OpenSearcher

NeurIPS 2026

Democratizing Deep Search through a Fully Offline Pipeline with Programmatic Verification

Zheng Chu[1*] Xiao Wang[3] Huiming Fan[1] Qianyu Wang[1] Zhuoyao Wang[1]
Xiang Cheng[3] Ming Liu[1,2] Bing Qin[1,2†] XingYu[3]

[1] Harbin Institute of Technology, Harbin, China
[2] Peng Cheng Laboratory, Shenzhen, China [3] Xiaohongshu Inc.

{zchu, qinb}@ir.hit.edu.cn

Code & resources

* Work completed during an internship at Xiaohongshu Dots. † Corresponding author.

Why train deep search agents offline?

Three practical barriers

Verification

Complex reasoning graphs make answer uniqueness difficult to check at scale.

Online dependency

Commercial search APIs add recurring cost during synthesis and RL.

Reproducibility

Closed data-generation pipelines are difficult to reproduce and extend.

OpenSearcher

Programmatic QA verification

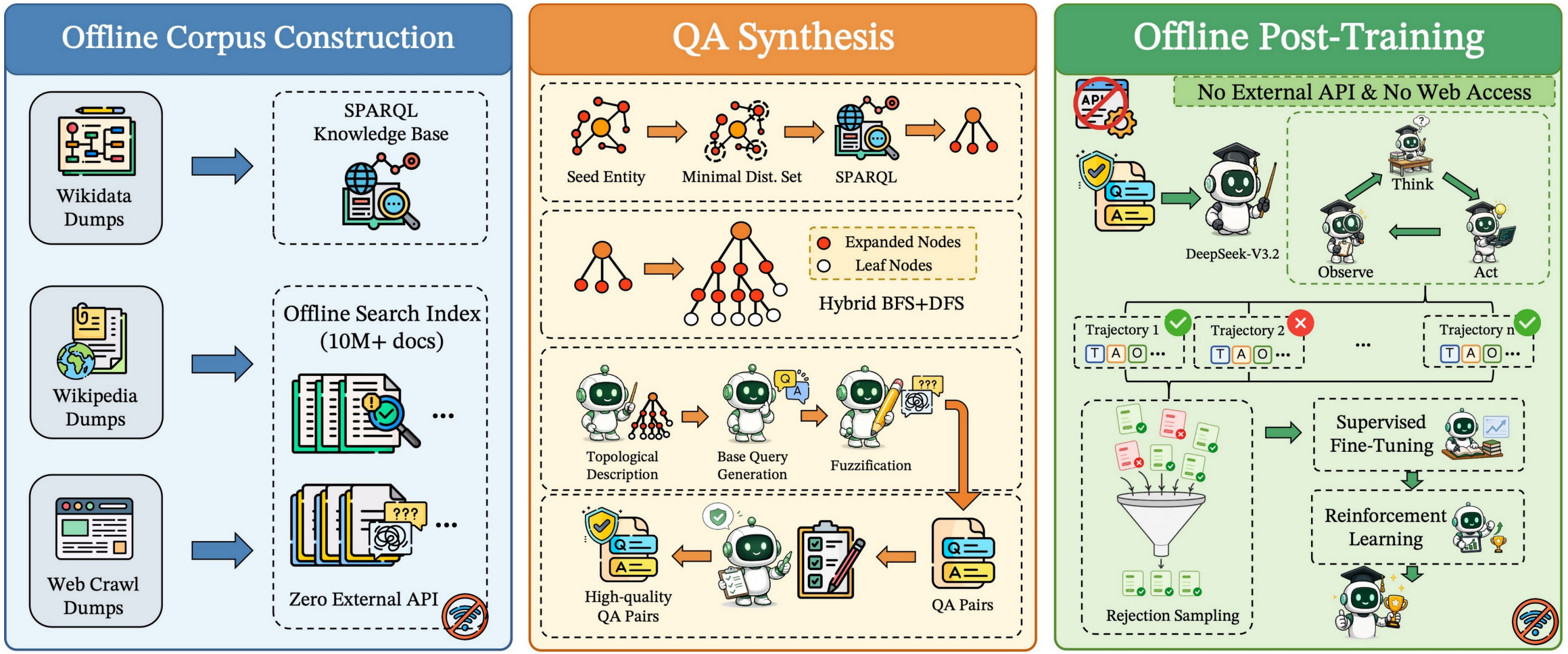
Knowledge-graph constraints identify a unique answer before question generation.

