

RL-Nav v1

Oracle-guided reinforcement learning for navigation under partial observability

Thesis: a blind agent can follow dense oracle embeddings and learn meaningful navigation behavior by following oracle commands.

2

training stages

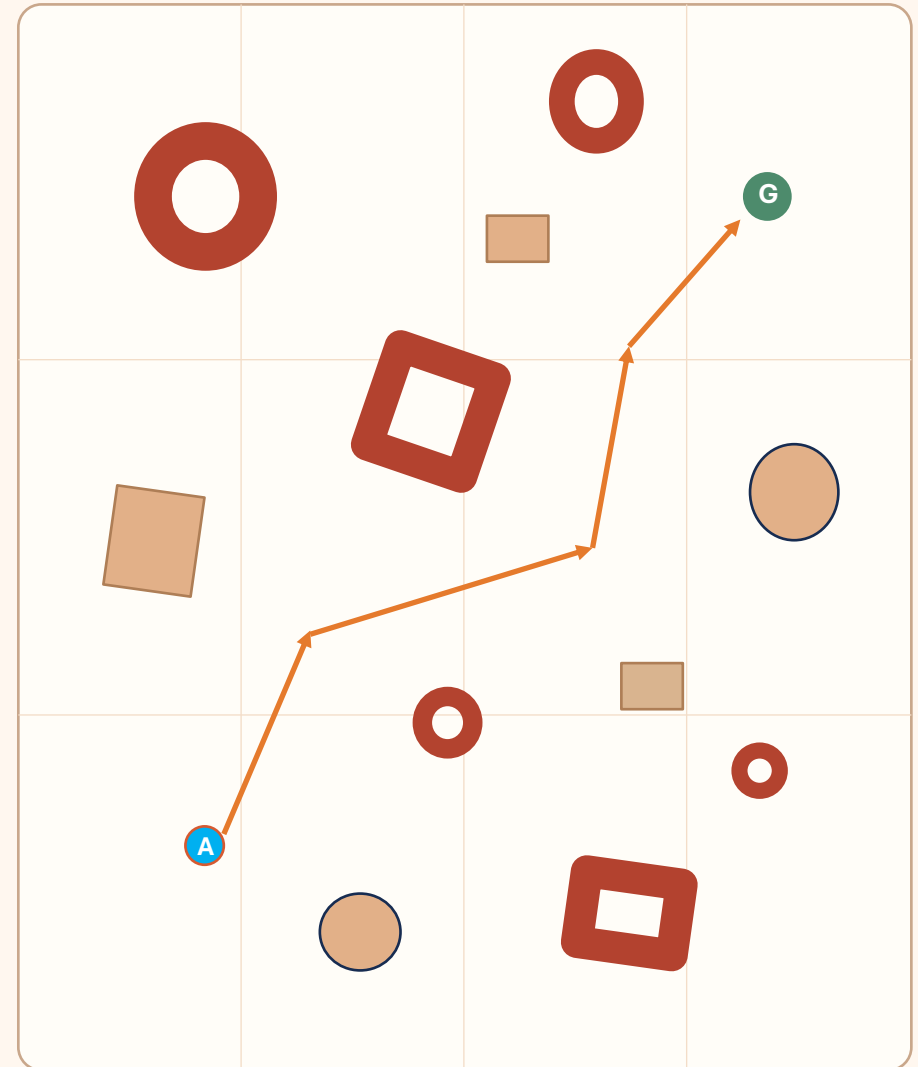
512-D

oracle embedding

62%

policy success

Lorenzo Marinelli – 2043092
Reinforcement Learning 25/26



Random map with obstacles, pits, noisy motion, and hidden goal

Why this problem is interesting

The offloading of planning to the LLM

What the agent does not know

- It cannot directly observe the goal location.
- Local sensing is only five binary hazard sectors.
- Advice is refreshed every 16 steps, not every action.
- Slippage and collisions make execution inconsistent.

"Can a policy learn to execute abstract guidance consistently over time, even when it cannot directly see the goal?"

