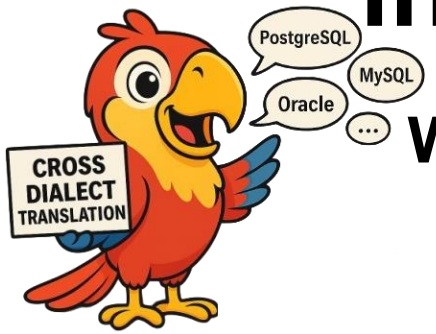
PARROT: A Benchmark for Evaluating LLM in Cross-System SQL Translation



Wei Zhou[1], Guoliang Li[2], Haoyu Wang[3], Yuxing Han[3],

Xufei Wu[1], Fan Wu[1], Xuanhe Zhou[✉1]

[1] Shanghai Jiao Tong University [2] Tsinghua University [3] ByteDance

https://code4db.github.io/parrot-bench/

11 / 2025

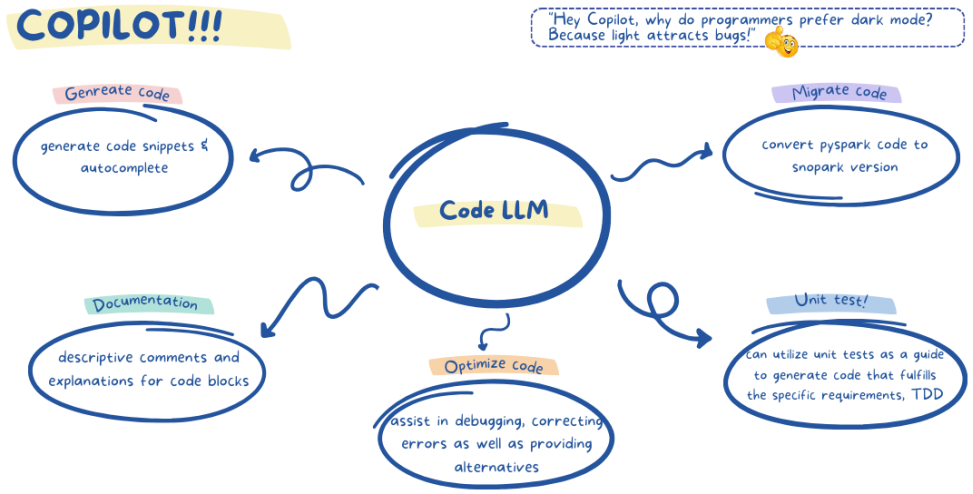
Outline

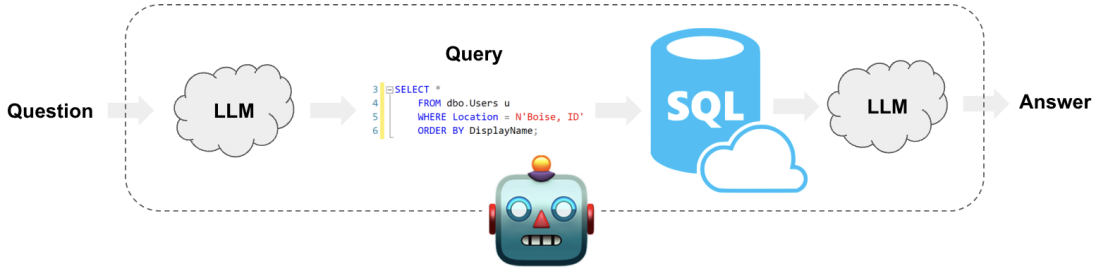
CONTENTS

1. Background
2. Preliminary
3. Framework
4. Experiment

Background

LLMs have advanced in diverse domains, e.g., it has **greatly improved the text-to-SQL accuracy** (81.67% comparable to 92.96% of humans).





