Uncovering the Impact of Chain-of-Thought Reasoning for Direct Preference Optimization: Lessons from Text-to-SQL

Hanbing Liu[1*], Haoyang Li[1*], Xiaokang Zhang[1], Ruotong Chen[1],
Haiyong Xu[2], Tian Tian[2], Qi Qi[1], Jing Zhang[1†]

[1] *Renmin University of China* [2] *China Mobile Information Technology Center*

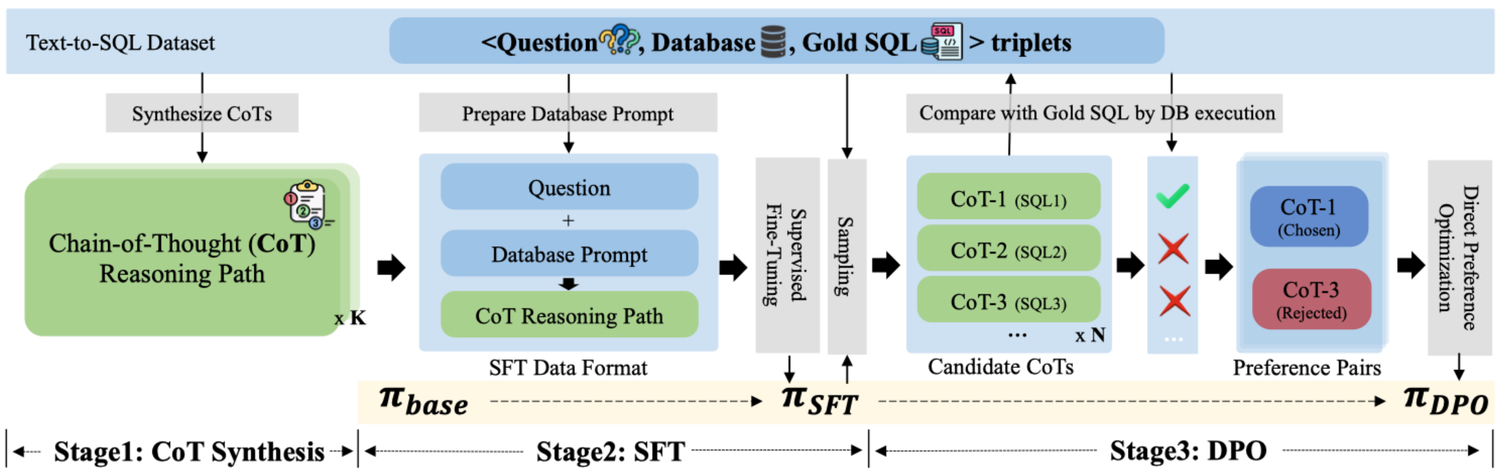
Github

ACL 2025 VIENNA JULY 27 - AUGUST 1

Introduction

- Direct Preference Optimization (DPO) and its variants have shown their power in further enhancing SFT models' capabilities on math problems or coding tasks. However, **DPO usually degenerate model performance on Text-to-SQL benchmarks**.
- After tons of tryouts (including hyper-parameter tuning, preference data collection, algorithms, etc.), we finally find the key factor: Unlike other complex reasoning tasks, **Text-to-SQL datasets lack Chain-of-Thought solutions, which is critical in DPO training**.
- We validate our findings through extensive experiments, and reveal that CoT in complex reasoning tasks helps DPO with **1)** More accurate reward model, **2)** More stable training process, and **3)** More reliable scaling behavior.

Pipeline

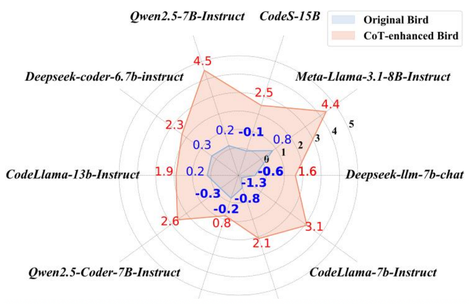


I. **CoT Synthesis**: Augment Text-to-SQL datasets with rationales by instruction-tuned models, creating multiple CoT solutions for each instance.

