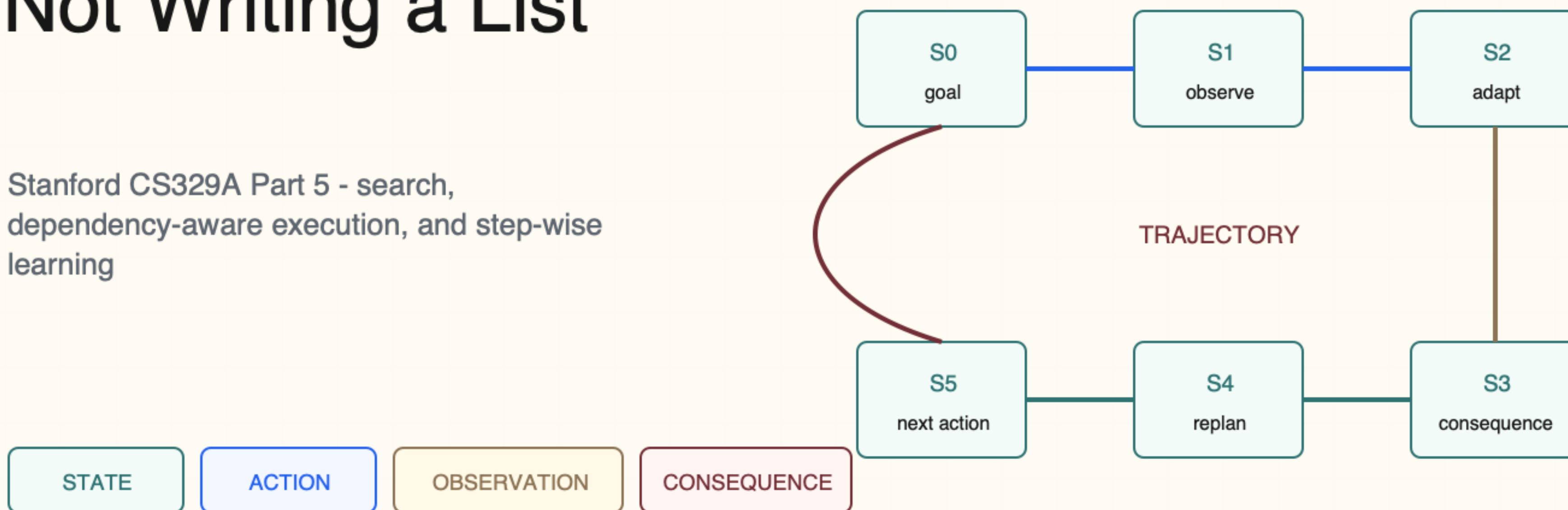


Planning Is Choosing a Trajectory, Not Writing a List

Stanford CS329A Part 5 - search,
dependency-aware execution, and step-wise
learning



Three Levers Improve Three Different Bottlenecks

Search the trajectory, expose independent work, or improve the next-action policy.