Overall Leaderboard | Single-Model Leaderboard

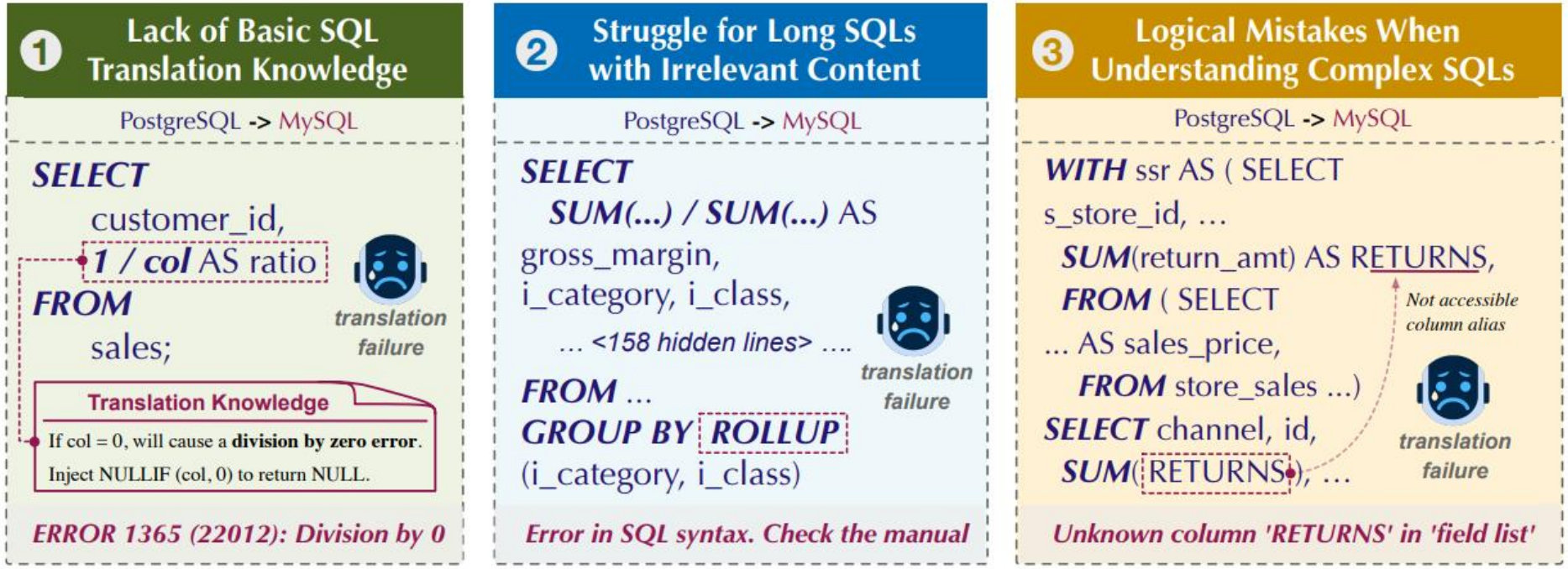
Leaderboard - Execution Accuracy (EX)

	Model	Code	Size	Oracle Knowledge	Dev (%)	Test (%)
	Human Performance *Data Engineers + DB Students*			✔		**92.96**
🏆1 Sep 25, 2025	Agentar-Scale-SQL *Ant Group* [Pengfei Wang et al. '25]	[link]	UNK	✔	74.90	**81.67**

[1] https://blogs.novita.ai/how-to-perform-code-generation-with-llm-models/

Background

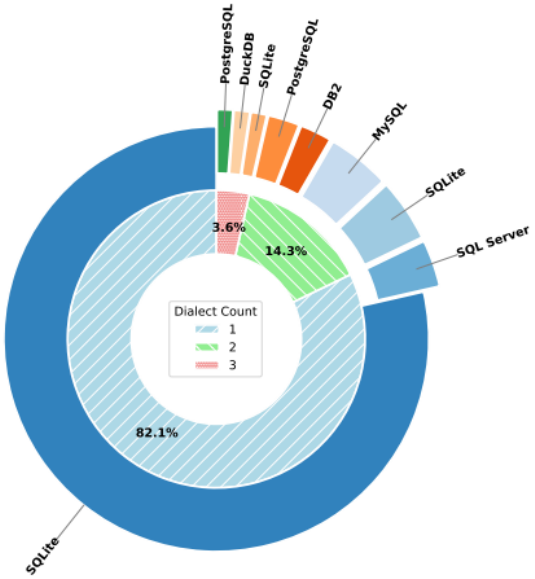
Problem: LLMs struggle with cross-system SQL translation.

