TraceTriage

A Benchmark for Cost-Aware Stop-or-Continue Decisions in Delayed-Outcome Workflows

Sihang Lei · Yihang Qiu · Xueyan Zhao · Yuwei Wang · Weiqiang Wang

University of Chinese Academy of Sciences
Institute of Computing Technology, CAS · Institute of Automation, CAS

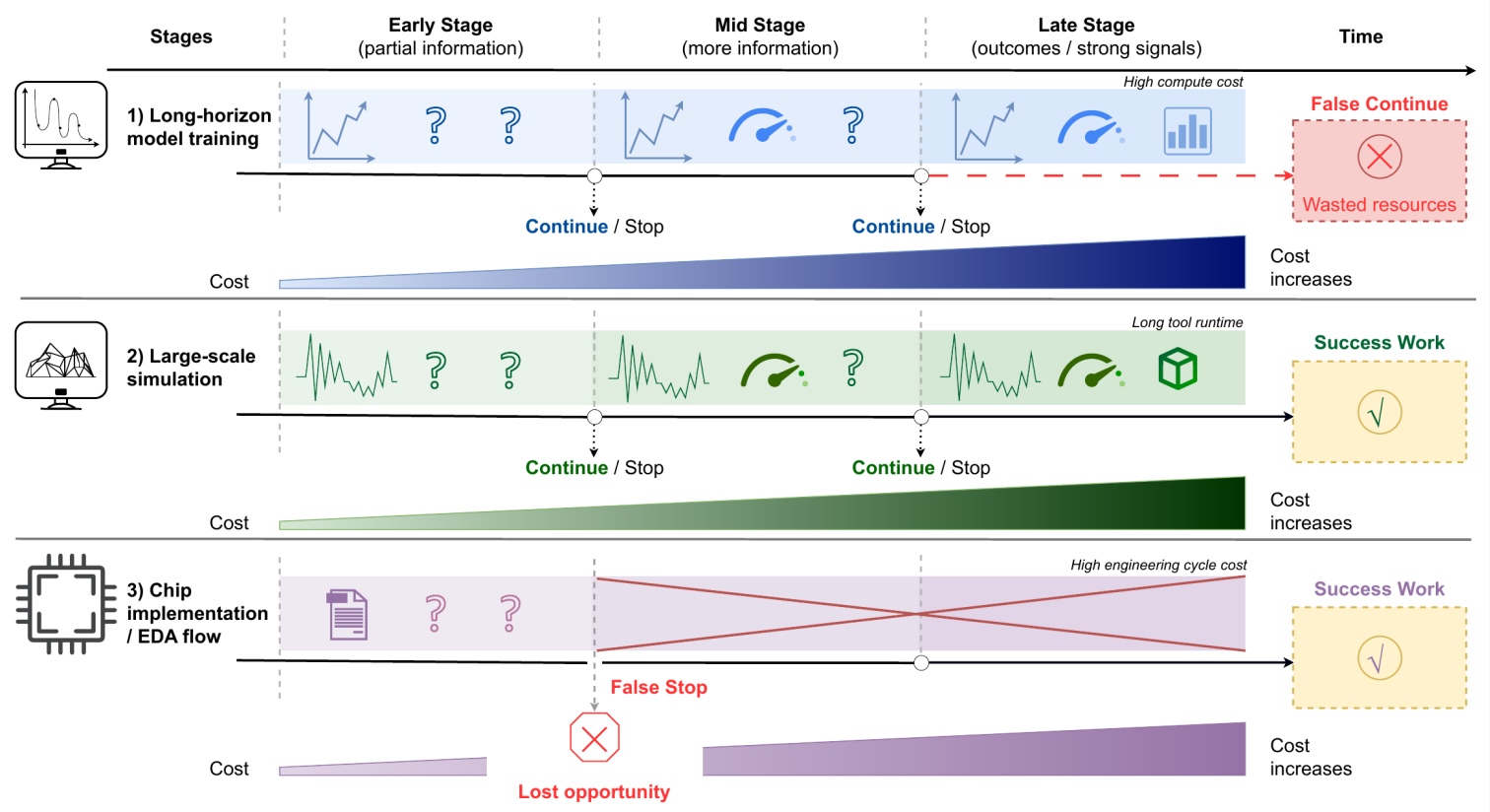
RESEARCH QUESTION

At a workflow checkpoint, should a run stop or continue?

