

ReflexSQL: A Self-Reflective Agent for Robust Text-to-SQL Generation

Handling enterprise-scale Text-to-SQL with execution-aware self-reflection

Wanxing WU, Kaiying WU
QueryPilots, Institut Polytechnique de Paris

Motivation

Text-to-SQL aims to translate natural language queries into executable SQL over arbitrary databases.

Despite recent advances in LLM-based approaches, Text-to-SQL systems remain unreliable in real-world settings, facing challenges:

- Errors in One-shot generation
- lack of execution-aware verification
- Poor schema generalization under complex databases
- Expensive target exploration

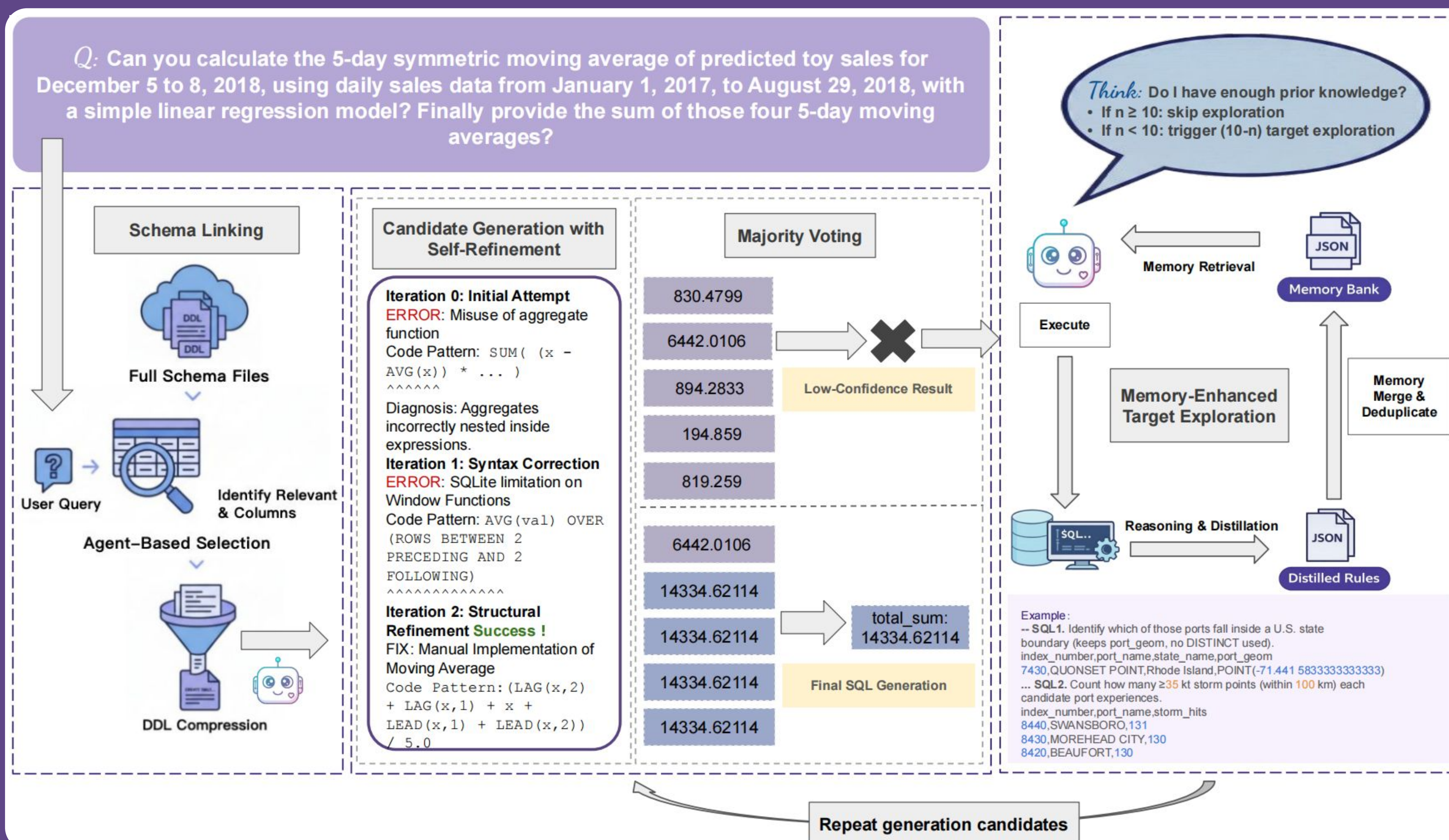
Reflex Agent: An execution-aware framework that iteratively refines SQL via Candidate Voting, Execution Feedback, and Memory-augmented Exploration.

Future Work

- ❖ **Efficient reflex design:**
Study which strategies are most useful for error correction
- ❖ **Cost-aware reflex:**
Reduce reflex overhead without sacrificing reliability
- ❖ **Robustness across settings:**
Test generalization to new data, schemas, and model

Towards an **efficient**, **cost-aware**, and **robust reflex-based** Text-to-SQL agent

Methods



Method Overview

1. Agent-Based Schema Linking

- **Input:** Full Database Schema + User Query
- **Process:** Agentic selection of relevant tables/columns
- **Output:** Query-based Compressed DDL

2. Iterative SQL Generation with Execution Feedback

Generate; Execute; Re-prompting the agent with error traces

3. Confidence Assessment via Majority Voting

- Majority voting over the execution results
- Queries without a clear consensus are flagged as low-confidence results

4. Memory-Controlled Exploration Loop

- Consult the Memory for prior knowledge
- Collect **query execution** for missing info if needed
- Update the Memory by distilling and deduplicating

5. Memory-Conditioned Answer Generation

- Inject the knowledge from the exploration step
- Produce trustworthy SQL

Experiments

Dataset & Evaluation Setup

1. **Benchmark:** Spider 2.0-Lite
2. **Databases:** 158 enterprise-scale databases
3. **Large-schema constraint:** Databases contain **800+ columns**; **~77 MB** context w/ rows+descriptions (~20M tokens)
4. **Task:** Generate executable SQL given schema and documentation
5. **Metrics:** EX; EX@8 (executable correctness if any of the top-8 generated SQLs executes correctly)

Limitations

- **Single Dataset:** Only evaluated on Spider 2.0 Lite; broader benchmarks needed for robust assessment.
- **General-Purpose Backbone:** Qwen3-8B and Qwen3-32B lacks specialized Text-to-SQL fine-tuning, potentially limiting performance compared to domain-specific models.