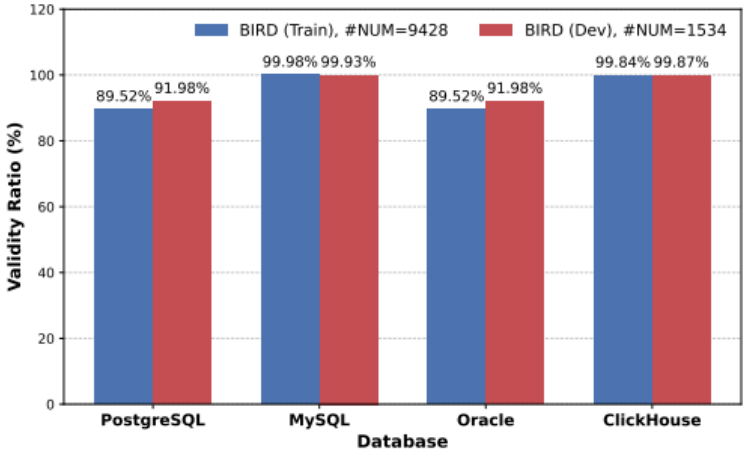


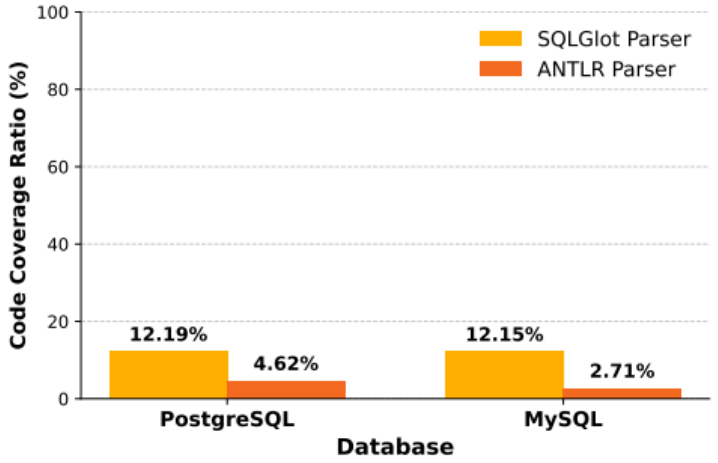
- **Failure 1:** Lack of Basic Knowledge (e.g., Division by zero error).
- **Failure 2:** Struggle with Long SQLs (e.g., Incorrect ROLLUP syntax).
- **Failure 3:** Logical Mistakes (e.g., Invalid column alias).

Background

Problem: Existing benchmarks are ill-suited for SQL-to-SQL evaluation.