01

LATS

search paths

inference

uncertain choice

02

SPRINT

expose width

runtime

independent work

03

SWiRL

learn steps

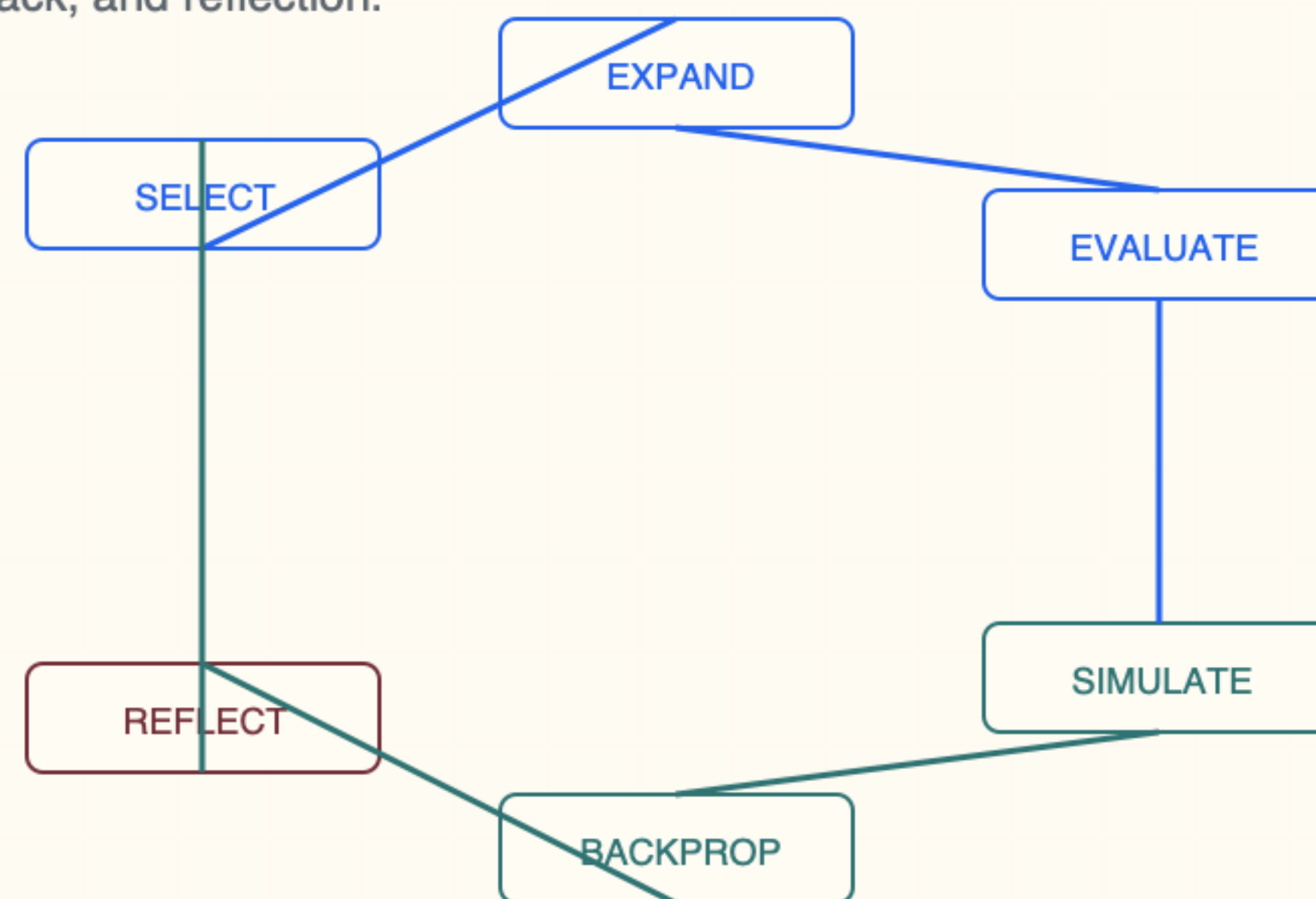
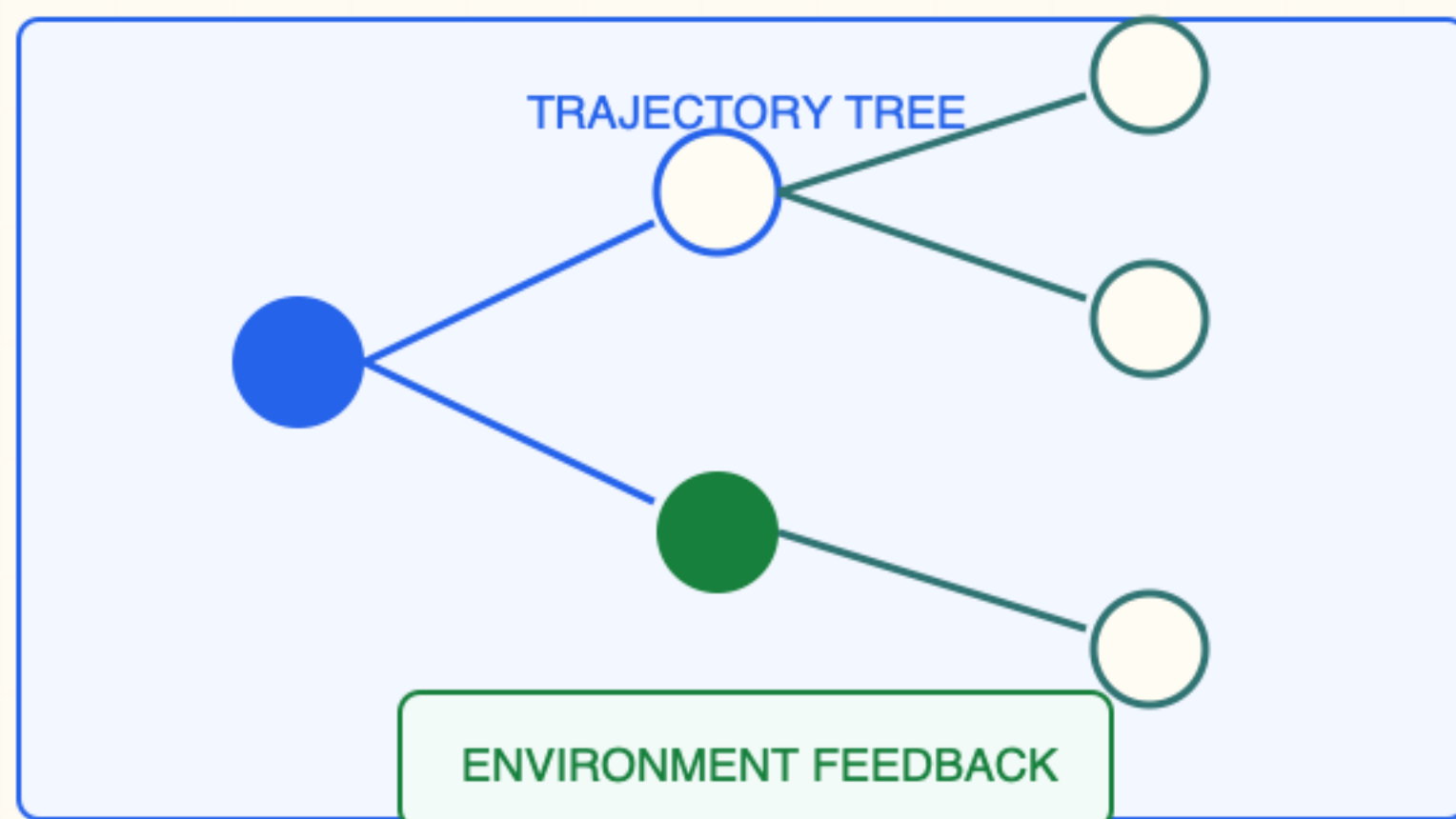
training