Fully offline synthesis and training

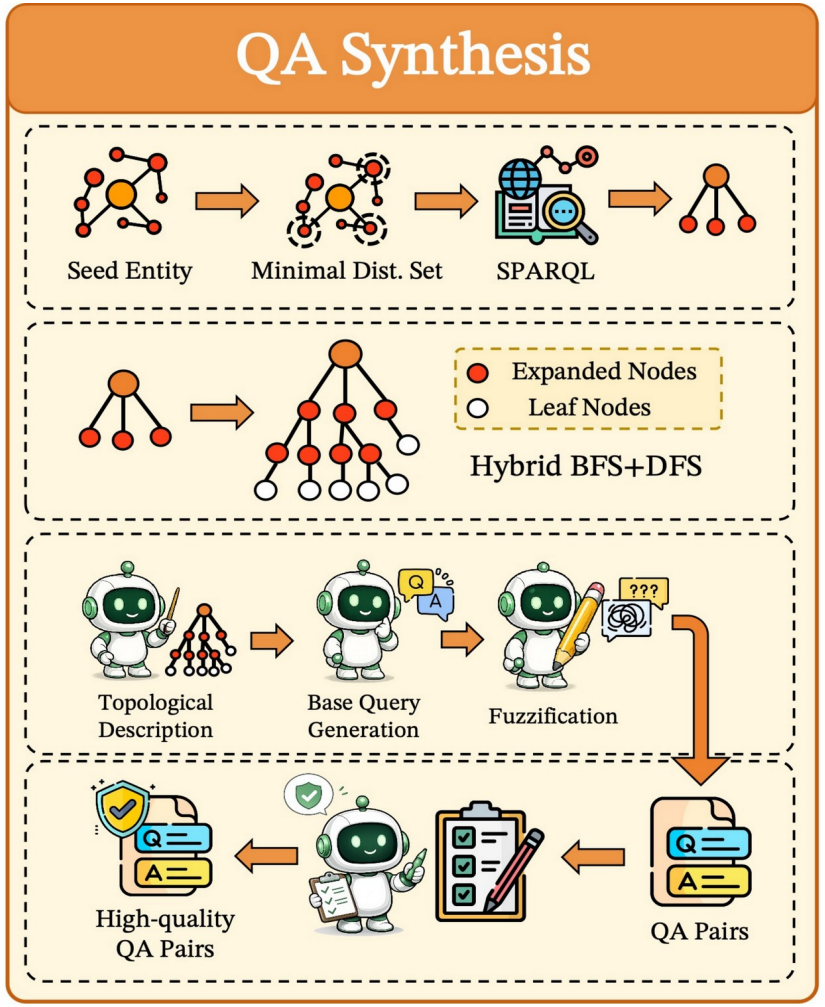
Public dumps and local retrieval support trajectory synthesis, SFT and RL.

A fully offline synthesis and training pipeline

Downloadable corpora, verified questions and a local search environment



Programmatically verified question synthesis



1 Minimal distinguishing set

Find relation–neighbor constraints that identify exactly one target entity.

2 Recursive reasoning DAG

Expand neighbors using a hybrid breadth-first / depth-first strategy.

3 Progressive generation

Build descriptions, write the query, then fuzzify searchable clues.

4 Final language-quality filter

Check factual consistency, answer leakage and ambiguity once per sample.