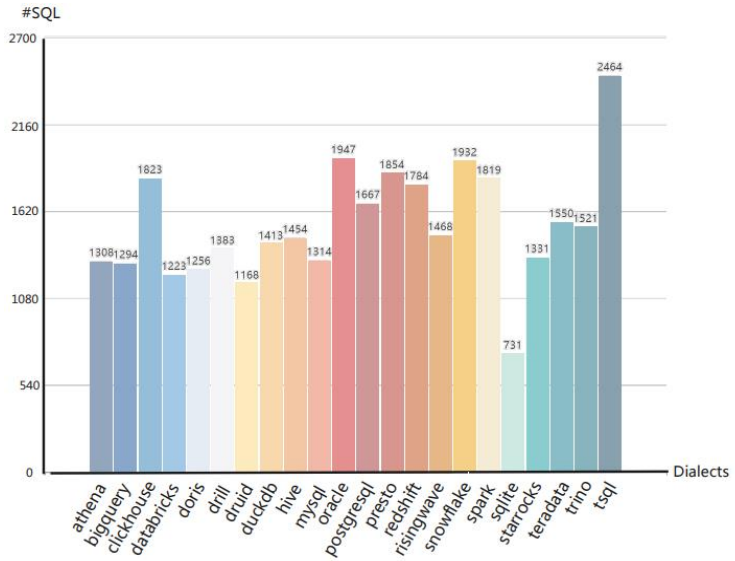


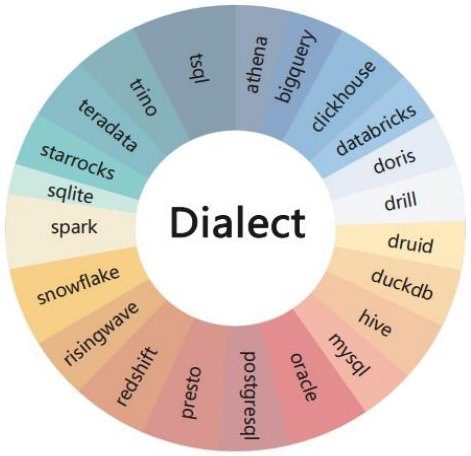


- **Limitation 1:** Limited System Diversity (e.g., 82.1% of benchmarks focus on SQLite).
- **Limitation 2:** Inadequate System Coverage (e.g., Over 89% of BIRD queries are system-agnostic).
- **Limitation 3:** Low Dialect Diversity (e.g., Fewer than 13% of queries test system-specific syntax).

Background

Dataset	#Dialect	#SQL	#Token / SQL			#Translation Type
			25th	Medium	75th	
PARROT	8	598	75.0	249.0	951.0	7
PARROT-Diverse	22	28,003	29.0	47.0	71.0	7
PARROT-Simple	22	5,306	4.0	6.0	10.0	7





We introduce PARROT, the first benchmark for cross-system SQL-to-SQL translation.

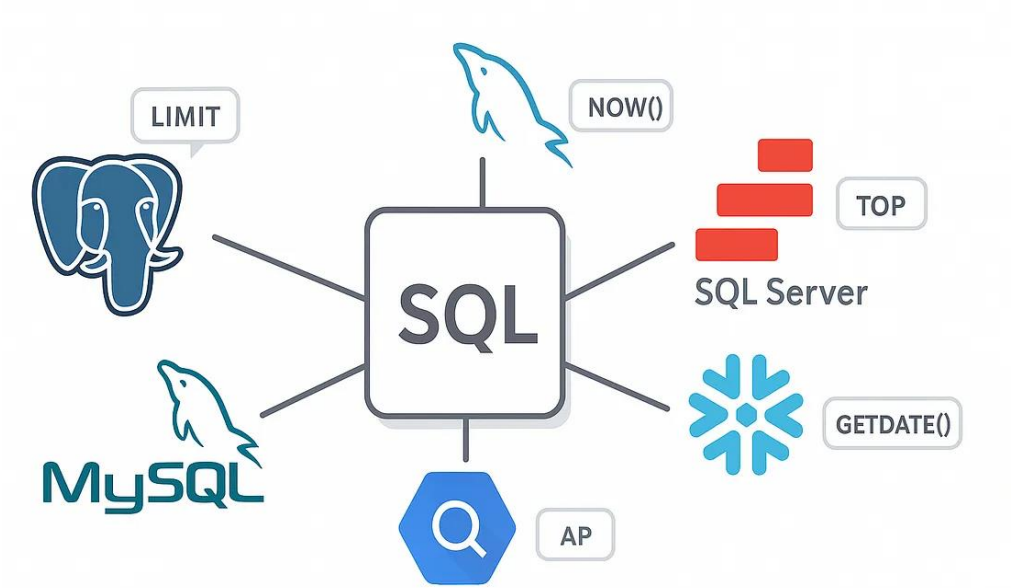
- **Core Dataset:** 598 translation pairs from 38 benchmarks and real-world services.
- **Multiple Variants:** (1) PARROT-DIVERSE: 28,003 samples for extensive syntax testing; (2) PARROT-SIMPLE: 5,306 unit-style samples for focused stress testing.

