



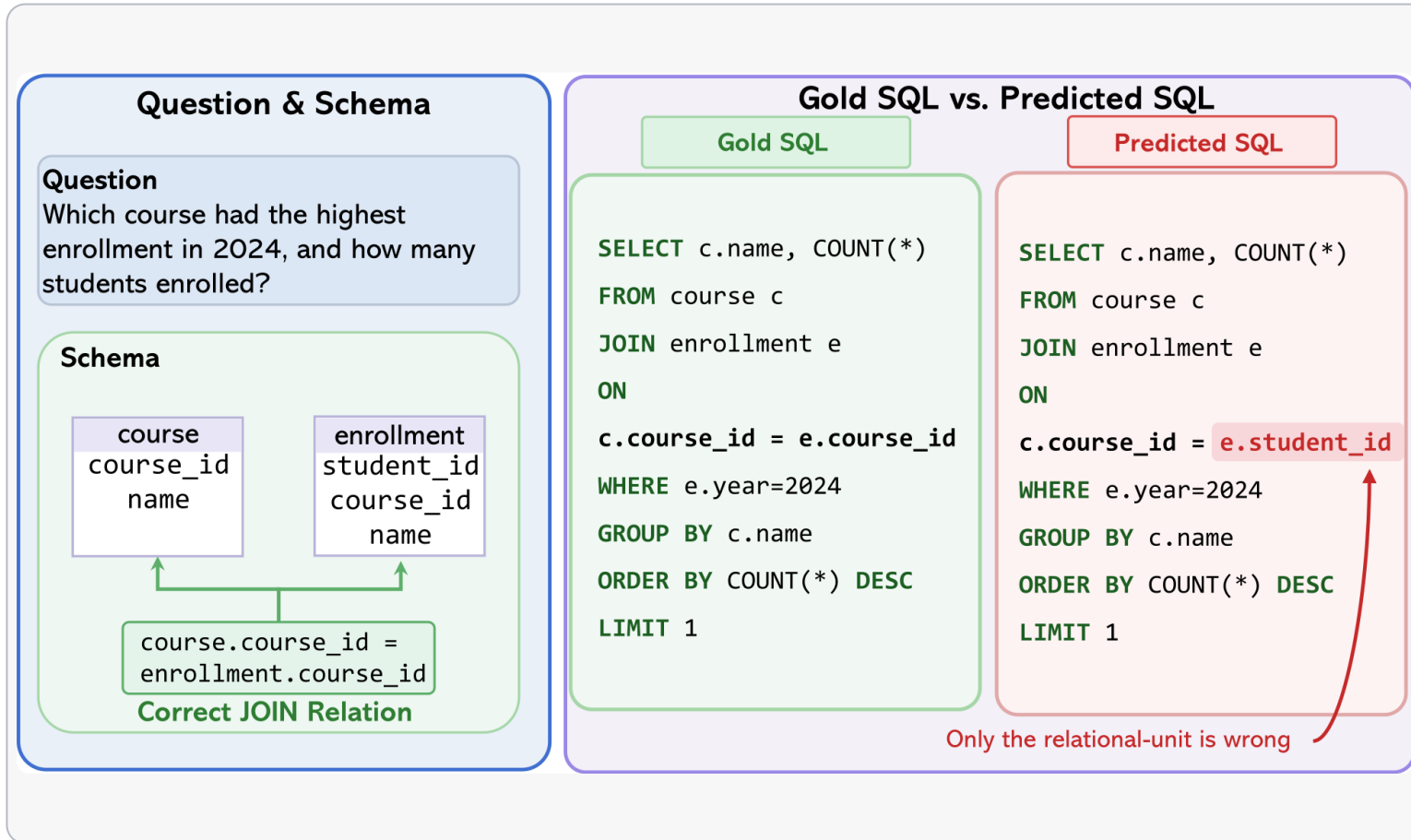
CAPER

Clause-Aligned Process Supervision for Text-to-SQL

Lujie Ban, Jiasheng Shi, Jinyang Li, Xiaolin Han, Tsz Nam Chan, Chenhao Ma

40th Conference on Neural Information Processing Systems

One wrong JOIN can sink the whole query

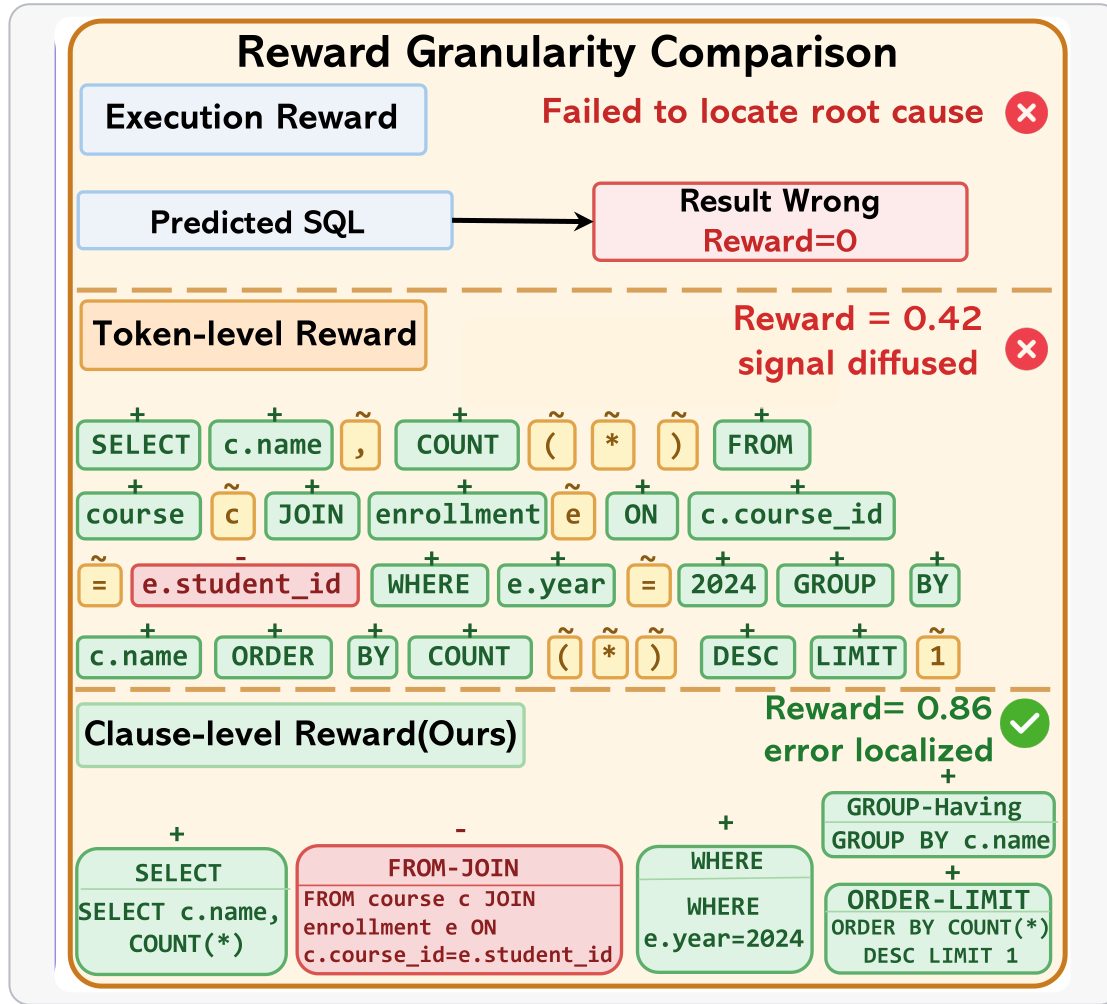


Nearly correct SQL.
One broken relation.

Reward = 0

The outcome says the query failed.
It cannot say why.

Existing rewards miss the semantic decision



Query-level

Sparse terminal signal

Token-level

Credit diffuses across symbols

Clause-level

Feedback matches SQL decisions

Clauses are the missing middle

Each unit is a complete top-level SQL decision.