II. **SFT**: Fine-tune base model on augmented dataset.

III. **DPO**: Sample SFT model on each training data multiple times; Evaluate correctness based on execution feedback, by which construct pair-wise preference dataset; Further train SFT model.

Key Findings



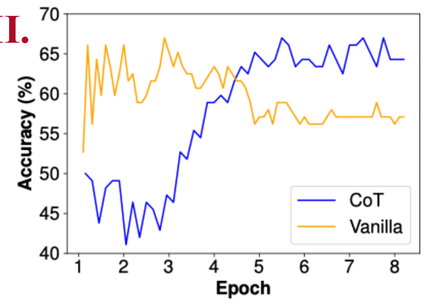
Model	Date	Size	EX
ExSL+granite-34b-code	Oct 27, 2024	34B	72.43
Qwen2.5-14B-Instruct + Syn CoT + DPO (Ours)	**Nov 12, 2024**	**14B**	**67.10**
XiYanSQL-QwenCoder-32B	Jan 09, 2025	32B	67.01
MSL-SQL+DeepSeek-V2.5	Oct 10, 2024	236B	66.82
ExSL+granite-20b-code	May 14, 2024	20B	65.38
SFT CodeS-15B	Oct 12, 2023	15B	58.47

I.

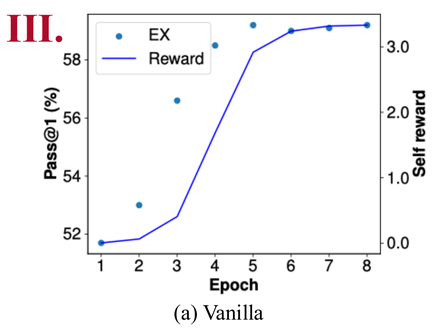
	Model	Bird Dev						ΔEX
		Greedy		Pass@1		Maj@16		
		SFT	DPO	SFT	DPO	SFT	DPO	
General Models								
Vanilla	Deepseek-llm-7b-chat	51.8	51.2 (-0.6)	47.9	49.1 (+1.3)	54.5	54.3 (-0.3)	-
	Meta-Llama-3.1-8B-Instruct	59.0	59.8 (+0.8)	56.1	57.2 (+1.1)	61.4	60.8 (-0.6)	-
	Qwen2.5-7B-Instruct	58.8	59.0 (+0.2)	55.1	55.7 (+0.6)	61.4	60.6 (-0.8)	-
	Qwen2.5-14B-Instruct	64.3	63.5 (-0.8)	62.3	62.6 (+0.3)	64.6	65.1 (+0.5)	-
Syn CoT	Deepseek-llm-7b-chat	54.3	55.9 (+1.6)	51.9	54.8 (+2.9)	59.1	61.0 (+1.9)	54.5 → 61.0 (**+6.5**)
	Meta-Llama-3.1-8B-Instruct	56.8	61.2 (+4.4)	57.5	59.0 (+1.5)	60.2	61.9 (+1.7)	61.4 → 61.9 (**+0.5**)
	Qwen2.5-7B-Instruct	57.4	61.9 (+4.5)	54.8	59.2 (+4.4)	63.0	64.9 (+1.9)	61.4 → 64.9 (**+3.5**)
	Qwen2.5-14B-Instruct	63.2	65.3 (+2.1)	61.8	64.7 (+2.9)	65.4	67.1 (+1.7)	64.6 → 67.1 (**+2.5**)
Coder Models								
Vanilla	Deepseek-coder-6.7b-instruct	60.6	60.9 (+0.3)	56.9	58.8 (+1.9)	59.8	61.0 (+1.2)	-
	CodeLlama-7b-Instruct-hf	57.0	55.7 (-1.3)	54.3	55.5 (+1.2)	59.1	58.5 (-0.6)	-
	CodeLlama-13b-Instruct-hf	60.0	60.2 (+0.2)	56.7	57.9 (+1.2)	61.9	62.0 (+0.1)	-
	Qwen2.5-Coder-7B-Instruct	61.6	61.3 (-0.3)	59.4	60.6 (+1.2)	61.3	62.7 (+1.4)	-
Syn CoT	Deepseek-coder-6.7b-instruct	61.5	63.8 (+2.3)	59.9	62.3 (+4.5)	64.3	65.4 (+1.1)	59.8 → 65.4 (**+5.6**)
	CodeLlama-7b-Instruct-hf	58.2	61.3 (+3.1)	56.9	60.4 (+3.5)	60.2	61.9 (+1.7)	59.1 → 61.9 (**+2.8**)
	CodeLlama-13b-Instruct-hf	62.0	63.9 (+1.9)	59.8	62.5 (+2.7)	63.6	65.8 (+2.2)	61.9 → 65.8 (**+3.9**)
	Qwen2.5-Coder-7B-Instruct	60.8	63.4 (+2.6)	59.1	62.8 (+3.7)	62.5	64.1 (+1.6)	61.3 → 64.1 (**+2.8**)
SQL-Specialized Models								
Vanilla	CodeS-7B	56.8	56.6 (-0.2)	53.7	54.6 (+0.9)	58.1	58.0 (-0.1)	-
	CodeS-15B	58.3	58.2 (-0.1)	55.6	56.2 (+0.6)	60.2	59.1 (-1.1)	-
Syn CoT	CodeS-7B	56.7	57.5 (+0.8)	54.2	55.3 (+1.1)	60.2	61.7 (+1.5)	58.1 → 61.7 (**+2.6**)
	CodeS-15B	58.6	61.1 (+2.5)	56.6	60.5 (+3.9)	62.4	63.2 (+0.8)	60.2 → 63.2 (**+3.0**)

