As the number of tables grows, contexts become longer, relevant evidence becomes easier to miss (Liu et al., 2024), and small mistakes in schema linking or constraint checking can lead to incorrect final answers. Although recent work has improved LLMs on single-table reasoning tasks (Wang et al., 2024; Lei et al., 2025), multi-table Q&A remains comparatively underdeveloped.

Reasoning traces are a natural source of supervision for this setting. Prior work on chain-of-thought reasoning shows that step-by-step rationales can improve multi-step problem solving (Wei et al., 2022),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

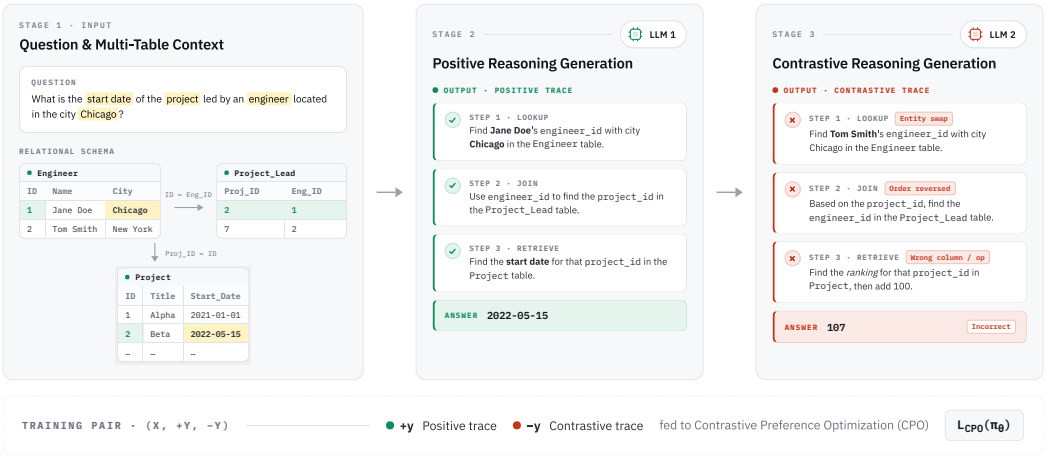

Figure 1: Illustration of our approach for generating positive & contrastive reasoning traces for multi-table Q&A

and table reasoning often requires exactly this kind of explicit decomposition. However, existing multi-table Q&A datasets generally provide questions and answers without annotated reasoning paths, and they do not provide paired examples of plausible but incorrect reasoning that can be used for preference optimization.

In this work, we use multi-table Q&A as a testbed to analyze what makes preference optimization on contrastive reasoning traces effective. We produce synthetic contrastive reasoning traces for MMQA (Wu et al., 2025a), using strong foundation models to generate positive traces that derive the gold answer from the provided tables and negative traces that are fluent and plausible but contain factual or logical errors, deliberately choosing heterogeneous LLMs for the two roles. Holding these preference pairs fixed, we then vary the training objective and the properties of the data when fine-tuning three open-weight LLMs in order to analyze the relative importance of the optimization and data for this task.

Optimization. Given identical $(+y, -y)$ pairs, we investigate **(RQ1)** whether adding regularization to DPO improves its performance for this task (Section 4.2); and **(RQ2)** what the effect is of removing the reference model from DPO, i.e., CPO (Section 4.3). **Data.** Given a fixed objective, we investigate **(RQ3)** how diverse the contrastive traces are and whether the choice of negative generator matters (Sections 5.1 and 5.2); **(RQ4)** how correct, faithful, and coherent the traces are (Section 5.3); and **(RQ5)** whether open-weight LLMs are a viable alternative to proprietary models for trace generation (Section 5.4). Finally, we investigate **(RQ6)** what the negatives teach the model (Section 6).

Our experiments show that this contrastive supervision substantially improves multi-table Q&A, and that the choice of objective is decisive. Across Qwen3-14B, Mistral-8B, and Llama-3.1-8B, CPO achieves absolute average improvements over Q&A supervised fine-tuning ranging from 10.7 to 19.3 percentage points. On MMQA, the largest absolute gain over Q&A supervised fine-tuning is 21 percentage points. DPO trained on the same preference pairs degrades two of the three models relative to positive-trace SFT; adding an NLL regularizer recovers part of the loss, while removing the reference model improves every model and is the larger of the two changes for all three. On the data side, the generated traces are diverse and largely reliable under automated and human evaluation, generating negatives with a different LLM than the positives gives a small but consistent advantage, and an open-weight generator comes within 1–2 points of the proprietary one on every comparison. We will make our code and dataset publicly available.

2 Related Work

Table question answering. Table question answering requires models to understand semi-structured data and perform reasoning over it. Early approaches such as TaBERT (Yin et al., 2020) pretrain transformers jointly on textual and tabular data. Spider (Yu et al., 2018) evaluates cross-domain

text-to-SQL parsing over multi-table databases, while TableBench (Wu et al., 2025b) focuses on single-table reasoning through direct answer generation. In the multi-table setting, MMQA (Wu et al., 2025a) introduces a multi-hop benchmark that requires identifying primary and foreign key relationships across tables, MultiTabQA (Pal et al., 2023) generates tabular answers over multiple related tables, TQA-Bench (Qiu et al., 2024) scales context length to 64K tokens, and MMTU (Xing et al., 2025) offers a massive multi-task benchmark for table understanding and reasoning. Most of these methods rely on implicit reasoning within the model and do not produce explicit multi-step reasoning traces.

Reasoning with LLMs for tables. Chain-of-Thought prompting (Wei et al., 2022) elicits step-by-step reasoning from LLMs, and Chain-of-Table (Wang et al., 2024) uses table operations as intermediate reasoning steps. Table-specialized LLMs include TableLlama (Zhang et al., 2024), fine-tuned on diverse table tasks; TableGPT2 (Su et al., 2024), which integrates tabular data into multimodal models; and Table-R1 (Yang et al., 2025b), which explores inference-time scaling for table reasoning. These methods target single-table reasoning, and synthetic reasoning traces for multi-table Q&A remain underexplored.