SPARQL(D(e)) = {e}

Offline trajectories, SFT and reinforcement learning

Local search environment

10M+ documents from public dumps
Tools: Search, Open and Python

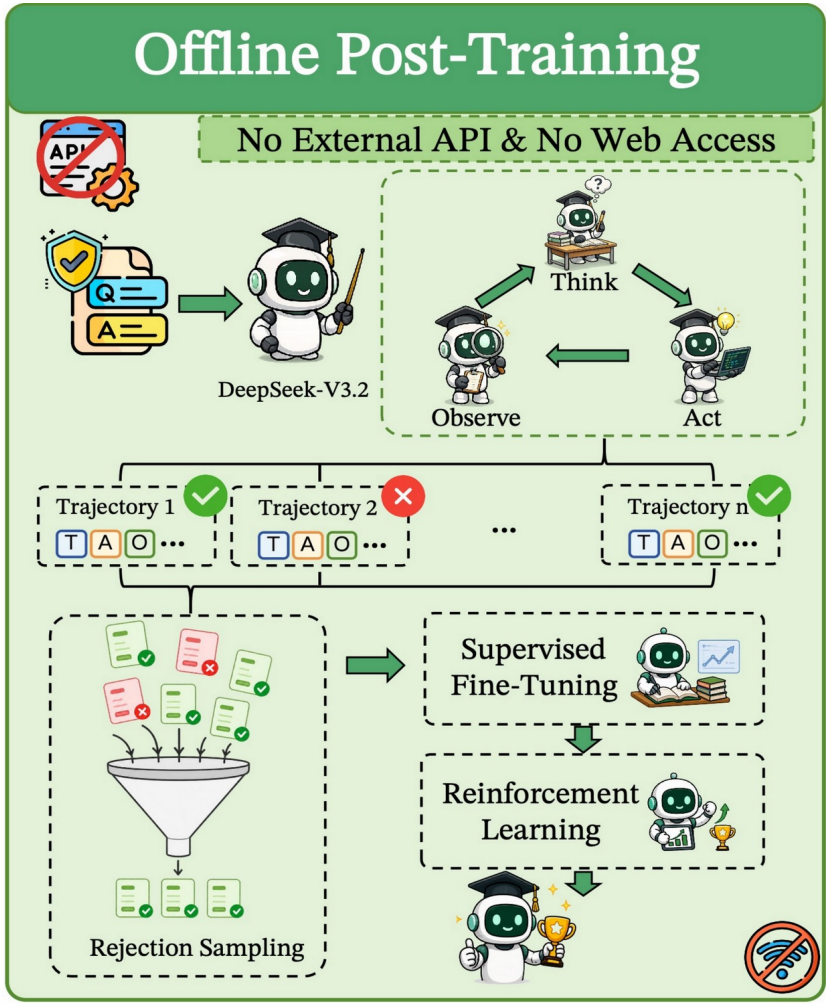
Teacher trajectories

DeepSeek-V3.2 generates multi-turn rollouts.
Rejection sampling retains correct answers.

