Learning Rewrite-Invariant Reasoning with Targeted Alternation Training

Mousa Arraf[1] Ido Guy[2] Kira Radinsky[1]

[1]Technion – Israel Institute of Technology [2]Ben-Gurion University of the Negev

1. Motivation

Meaning-preserving rewrites should not change the answer. In practice, an LLM can solve one wording and fail on another wording of the same task.

Failure mode. A rewrite can push the sampled reasoning trajectory across a point after which the model no longer recovers.

Training issue. Uniform paraphrase augmentation spends budget on easy or redundant rewrites rather than on the variants that repeatedly trigger recoverable-to-irrecoverable transitions.

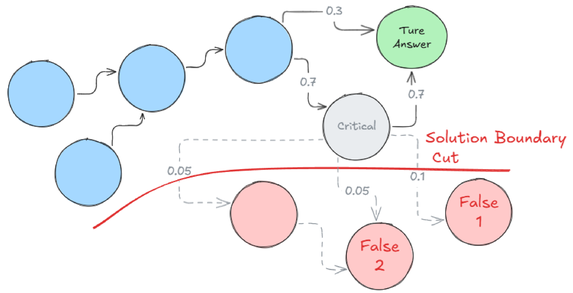
2. Contributions

1. **Reasoning graphs:** aggregate traces from the original question and accepted rewrites into a per-question graph.
2. **Solution Boundary Cut:** identify edges that leave the recoverable region.
3. **Boundary predictor:** learn recurring, model-specific crossing patterns.
4. **Targeted alternation training:** convert high-confidence crossings into contrastive rewrite examples.

Core object: Solution Boundary Cut

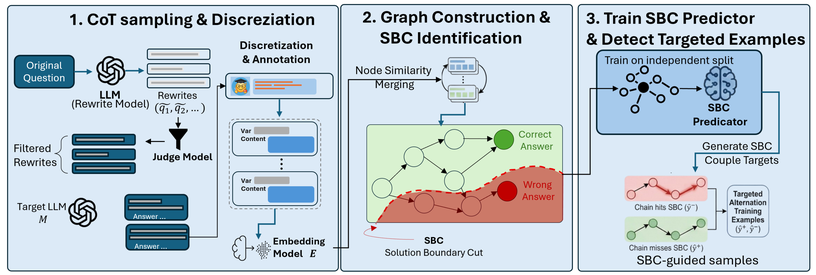
A node is **recoverable** if it can reach a correct terminal answer; otherwise it is **irrecoverable**. The boundary contains edges from recoverable to irrecoverable states:

$$E_{SBC} = \{(u, v) \in E_q : u \in V_{rec},\ v \in V_{irr}\}.$$



SBC localizes the first observed transition after which sampled reasoning does not recover.

3. Paper idea



Pipeline: sample rewrites, build reasoning graphs, detect SBC edges, and train on targeted pairs.

1. **Sample.** Accepted rewrites and target-model traces.
2. **Discretize.** Comparable steps with operation and variable tags.
3. **Build graph.** Merged states and observed transitions.
4. **Locate SBC.** Reverse reachability from correct terminals.
5. **Train.** DPO pairs, or in-context examples for GPT-5.2.

Protocol. Source benchmarks are MMLU-Pro, Big-MATH, and DROP; HLE is transfer-only. Each base question uses five accepted rewrites and $K = 2$ traces for the original plus each rewrite, yielding **12 traces**. Evaluation is leave-one-benchmark-out.

5. Targeted Alternation Training

Targeted alternations are necessary. Training without alternations remains at the base-model level, while training on boundary-crossing rewrites substantially improves performance.

A. Alternations matter: GPT-4.1-mini training ablation

Setting	Big-Math	MMLU-PRO	DROP	Avg.
Base	67.1	76.5	85.9	76.5
No alternations	66.7	76.6	86.1	76.5
With alternations	**74.8**	**78.0**	**91.9**	**81.6**

4. Results

Main result. Targeted alternation training is best or tied-best across nearly all model/benchmark combinations.

Model	Train	Big-Math	MMLU-PRO	DROP
DeepSeek-V2-Lite	Base	52.7± 0.61	60.1± 1.04	48.1± 1.15
	SFT	54.3± 0.61	53.8± 1.19	49.8± 1.28
	Hard	54.4± 0.61	60.3± 1.34	50.6± 1.43
	Random	54.6± 0.62	54.5± 1.27	50.0± 1.35
	TT vote	53.7± 0.61	60.6± 1.43	51.2± 1.51
	Positive	56.0± 0.60	59.6± 1.26	53.1± 1.35
	Targeted	**58.1± 0.59**	**61.8± 1.26**	**54.4± 1.35**
Phi-4	Base	63.4± 0.58	71.1± 0.96	76.8± 0.98
	SFT	65.0± 0.58	70.2± 1.11	76.4± 1.10
	Hard	65.1± 0.59	70.5± 1.26	75.7± 1.23
	Random	65.0± 0.51	70.8± 1.19	75.0± 1.19
	TT vote	64.2± 0.59	71.5± 1.34	82.2± 1.26
	Positive	67.2± 0.57	71.8± 1.19	81.4± 1.23
	Targeted	**68.2± 0.56**	**72.3± 1.18**	**84.2± 1.15**
LLaMA-3.3-70B	Base	61.5± 0.58	71.9± 0.96	67.1± 1.07
	SFT	62.0± 0.57	72.1± 1.11	67.9± 1.13
	Hard	62.2± 0.59	69.7± 1.28	71.5± 1.20
	Random	62.3± 0.59	69.6± 1.28	69.0± 1.16
	TT vote	61.9± 0.52	**73.0± 1.33**	74.1± 1.26
	Positive	64.4± 0.53	71.8± 1.19	74.2± 1.28
	Targeted	**65.3± 0.58**	72.8± 1.18	**76.5± 1.24**
GPT-4.1-mini	Base	67.1± 0.51	76.5± 0.91	85.9± 0.87
	SFT	69.9± 0.56	76.7± 1.05	86.4± 0.96
	Hard*	70.0± 0.56	76.7± 1.27	86.5± 1.23
	Random	70.5± 0.56	77.2± 1.18	86.0± 1.14
	TT vote	69.1± 0.57	77.0± 1.36	85.0± 1.34
	Positive	70.5± 0.56	77.8± 1.17	90.1± 1.09
	Targeted	**74.8± 0.53**	**78.0± 0.86**	**91.9± 0.89**

Accuracy (%) with estimated 95% confidence intervals.

6. Boundary Prediction and Graph Construction

B. Boundary prediction: graph structure improves SBC detection

Predictor	DeepSeek	GPT	LLaMA	Phi
Logistic regression	70.4	70.6	69.9	72.3
Graph Transformer	75.4	74.6	73.7	77.2
GNN	**80.2**	**78.1**	**76.9**	**83.8**

C. Node annotations improve graph construction

Setting	No var. tag	Var. tag	△	Best
No operation label	0.87	0.91	+0.04	
Operation label	0.92	**0.94**	+0.02	**yes**