Preference optimization for reasoning. Direct Preference Optimization (DPO) (Rafailov et al., 2023) trains language models to satisfy preferences without an explicit reward model, but it requires a reference model and treats all preference pairs equally regardless of the quality gap between responses (Amini et al., 2024; Wu et al., 2024). A family of objectives modifies DPO by removing the reference model or adding a likelihood term: CPO (Xu et al., 2024) drops the reference model and adds a behavior cloning regularizer; SimPO (Meng et al., 2024) uses a length-normalized, reference-free margin; ORPO (Hong et al., 2024) adds an odds-ratio penalty to the SFT loss; IPO (Azar et al., 2024) replaces the logistic loss with a squared loss on the preference margin; KTO (Ethayarajh et al., 2024) removes the need for paired data; and RPO (Pang et al., 2024) adds an NLL term on the chosen response. Plain DPO has been reported to degrade long-form and structured reasoning (Pang et al., 2024; Liu et al., 2025). Setlur et al. (2024) and Chia et al. (2023) show that training on negative synthetic traces improves reasoning. We do not propose a new objective; rather, we hold the preference data fixed to compare DPO, DPO+NLL, and CPO on long, structured multi-table reasoning traces, and we characterize the synthetic negatives themselves, which prior work treats as given.

3 Experimental Setup

Multi-table Q&A involves answering a question $[Question]$ given a set of relational tables $[Tables] = [Table_1, Table_2, \ldots, Table_N]$; the task requires identifying and linking relevant information across tables and performing compositional reasoning to produce the correct answer $[Answer]$. We detail the synthetic contrastive reasoning traces, training objectives, and evaluation suite that form the basis of our analysis; Figure 1 illustrates the trace generation pipeline.

3.1 Synthetic Positive Reasoning Traces

To help LLMs solve complex problems involving multiple tables, we generated step-by-step "reasoning traces" motivated from Chain-of-Thought (CoT) prompting (Wei et al., 2022; Wang et al., 2024) instead of just using simple Q&A pairs. We primarily used GPT-4o (Hurst et al., 2024) to create these traces because it is capable of understanding structured data and following logic, but demonstrate how our approach is also effective with open LLMs (see Section 5.4). The generation prompt (Table 11) instructs the model to reason only over the provided tables, to name the table used at every step, to format the steps as a markdown table, and to use short identifiers rather than copying rows; Appendix B gives the rationale for each rule.

Validation and filtering. We generated candidate reasoning traces for the question and table pairs in our dataset $\mathcal{T}$ using GPT-4o. A trace is defined as "positive" $(+y)$ if and only if it leads to the correct ground truth answer $[Answer]$. Because valid answers often differ in formatting or phrasing (e.g., aliases), exact string matching is insufficient; following the evaluation prompt proposed in MMQA (Wu et al., 2025a), we use Gemini 2.0 Flash as a judge of semantic equivalence between the generated and gold answers.

As detailed in Table 12 of Appendix C, the evaluator is prompted to return a score of **1** if the answers are semantically equivalent and **0** otherwise. Any trace that receives a score of 0 or fails to adhere to the required markdown formatting is strictly discarded. Thus, our pipeline is specifically designed to mitigate any potential errors by GPT-4o in trace generation; 58.3% of generated traces pass this filter. We further validate the reliability of this generate-and-filter strategy through human calibration of the judge (Appendix F), which shows that the filter is conservative (it rejects more correct traces than it lets bad ones through), so the resulting training set is biased toward valid reasoning paths.

3.2 Synthetic Contrastive Reasoning Traces

To enable contrastive learning, we require pairs of correct $(+y)$ and incorrect $(-y)$ reasoning paths. For every validated positive trace generated in the previous step, we synthesized a corresponding negative trace that leads to an incorrect answer. We used Gemini 2.0 Flash (Google) to generate these negative traces. Using the prompt template shown in Table 13 of Appendix C, we provided the model with the correct trace and the original question. The model was instructed to generate a "contrastive" trace that is plausible but factually or logically flawed. The prompt (Table 13) requires the model to alter numbers, logic, or entities at every step so that the conclusion is wrong, allows it to reorder steps, asks for a trace that remains fluent and plausible rather than nonsensical, and requires the markdown format of the positive trace; Appendix B details these constraints. We deliberately use a different model for negative traces (Gemini 2.0 Flash) than for positive traces (GPT-4o); Section 5.2 analyzes this choice. We perturb the validated positive trace rather than sampling failures from the student model. This yields exactly one negative per positive, keeps each negative matched in format and length to its positive (median step difference 0, Section 5.1), so that the two traces differ in reasoning content rather than presentation, and makes the data independent of any particular student.

3.3 Training Data

We used the MMQA two-table dataset (Wu et al., 2025a) as a basis for generating our synthetic reasoning traces, which contains 2591 examples of two-table Q&A pairs. We constructed an evaluation set $\mathcal{E}$ of size 355 and a training set $\mathcal{T}$ of size 1920. A few examples were excluded due to context length limitations (128k tokens). The positive and contrastive traces $(+y, -y)$ were generated over the training set $\mathcal{T}$. All models were trained on this dataset $(+y, -y)$.

3.4 Training Objectives

Given a dataset $\mathcal{D} = \{x_i, +y_i, -y_i\}_{i=1}^{N}$, x is the input (for table Q&A this is the tables and the question), $+y$ is the positive trace, and $-y$ is the negative trace. DPO (Rafailov et al., 2023) trains a policy π_θ using the loss

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = \\ - \mathbb{E}_{(x,+y,-y)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(+y|x)}{\pi_{ref}(+y|x)} - \beta \log \frac{\pi_\theta(-y|x)}{\pi_{ref}(-y|x)} \right) \right] \quad (1)$$

Here, π_{ref} is a pre-trained reference model. DPO has two drawbacks. It must load and run both the policy and the reference model at every step, and it treats all pairs the same regardless of how different the two responses are (Amini et al., 2024; Furuta et al., 2024; Wu et al., 2024), so the model does not learn the real preference gap and can drift from the preferred data distribution. Both matter for multi-table Q&A: reasoning chains are long, negative traces share the format, step structure, and much of the wording of their positive traces (median embedding cosine similarity 0.85, Section 5.1), and DPO has been shown to fail or even degrade performance on structured reasoning tasks (Liu et al., 2025).

CPO (Xu et al., 2024) helps with both problems. It removes memory and speed issues by setting π_{ref} to a uniform prior U, so only one model is needed. It also adds a behavior cloning

(BC) regularizer (Hejna et al., 2023) to keep π_θ close to the preferred data. The CPO loss is

$$\begin{aligned}&\mathcal{L}_{CPO}(\pi_\theta) = \\ &- \mathbb{E}_{(x,+y,-y)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\pi_\theta(+y|x) - \beta\log\pi_\theta(-y|x)\right)\right] \\ &- \mathbb{E}_{(x,+y)\sim\mathcal{D}}\left[\log(\pi_\theta(+y|x))\right]\end{aligned} \tag{2}$$

These properties make CPO an attractive objective for multi-table Q&A. However, CPO differs from DPO in two ways at once (no reference model, and an added BC regularizer), so its advantage cannot be attributed to either change alone; Section 4 separates the two.