Partial observability

No direct goal view for the agent.
Only local cues

Omniscient Oracle

LLM has complete view of world map and agent state

Stochastic control

Linear + lateral slip
Angular noise, agent must learn to compensate

Temporal consistency

Advice ages
Memory matters

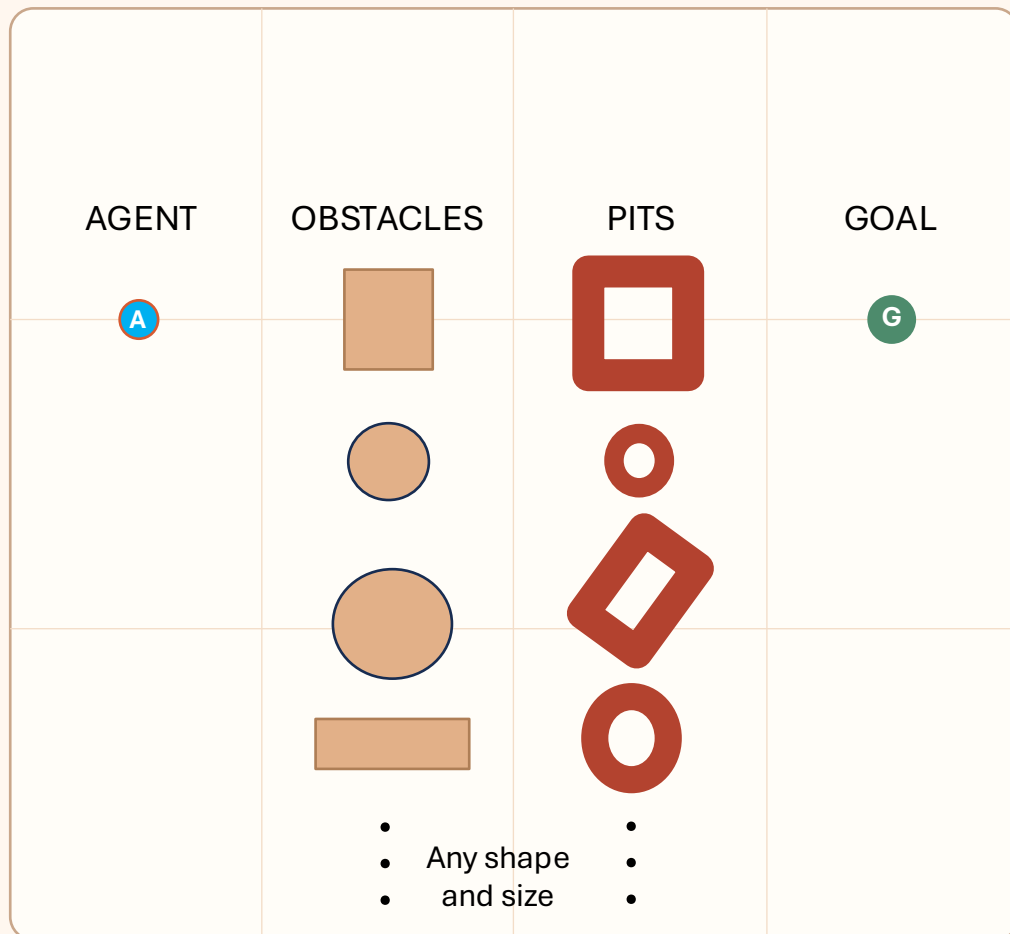
Research contribution

Oracle guidance
For robotics

The interesting question is not perception alone; it is instruction following under uncertainty.

Environment and task design

Each episode is a fresh random map, so the policy cannot memorize one fixed layout.



Bounded square world: random spawn, goal, obstacles, and pits

Episode mechanics

- World: bounded square map with random spawn and goal positions.
- Hazards: obstacles block motion; pits are terminal failures.
- Continuous control: normalized turn + forward action.
- Noise: linear, lateral, and angular slippage at execution time.

Why this is harder than standard navigation

- The goal is hidden from the policy.
- The agent must use memory between guidance refreshes.
- Local mistakes and catastrophic failures are both possible.