weak policy

DIAGNOSE THE BOTTLENECK BEFORE COMBINING THE METHODS

LATS Searches Over Agent Behavior

Six search operations combine model reasoning, environment feedback, and reflection.

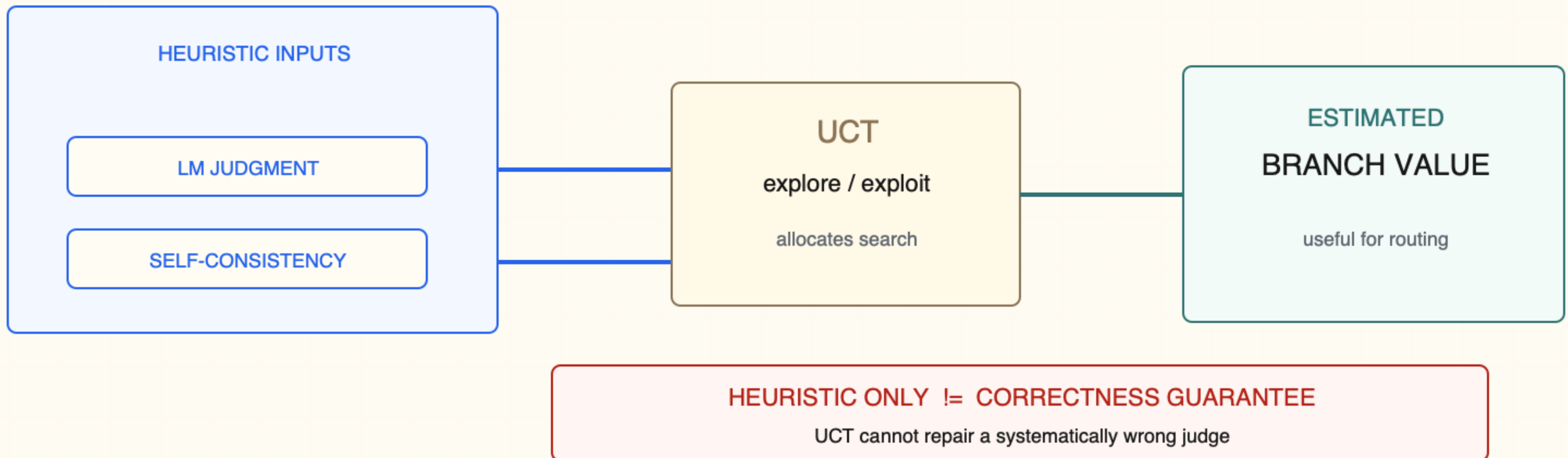


CAVEAT

Reflection can inform another attempt; it does not prove the reflected diagnosis is correct.