DPO+NLL. Pure DPO optimizes only the *relative* margin between $+y$ and $-y$: the margin can grow while the likelihood of both traces falls, degrading the model's ability to generate valid traces, a failure mode documented for long reasoning chains (Pang et al., 2024; Liu et al., 2025). DPO+NLL (RPO; Pang et al., 2024), used in Llama-3 post-training (Grattafiori et al., 2024), adds the same BC regularizer as CPO on the positive trace, with weight $\alpha = 1$ (untuned), while retaining π_{ref} (loss in Appendix D). It is the natural intermediate point between DPO and CPO, so comparing DPO with DPO+NLL isolates the regularizer (RQ1) and comparing DPO+NLL with CPO isolates the reference model (RQ2). The fourth cell, a reference-free objective without the NLL term, is not part of our study (Appendix A).

3.5 Models, Training Protocol, and Evaluation

We performed a comparative analysis of five fine-tuning strategies across three evaluation datasets and three foundational models (Meta-Llama-3.1-8B, Qwen3-14B, and Mistral-8B[1]). The strategies evaluated were: (1) Q&A SFT (standard Question-Answer pairs), (2) Trace SFT (using positive reasoning traces), (3) DPO, (4) DPO+NLL, and (5) CPO, the latter three utilizing the same contrastive reasoning preferences $(+y, -y)$ and the same SFT checkpoint. All experiments were conducted on a single NVIDIA H100 GPU. The evaluation method is the same as discussed in Section 3.1. Because the answer judge has high positive-class precision but lower recall on our human audit (Appendix F), reported accuracies are conservative. Refer to Appendix D for comprehensive implementation details. We evaluate on three datasets: the held-out MMQA evaluation set $\mathcal{E}$, MMTU (Xing et al., 2025), and TableBench (Wu et al., 2025b). In addition, we construct a multi-table Q&A evaluation set from the BIRD benchmark (Li et al., 2024) (Appendix E) and use it as an out-of-distribution testbed (Section 6.1). Dataset statistics are provided in Table 4.

4 Optimization: Regularization and Reference Model

We first hold the contrastive data fixed and vary the objective (Table 1), then read off the effect of regularization (RQ1) and of the reference model (RQ2).

4.1 Main Results

Table 1 reports results across the three base models and three evaluation datasets, together with Table-R1 (Yang et al., 2025b), a recent tabular foundation model, as an external reference point. All three base LLMs reach similar accuracy when fine-tuned only on the original Q&A pairs from the MMQA training set. Training on positive reasoning traces generated by GPT-4o improves over Q&A pairs in nearly all cases, often with the largest gains out of domain: Qwen3-14B-Base gains 3 points on the in-domain MMQA test set but 15 points on MMTU.

Compared to positive trace SFT, DPO[2] produces inconsistent results across models. Qwen3-14B matches Trace SFT on MMTU and improves on TableBench, but Mistral-8B falls below even Q&A-only SFT on MMQA and MMTU, and Llama-3.1-8B loses accuracy relative to Trace SFT on MMQA and MMTU. We return to the causes of this behavior in Sections 4.2 and 4.3.

In contrast, CPO improves over both SFT variants for every model and dataset, with the highest accuracy in every column for Mistral-8B and Llama-3.1-8B and for Qwen3-14B on all datasets but

[1] Mistral-8B refers to Ministral-8B-Instruct-2410 throughout.

[2] DPO† uses the same configuration and SFT checkpoint as CPO.

Model	Method	MMQA	MMTU	TableBench	Avg.
Qwen3-14B-Base	Q&A SFT	29.0	29.0	20.0	26.0
	Trace SFT	32.0	44.0	23.0	33.0
	DPO†	33.2	43.9	29.9	35.7
	DPO+NLL†	33.5	42.9	**31.2**	35.9
	CPO	**50.0**	**56.0**	30.0	**45.3**
Mistral-8B-Instruct	Q&A SFT	30.0	30.0	17.0	25.7
	Trace SFT	32.0	33.0	18.0	27.7
	DPO†	20.8	27.0	18.2	22.0
	DPO+NLL†	28.2	36.5	21.7	28.8
	CPO	**43.0**	**46.0**	**22.0**	**37.0**
Llama-3.1-8B-Instruct	Q&A SFT	28.0	31.0	18.0	25.7
	Trace SFT	38.0	40.0	19.0	32.3
	DPO†	28.5	35.7	19.1	27.8
	DPO+NLL†	33.5	37.8	22.5	31.3
	CPO	**42.0**	**43.0**	**24.0**	**36.3**
Table-R1		33.6	51.0	30.0	38.2

Table 1: Main results across three base models and three evaluation datasets. Accuracy (%) on each evaluation set; Avg. is the mean over the three. CPO, DPO, and DPO+NLL use our synthetic contrastive reasoning traces. Each entry is a single training run; 95% binomial confidence intervals are at most ± 5.2 (MMQA, n=355), ± 2.3 (MMTU, n=1793), and ± 3.3 (TableBench, n=886) points. Out-of-distribution results on BIRD are reported in Table 3. †Same SFT checkpoint and LoRA configuration as CPO.

TableBench, where DPO+NLL is marginally higher (31.2 vs. 30.0, within noise). On MMQA, CPO gains 13–21 points over Q&A SFT for all three models, whereas Trace SFT gains 2–10, indicating that the CPO stage, which exposes the model to incorrect reasoning traces, provides a stronger learning signal than positive traces alone. For context, GPT-4o produced a judged-correct trace for 58.3% of the MMQA training questions (Section 3.1), so the best 14B student approaches its teacher's accuracy on this task.

Table-R1 was not fine-tuned on MMQA, so the comparison with it is not like-for-like. Qwen3-14B with CPO exceeds it on MMQA and MMTU and matches it on TableBench; Mistral-8B and Llama-3.1-8B exceed it on the multi-table MMQA set but not on MMTU and TableBench, where Table-R1 is strong.

4.2 Does Regularization Repair DPO? (RQ1)

DPO+NLL adds the BC regularizer of CPO to DPO while retaining the reference model (Section 3.4). Averaged over the three evaluation sets (last column of Table 1), the regularizer helps most where DPO degraded most: the average rises from 22.0% to 28.8% for Mistral-8B and from 27.8% to 31.3% for Llama-3.1-8B, recovering part of the MMQA accuracy that pure DPO had lost. For Qwen3-14B, where DPO did not degrade, the regularizer has no measurable effect (35.7% vs. 35.9%). DPO+NLL nevertheless stays close to Trace SFT (2.9 and 1.1 points above it for Qwen3-14B and Mistral-8B, and 1.0 point below it for Llama-3.1-8B) and well below CPO. Anchoring the policy to the positive traces therefore prevents the likelihood collapse described in Section 3.4, but it does not by itself produce the gains of contrastive training.

