THEME BASED RESEARCH SCHEME T32-615/24-R
Enhancing Hong Kong's Role in Sustainable Supply Chain Finance via Technology Transformation

NeurIPS 2026

From Expert Knowledge to Optimization Modeling:

Prototype-Based Data Synthesis and Logical Reinforcement Learning

Bo Hu[1] Qingcan Kang[2]* Qian Liu[1] Mingxuan Yuan[2]

[1] Department of Industrial Engineering and Decision Analytics, HKUST
[2] Huawei Noah's Ark Lab, Hong Kong

HUAWEI NOAH'S ARK LAB

PAPER

Research question and contributions

Research gap

LLMs often construct optimization models from scratch. Expert modelers first recognize a canonical problem structure, preserve its bottleneck constraints, then refine the model for the task.

Research question

Can human expert knowledge improve LLMs' OR modeling ability?

1 Prototype-anchored data

Seventeen expert OR prototypes guide problem and solution synthesis; bottleneck constraints are preserved and variables precede expressions.

Training data and reward signal

100 curated pairs augment OR-Instruct; synthetic-problem trajectories train the reward model.

2 Process-aware RL

Logic-TGRPO adapts an SFT-initialized 8B model on unlabeled OR problems using answer consensus, trajectory logic and prototype fidelity.

Empirical evidence

80.6% macro-average accuracy after transductive adaptation: +5.8 pp vs. frozen SFT + Thinking (74.8%).

Two-stage modeling framework

17 OR prototypes → **Prototype-guided SFT** → **Logic-TGRPO adaptation**

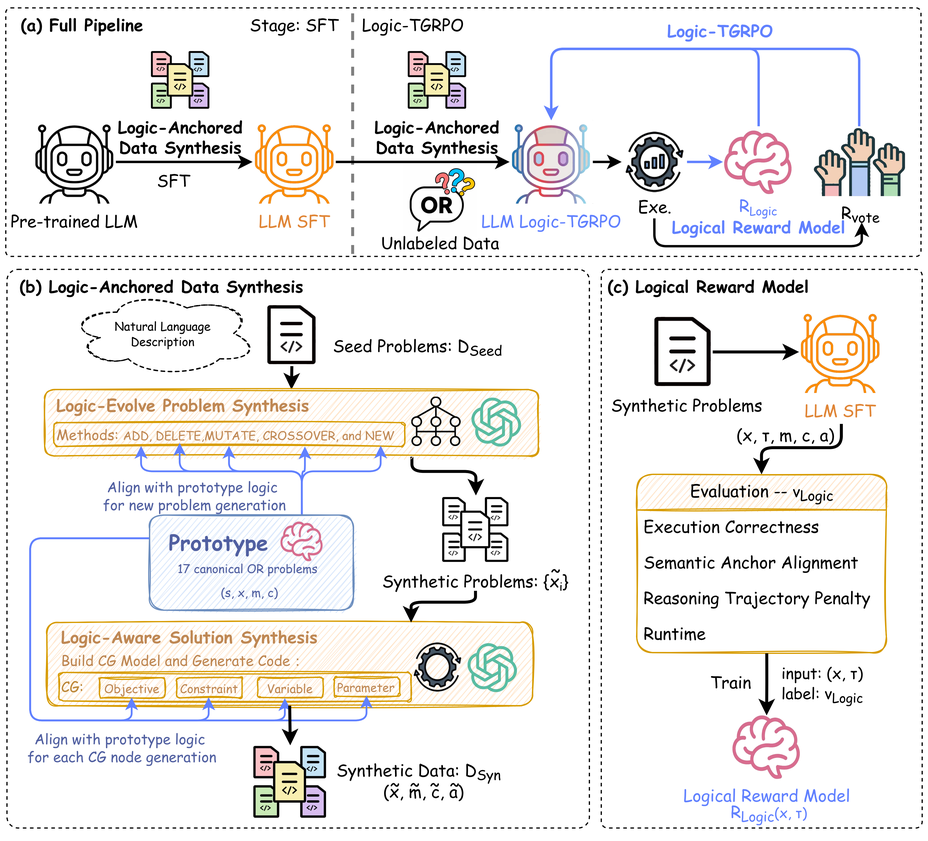

Figure 1. Logic-Anchored Data Synthesis and Logic-TGRPO

Stage 1 Prototype-guided SFT

100 curated synthetic pairs augment OR-Instruct to initialize Qwen3-8B.