8–12

obstacles

2–4

pits

600

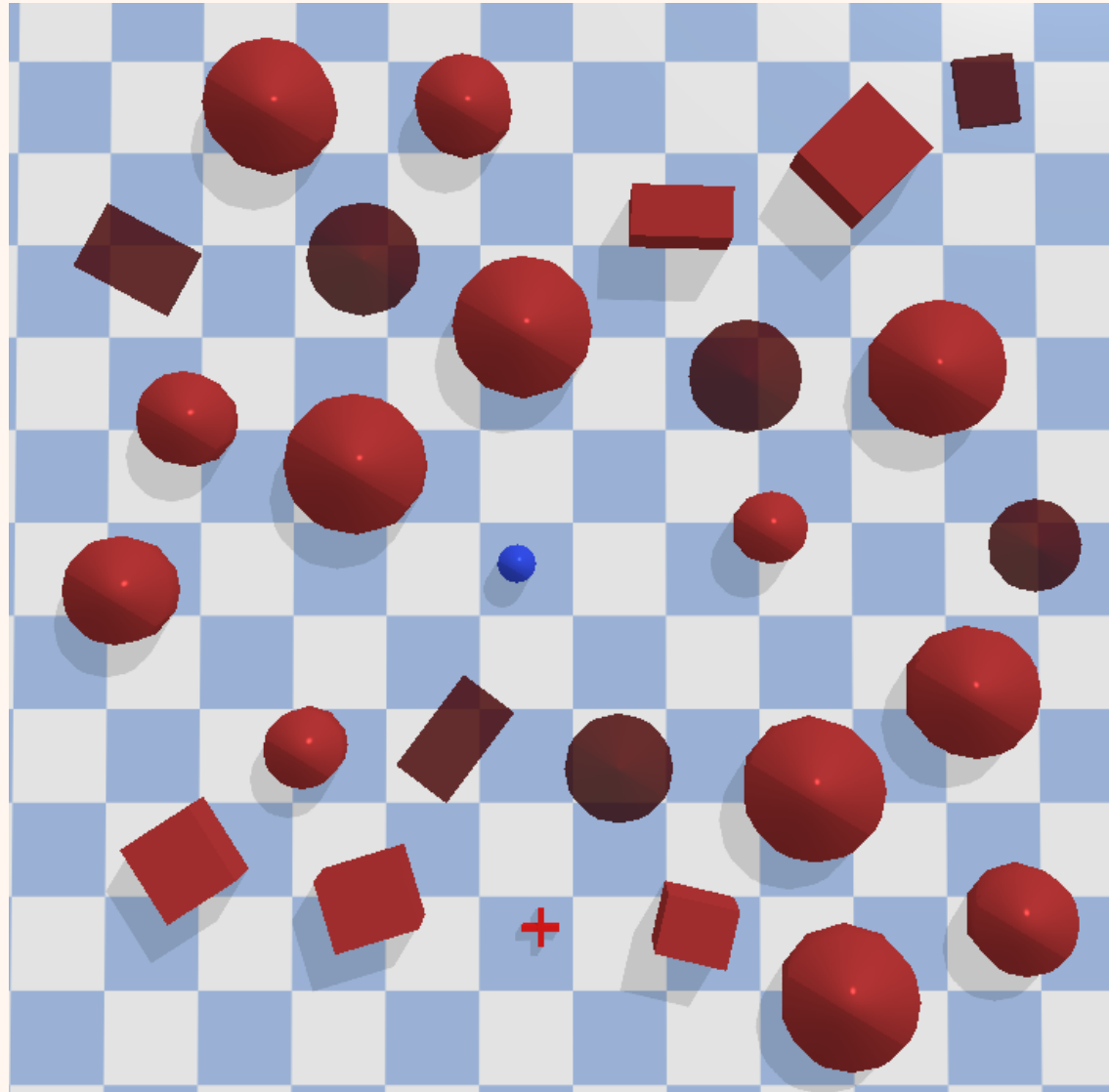
max steps

16

step advice
cadence

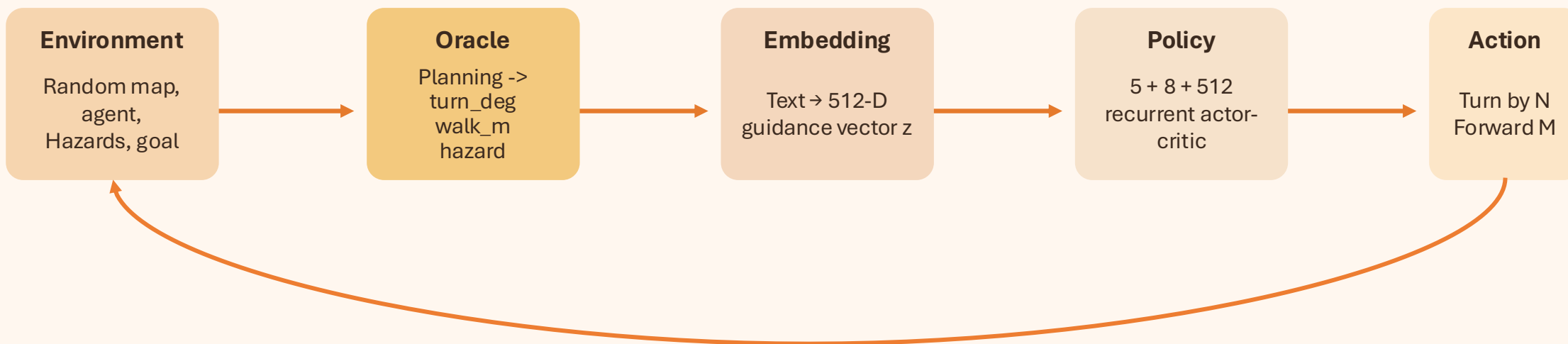
Environment and task design

Screenshot of the actual pybullet environment with more hazards than normal

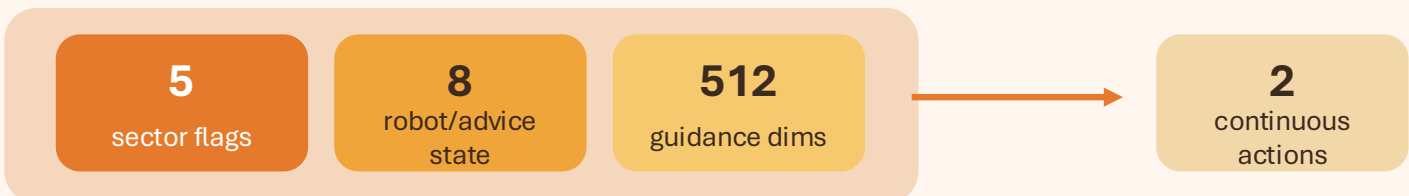


Full oracle-guided control loop

Full stack: environment design, oracle design, dataset engineering, model training, and evaluation tooling.



Loop intuition: the oracle provides high-level intent, while the policy must convert it into low-level robust execution over multiple noisy steps.



What the policy sees, decides, and optimizes

The design deliberately mixes tiny local sensing with a dense high-level conditioning vector.

Agent Observation

5

hazard
sectors

8

robot state

512

guidance z

- Sector flags: front, front-left, left, front-right, right.
- Robot state includes x, y, heading sin, heading cos, advice age, and remaining instruction progress.
- The policy knows not only the advice, but where it currently stands relative to it.

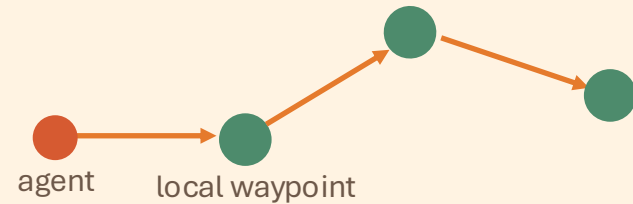
Action

2-D continuous
output

turn $\in [-1, 1]$
forward $\in [-1, 1]$

- Turn is scaled by max_turn_deg.
- Forward is remapped from $[-1, 1]$ to $[0, 1]$.
- Execution noise makes the same command produce variable outcomes.

Reward shaping



- Per-step penalty, collision penalty, pit penalty, and final goal bonus.
- Most interesting (and necessary for training) choice: each oracle refresh defines a temporary local goal.
- The policy is rewarded for making coherent progress toward the current instruction.

This local-goal mechanism is the key bridge between abstract advice with sparse and rare reward and step-by-step control with good learning signal.

Two oracle modes, one shared interface

This abstraction is a strong engineering choice because both oracle paths end in the same 512-D conditioning signal.

Live LLM (openai gpt-5-mini API) oracle

- Receives the complete state of the environment.
- Receives strict prompt with frame conventions.
- Returns structured JSON and clamps the answer.
- Embeds final advice text into $z \in \mathbb{R}^{512}$.

no raw free-form output, lower left/right ambiguity, same final interface as the dataset oracle. Model could work only with natural language, but no reason to not request the LLM for formatted output

Shared output contract

Advice fields

turn_deg • walk_m • hazard

Final representation

dense embedding z (512-D)

Dataset oracle

- Loads premade embeddings and corresponding advice.
 - Matches candidates by turn/walk geometry and hazard filtering.
- Limitation: heuristic controller for movement, not true semantic retrieval, so “not advanced thinking like LLM” .
- Advantage: much faster, basically necessary in training to avoid API latency.

Best practical choice for PPO training because it removes latency and cost from the loop.