I. Synthesized CoT reasoing leads to **stable and significant improvements in the DPO stage**, consistently beats vanilla models.

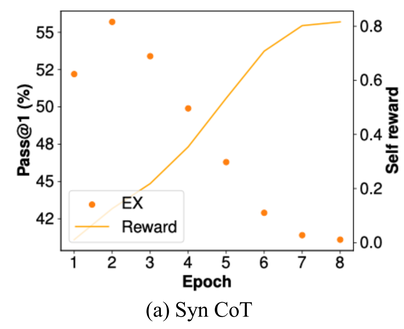
II.



III.



(a) Vanilla

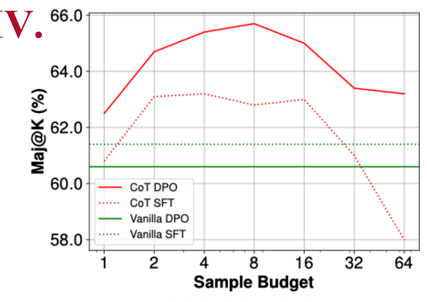


(a) Syn CoT

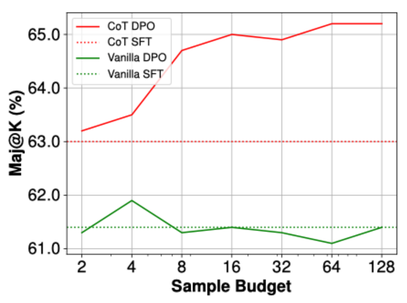
II. When enhanced with CoT reasoning, model's **implicit reward model performs more accurate** in the DPO training progress, providing better reward signal for improving generation.

III. CoT reasoning reveals model's logic behind SQL generation, making it more precise in locating the token where mistake takes place, which in turn makes **model less prone to reward hacking**.

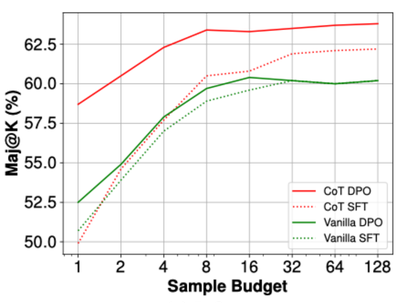
IV.



(a) CoT Synthesis



(a) Preference Data



(a) Inference

IV. Model trained on CoT-enhanced Text-to-SQL datasets **benefites from increased sample budgets aross stages**, including reasoning synthesis, preference data collection and inference voting scale.