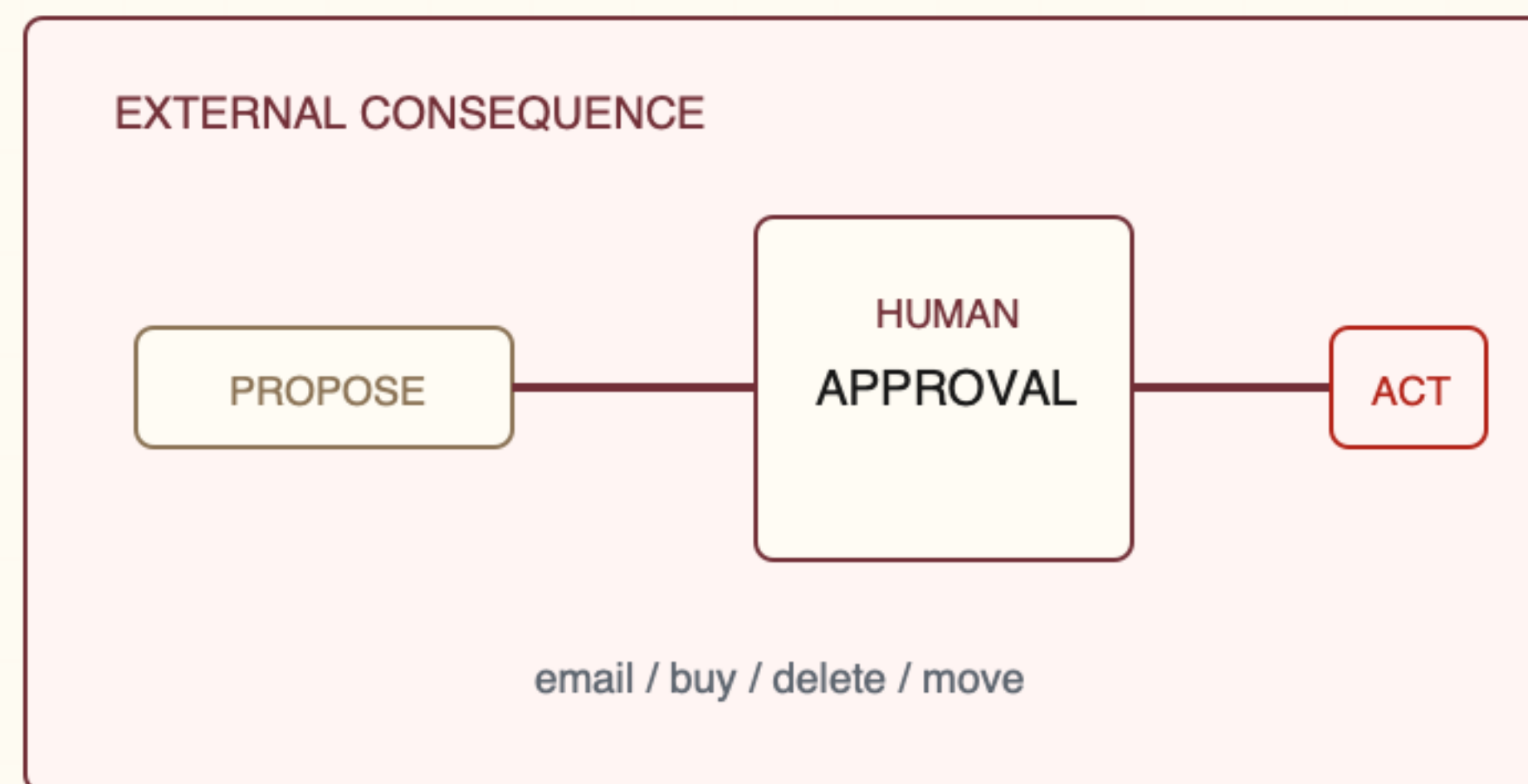
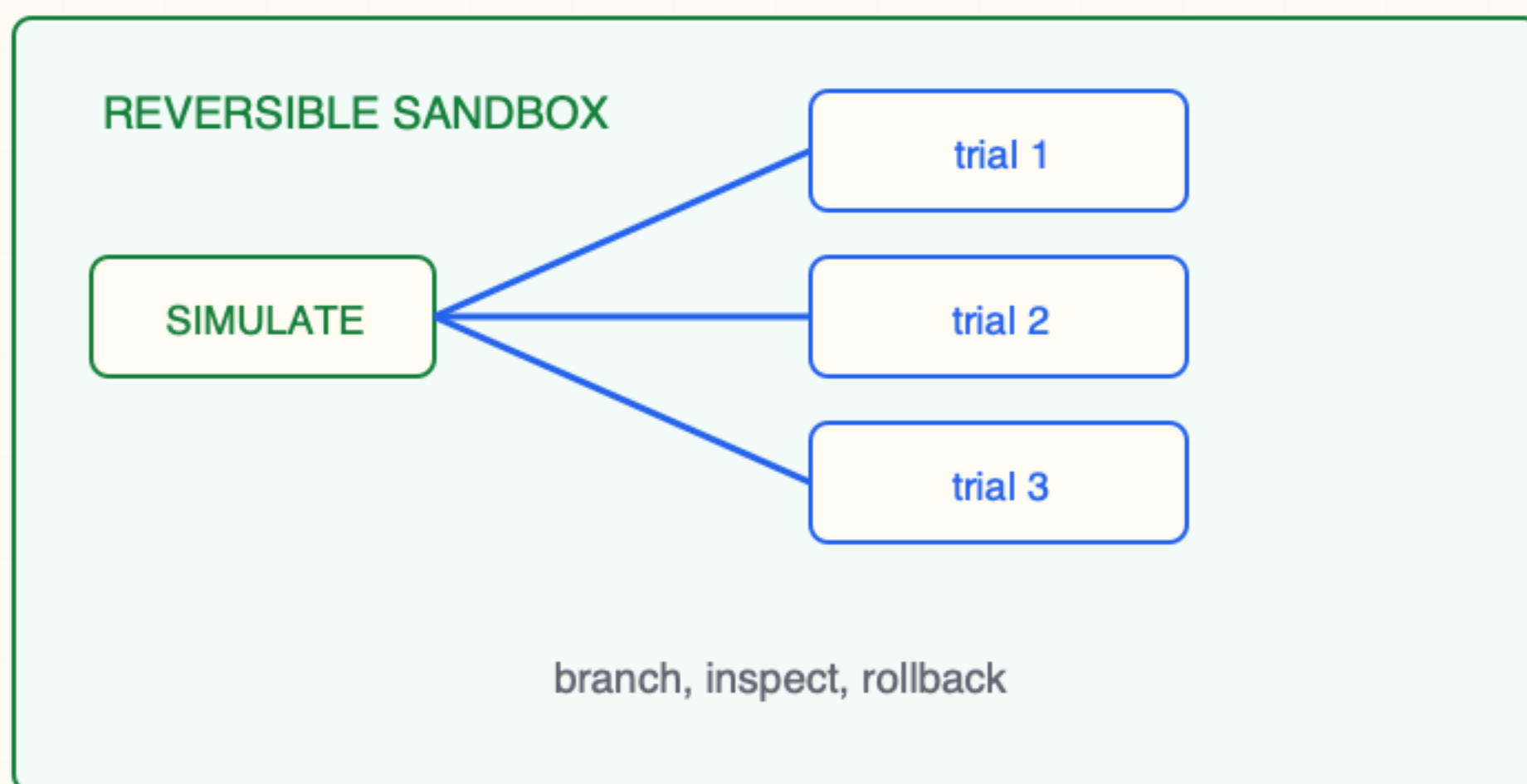
Value Is Estimated, Not Known