Two synthesis anchors

Preserve bottleneck constraints in problems; define variables before objectives and constraints in solutions.

Stage 2 Logic-TGRPO

Sample four trajectories per input; optimize raw rewards without advantage normalization. For outputs passing format/code gates:

Rtotal = Rvote + Rlogic + Rproto

Rvote Answer consensus

Agreement with the group's majority answer.

Rlogic Trajectory quality

Qwen3-1.7B scores trajectories, trained on execution, semantic and runtime feedback.

Rproto Prototype fidelity

Score core structural fidelity in [0, 1]; no prototype match gives Rproto = 0.

Format/code gate: bad format −0.5; other failures 0. GPT-5-mini judges semantic alignment and prototype fidelity.

Label-free transductive adaptation

100 unlabeled statements from 1,510 evaluated test inputs (~6.6%). No gold formulations, code, objective values or answers enter RL. Freeze for greedy pass@1.

Experiment results

Seven-benchmark macro-average accuracy **+5.8 pp**

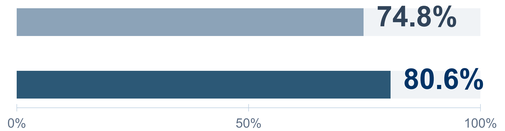

Table 1. Qwen3-8B variants · solution accuracy (%) · AVG: macro

Model / configuration	NL4Opt	MAMO EasyLP	MAMO ComplexLP	NLP4LP	Complex OR	Industry OR	ReSocratic	AVG
Frozen models / SFT and thinking references								
Qwen3-8B	76.5	87.3	25.2	74.7	16.7	31.0	77.9	**55.6**
SFT, no synthetic data	85.9	93.9	45.0	87.6	44.4	45.2	78.9	**68.7**
SFT, synthetic only	80.3	90.1	55.0	79.2	44.4	47.6	76.4	**67.6**
SFT, combined	90.1	93.0	51.4	91.0	33.3	45.2	82.1	**69.4**
Native thinking	79.3	84.2	38.7	93.3	33.3	42.9	82.6	**64.9**
SFT + Thinking	94.4	93.0	53.2	96.6	44.4	52.4	89.6	**74.8**
Label-free transductive adaptation								
Logic-TGRPO, no synthetic	91.5	94.3	57.7	98.9	55.6	59.5	90.8	**78.3**
Logic-TGRPO (Full)	**92.5**	**93.8**	**66.7**	**97.2**	**61.1**	**61.9**	**91.3**	**80.6**

Hard-benchmark gains vs. frozen SFT + Thinking

MAMO ComplexLP +13.5 pp · ComplexOR +16.7 pp · IndustryOR +9.5 pp

Ablation study

Table 4. Macro-average accuracy (%) · bold: best in each block

SFT: prototype anchors	
Combined	69.4
Without problem prototype	69.5
Without solution prototype	**70.2**
Without both	67.7

RL: logical rewards	
Full framework	**80.6**
Without Rproto	79.9
Without Rlogic	78.5
Without both rewards	78.9

SFT anchors have task-dependent effects. Removing Rlogic reduces MAMO ComplexLP from 66.7% to 59.5%.

Inference latency: 7.7 s vs 17.5 s (OptiTree)

Table 3: published OptiTree reference; different backbones; excludes adaptation.

Next steps

Explore compact LLMs, automated prototype discovery and step-level rewards for efficient optimization modeling; evaluate accuracy, latency, total compute and beyond-library generalization.

neurips.cc/virtual/2026/poster/153007