WITH

SELECT

FROM / JOIN

WHERE

GROUP / HAVING

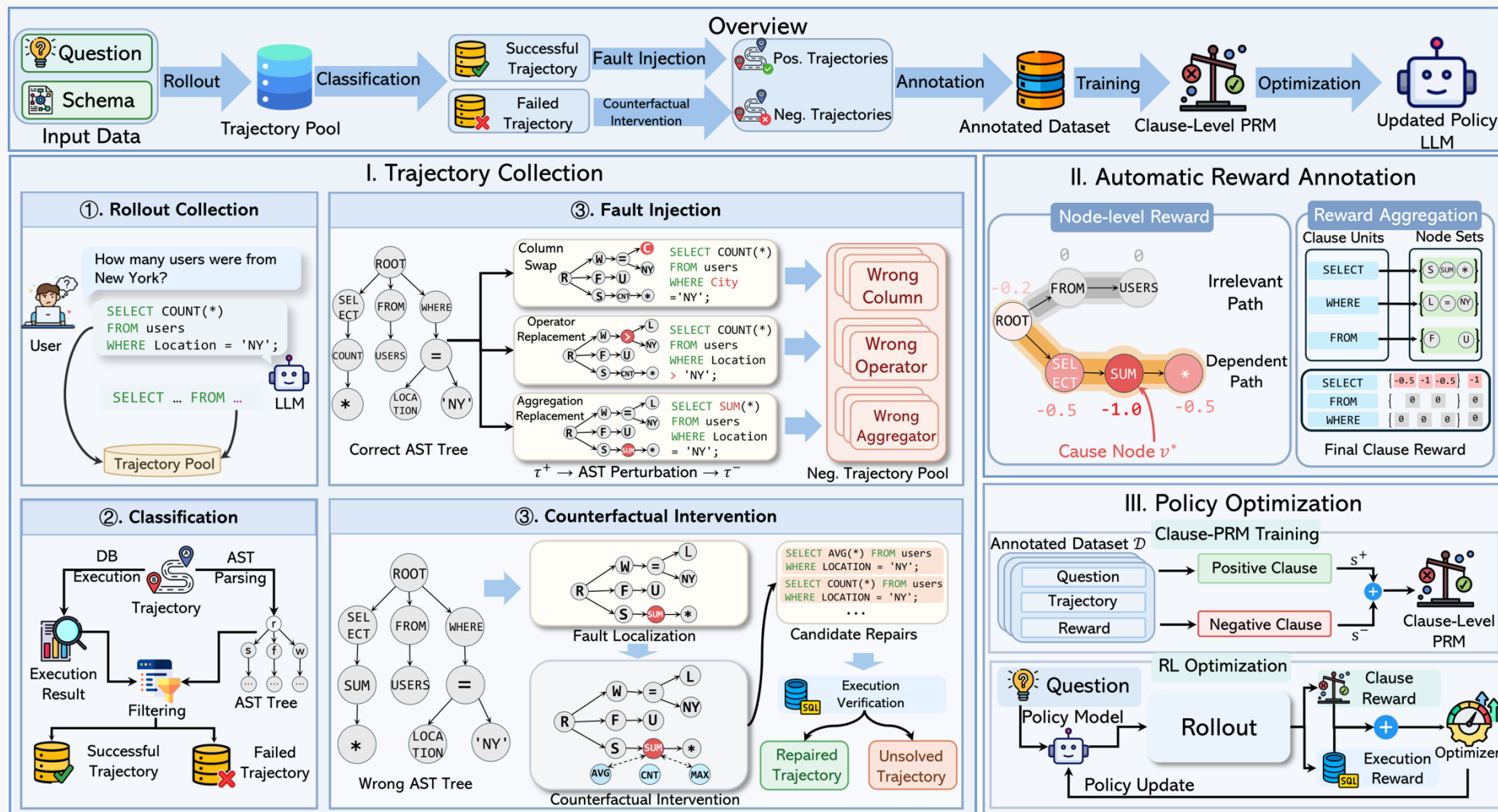
WINDOW

ORDER / LIMIT

SET-OP

Denser than execution feedback · more semantic than token feedback

CAPER turns outcomes into clause supervision



Clause boundaries define where feedback lands

One score evaluates one semantic continuation under its shared prefix.