4.3 What Does Removing the Reference Model Do? (RQ2)

CPO differs from DPO+NLL only in replacing π_{ref} with a uniform prior, so the gap between the two rows of Table 1 isolates the reference model. Averaged over the three evaluation sets, CPO exceeds DPO+NLL by 9.4 points for Qwen3-14B (45.3% vs. 35.9%), 8.2 points for Mistral-8B (37.0% vs. 28.8%), and 5.0 points for Llama-3.1-8B (36.3% vs. 31.3%); the only per-dataset exception is TableBench for Qwen3-14B, where the two are within noise. Unlike the regularizer, whose effect ranges from negligible to large depending on whether DPO had collapsed (0.2, 6.8, and 3.5 points, respectively), removing the reference model improves every model and is the larger of the two changes for all three, although by a wide margin only for Qwen3-14B. Because DPO+NLL uses the identical NLL term, LoRA configuration, initialization, and schedule as CPO, the gap between them cannot be explained by additional likelihood training on the positive traces alone.

Eval.	Base model	Generator (positive / negative)		
		Qwen-3 30B / Qwen-3 30B	GPT-4o / GPT-4o	GPT-4o / Gemini 2.0
MMQA	Qwen3-14B	45	46	50
	Mistral-8B	40	42	43
	Llama-3.1-8B	40	42	42
MMTU	Qwen3-14B	54	55	56
	Mistral-8B	44	45	46
	Llama-3.1-8B	42	43	43

Table 2: CPO accuracy (%) by trace-generator configuration; the last column is our default. 95% confidence intervals are about ± 5 points on MMQA and ± 2.3 on MMTU.

This pattern is consistent with the argument in Section 3.4 that the reference-normalized margin provides a weak signal when negatives closely resemble their positives in form, with two caveats. β (0.1 for all objectives) is not directly comparable across the objectives: in DPO it scales a reference-normalized margin, in CPO a raw log-likelihood difference summed over roughly 60 tokens. And negatives are longer than positives (median 68 vs. 60 tokens, Table 7), which biases the raw CPO margin toward the positive trace independently of content. A β sweep, a length-normalized variant, and the reference-free objective without the NLL term would be needed to attribute the gap to the reference model alone.

5 Data: Diversity, Quality, and Source of the Traces

We now hold the objective fixed (CPO) and analyze the traces themselves: their diversity (Section 5.1), the choice of negative generator (Section 5.2), their quality (Section 5.3), and open-weight versus proprietary generators (Section 5.4). These analyses characterize the data; establishing causal effects on accuracy would require training on controlled subsets, which we have not done.

5.1 How Diverse and Contrastive Are the Traces? (RQ3)

We characterize the traces along three dimensions: surface diversity, structural diversity, and pairwise contrastiveness (Tables 7 and 8, Appendix G). Both positive and negative traces are almost entirely unique (99.7% and 100%). Negative traces are lexically more diverse than positive ones (vocabulary 8,064 vs. 5,799 tokens; distinct-4 0.736 vs. 0.590), consistent with their being produced by a different generator. Classifying each step into operation types (JOIN, FILTER, AGGREGATE, RETRIEVE) by keyword matching, negatives also show higher structural diversity (40.5% vs. 26.9% unique operation sequences) while preserving the same reasoning depth (median of 5 steps), so they are not simply truncated or shallow. Within pairs, the median token-level Jaccard similarity is 0.47 and the median sentence-embedding cosine similarity is 0.85; the median step difference is 0, and 75% of pairs differ by at most one step. Negatives therefore remain topically anchored to the question while differing in roughly half their vocabulary, as expected from the instruction to alter every step. Whether this balance is optimal is untested; we have not trained on subsets of differing similarity. A further 4.4% of pairs share the same final answer despite different traces; we retain them (Appendix G).

5.2 Does the Choice of Negative Generator Matter?

Table 2 compares three generator configurations for CPO training: GPT-4o for both roles, our default of GPT-4o positives with Gemini 2.0 Flash negatives, and the open-weight Qwen-3 30B for both roles. Using Gemini rather than GPT-4o for the negatives raises MMQA accuracy by 4 points for Qwen3-14B and 1 point for Mistral-8B, with no change for Llama-3.1-8B; on MMTU the differences are 1, 1, and 0 points. The effect is directionally consistent, since the heterogeneous configuration is never worse and is better in four of six comparisons, but it is small: only the Qwen3-14B MMQA difference approaches the confidence interval, so we regard it as a modest advantage rather than an established effect. Our working hypothesis for the direction of the effect is that a distinct negative generator increases the separation between correct and incorrect reasoning distributions, in line with the higher lexical and structural diversity of the negatives (Section 5.1). Testing it would require a condition with Gemini in both roles, since the present comparison changes the generator's identity together with its heterogeneity.

5.3 How Correct, Faithful, and Coherent Are the Traces? (RQ4)

We assess trace quality using GPT-5-mini as an LLM-as-judge, evaluating each contrastive pair along multiple dimensions.

Evaluation Criteria. The judge scores each trace for **correctness** of the final answer given the tables, **faithfulness** of its claims to table evidence, and **coherence** of the reasoning (each 0–5), flags **hallucination** (whether any step contains unsupported claims), and rates the **contrastiveness** (0–5) of each pair, i.e., whether the rejected trace is an effective negative example.

Results. Table 6 (Appendix G) shows that chosen traces achieve high correctness (4.71/5) and minimal hallucination (3.1%), while rejected traces show substantially lower correctness (0.92) and higher hallucination (46.1%). The judge prefers chosen traces in 97.2% of pairs, with a mean contrastiveness score of 4.03, confirming our pipeline produces well-separated and plausible contrastive pairs.

Human calibration of the trace quality judge. Our pipeline uses three LLM judges: the Gemini answer evaluator (Section 3.1), the GPT-5 semantic consistency judge for BIRD (Appendix E), and the trace quality judge above; each was audited against human annotations, the first two in Appendix F. For the trace quality judge we annotated 30 pairs on the same three dimensions. Human and LLM mean scores agree closely for both positive and negative traces (Table 9, Appendix G), supporting its use at scale; we compare means only and did not compute category-level agreement statistics.