Outline

CONTENTS

1. Background
2. Preliminary
3. Framework
4. Experiment

Preliminary



Definition: Translating a SQL query from a source to a target dialect, ensuring two properties:

- **Standard Compatibility**
- **Equivalent Functionality**

CASE-1: PG → Oracle

- ✓ tbl AS tbl_alias
- ✓ tbl tbl_alias

CASE-2: PG → MySQL

- ✓ GROUP BY ROLLUP(col_list)
- ✓ GROUP BY col_list WITH ROLLUP

CASE-3: PG → MySQL

- ✓ TO_TIMESTAMP(unix_time)
- ✓ FROM_UNIXTIME(unix_time)

[1] https://medium.com/@remisharoon/why-sql-has-so-many-dialects-and-how-to-deal-with-them-dc9e8698b2e4

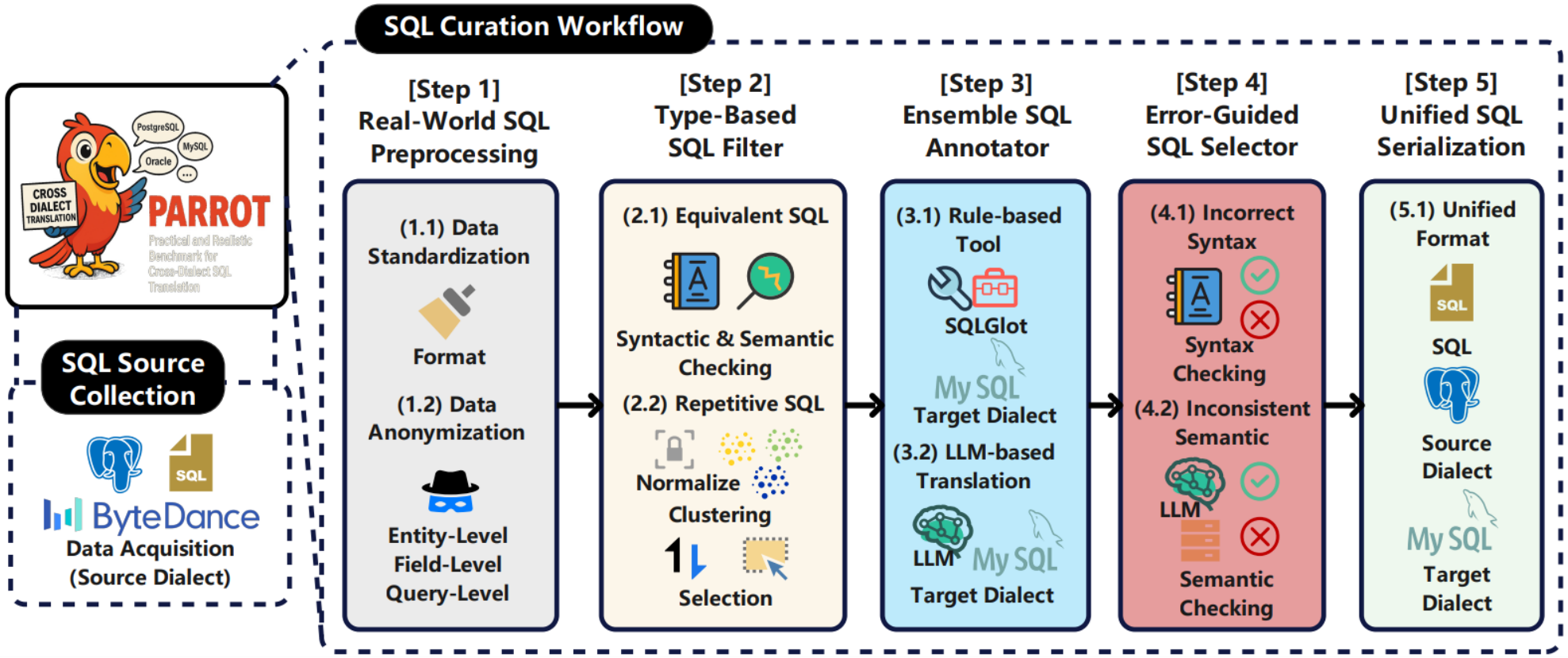
Outline

CONTENTS

1. Background
2. Preliminary
3. Framework
4. Experiment

Framework

Workflow: Five-Step SQL Curation Workflow that anonymizes, filters, annotates, and validates real-world SQL.