SFT cold start, then GRPO

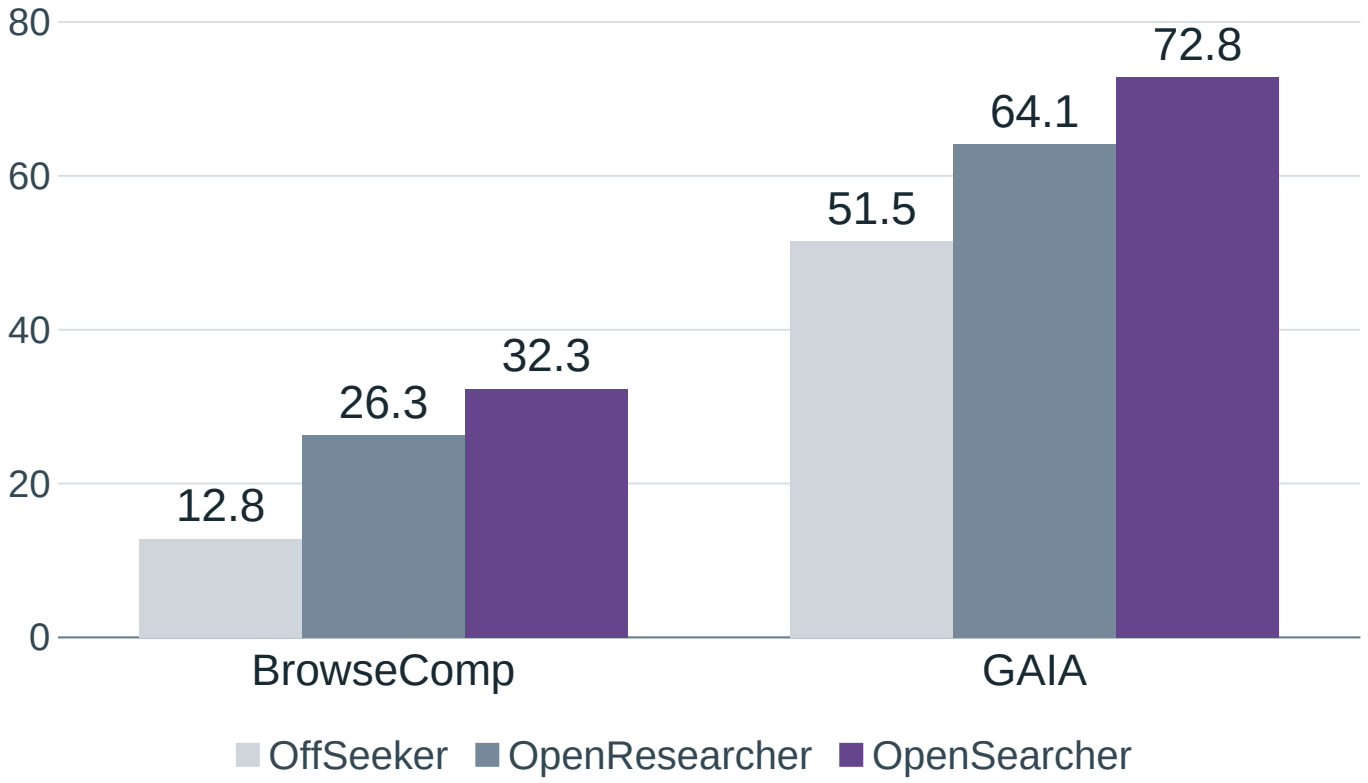
Base model: Qwen3-30B-A3B-Thinking
Outcome rewards; tool responses masked from loss.

Synthesis + training: offline. Evaluation: online search APIs.



Main benchmark results

Offline-method comparison, accuracy (%)



Additional benchmarks

Benchmark	Ours
HLE	**30.0**
BrowseComp-zh	**34.3**
xBench-DS	**66.0**
xBench-DS-2510	**35.0**

32.3% BrowseComp **72.8% GAIA**

Offline RL improves all six benchmarks

Benchmark	SFT	SFT + RL	Gain (pp)
BrowseComp	26.2	32.3	**+6.1**
GAIA	61.2	72.8	**+11.6**
HLE	25.0	30.0	**+5.0**
BrowseComp-zh	20.8	34.3	**+13.5**
xBench-DS	60.0	66.0	**+6.0**
xBench-DS-2510	29.0	35.0	**+6.0**