5.4 Can Open-Weight Generators Replace Proprietary Ones? (RQ5)

Using the open-weight Qwen-3 30B for both roles yields accuracy within 1–2 points of GPT-4o for both roles on every base model and both datasets (Table 2). GPT-4o is ahead in all six comparisons, so the proprietary generator likely holds a small edge, but every difference is inside the confidence interval. An open-weight generator is therefore a viable alternative for producing contrastive traces on this task, at a small and statistically unresolved cost in accuracy.

6 What Do the Negatives Teach? (RQ6)

We hypothesize that models benefit not only from seeing correct multi-table reasoning paths but also from learning to reject plausible errors such as missed constraints, incorrect joins, and unsupported aggregations. Two observations are consistent with this hypothesis; none establishes it causally.

Negatives are plausible failures, not noise. Despite their errors, rejected traces maintain moderate coherence (mean 3.40, Table 6) and the same reasoning depth as positive traces (Table 7), remaining largely readable rather than incoherent. This helps models learn to distinguish plausible but flawed reasoning from correct inference.

Case study: skipped constraint verification. Table 10 (Appendix G) shows a training pair for a question that filters products on two conditions, an attribute in `TABLE2` and a catalog level in `TABLE1`. The negative trace finds both items matching the attribute but never checks the level constraint and sums both prices, including an item that should have been excluded; the positive trace verifies the level of each candidate first. Failing to verify every constraint before aggregating is a common error pattern in the negatives, and training on such pairs penalizes exactly this shortcut.

6.1 Out-of-Distribution Transfer: BIRD and Three-Table Queries

On our full BIRD evaluation set (Appendix E), built from real-world databases in domains absent from MMQA, CPO is the best training strategy for every model, reaching 29.0%, 26.0%, and 23.0% for Qwen3-14B, Mistral-8B, and Llama-3.1-8B against at most 23.0%, 17.0%, and 18.0% for the two SFT baselines (Table 3); Table-R1 reaches 10.3%.

Our training data is drawn from the two-table subset of MMQA (Section 3.3), but the approach itself is not restricted to $N = 2$. To assess generalization, we evaluated all three base models on the three-table subsets of MMQA and our BIRD evaluation set. Table 3 reports these results. CPO

Dataset	Model	Q&A SFT	Trace SFT	CPO
MMQA (3-table)	Qwen3-14B	28.53	27.00	**40.88**
	Mistral-8B	21.47	24.85	**34.32**
	Llama-3.1-8B	18.97	28.84	**35.19**
BIRD (all)	Qwen3-14B	22.0	23.0	**29.0**
	Mistral-8B	17.0	12.0	**26.0**
	Llama-3.1-8B	16.0	18.0	**23.0**
BIRD (3-table)	Qwen3-14B	50.00	50.00	**56.25**
	Mistral-8B	31.25	18.75	**43.75**
	Llama-3.1-8B	31.25	43.75	**56.25**

Table 3: Out-of-distribution accuracy (%) on our full BIRD evaluation set (n=1024) and on the three-table subsets of MMQA and BIRD (Section 6.1). All models are trained on the two-table MMQA subset only. The BIRD three-table subset contains only 16 examples, so each example shifts accuracy by 6.25 points; those results are indicative only.

outperforms both Q&A SFT and Trace SFT on every model and on both datasets, indicating that reasoning patterns learned through CPO on two-table data transfer to harder three-table queries; the BIRD subset is very small, however, and only the MMQA differences should be read as more than indicative. On MMQA, absolute accuracies are lower than on the two-table evaluation set, which is expected as both the input context length and the number of reasoning hops increase with more tables; the 16-example BIRD subset is too small for such a comparison.

7 Conclusion

We used multi-table Q&A as a testbed to analyze what makes preference optimization on synthetic contrastive reasoning traces effective. On the optimization side, CPO consistently outperformed Q&A-only and positive-trace SFT across three open-weight LLMs, three evaluation datasets, and an out-of-distribution BIRD set, whereas DPO on the same pairs degraded two of the three models relative to positive-trace SFT; an NLL regularizer recovered part of the loss, and removing the reference model improved every model and contributed the larger share of CPO's advantage for all three. On the data side, the traces are diverse and largely faithful under automated and human evaluation, generating negatives with a different LLM than the positives gives a small but consistent advantage, and an open-weight generator comes within 1–2 points of the proprietary one. The gains are consistent with models learning not only correct multi-table reasoning paths but also to reject plausible errors such as missed constraints, incorrect joins, and unsupported aggregations.

Reproducibility statement

All prompts used for trace generation, answer validation, negative-trace generation, and BIRD semantic-consistency verification are given verbatim in Appendix C. LoRA configurations, training schedules, and hyperparameters for every stage and objective are listed in Appendix D, the construction of the BIRD evaluation set with per-stage sample counts in Appendix E, and the human calibration protocols in Appendix F. Code, the contrastive trace dataset, and the BIRD evaluation set will be released upon publication.

Ethics statement

This work uses publicly released academic benchmarks of structured table data that contain no personal or sensitive information. Human annotation was performed by the authors on synthetic model outputs; no external participants were recruited. We do not foresee societal risks specific to this work.

AI use statement

We used generative AI for writing assistance in preparing this manuscript, including formatting tables & figures and improving clarity of our original writing. AI assistants were also used to help with code implementation during the course of our work. Our use of AI did not impact the core methodology, scientific rigor, or originality of our research.

REFERENCES

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. Contrastive chain-of-thought prompting. *arXiv preprint arXiv:2311.09277*, 2023.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Hiroki Furuta, Kuang-Huei Lee, Shixiang Shane Gu, Yutaka Matsuo, Aleksandra Faust, Heiga Zen, and Izzeddin Gur. Geometric-averaged preference optimization for soft preference labels. *Advances in Neural Information Processing Systems*, 37:57076–57114, 2024.

Google. Gemini 2.0 flash. `https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Fangyu Lei, Jinxiang Meng, Yiming Huang, Tinghong Chen, Yun Zhang, Shizhu He, Jun Zhao, and Kang Liu. Reasoning-table: Exploring reinforcement learning for table reasoning. *arXiv preprint arXiv:2506.01710*, 2025.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi 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, 2024.

Hanbing Liu, Haoyang Li, Xiaokang Zhang, Ruotong Chen, Haiyong Xu, Tian Tian, Qi Qi, and Jing Zhang. Uncovering the impact of chain-of-thought reasoning for direct preference optimization: Lessons from text-to-sql. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21223–21261, 2025.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Meta. Meta llama 3.1 8b instruct. `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`, 2024.

MistralAI. Ministral-8b-instruct. https://huggingface.co/mistralai/Ministral-8B-Instruct-2410.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. Multitabqa: Generating tabular answers for multi-table question answering. *arXiv preprint arXiv:2305.12820*, 2023.