Outline

CONTENTS

1. Background
2. Preliminary
3. Framework
4. Experiment

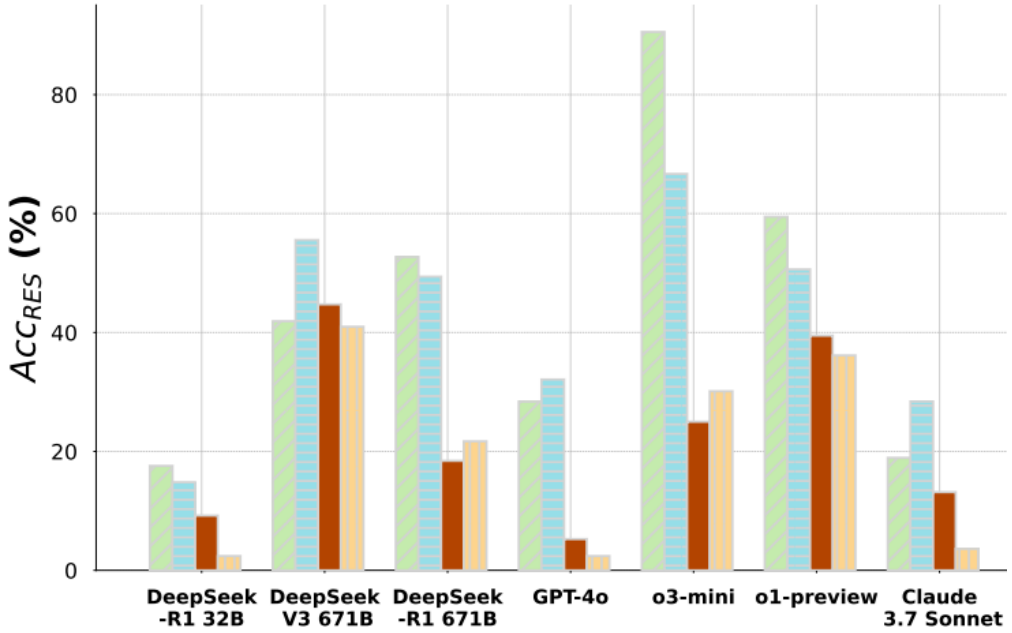
Experiment

Model	* ↓ PostgreSQL	* ↓ MySQL	* ↓ Oracle	* ↓ DuckDB	* ↓ SQL Server
Open-Source LLM					
DeepSeek-R1 7B	17.24	20.59	17.24	14.29	15.79
DeepSeek-R1 32B	**58.62**	**58.82**	39.66	10.71	**42.11**
DeepSeek-Coder-V2 Lite	34.48	32.35	32.76	3.57	21.05
DeepSeek V3 671B	55.17	55.88	51.72	53.57	36.84
DeepSeek R1 671B	48.28	44.12	50.00	42.86	36.84
Proprietary LLM					
GPT-4o	**58.62**	50.00	55.17	**60.71**	**42.11**
o3-mini	31.03	8.82	43.10	35.71	21.05
Claude 3.7 Sonnet	**58.62**	44.12	**58.00**	42.86	36.84

- **(Performance Oscillation):** LLM performance is unstable across dialects.
- **(Scale Isn't Everything):** A larger model does not guarantee better performance.

Experiment

Model	Acc_{EX}	Acc_{RES}
Open-Source LLM		
DeepSeek-R1 32B	21.00	16.91
DeepSeek-V3 671B	39.94	32.65
DeepSeek-R1 671B	46.94	40.52
Proprietary LLM		
GPT-4o	23.91	21.87
o3-mini	**58.60**	**54.23**
o1-preview	56.26	48.69
Claude 3.7 Sonnet	24.20	22.74



- **(Complexity Penalty):** LLMs struggle to obtain accurate translation when the SQLs become more lengthy with more complex operations.