WITH

SELECT

FROM / JOIN

WHERE

GROUP / HAVING

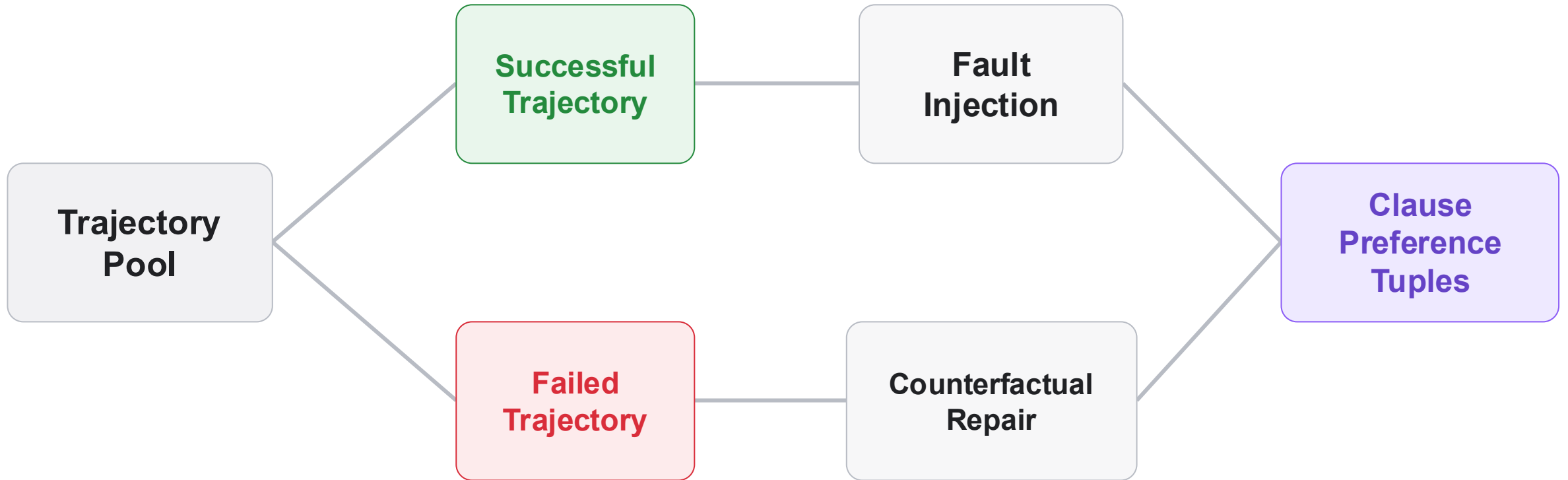
WINDOW

ORDER / LIMIT

SET-OP

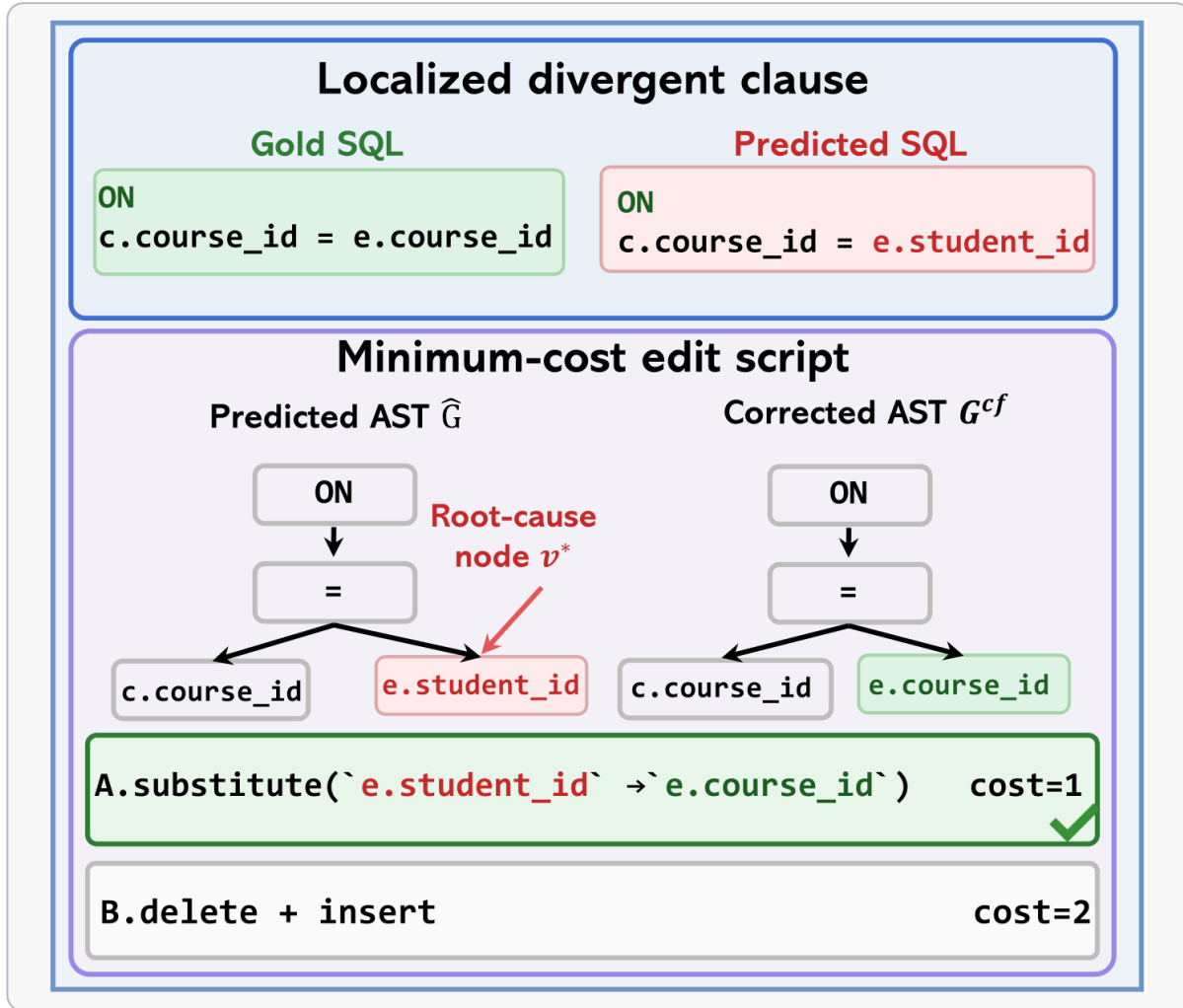
Nested subqueries stay inside their enclosing top-level unit.

Two trajectory types create reliable supervision



Execution checks retain only genuine semantic changes.