Richard Y Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

Zipeng Qiu, You Peng, Guangxin He, Binhang Yuan, and Chen Wang. Tqa-bench: Evaluating llms for multi-table question answering with scalable context and symbolic extension. *arXiv preprint arXiv:2411.19504*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37:43000–43031, 2024.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, et al. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*, 2024.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *International Conference on Learning Representations*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jian Wu, Linyi Yang, Dongyuan Li, Yuliang Ji, Manabu Okumura, and Yue Zhang. Mmqa: Evaluating llms with multi-table multi-hop complex questions. In *The Thirteenth International Conference on Learning Representations*, pp. 1, 2025a.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct Preference Optimization with Dynamic β. *Advances in Neural Information Processing Systems*, 37:129944–129966, 2024.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. Tablebench: A comprehensive and complex benchmark for table question answering. 39(24):25497–25506, 2025b.

Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and HV Jagadish. Mmtu: A massive multi-task table understanding and reasoning benchmark. *arXiv preprint arXiv:2506.05587*, 2025.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. Table-r1: Inference-time scaling for table reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20616–20635, 2025b.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

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.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. Tablellama: Towards open large generalist models for tables. pp. 6024–6044, 2024.

A LIMITATIONS

Our highest-quality contrastive traces are produced using GPT-4o for positive traces and Gemini 2.0 Flash for negative traces. Both are commercial APIs, which imposes cost and reproducibility constraints on dataset construction at larger scales. We mitigate this in Section 5.4 by showing that an open-weight generator (Qwen-3 30B) yields competitive performance.

Three stages of our pipeline (answer validation during trace filtering, semantic consistency verification during BIRD evaluation set construction, and automated dataset-quality evaluation) use LLM-as-judge. Our human studies (Section 5.3 and Appendix F) show that these judges are conservative and agree with human annotators on the majority of samples, but the audits are small ($n = 30$–57), and any residual judge bias propagates into both training data and evaluation.

Our positive trace filter (Section 3.1) accepts a candidate trace only if its final answer matches the ground truth. It does not verify the correctness of every intermediate reasoning step, so a trace whose intermediate steps are flawed but whose final answer is coincidentally correct can be retained. Our automated and human dataset-quality evaluations indicate that this failure mode is rare ($\geq 92\%$ of chosen traces score ≥ 4 on correctness, Table 6).

B PROMPT DESIGN AND NEGATIVE-TRACE CONSTRAINTS

Positive traces. A major challenge in generating synthetic reasoning traces for our task is mitigating hallucinations and ensuring that the model does not rely on outside knowledge when solving the problem; instead, it should only reason over the provided tables to produce the final answer. Therefore, we designed our prompt to encourage the model to focus on the provided input data, instructing it to act as an *"expert at multi-table question answering"*. Our full prompt (detailed in Table 11) includes three rules to ensure the traces are high-quality:

Explicit References: Every reasoning step must state exactly which table ($TABLE1$ or $TABLE2$) is being used. This forces the model to support its logic with evidence and prevents it from guessing.

Step-by-Step Structure: The output must be formatted as a markdown table (|Step |Output |). This breaks complicated questions down into simple steps, making the process easier for smaller models to learn later.

Concise Summaries: We instructed the model not to copy full rows of data. Instead, it must use short identifiers. This ensures the trace focuses on the logic (e.g., comparing numbers, adding values) rather than just copying text.

Negative traces. Key constraints in the prompt ensure the quality of the negative examples:

Active Error Induction: The model must alter numbers, logic, or entities at every step to force an incorrect conclusion, and may reorder steps.

Plausibility: The trace must remain fluent and believable. It should not be nonsensical, as the goal is to teach the student model to detect subtle reasoning errors rather than simple formatting mistakes.

Structural Consistency: The negative trace must follow the same markdown table format as the positive trace (steps may be reordered) to ensure a fair comparison.

C PROMPT

We provide complete details of our prompts in Tables 11, 12, and 13.

D EXPERIMENTAL DETAILS

All experiments were conducted on a single NVIDIA H100 80GB GPU.

Models and Training Stages We conducted five sets of fine-tuning experiments on three foundational models: Meta-Llama-3.1-8B-Instruct (Meta, 2024; Grattafiori et al., 2024), Qwen3-14B-Base (Yang et al., 2025a), and Mistral-8B-Instruct (MistralAI).

1. **Stage 1 (Q&A SFT):** The models were fine-tuned using Supervised Fine-Tuning (SFT) on standard Q&A pairs ($[Question]$, $[Tables]$, $[Answer]$) from the training set $\mathcal{T}$.
2. **Stage 2 (Trace SFT):** We fine-tuned the models on the question and synthetically generated positive reasoning traces ($[Question]$, $[Tables]$, $+y$).
3. **Stage 3 (Preference Optimization):** We trained CPO (Xu et al., 2024), DPO, or DPO+NLL on the contrastive reasoning traces $(+y, -y)$. This stage was initialized using the model weights from Stage 2 in all three cases.

Hyperparameters and Configuration We utilized Low-Rank Adaptation (LoRA) (Hu et al., 2022) for parameter-efficient fine-tuning across all stages, though with distinct configurations for SFT and CPO to balance plasticity and stability.

- **SFT Configuration (Stage 1 & 2):** To facilitate the learning of complex reasoning structures, we used a LoRA configuration with $r = 1056$ and $\alpha = 1056$. We targeted all linear projection layers, including attention (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and MLP modules (`gate_proj`, `up_proj`, `down_proj`). Training was performed for 10 epochs with a learning rate of $2\text{e}{-5}$, a cosine learning rate scheduler, 100 warmup steps, and an effective batch size of 8.
- **CPO Configuration (Stage 3):** For preference alignment, we adopted a more constrained LoRA configuration with $r = 256$ and $\alpha = 256$, targeting only the attention modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`) while freezing the MLP layers. The CPO training ran for 2 epochs with a learning rate of $2\text{e}{-5}$, a cosine scheduler, 100 warmup steps, and an effective batch size of 8. We use the sigmoid loss with $\beta = 0.1$ and weight the NLL (BC) term 1.0 relative to the preference loss, the same settings as DPO and DPO+NLL.
- **DPO Configuration:** We train DPO using the same $(+y, -y)$ contrastive pairs, LoRA configuration, schedule, and effective batch size as CPO, initialized from the same SFT checkpoint, with sigmoid loss and $\beta = 0.1$.
- **DPO+NLL Configuration:** Identical to DPO, with an added NLL loss on the chosen response weighted $\alpha = 1$ relative to the preference loss (loss weights $1.0/1.0$), without tuning. The loss is

$$\begin{aligned}&\mathcal{L}_{DPO+NLL}(\pi_\theta; \pi_{ref}) = \\ &-\mathbb{E}_{(x,+y,-y)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(+y|x)}{\pi_{ref}(+y|x)} - \beta\log\frac{\pi_\theta(-y|x)}{\pi_{ref}(-y|x)}\right)\right] \\ &-\alpha\,\mathbb{E}_{(x,+y)\sim\mathcal{D}}\left[\log(\pi_\theta(+y|x))\right]\end{aligned} \tag{3}$$

  where the first term is the DPO objective (Eq. 1) and the second is the NLL (BC) loss on $+y$.