Premade advice dataset: the strongest systems component

```

1 ('id': 0, 'label': ('turn_deg': 45.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'left'), 'text': 'turn left 45 degrees and go forward 3.0 m, watch out for obstacles on the left')
2 ('id': 1, 'label': ('turn_deg': -5.0, 'walk_m': 3.0), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'turn right 5 degrees and proceed 3 m; avoid the pit on the right')
3 ('id': 2, 'label': ('turn_deg': 65.0, 'walk_m': 1.5), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'turn left 65 degrees and go forward 1.5 m; watch out for the pit on the right')
4 ('id': 3, 'label': ('turn_deg': 5.0, 'walk_m': 2.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn left 5 degrees and proceed 2.0 m; avoid the pit on the left')
5 ('id': 4, 'label': ('turn_deg': -5.0, 'walk_m': 3.1), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate 5 degrees to the right and go forward 3.1 m, beware obstacles on the right')
6 ('id': 5, 'label': ('turn_deg': 18.0, 'walk_m': 2.5), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate left 18 degrees and walk 2.5 m, beware obstacles on the right')
7 ('id': 6, 'label': ('turn_deg': -18.0, 'walk_m': 1.5), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate 18 degrees to the right and go forward 1.5 m, watch out for the pit on the right')
8 ('id': 7, 'label': ('turn_deg': -18.0, 'walk_m': 1.1), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate 18 degrees to the right and go forward 1.1 m; watch out for obstacles on the right')
9 ('id': 8, 'label': ('turn_deg': 48.0, 'walk_m': 2.1), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn left 48 degrees and proceed 2.1 m, beware obstacles on the right')
10 ('id': 9, 'label': ('turn_deg': 55.0, 'walk_m': 2.5), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'turn left 55 degrees and go forward 2.5 m, watch out for obstacles ahead')
11 ('id': 10, 'label': ('turn_deg': 15.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'left'), 'text': 'rotate left 15 degrees and walk 3.0 m; beware obstacles on the left')
12 ('id': 11, 'label': ('turn_deg': 48.0, 'walk_m': 8.0), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'rotate left 48 degrees and walk 8.0 m; beware of obstacles in front')
13 ('id': 12, 'label': ('turn_deg': 80.0, 'walk_m': 2.7), 'hazard': ('kind': 'pit', 'where': 'front'), 'text': 'turn left 80 degrees and proceed 2.7 m, beware of the pit in front')
14 ('id': 13, 'label': ('turn_deg': -20.0, 'walk_m': 3.7), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate 20 degrees to the right and go forward 3.7 m, watch out for obstacles on the right')
15 ('id': 14, 'label': ('turn_deg': 65.0, 'walk_m': 2.1), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate right 65 degrees and walk 2.1 m; beware obstacles on the right')
16 ('id': 15, 'label': ('turn_deg': 65.0, 'walk_m': 1.8), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate 65 degrees to the left and go forward 1.8 m; avoid the pit on the right')
17 ('id': 16, 'label': ('turn_deg': -18.0, 'walk_m': 1.1), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn right 18 degrees and proceed 1.1 m, beware obstacles on the right')
18 ('id': 17, 'label': ('turn_deg': 18.0, 'walk_m': 2.9), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate left 18 degrees and walk 2.9 m')
19 ('id': 18, 'label': ('turn_deg': 18.0, 'walk_m': 4.0), 'hazard': ('kind': 'obstacle', 'where': 'left'), 'text': 'turn 18 degrees and go forward 4 m, beware obstacles on the left')
20 ('id': 19, 'label': ('turn_deg': -5.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'turn right 5 degrees and go forward 3.0 m; watch out for obstacles ahead')
21 ('id': 20, 'label': ('turn_deg': -75.0, 'walk_m': 1.1), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate 75 degrees to the right and go forward 1.1 m, avoid the pit on the right')
22 ('id': 21, 'label': ('turn_deg': 25.0, 'walk_m': 1.7), 'hazard': ('kind': 'obstacle', 'where': 'left'), 'text': 'rotate 25 degrees and walk 1.7 m')
23 ('id': 22, 'label': ('turn_deg': -60.0, 'walk_m': 1.1), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'turn right 45 degrees and go forward 1.1 m; beware of obstacles in front')
24 ('id': 23, 'label': ('turn_deg': 90.0, 'walk_m': 1.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'rotate left 90 degrees and walk 1 m, watch out for the pit on the left')
25 ('id': 24, 'label': ('turn_deg': -20.0, 'walk_m': 1.5), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn right 20 degrees and go forward 1.5 m, avoid the pit on the left')
26 ('id': 25, 'label': ('turn_deg': -48.0, 'walk_m': 3.0), 'hazard': ('kind': 'pit', 'where': 'front'), 'text': 'rotate right 48 degrees and walk 3.0 m, beware of the pit in front')