A minimal counterfactual exposes the root cause



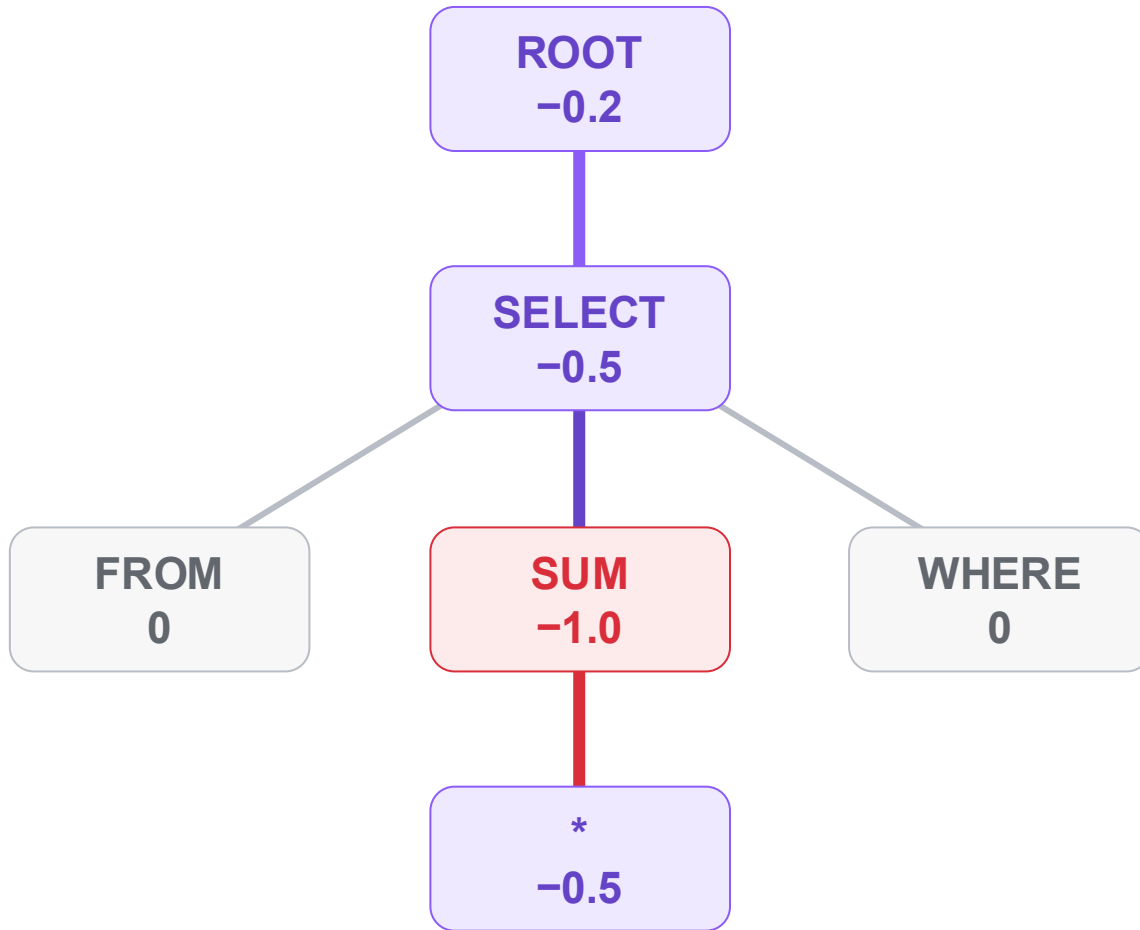
01 Find the earliest divergent clause

02 Search the minimum-cost AST edit script

03 Anchor the first causal edit

Retain only valid, executable repairs.