Checkpoint-local evidence → success score → STOP / CONTINUE

Decisions happen before the final outcome

Logs and reports arrive during a run; the success label arrives only after completion.



False continue

Keep spending on a run that ultimately fails.

Cost: wasted downstream resources.

False stop

Terminate a run that would have succeeded.

Cost: a lost successful outcome.

Evaluated instantiation

OpenROAD chip implementation; training and simulation motivate the broader problem.

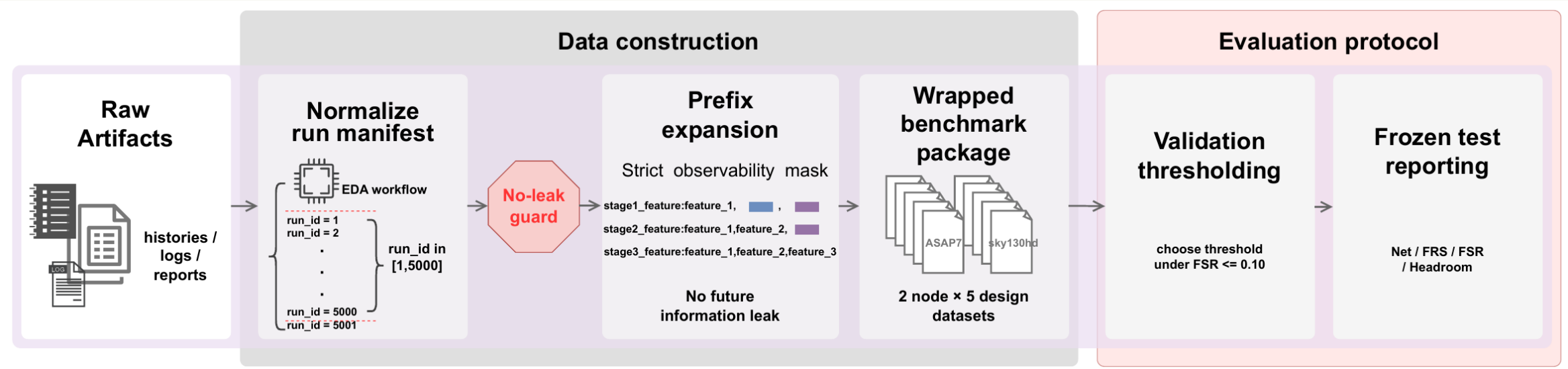
The policy may use only evidence available at the current checkpoint.

Source: TraceTriage paper, Figure 1 and §1

81,041 checkpoint prefixes make early decisions testa

50,000 campaign trials | **31,572** retained runs | **81,041** prefixes

2 process nodes · 5 designs per node · 4 checkpoints: place, cts, grt, drt



Checkpoint-local observability

Metrics, reports, logs, configuration, elapsed time and missingness.

No-leak information boundary

All prefixes from one run share a split. Future-stage evidence and remaining runtime are excluded from model inputs.

Source: TraceTriage paper, Figure 2 and §3.4–3.5

Validation selects the policy; test reports utility once

A single protocol connects success scores to deployable stopping actions.

1 Score

Fit supervised models or
Obtain prompted LLM scores

2 Select

Maximize validation Net
Subject to FSR ≤ 0.10

3 Freeze

Fix the model and threshold;
No test tuning

4 Evaluate

Report test Net, FRS, and FSR

For Qwen, the training label is final-run strict success--not a directly annotated stopping decision

STOP when the predicted success score falls below the selected threshold.