27 ('id': 26, 'label': ('turn_deg': -65.0, 'walk_m': 0.9), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'turn right 65 degrees and proceed 0.9 m, avoid the pit on the right')
28 ('id': 27, 'label': ('turn_deg': 38.0, 'walk_m': 2.7), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'turn left 38 degrees and go forward 2.7 m, watch out for obstacles ahead')
29 ('id': 28, 'label': ('turn_deg': -5.0, 'walk_m': 1.8), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'rotate 5 degrees to the right and go forward 1.8 m; avoid the pit on the left')
30 ('id': 29, 'label': ('turn_deg': -75.0, 'walk_m': 1.0), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate right 75 degrees and walk 1.0 m')
31 ('id': 30, 'label': ('turn_deg': 0.0, 'walk_m': 1.1), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'go forward 1.1 m; watch out for obstacles ahead')
32 ('id': 31, 'label': ('turn_deg': 48.0, 'walk_m': 2.5), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate right 48 degrees and walk 2.5 m, avoid the pit on the right')
33 ('id': 32, 'label': ('turn_deg': -20.0, 'walk_m': 3.0), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'turn right by 20 degrees, then walk 3.0 m forward; watch out for the pit on the right')
34 ('id': 33, 'label': ('turn_deg': -30.0, 'walk_m': 2.0), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'rotate right 30 degrees and walk 2 m, watch out for obstacles on the right')
35 ('id': 34, 'label': ('turn_deg': 88.0, 'walk_m': 0.8), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate 88 degrees to the left and go forward 0.8 m, watch out for the pit on the right')
36 ('id': 35, 'label': ('turn_deg': -10.0, 'walk_m': 3.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'rotate right 10 degrees and walk 3.0 m, avoid the pit on the left')
37 ('id': 36, 'label': ('turn_deg': -30.0, 'walk_m': 1.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'rotate 30 degrees to the right and go forward 1.0 m; avoid the pit on the left')
38 ('id': 37, 'label': ('turn_deg': -58.0, 'walk_m': 1.1), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn right 58 degrees and proceed 1.1 m; watch out for the pit on the left')
39 ('id': 38, 'label': ('turn_deg': -68.0, 'walk_m': 2.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn right 68 degrees and go forward 2.0 m; watch out for the pit on the left')
40 ('id': 39, 'label': ('turn_deg': 15.0, 'walk_m': 1.8), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn left 15 degrees and go forward 1.8 m; watch out for obstacles on the right')
41 ('id': 40, 'label': ('turn_deg': 28.0, 'walk_m': 1.2), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'rotate 28 degrees to the left and go forward 1.2 m, avoid the pit on the right')
42 ('id': 41, 'label': ('turn_deg': 85.0, 'walk_m': 1.8), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn 85 degrees and go forward 1.8 m; avoid the pit on the left')
43 ('id': 42, 'label': ('turn_deg': -48.0, 'walk_m': 1.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn right by 48 degrees, then walk 1.0 m forward; watch out for the pit on the left')
44 ('id': 43, 'label': ('turn_deg': -48.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn right 48 degrees and go forward 3.0 m')
45 ('id': 44, 'label': ('turn_deg': -88.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'left'), 'text': 'rotate 88 degrees to the right and go forward 3.0 m, beware obstacles on the left')
46 ('id': 45, 'label': ('turn_deg': -60.0, 'walk_m': 0.8), 'hazard': ('kind': 'obstacle', 'where': 'front'), 'text': 'turn right 45 degrees and proceed 0.8 m; beware obstacles in front')
47 ('id': 46, 'label': ('turn_deg': -68.0, 'walk_m': 1.0), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn right by 68 degrees, then walk 1.0 m forward; avoid the pit on the left')
48 ('id': 47, 'label': ('turn_deg': 0.0, 'walk_m': 3.0), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'move forward 3.0 m')
49 ('id': 48, 'label': ('turn_deg': 35.0, 'walk_m': 1.2), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn 35 degrees and go forward 1.2 m; beware obstacles on the right')
50 ('id': 49, 'label': ('turn_deg': -30.0, 'walk_m': 1.0), 'hazard': ('kind': 'obstacle', 'where': 'right'), 'text': 'turn right 30 degrees and proceed 1.0 m')
51 ('id': 50, 'label': ('turn_deg': 15.0, 'walk_m': 2.1), 'hazard': ('kind': 'pit', 'where': 'left'), 'text': 'turn left by 15 degrees, then walk 2.1 m forward; watch out for the pit on the left')
52 ('id': 51, 'label': ('turn_deg': 30.0, 'walk_m': 2.5), 'hazard': ('kind': 'pit', 'where': 'right'), 'text': 'turn right 30 degrees and proceed 2.5 m; watch out for the pit on the right')