AST topology converts localization into reward

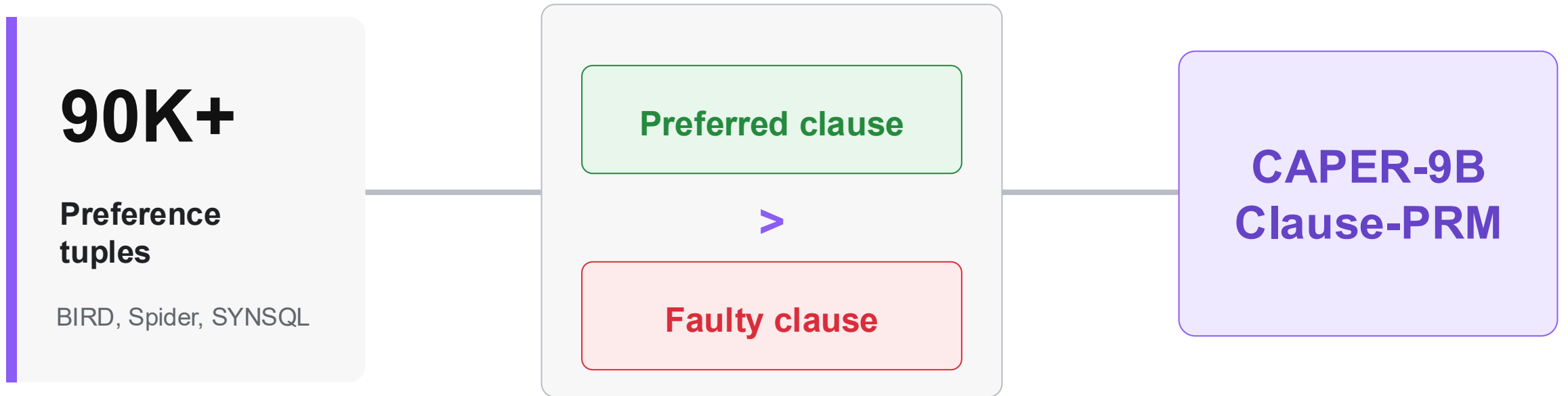


**Strongest penalty
at the cause**

**Ancestors decay with
AST distance**

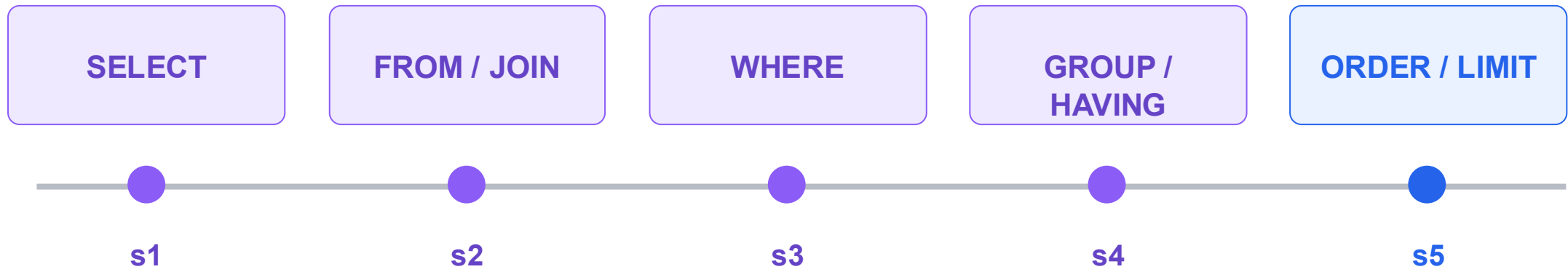
Unrelated branches are masked to
zero.

CAPER-9B learns clause preferences



Larger structural margins receive stronger preference weight.