Evaluation Datasets We evaluated on four datasets: MMQA evaluation set $\mathcal{E}$, MMTU (Xing et al., 2025), TableBench (Wu et al., 2025b), and our multi-table Q&A evaluation set constructed from the BIRD benchmark (Li et al., 2024). We provide the dataset statistics in Table 4.

License details Licenses for datasets and models utilized in our study are as follows: MMQA (CC BY 4.0), MMTU (CC BY 4.0), TableBench (Apache 2.0), BIRD (CC BY-SA 4.0). Models: Qwen3-14B-Base (Apache 2.0), Ministral-8B-Instruct-2410 (Mistral Research License), Llama-3.1-8B-Instruct (Llama 3.1 Community License). All assets are used for research purposes only.

E BIRD Evaluation Set Construction

To further evaluate model generalization across diverse domains and data sources, we construct an additional evaluation set derived from the BIRD benchmark (Li et al., 2024), a large-scale text-to-SQL

Dataset	Total Tables	Avg Cols	Avg Rows	Total Samples
MMQA	960	5.91	522.64	355
MMTU	1793	5.50	16.43	1793
BIRD	2213	6.98	588.40	1024
TableBench*	886	6.68	16.71	886

Table 4: Dataset Statistics used in our evaluation.
**TableBench contains single-table queries. For evaluation, the second table input is left empty.*

Stage	Samples
BIRD train (original)	9,428
Multi-table filtering (2-3 tables)	6,826
Evidence extraction with size constraints	2,258
Semantic consistency verification	1,024

Table 5: BIRD evaluation set construction pipeline. Each stage progressively filters samples to ensure multi-table structure, complete evidence, and semantic consistency.

dataset containing 12,751 question-SQL pairs across 95 databases spanning 37 professional domains. Unlike MMQA (which originates from Spider with Wikipedia-based tables), BIRD uses real-world databases sourced from Kaggle with authentic schema complexity and diverse domain coverage, serving as a challenging out-of-distribution testbed for multi-table reasoning. We employed three filtering stages to ensure data quality, introducing two key contributions: (1) *full-table evidence extraction* to guarantee answer derivability, and (2) *automated semantic consistency verification* via LLM-as-judge to filter question-SQL misalignments. Table 5 summarizes the filtering pipeline and sample counts at each stage.

Stage 1: Multi-Table Filtering. From the BIRD training split (9,428 samples), we select queries involving exactly 2 or 3 tables based on the number of JOIN clauses in each SQL query. This yields 6,826 candidates (5,300 two-table and 1,526 three-table), aligning with the multi-hop reasoning setting evaluated on MMQA.

Stage 2: Evidence Extraction with Size Constraints. A key challenge in table Q&A is ensuring that provided evidence is complete: truncated tables may exclude rows necessary to derive the answer, creating unanswerable samples. Unlike prior work that uses fixed row limits (e.g., top-k rows), we extract *full table contents* for each sample, applying size thresholds only to exclude prohibitively large tables (maximum 5k rows per table and 120k total cells per sample). This ensures that any sample in our dataset is answerable from the provided table evidence alone. We execute each gold SQL query on the source database and exclude samples that return empty results. After this stage, 2,258 samples remain with complete, self-contained table evidence.

Stage 3: Semantic Consistency Verification. Text-to-SQL datasets may contain misalignments between natural language questions and their SQL annotations (Li et al., 2024). We employ GPT-5 as a semantic judge to verify that each SQL query exactly captures the question's intent. The judge receives the question, SQL, optional evidence hints from BIRD, and a schema summary (column names and key constraints), outputting "Yes" only when confident of semantic equivalence. The full prompt template is provided in Table 14. Of 2,258 samples, 1,024 (45.4%) pass this verification, and these form our final BIRD evaluation set.

Semantic consistency verification prompt. Table 14 presents the prompt template used for semantic consistency verification during BIRD evaluation set construction. The judge receives the natural language question, the gold SQL query, optional evidence hints provided by BIRD, and a schema summary containing table names, column names, and key constraints. The model outputs "Yes" only when confident that the SQL query exactly captures the question's intent, and "No" otherwise (including cases of uncertainty).

Metric	Chosen	Rejected
Correctness (mean)	4.71	0.92
Correctness ≥ 4 (%)	92.2	3.8
Faithfulness (mean)	4.80	2.74
Faithfulness ≥ 4 (%)	96.6	25.9
Coherence (mean)	4.68	3.40
Coherence ≥ 4 (%)	97.8	49.3
Hallucination (%)	3.1	46.1
Pairwise Judgments		
Chosen wins (%)	97.2	
Contrastiveness (mean)	4.03	
Contrastiveness ≥ 3 (%)	99.8	

Table 6: LLM-as-judge evaluation results. Correctness, faithfulness, and coherence are scored 0-5 (higher is better). Hallucination is reported as the percentage of traces containing unsupported claims (lower is better).

F Calibration of Pipeline Judges

The answer evaluator used to filter positive traces (Section 3.1) and the semantic consistency judge used to construct the BIRD evaluation set (Appendix E) are both LLM-as-judge filters. We calibrate each against human annotations.

Gemini answer evaluator. We audited 57 randomly sampled $(\text{generated answer}, \text{gold answer})$ pairs labeled by the Gemini answer evaluator and compared its judgments to human annotations. Of the 27 pairs that humans judged correct, the evaluator accepted 19 (recall 70.4%); of the 30 pairs that humans judged incorrect, it rejected 29 (specificity 96.7%). Its precision is therefore 95.0% $(19/20)$ on the positive class and 78.4% $(29/37)$ on the negative class. The error distribution is asymmetric: 8 false negatives versus 1 false positive. In other words, the judge is conservative—it tends to discard correct answers rather than accept incorrect ones. For the construction of our training set, this skew is favorable: the dominant failure mode is rejecting valid traces (reducing dataset size), not admitting bad traces (contaminating the positive class). As a result, the composition of the retained positive training set is biased toward valid reasoning paths, and our reported performance is likely a lower bound.