+13.5

BrowseComp-zh

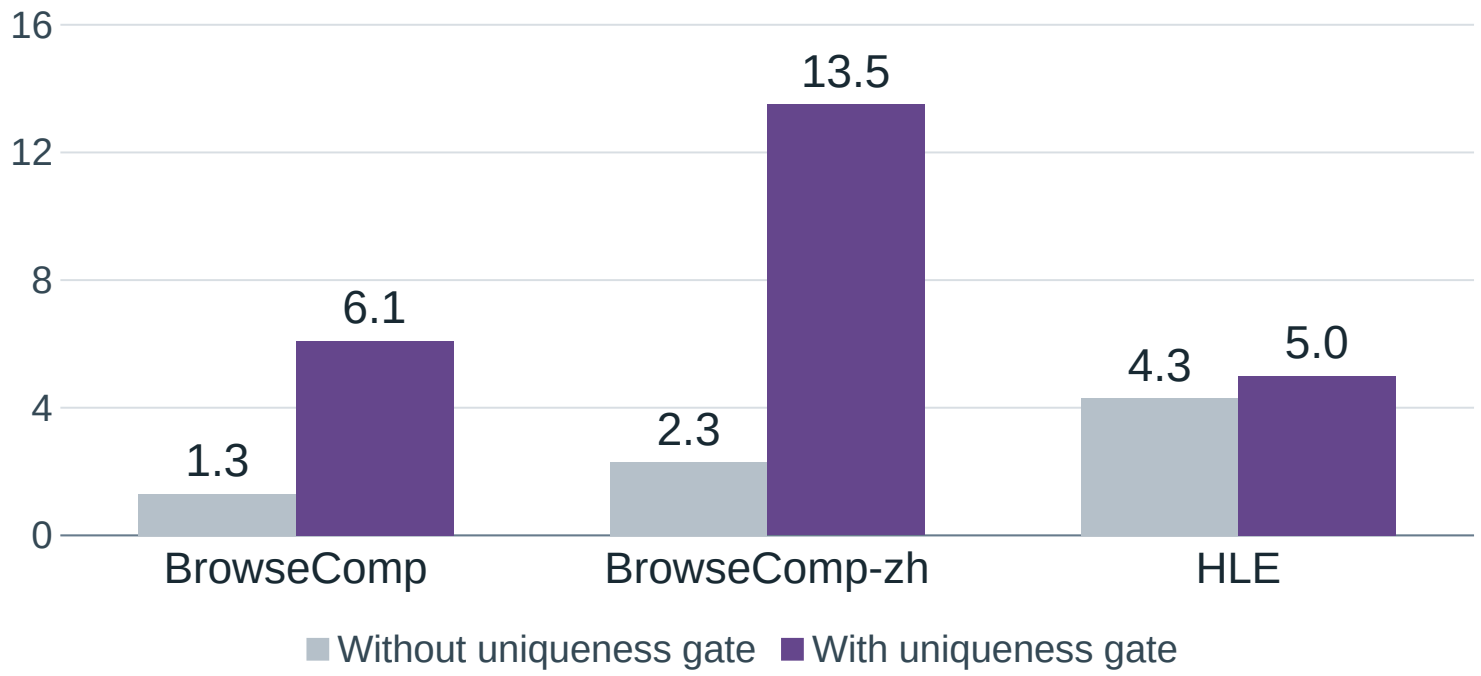
+11.6

GAIA

Accuracy (%). Gains are percentage points from SFT to SFT + RL.

Answer uniqueness and the benefit of offline RL

SFT to SFT + RL gain (percentage points)



Larger gains on search-intensive tasks

The largest difference appears on BrowseComp-zh.

Comparison scope

The no-gate condition uses an earlier internal synthesis variant.
This is a pipeline-stage comparison, not an otherwise identical single-variable ablation.

Open deep search research with downloadable data

Verified QA

Knowledge-graph constraints make answer uniqueness programmatically checkable.

Fully offline synthesis and training

Local retrieval supports trajectory synthesis, supervised fine-tuning and RL.

Code

Paper

github.com/RedSearchAgent/REDSearcher/tree/main/OpenSearcher

{zchu, qinb}@ir.hit.edu.cn

Thank you