UCT organizes exploration around heuristic signals; it does not turn them into truth.



Search Stops at Cost, Safety, and Reversibility

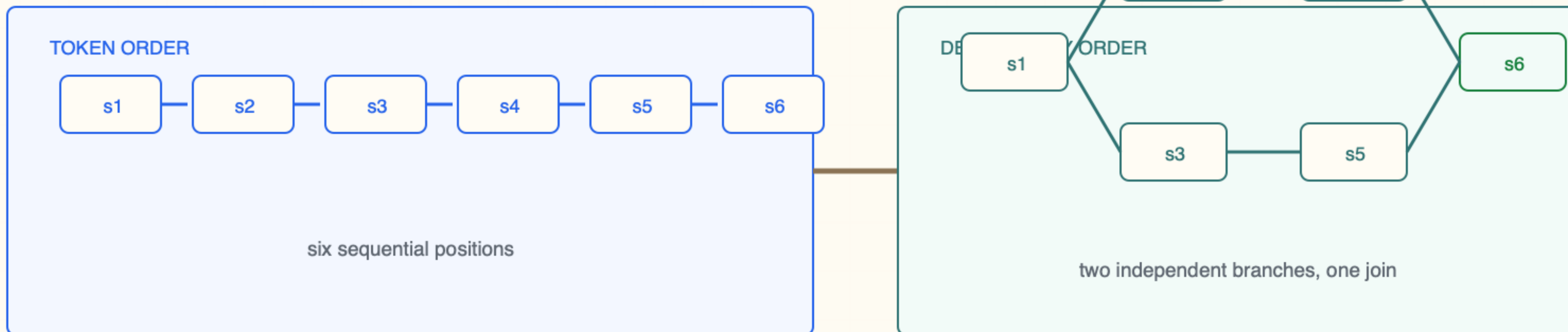
A branch is useful only when it can be evaluated without unacceptable consequences.