GPT-5 semantic consistency judge. For the BIRD evaluation set construction, we audited 30 samples (15 accepted, 15 rejected) by the GPT-5 semantic consistency judge against human annotations. Human and judge labels agree on 70% of samples. Here too the judge is conservative: it rejects 7 samples that humans would accept, while only 2 samples accepted by the judge were judged by humans as actually misaligned. Since this judge is used as a quality filter to construct the evaluation set, the conservative behavior means the resulting BIRD evaluation set is a high-precision subset of valid question-SQL pairs at the cost of some recall.

G Additional Tables

Tables 7 and 8 report the diversity and pairwise-contrastiveness metrics discussed in Section 5.1; Table 6 reports the LLM-as-judge quality scores and Table 9 compares them with human scores (Section 5.3); Table 10 is the case study of Section 6.

Same-Answer Pairs. 4.4% of pairs share identical answer objects despite different reasoning traces: flawed reasoning coincidentally arrives at the correct answer. Such pairs provide weaker preference signal. We retain them because they represent authentic reasoning errors that a model should still learn to reject, but their effect on training is untested.

Metric	Chosen	Rejected
Surface Diversity		
Unique Rate (%)	99.7	100.0
Vocabulary Size	5,799	8,064
Distinct-1	0.025	0.031
Distinct-2	0.181	0.237
Distinct-3	0.396	0.509
Distinct-4	0.590	0.736
Median Length (tokens)	60	68
Structural Diversity		
Median Steps	5	5
Mean Steps	5.50	5.83
Unique Op-Sequence Rate (%)	26.9	40.5

Table 7: Surface and structural diversity metrics for chosen (positive) and rejected (negative) reasoning traces.

Metric	Min	P25	Median	P75	Max
Token Jaccard	0.07	0.40	0.47	0.55	0.96
Semantic Cosine	0.41	0.80	0.85	0.90	0.99
Step Difference	0	0	0	1	26

Table 8: Pairwise contrastiveness between chosen and rejected traces. Token Jaccard measures lexical overlap; Semantic Cosine measures embedding similarity using sentence-transformers. P25/P75 denote the 25th/75th percentiles.

	Positive ($+y$)		Negative ($-y$)	
Metric	**Human**	**LLM**	**Human**	**LLM**
Correctness	5.00	4.71	1.10	0.92
Faithfulness	4.93	4.80	3.35	2.74
Coherence	4.97	4.68	3.90	3.40

Table 9: Human vs. LLM-judge scores on a sample of 30 contrastive trace pairs. Scores are 0–5; higher is better.

Question: What is the total price for products in **level 8** with **attribute 4** and value '1'?
Gold Answer: 687.59

Positive Reasoning Trace ($+y$)	**Negative Reasoning Trace** ($-y$)
1. Identify entries in `TABLE2` with attribute_id 4 and value '1'. Result: **ID 2, ID 5**.	1. Identify entries in `TABLE2` with attribute_id 4 and value '1'. Result: **ID 2, ID 5**.
2. Look up these IDs in `TABLE1` to check `catalog_level_number`.	2. **Retrieve prices** for these IDs from `TABLE1`. *(Skipped constraint verification)*
3. **Verification:** - ID 2 is Level 8 (✓Match) - ID 5 is Level 9 (✗Mismatch - Exclude)	3. Price for ID 2 is 687.59. 4. Price for ID 5 is 616.22.
4. Sum prices of matching entries (only ID 2).	5. **Sum prices** of both entries.
5. Total: 687.59	6. Total: $687.59 + 616.22 = 1303.81$
Final Answer: 687.59 ✓	**Final Answer:** 1303.81 ✗

Table 10: Comparison of constraint verification.

Table 11: Prompt for generating positive reasoning traces.

System Prompt

You are an expert at multi-table question answering. You are given a [Question] and two tables [TABLE1] and [TABLE2]. Your job is to reason step-by-step and arrive at the correct answer.

Each reasoning step must:

- Refer to either TABLE1 or TABLE2 explicitly
- Explain the logical operation concisely
- Avoid including full table rows; use short identifiers

Respond in a markdown table: | Step | Output |, followed by {"Answer": [...]}.

Inputs

[Question]	The given multi-hop question.
[TABLE1], [TABLE2]	The input tables (serialized in JSON).

Table 12: Prompt template for the LLM-as-a-Judge answer evaluation.

System Prompt

You are an Answer evaluator. Measure the semantic similarity between [Generated Answer] and [Gold Answer]. Return 1 if they mean the same (allowing for aliases like "Messi" == "Lionel Messi"), else return 0.

Inputs

[Generated Answer]	The model-generated answer to be evaluated.
[Gold Answer]	The gold (reference) answer.

Table 13: Prompt template for generating contrastive (negative) reasoning traces.

System Prompt

You are an expert reasoning trace generator. Your task is to generate a contrastive reasoning trace that is wrong.

Requirements:

- Make changes at each step to induce errors (alter numbers, logic, or entities).
- The contrastive trace should be plausible but incorrect.
- You may reorder steps if it helps make the trace more different.
- Follow the markdown table format: | Step | Output | ... followed by {"Answer": [...]}

Inputs

[Prompt]	The original multi-table Q&A prompt.
[Correct Trace]	The correct reasoning trace (positive example).

Table 14: Prompt template for BIRD semantic consistency verification using GPT-5 as judge.

System Prompt

```
You are a strict SQL semantics auditor.

Task:
- Given a natural language Question and a gold SQL query, decide whether the SQL's
semantics EXACTLY match the Question's intent.
- Evidence is optional; if provided, use it only to disambiguate terms/constraints.
- Schema is optional; if provided, it contains column names (and sometimes key hints).
Use it only to interpret the SQL correctly.

Decision rule:
- Answer "Yes" ONLY if you are confident there is NO semantic mismatch.
- Otherwise answer "No" (including any uncertainty).

Treat as mismatch (non-exhaustive):
- Missing/extra filtering conditions or constraints
- Wrong aggregation level/unit (rows vs entities), join multiplicity causing miscounts
- Incorrect join logic, grouping, DISTINCT, LIMIT/OFFSET, ORDER BY direction
- Answering a related but different question

Output format:
- Output exactly one token: Yes or No.
```

User Prompt Structure

[db_id]	The database identifier from BIRD.
[Question]	The natural language question from BIRD.
[Evidence]	Optional evidence/hints provided by BIRD for disambiguation.
[SQL]	The gold SQL query to be verified.
[Schema]	Schema summary: table names with column lists, plus primary/foreign key annotations.