```

```

embeddings= np.load("embedded_openai/embeddings.npy")
embeddings.shape

```

✓ 0.0s

(27972, 512)

Why the dataset matters

27,972

rows

512

embedding dim

219

embedding
batches

531,695

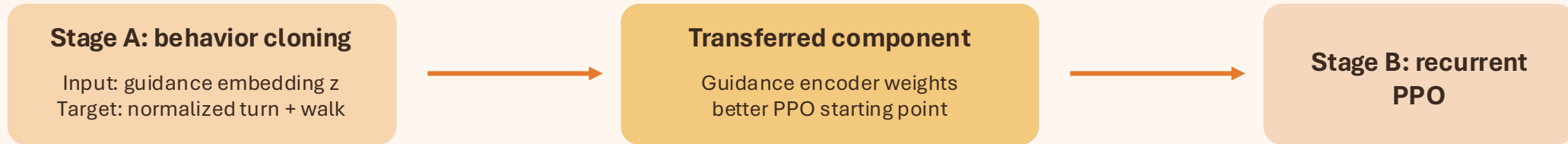
tokens embedded

text-embedding-
3-large
embedding model

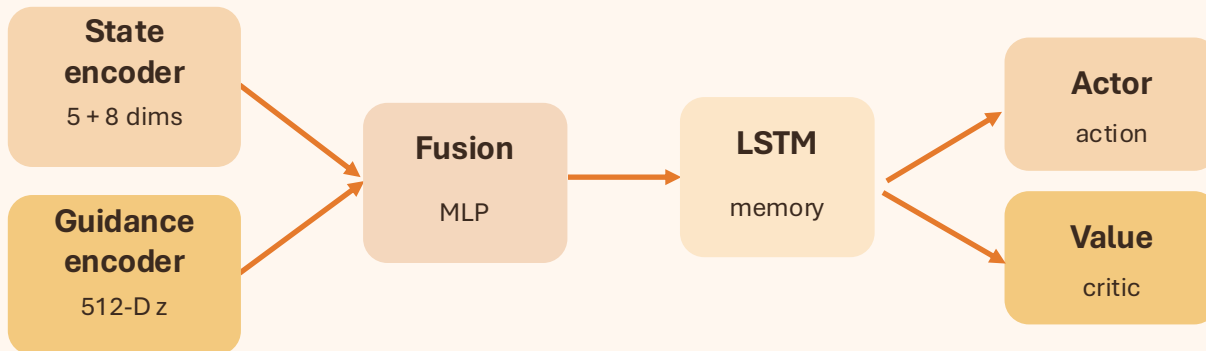
- Each row couples text advice with structured targets: turn angle, walk distance, hazard tag.
- Before training, the code checks dtype, shape, row alignment, and value normalization.
- Practical benefit: PPO can train without live API calls.

Two-stage learning strategy

Behavior cloning gives the guidance pathway a useful initialization before recurrent PPO takes over.



Recurrent policy architecture



Why PPO needs help

- Random guidance processing at PPO start is unstable.
- The imitation loss keeps the policy aligned with current oracle intent.
- The LSTM handles delayed execution between refreshes.

BC does not solve navigation; it only bootstraps the guidance pathway.

Loss functions

Behaviour Cloning Loss

$$L_{BC} = \frac{1}{N} \sum_{i=1}^N [(\hat{t}_i - t_i)^2 + (\hat{w}_i - w_i)^2]$$

Terms

t_i = normalized turn targets

w_i = normalized walk target

\hat{t}_i and \hat{w}_i = model prediction from the 512-D advice embedding

PPO Loss

$$L_{PPO} = L_{clip} + L_{value} - H(\pi) + L_{imitation}$$

Terms

L_{clip} = Clipped ppo policy loss

L_{value} = critic/value loss

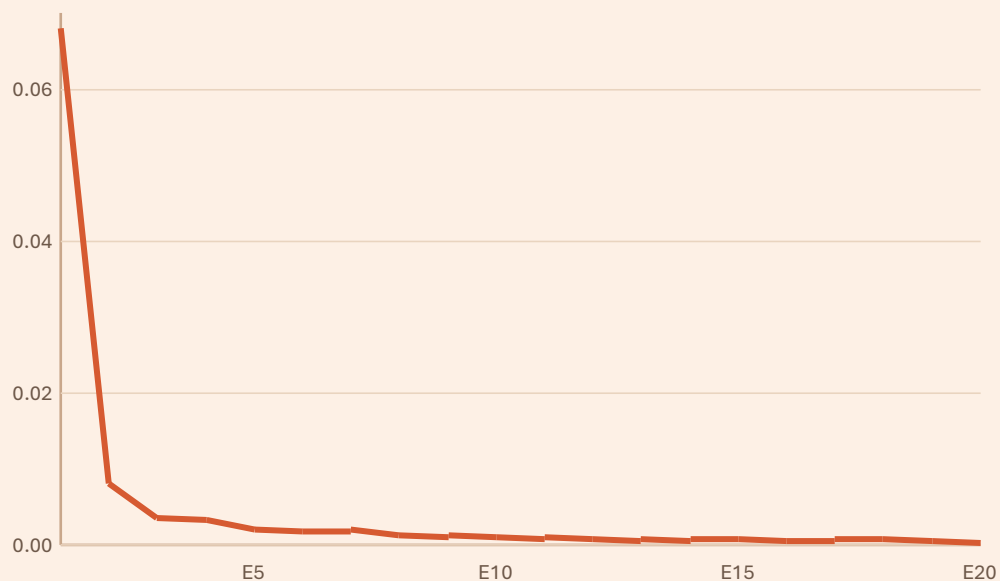
$H(\pi)$ = entropy regularization

$L_{imitation}$ = auxilliary imitation loss

What training suggests before held-out evaluation

The warm-start works very well offline, while PPO shows strong improvement on rolling training metrics.