BOUNDED BUDGET + EVALUATOR + SAFE ACTION MODEL

A Sequential Trace Can Hide a Dependency DAG

Token order is not always dependency order; independent work can execute concurrently.

**CAVEAT**

Parallelism is valid only when the inferred dependency edges are correct.

SPRINT Compiles Demonstrations Into Parallel Stages

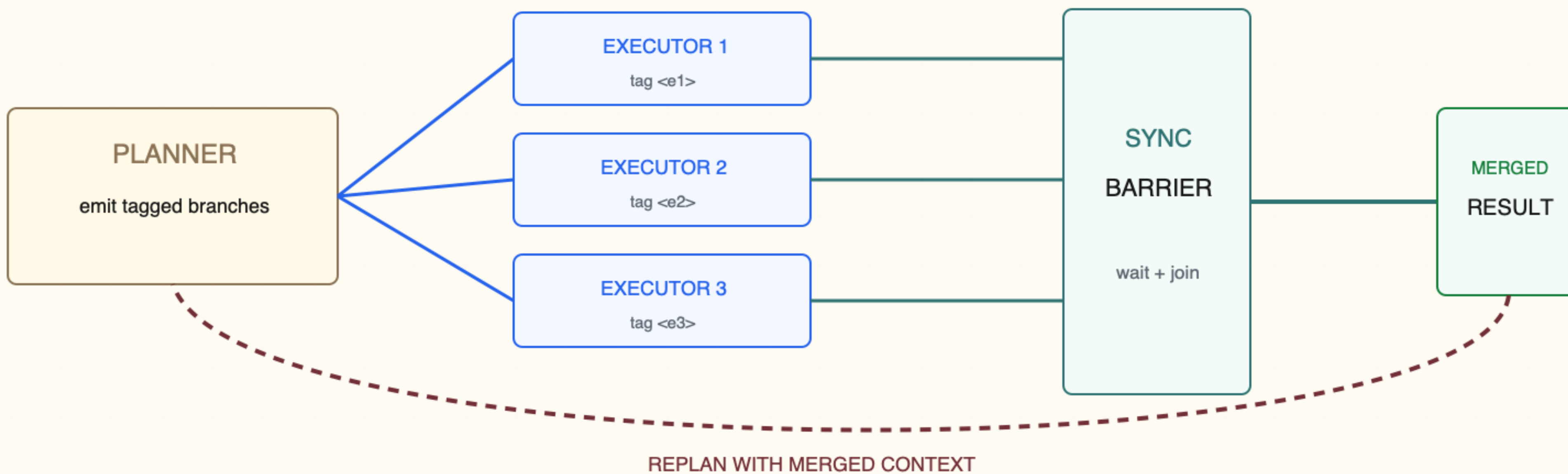
Decomposition and dependency labeling convert a trace into staged planner-executor training data.

**CAVEAT**

Dependency inference is load-bearing: a false independence label teaches an invalid schedule.

Parallel Execution Needs an Orchestrator

The model emits tagged branches; a runtime launches, synchronizes, and rejoins them.