Practical Insights for Text-to-SQL

Category	Description	Type	Vanilla DPO Fix (%)	Syn CoT DPO Fix (%)	Δ(%)
External Knowledge	Neglect of hints	**[A1] EK**	0.0 (0/3)	**37.5 (3/8)**	+37.5
Schema Linking	Fails to match the question with its concerning table and columns	[B1] Table	13.0 (12/92)	15.9 (11/69)	+2.9
		[B2] JOIN	15.6 (12/77)	**32.1 (18/56)**	+16.5
		[B3] Column	10.3 (7/68)	16.1 (10/62)	+5.8
		[B4] Hallucination	23.7 (14/59)	**27.2 (28/102)**	+3.5
		[B5] Condition	16.7 (10/60)	23.2 (16/69)	+6.5
		[B6] NULL/DISTINCT	9.7 (3/31)	**40.0 (12/30)**	+30.3
Value Retrieval	Mismatch of condition with its storage format	[C1] String/Number	4.5 (1/22)	21.1 (4/19)	+16.6
		[C2] Date	23.1 (6/26)	**30.4 (7/23)**	+7.3
Operation	Misunderstands required operation in the question.	[D1] Mathematical Formula	13.3 (6/45)	18.2 (8/44)	+4.9
		[D2] Aggregation	6.7 (5/75)	18.2 (12/66)	+11.5
		[D3] Complex Operation	5.6 (1/18)	12.5 (3/24)	+6.9
Information	Fails to organize information in the right way	[E1] Redundant/Incomplete	11.8 (4/34)	19.2 (5/26)	+7.4
		[E2] Column Sequence	0 (0/5)	**42.9 (3/7)**	+42.9
		[E3] ORDER BY/LIMIT	9.1 (1/11)	12.5 (1/8)	+3.4
		[E4] Format	**66.7 (2/3)**	**33.3 (2/6)**	-33.4
Syntax Error	Inexecutatble SQL	[F1] Syntax	14.3 (2/14)	13.3 (2/15)	-1.0

Type	Vanilla DPO Fix (%)	Syn CoT DPO Fix (%)	Δ(%)↓
[E2] Column Sequence	0 (0/5)	**42.9 (3/7)**	**+42.9**
[A1] EK	0.0 (0/3)	**37.5 (3/8)**	**+37.5**
[B6] NULL/DISTINCT	9.7 (3/31)	**40.0 (12/30)**	**+30.3**
[C1] String/Number	4.5 (1/22)	21.1 (4/19)	**+16.6**
[B2] JOIN	15.6 (12/77)	**32.1 (18/56)**	**+16.5**
[D2] Aggregation	6.7 (5/75)	18.2 (12/66)	**+11.5**
[E1] Redundant/Incomplete	11.8 (4/34)	19.2 (5/26)	**+7.4**
[C2] Date	23.1 (6/26)	**30.4 (7/23)**	**+7.3**
[D3] Complex Operation	5.6 (1/18)	12.5 (3/24)	**+6.9**
[B5] Condition	16.7 (10/60)	23.2 (16/69)	**+6.5**
[B3] Column	10.3 (7/68)	16.1 (10/62)	**+5.8**
[D1] Mathematical Formula	13.3 (6/45)	18.2 (8/44)	**+4.9**
[B4] Hallucination	23.7 (14/59)	**27.2 (28/102)**	**+3.5**
[E3] ORDER BY/LIMIT	9.1 (1/11)	12.5 (1/8)	**+3.4**
[B1] Table	13.0 (12/92)	15.9 (11/69)	**+2.9**
[F1] Syntax	14.3 (2/14)	14.3 (2/14)	**-1.0**
[E4] Format	66.7 (2/3)	**33.3 (2/6)**	**-33.4**

I. **DPO excels at correcting errors caused by ignoring detail** requirements, such as deduplication and returned column sequence.

II. **CoT largely improves DPO's correction ability for explicit logic** required tasks, like multiple JOINs and string operations.

III. **DPO is not very good at fixing schema linking mistakes and syntax errors.**

Contact

Wechat

Arxiv

Github: /RUCKBReasoning/DPO_Text2SQL
Homepage: tokgoleo.github.io/home/
Email: liuhanbing@ruc.edu.cn

Let's Connect!

ACL 2025 VIENNA JULY 27 - AUGUST 1