GeoX: Mastering Geospatial Reasoning Through Self-Play and Verifiable Rewards

Kyeongjin Ahn[1,3] · Seungeon Lee[1] · Krishna P. Gummadi[1] · Meeyoung Cha[2,3]
[1]Max Planck Institute for Software Systems · [2]Max Planck Institute for Security and Privacy · [3]KAIST

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS
MAX PLANCK INSTITUTE FOR SECURITY AND PRIVACY

Paper LinkedIn

NEURAL INFORMATION PROCESSING SYSTEMS

01 THE CHALLENGE

2 MIN — ONLY 2 MINUTES? READ THIS FIRST.

ONE IMAGE.
THOUSANDS OF HARD QUESTIONS.
ZERO HUMAN LABELS.



LEVEL 01	EXISTENCE Is there a ship in the image?	→ yes
LEVEL 02	COMPARISON · DIRECTION Where is the largest ship?	→ top right
LEVEL 03	COMPARISON · RELATION Which building is closest to the 3rd-largest ship?	→ Blue warehouse [x_1, y_1, x_2, y_2]

ONE QUESTION → AN EXECUTABLE ANSWER

```
program.py · executed on image I                                  # returned value
1 ships, bldgs = segment(I, "ship", "building")        → 62 ships · 14 buildings
  # ships + buildings
2 s3 = sort_by_area(ships)[-3]                                    → ship_17
  # select the 3rd-largest ship
3 d = distance(s3, bldgs)                                         → 48 m vs 112 m
  # compute candidate buildings
4 return argmin(d)                                                → "Blue warehouse"
  # the nearest building
```

✓ Run the program. Its output becomes the training label.

02 THE GEOX METHOD

SELF-PLAY DEMO · SCAN TO WATCH

Watch GeoX ask a question and solve it.

Scan with your phone camera — no app needed.

Harder questions make a better solver, which earns harder questions.

SELF-PLAY WITH A SHARED POLICY $\pi\vartheta$

PROPOSER
Construct an executable problem.

PROBLEM
q "Closest building to the 3rd-largest ship?"
a "ship", "building"
p
```
S = segment(I, a[0])
s3 = sort_by_area(S)[-3]
B = segment(I, a[1])
argmin(B, distance(s3))
```
o "Blue warehouse"

✓ Execute → validity check

TOOL CALL INTERFACE
Used for proposer.

INTERNAL TOOLS
- **Geometry** — area · centroid · orientation
- **Topology** — distance · adjacent · overlaps
- **Aggregation** — count · argmin · compare

EXTERNAL TOOLS
- **Open-vocab segmenter** — phrase → masks · SegEarth-OV3
- **Future extensions** — vertical, graph, meta reasoning (depth · road · OSM)

SOLVER
One problem. Three missing pieces.

REASONING MODES
- ABDUCTION — infer a: p (?) = o → â
- DEDUCTION — predict o: p (a) = ? → ô
- INDUCTION — synthesize p: ? (a_t) = o_t → p̂

LEARNABILITY REWARD
Prefer hard-but-solvable questions.

	s	reward
Too easy	s = 1.00	0
Too hard	s = 0.00	0
Hard but solvable	s = 0.125	0.875

$r^{proposer} = 1[s > 0] \cdot (1 - s)$

CORRECTNESS REWARD
Check the solver's answer by execution.

		reward
ABDUCTION	Infer the missing input â. VERIFY BY EXECUTION	1
DEDUCTION	Predict the missing output ô. VERIFY BY COMPARISON	1
INDUCTION	Synthesize the missing program p̂. VERIFY ON HELD-OUT SET	1

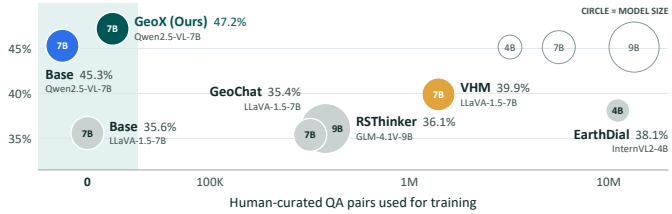
$r^{solve} = 1[\hat{o} = o]$

Learnability → JOINT POLICY UPDATE ← Correctness

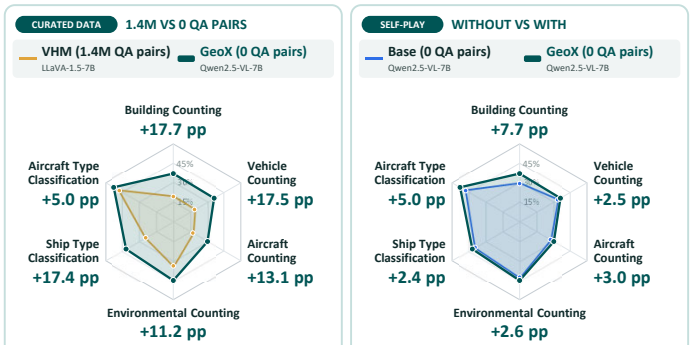
03 RESULTS

PERFORMANCE COMPARISON

Mean VQA accuracy · RSVQA-HR, EarthVQA, GEOBench-VLM



PERFORMANCE DETAILS



COMPLEMENTARY AXES

Reasoning-dimension pairs per question: three benchmarks vs. GeoX self-play
Node size = frequency · chord width = co-occurrence