Critical Path Is Not Total Work

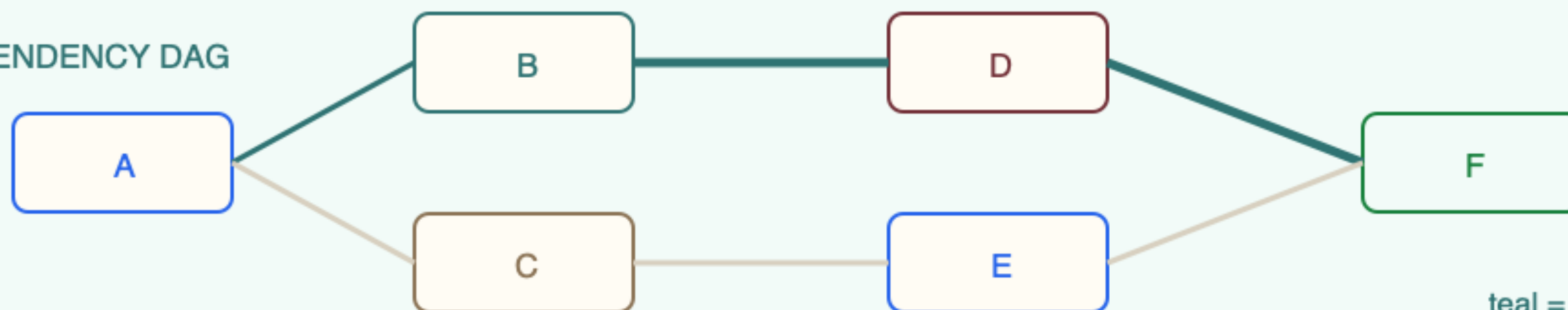
Sequential depth can fall while total tokens, overhead, and straggler latency remain.