Rewards arrive at clause boundaries



Clause reward at every boundary

Execution reward only at the terminal boundary

Clause return-to-go → position-wise GRPO advantage

Three tests connect attribution to usefulness

01

Policy optimization

Execution accuracy

BIRD + Spider

02

Failure localization

Acc · Hit@3 · MRR

Held-out failures

03

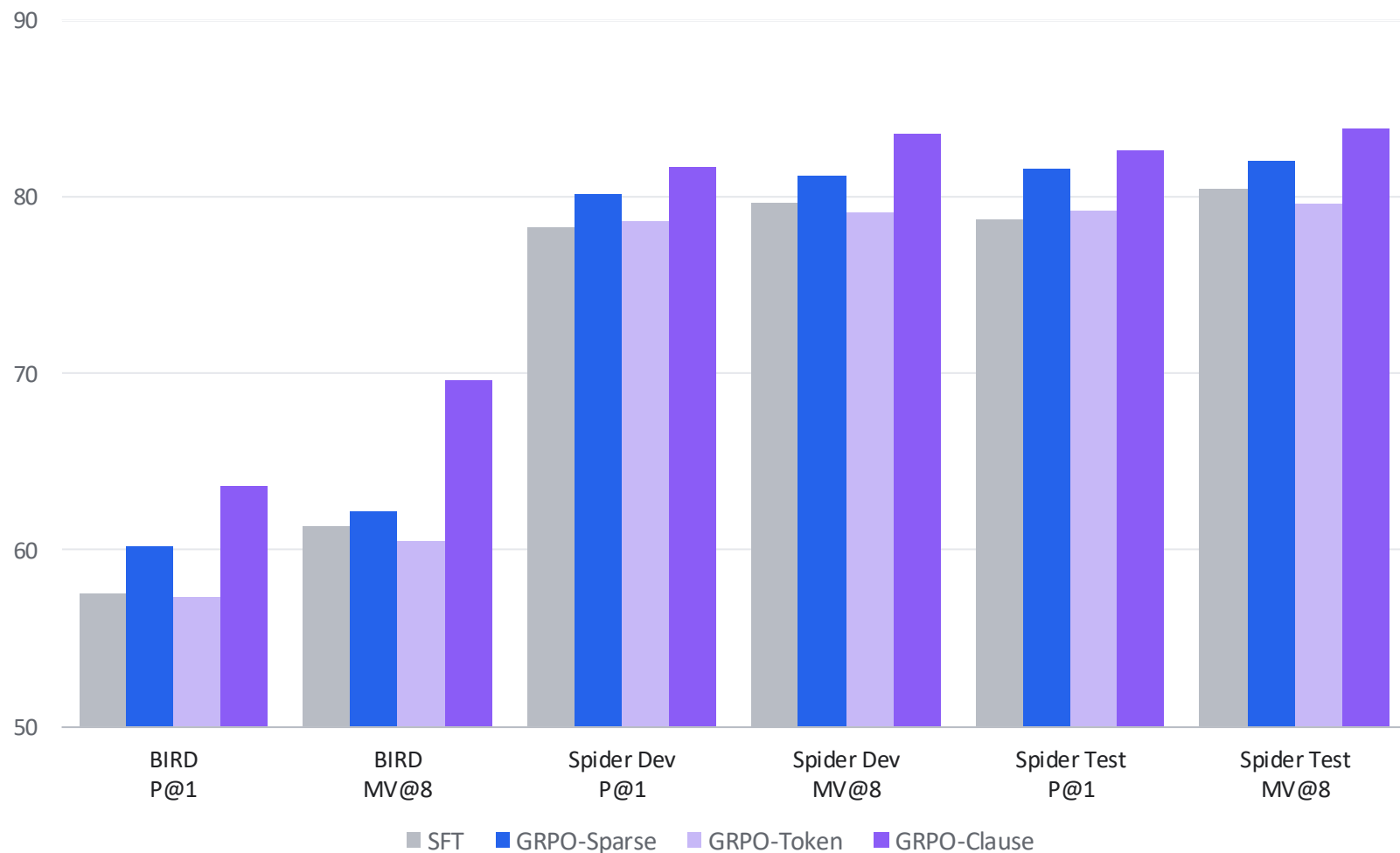
Candidate verification

Delta EX over MV@8

Four black-box LLMs