NET UTILITY PER CHECKPOINT

$$\mathrm{Net}(\lambda,\mu)=\frac{1}{n}\sum_{i=1}^{n}[\mathbf{1}(d_i=1,y_i=0)R_i-\lambda\mathbf{1}(d_i=1,y_i=1)R_i-\mu\mathbf{1}(d_i=0,y_i=0)R_i]-c.$$

Default penalty weight: λ = 1. Continue-cost weight: μ = 0

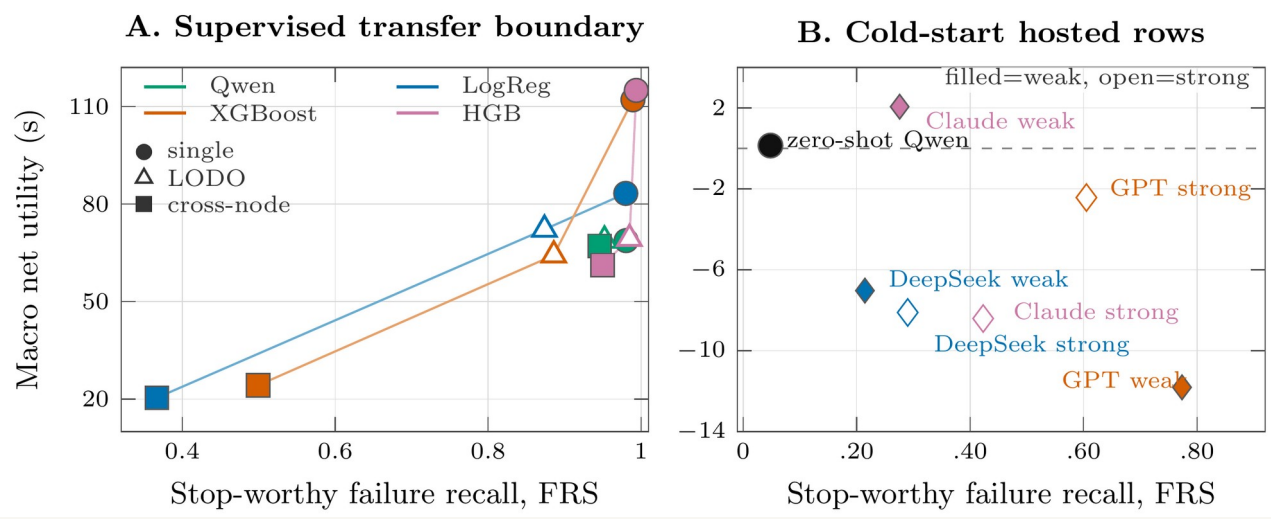
Metric	Interpretation
FSR · false-stop rate	Fraction of successful prefixes incorrectly stopped
FRS · failure recall	Fraction of failed prefixes selected for stopping

Validation risk control is empirical; test FSR can differ. Utility is prefix-level, not an end-to-end rollout simulation.