Main Results

Model	Method	EX	EX@8
Qwen3-8B	Base Model	11.85	18.34
	Majority Voting	18.51	31.85
	ReflexSQL (Ours)	40.74	41.48
Qwen3-32B	Base Model	16.29	24.51
	Majority Voting	26.86	42.53
	ReflexSQL (Ours)	40.62	46.87

- Sampling alone yields limited gains, while execution-aware reflex introduces a qualitative improvement.

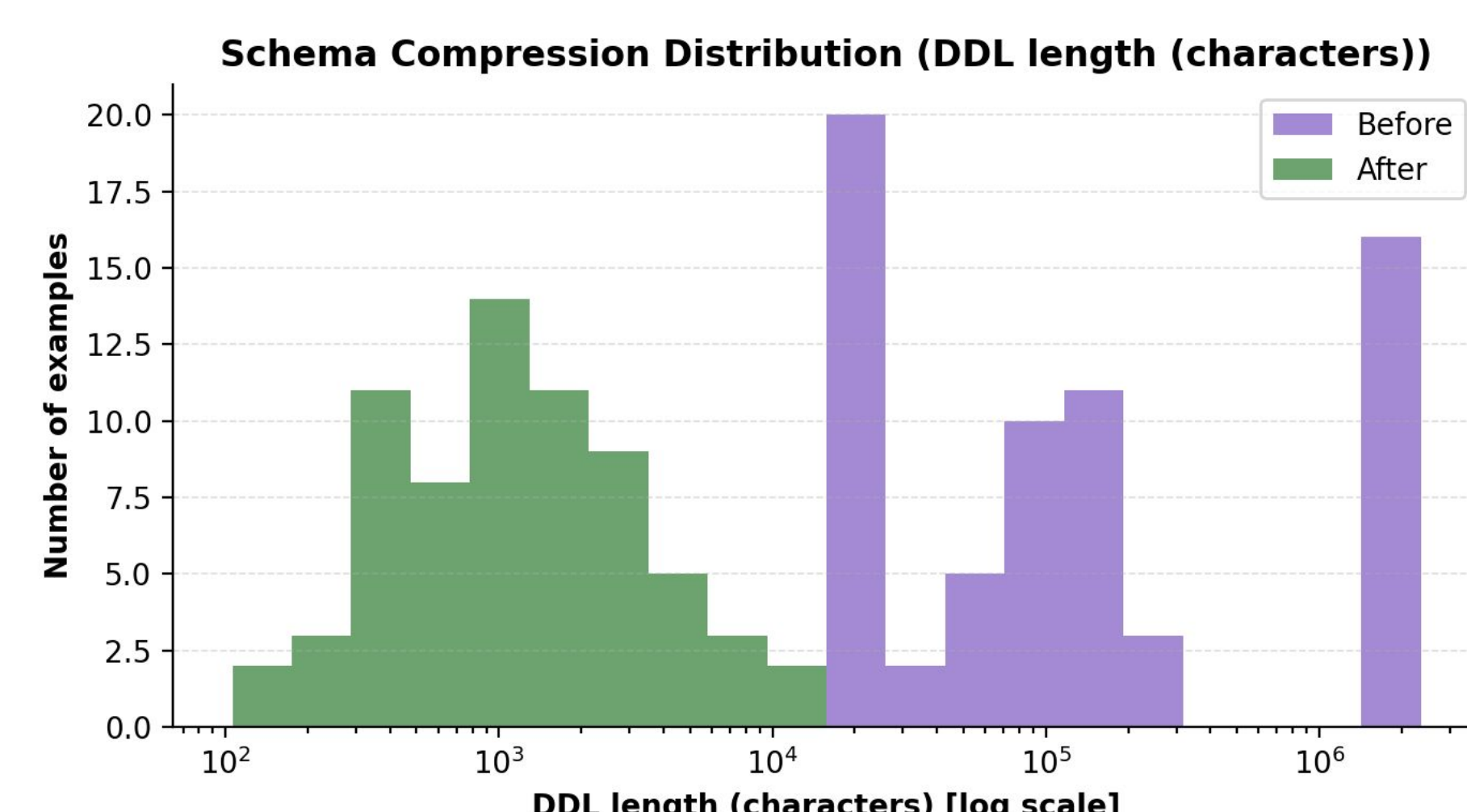
Ablation Study with Qwen3-32B

Method	EX	EX@8
w/o Info Compression	40.18	42.96
w/o Majority Voting	37.77	41.70
w/o Target Exploration	31.85	42.53
ReflexSQL (Ours)	42.53	47.40

- **Memory-based target exploration is critical under large-schema settings.**

Analysis

- **Compression reduces schema length by orders of magnitude, enabling cost-effective prompting for massive databases.**



- **Reflex improves both executability and correctness among executable queries.**

Setting	Exec (%)	Corr@Exec (%)
Single-pass	38.51	36.53
Reflex	86.66	47.01