TOTAL WORK

sum of all blocks



DEPENDENCY DAG



teal = critical path

WHAT REMAINS

scheduling

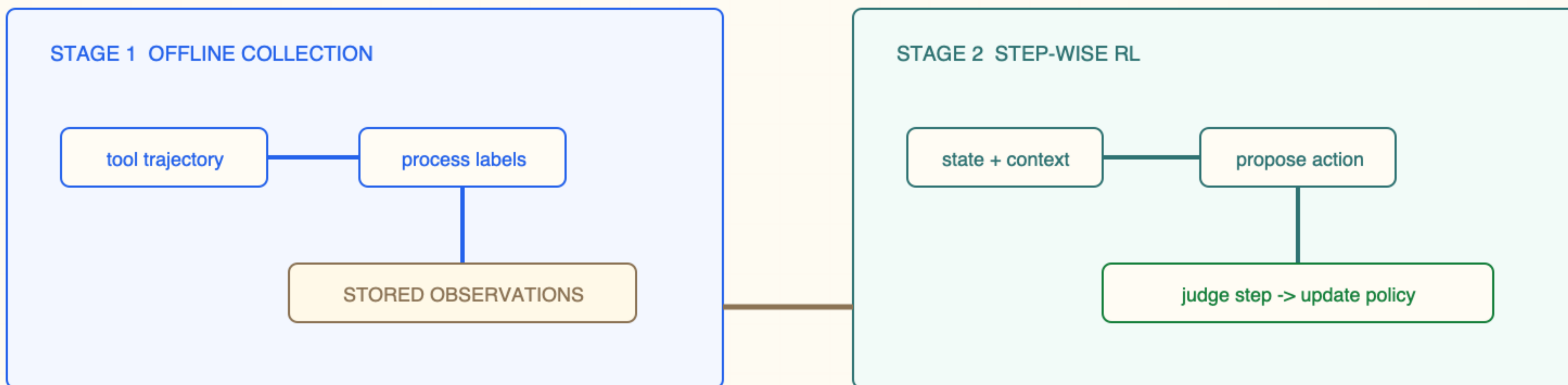
stragglers

short-task overhead

latency must be measured

SWiRL Learns at the Step Level

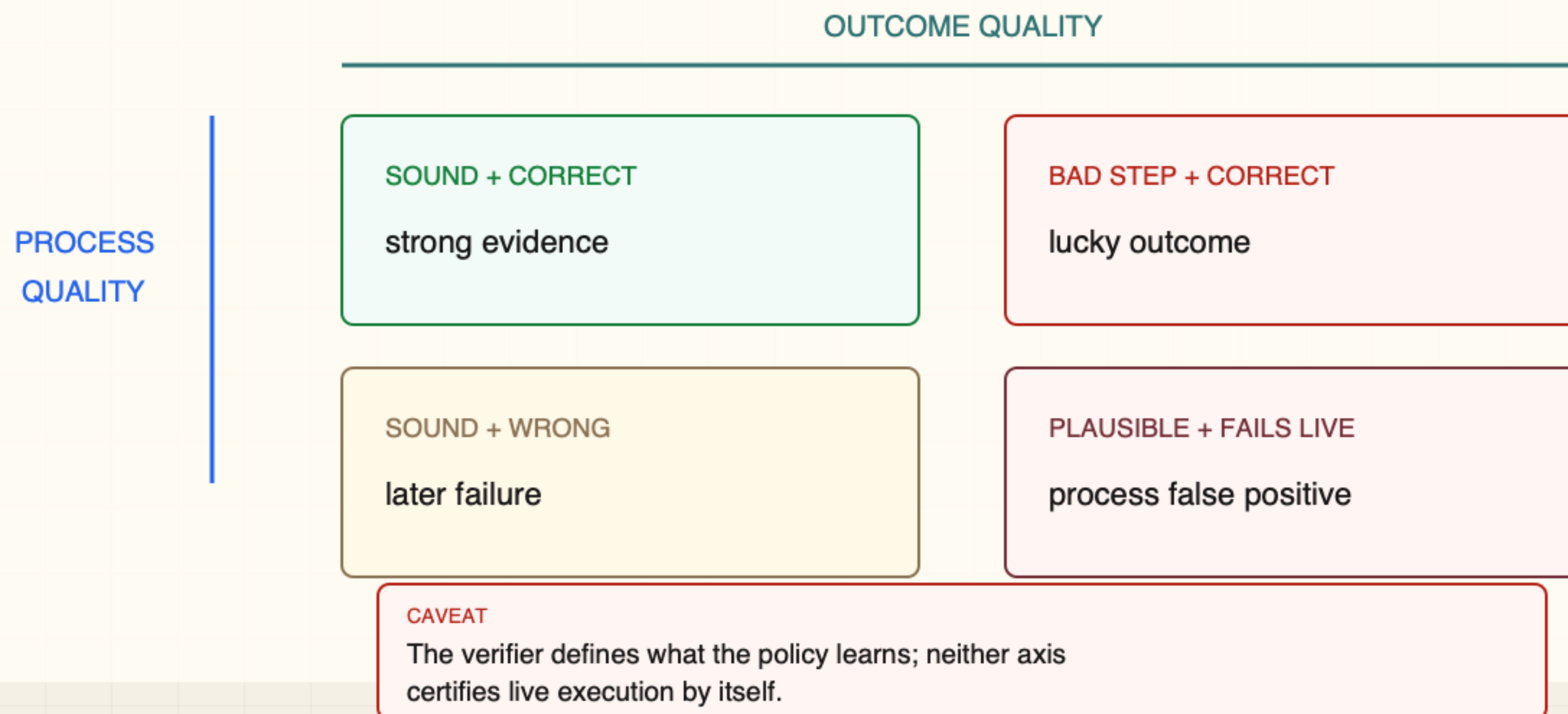
Offline tool trajectories supply intermediate feedback without a live call in every rollout.

**CAVEAT**

Stored feedback avoids live-tool instability but can become stale or off-policy.

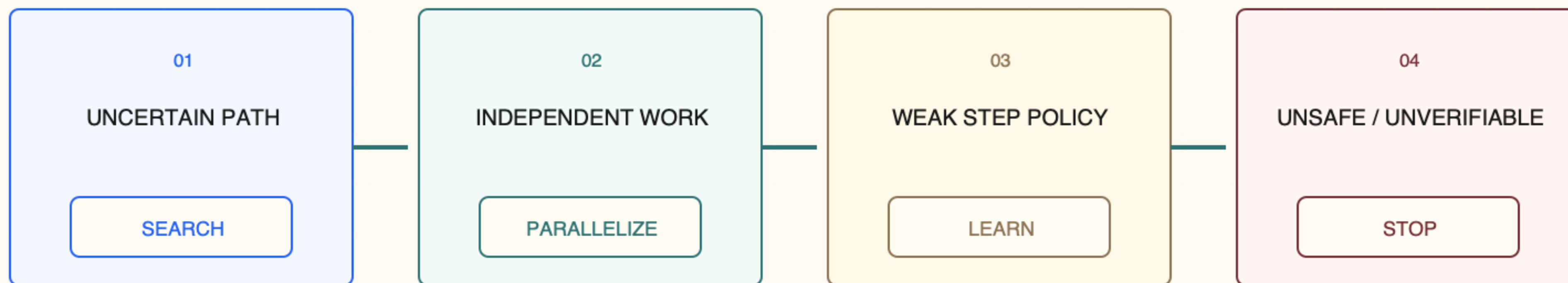
Process and Outcome Signals Fail Differently

A correct answer can hide bad steps; a plausible step can still fail in the live environment.



Choose the Bottleneck Before You Choose the Method

Search, parallelize, learn, or stop: evaluator quality and reversibility gate every route.



GATE EVERY ROUTE: EVALUATOR QUALITY + REVERSIBILITY

No added autonomy is a valid engineering choice.