Backbone-matched reward comparison isolates supervision granularity.

Clause rewards improve end-to-end SQL



6 / 6

matched settings led
by GRPO-Clause

75.94

Average P@1

79.01

Average MV@8

CAPER-9B finds the faulty clause

Held-out failed trajectories · strongest baseline: Qwen3.5-9B self-debug

84.53%

Top-1 localization

Self-debug: 82.78%

96.58%

Hit@3

Self-debug: 93.56%

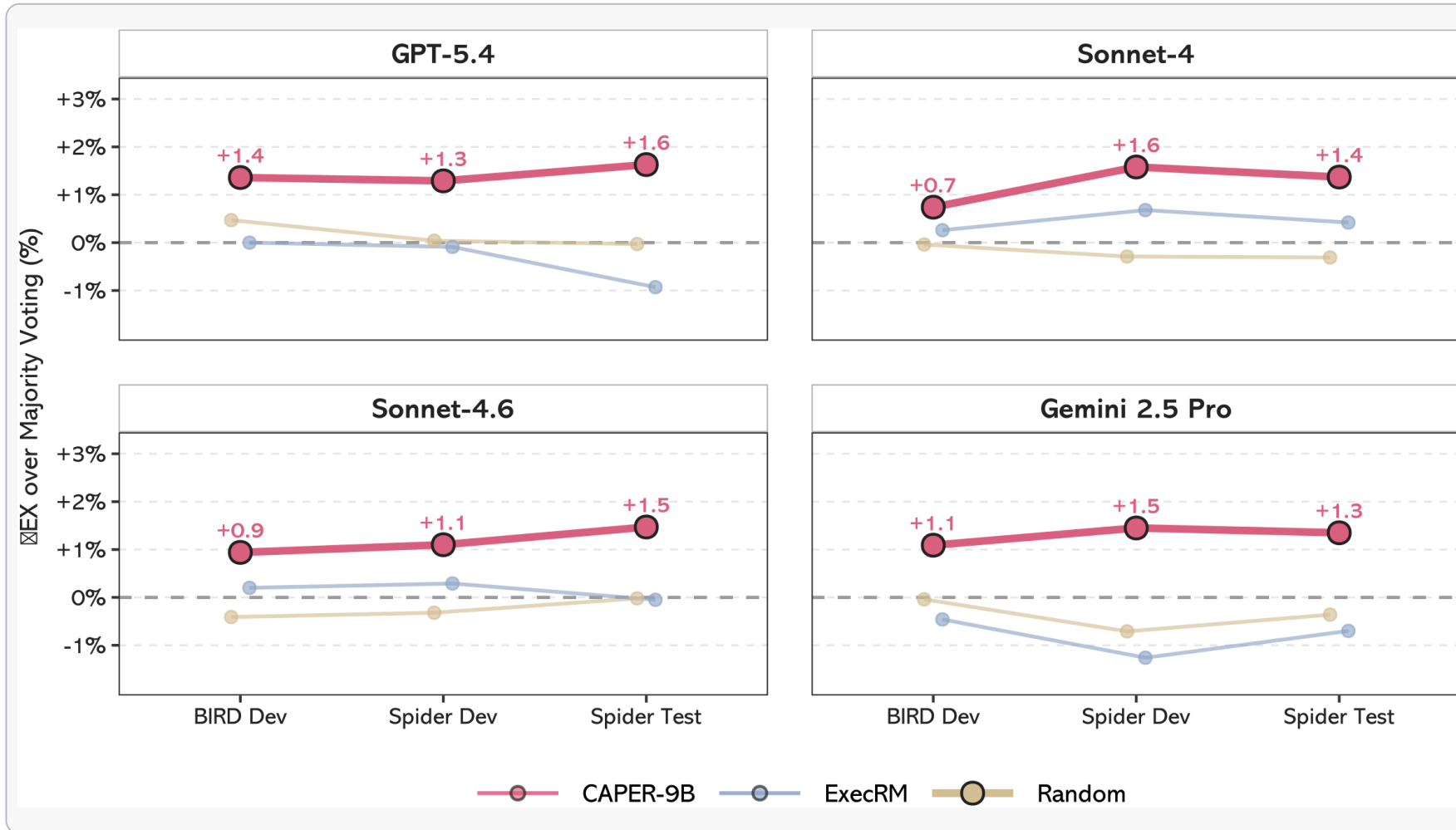
90.60%

MRR

Self-debug: 86.35%

Explicit clause supervision improves every ranking metric.

