TinySQL: A Progressive Text-to-SQL Dataset for Mechanistic Interpretability Research

Abir Harrasse*, Philip Quirke*, Clement Neo*, Dhruv Nathawani, Luke Marks, Amir Abdullah

WHY TEXT-TO-SQL?

TEXT-TO-SQL: THE SWEET SPOT

User Query (Natural Language): *"Show me employees earning over 50k"*

SQL Output: *SELECT * FROM employees WHERE salary > 50000*

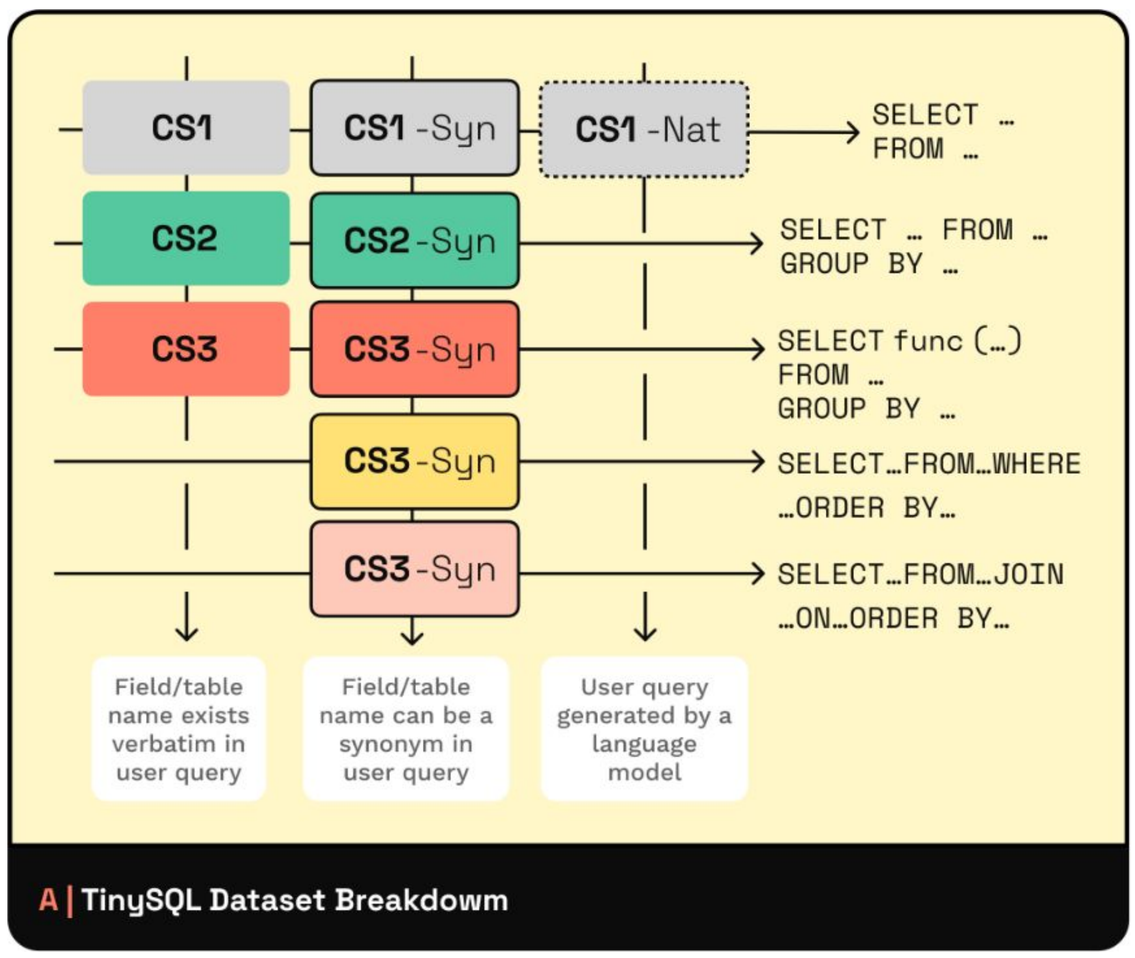
From Toy Tasks:
- Formal structure (ground truth answers).
- Systematic patterns.

From Real World:
- Natural language understanding
- Schema grounding.
- Compositional reasoning.

But, existing datasets are noisy…

THE TINYSQL DATASET

5 complexity levels, 3 model scales, 20 trained checkpoints…



We finetune TinyStories (33M), Qwen 2.5 (0.5B) and Llama 3.2 1B on each of the datasets

HOW WE STUDY CIRCUITS

Edge Attribution Patching based circuits

Create Corrupted Prompts → **Measure Causal Impact** → **Select Minimal Circuit**

Original: "Show me name and age from employees"
Corrupted: "Show me name and age from guests"

- Clean input x_c
- Corrupted input x_d

How does each edge E affect the model's behavior?

$$\Delta_E L = (e_d - e_c) \cdot \nabla_e L(x_c)$$

Task-specific metric

$$L = \frac{\ell(x_{\text{clean}}|\text{do}(E = e_{\text{patch}})) - \ell(x_{\text{corr}})}{\ell(x_{\text{clean}}) - \ell(x_{\text{corr}})}$$

- Process 15 batches of 100 prompts.
- Selected 10 most important edges ranked by $|\Delta_E L|$.
- Keep the top-k edges.

Sparse Autoencoders (SAEs) based circuits

Collect Activations → **Encode via SAE** → **Similarity- based Selection** → **AUC Mapping**

$\{x^{(j)}\}_{j=1}^{N}$ $\{z^{(j)}\}_{j=1}^{N}$ k features

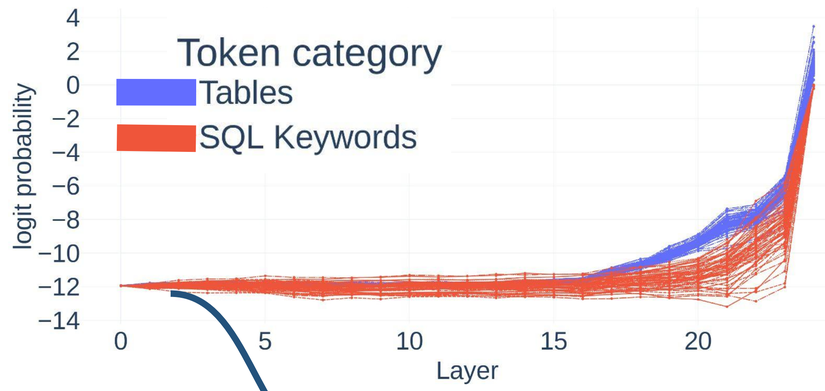
Map back to neurons or attention heads

$\|W_{\text{dec}}[f]\|$

WHAT WE DISCOVERED

- Minimal circuits use only **12-30%** of model components.
- **Small models** (BM1): More compact circuits.
- Larger models (BM2): SAEs essential for **fine granularity**.
- Circuits are not **uniquely identifiable**.
- There's some **shared causal structure** but methods diverge.

EAP Circuits — **Shared Components** — **SAE Circuits**



Intent first, Grounding Later

Takeaways

- TinySQL is the **first testbed that bridges toy tasks and real-world settings**, letting us study circuits in controlled but realistic settings.
- We **systematically compared** EAP, SAE, and Logit Lens to see where each method works and where it breaks.
- Circuits exist, but they're not unique. Methods only agree ~60% of the time, showing models use **distributed computation instead of clean modules.**

EMNLP 2025