Experiment

Model	DeepSeek-R1 32B	DeepSeek V3 671B	DeepSeek-R1 671B	GPT-4o	o3-mini	o1-preview	Claude 3.7 Sonnet
Syntax Parsing	0.05	**0.28**	0.10	0.00	0.11	0.03	0.03
Identifier Resolution	0.02	0.00	0.07	0.00	0.01	0.06	0.00
Function Resolution	**0.28**	0.00	0.05	**0.40**	**0.64**	**0.37**	0.04
Function Usage	**0.62**	**0.72**	**0.70**	**0.60**	**0.24**	**0.54**	**0.93**
Type Compatibility	0.01	0.00	0.08	0.00	0.00	0.00	0.00
Other Errors	0.02	0.00	0.00	0.00	0.00	0.00	0.00

Original PostgreSQL SQL

```
SELECT
  TO_CHAR(
    TO_TIMESTAMP(
      virtual_T1."day" || '',
      'YYYYMMDD' ), 'YYYY'
  ) AS "col1_2", ...
FROM ( ...
    UNION ALL SELECT ...
      CASE WHEN NOT
        t1.p_rate IS NULL
        THEN CONCAT(
          t1.p_rate, '%'
        ) ELSE ''
      END AS p_rate, ...
    FROM ... AS t1
    WHERE t1.rn = 1
  ) AS virtual_T1
```

Correct ClickHouse SQL ✓

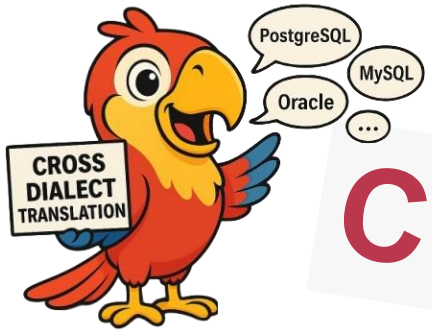
```
SELECT
  formatDateTime(
    parseDateTimeOrNull(
      virtual_T1."day" || '',
      '%Y%m%d' ), '%Y'
  ) AS "col1_2", ...
FROM ( ...
  UNION ALL SELECT ...
    CASE WHEN NOT (
      t1.p_rate IS NULL
    ) THEN CONCAT(
      t1.p_rate, '%'
    ) ELSE ''
    END AS p_rate, ...
  FROM ... AS t1
  WHERE t1.rn = 1
) AS virtual_T1
```

Translation by o3-mini ✗

```
SELECT
  formatDateTime(
    parseDateTimeBestEffort(
      virtual_T1.day
    ), '%Y') AS col1_2, ...
FROM ( ...
  UNION ALL SELECT ...
    if( t1.p_rate != '',
      concat(t1.p_rate, '%'),
      '' ) AS p_rate, ...
  FROM ... AS t1
  WHERE rn = 1
) AS virtual_T1
...
```

- **(Function Failures):**

The vast majority of failures stem from the misuse of **built-in functions** (Function Resolution and Function Usage).



About PARROT

PARROT (**P**ractical **A**nd **R**ealistic Benchma**R**k for Cr**O**ss-System SQL **T**ranslation) was created to support the task of Cross-System SQL Translation (i.e., SQL-to-SQL translation), which involves adapting a query written for one database system into its functionally equivalent form for another.

The main dataset comprises 598 translation pairs from 38 open-source benchmarks and real-world business services, specifically prepared to challenge system-specific SQL understanding.

News

Sept. 18, 2025: Our paper "PARROT: A Benchmark for Evaluating LLMs in Cross-System SQL Translation" has been accepted by NeurIPS 2025 !

May 15, 2025: We have released PARROT-1.0 (28,003 translation pairs from 38 open-source benchmarks for extensive syntax testing) and published the leaderboard.

Overall Leaderboard | Single-Dialect Leaderboard

PARROT

We have publicly released PARROT along with detailed usage instructions. For more details, please visit the GitHub repository. To update the leaderboard, ensure that your paper or resource is publicly accessible and submit a pull request.

Leaderboard - Dialect Compatability (Acc_{EX})

	Model	Code	Size	Accuracy (%)
	Human Performance			> 90.00
2	DeepSeek-V3 671B *DeepSeek*		671B	54
3	Claude 3.7 Sonnet *Anthropic*		UNK	48.09
4	DeepSeek-R1 671B *DeepSeek*		671B	44.42
5	DeepSeek-R1 32B *DeepSeek*		32B	41.98

Challenge the leaderboard!

https://code4db.github.io/parrot-bench/