One verifier improves four black-box generators



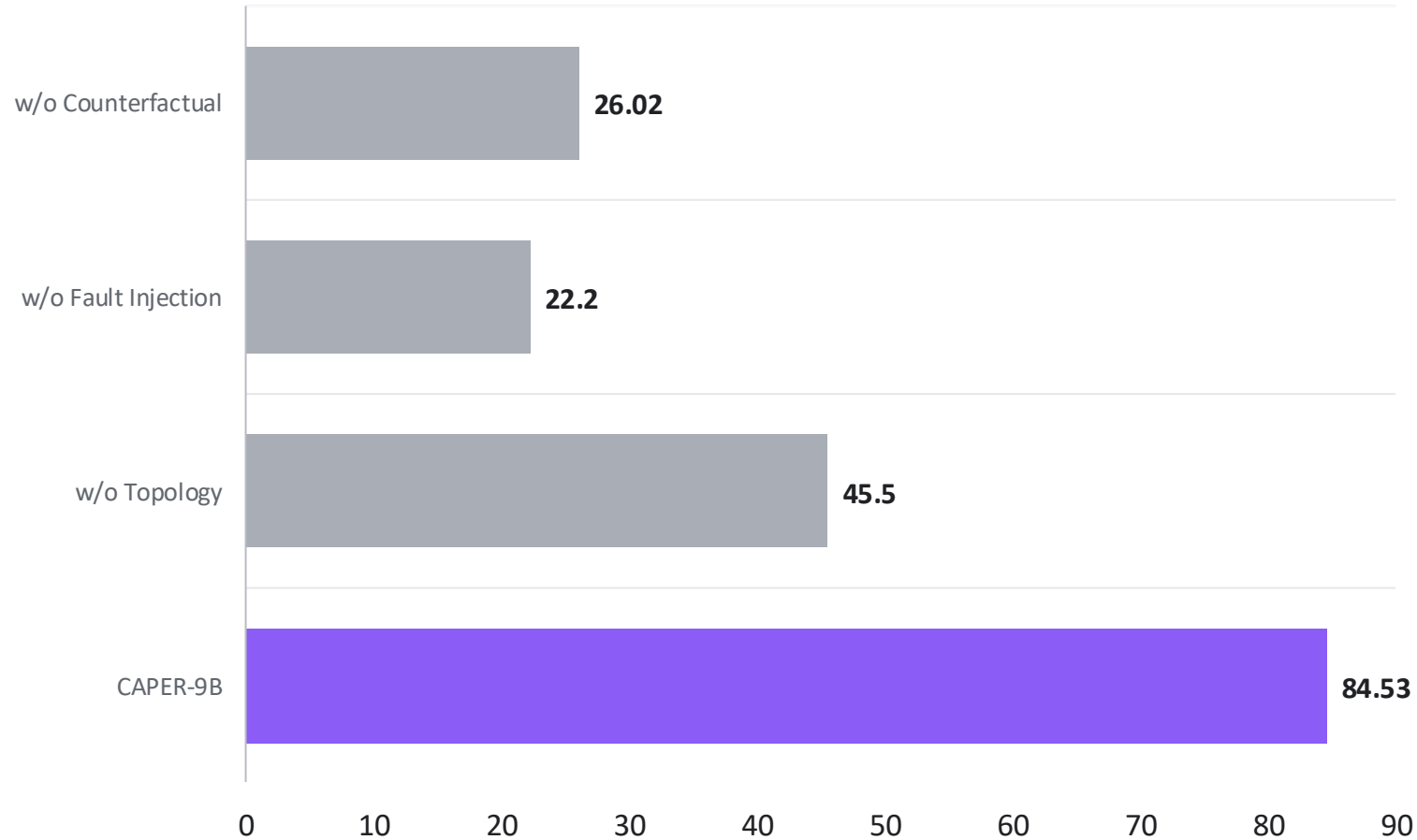
+0.7 to +1.6

**EX points over
Majority Vote@8**

**Same eight
candidates
No generator
updates**

4 models · 3 splits

The full annotation pipeline is necessary



**Topology sharpens
precise ranking.**

**Both annotation paths
are essential.**

**Dense reward alone
is insufficient.**

CAPER still has a bounded scope

01 Dataset scope

Evaluated on BIRD and Spider

02 SQL granularity

Top-level clause units only

03 Domain transfer

New dialects may need tuning

04 Offline cost

Parsing, repair, and execution checks

The current claim is clause-aligned supervision within this tested scope.

Localize before you optimize.

Outcomes	say whether
Clauses	explain where
Counterfactuals	teach why
CAPER	turns attribution into learning

15.3%

relative EX gain over GPT-5.4

84.53%

top-1 failure localization

90.60%

mean reciprocal rank



Questions?

Localize the clause.
Improve the query.