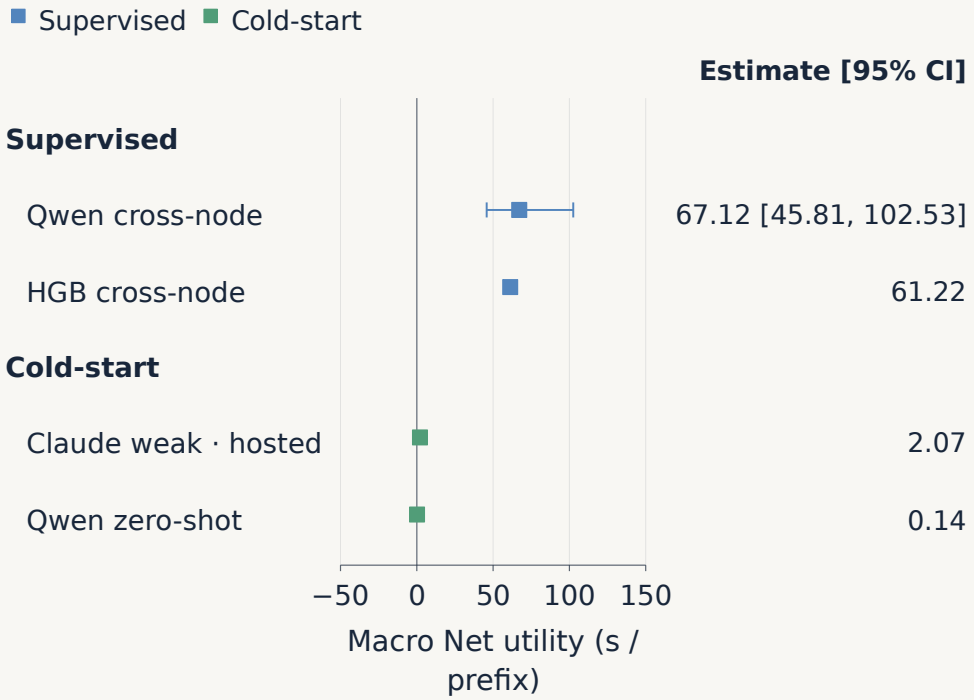
Source: TraceTriage paper, §2.3 and §4

Supervised policies recover runtime savings

Cold-start LLMs show a substantial prediction-to-utility gap under the same policy-selection protocol.



KEY OPERATING-POINT COMPARISON



Qwen only: 95% run-cluster bootstrap CI for the three-seed mean; models and thresholds fixed

Predictive signal does not automatically translate into useful intervention

hosted request latency is charged. Zero-shot Qwen identifies only 4.8% of failing prefixes(FRS = 0.0482), so its small positive Net is not effective triage.

Source: TraceTriage paper, Figure 3, Tables 2 and 4; corrected hosted values

Labels unlock utility; cold-start triage remains open

1 · Risk-constrained selection matters

Validation thresholds connect success scores to stopping decisions under an explicit false-stop constraint.

2 · Labels affect utility

With five completed target-domain runs, HGB rarely stops. Utility rises as more completed-run labels become available.

3 · Transfer is slice-specific

Macro averages can hide missed savings on high-oracle datasets; utility must be inspected by dataset.

COMPLETED-RUN LABEL BUDGET

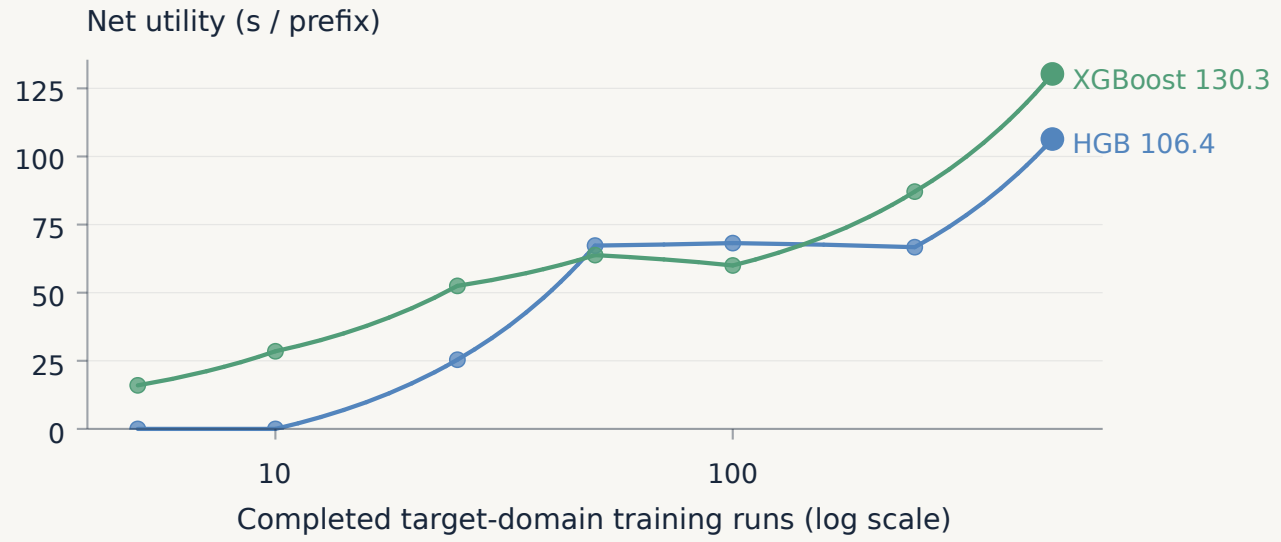


Table 24 · campaign-order training subsets; fixed validation/test.
All labels: HGB 114.9 s · XGBoost 110.5 s

TraceTriage separates prediction quality from deployable, cost-aware intervention.

Scope: OpenROAD · two process nodes · five designs per node · not a signoff or tapeout oracle