Behavior cloning validation loss



5.09e-4

best val loss

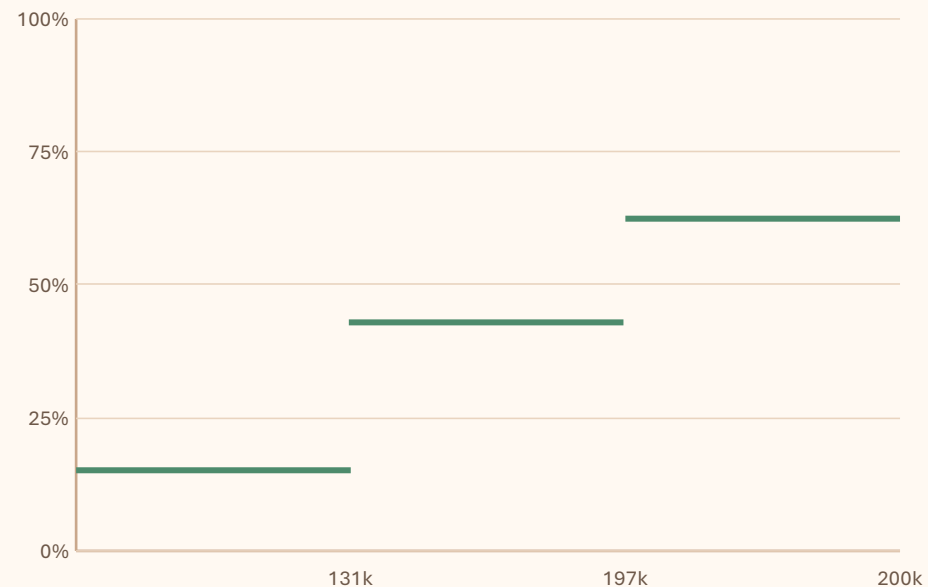
23,776

train rows

4,196

val rows

Rolling PPO success during training



200k

env steps

1166

episodes

0.35

imitation coef

Held-out evaluation

Held-out success rate

62%



Policy checkpoint details

-29.12
mean return

120.69
mean collisions

11.13
Oracle calls per episode

18%
pit fail

62.6
steps to goal

Acceptance target was 70% success over 100 episodes. Current result: 62%, so acceptance is not met.

Bottom line: the bottleneck is execution learning, not just advice generation.

What the results mean

What already works

- The guidance encoder tell us that the embeddings are expressive enough to be used as numerical instructions.
- The offline oracle and embedding pipeline are mature components.
- Oracle-only baselines provide a clean diagnostic reference.

This is stronger as a systems-and-methods project than as a final performance claim.

Where the policy still struggles

- Held-out success is below the 70% target.
- Collision count remains high, suggesting weak local avoidance.
- Agent can't back away when having collided with obstacle
- The dataset oracle uses heuristics rather than semantic retrieval.

The likely gap is reliable execution under noise, hazards, and memory pressure.

Conclusion and next steps

Main takeaway: the contribution is the framework itself, plus clear evidence of what already works and what should improve next.

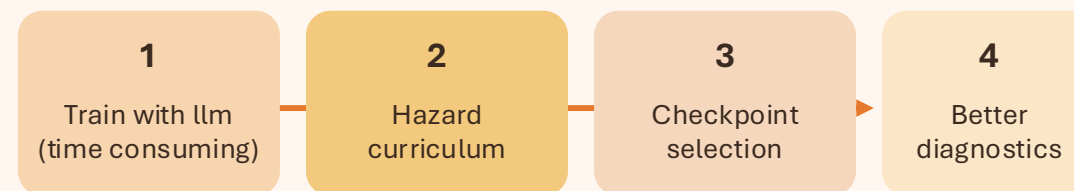
Takeaway

LLM advice can support RL training by extracting elements of the environment and aiding in task by acting as the brain and embedding models can out nuanced enough embeddings that can be mapped to actions

Most important contributions

- Unified live-vs-offline oracle abstraction.
- Balanced dataset and data-contract validation pipeline.
- BC warm-start + recurrent PPO training strategy.
- Oracle-only baseline for diagnosis and comparison.

Logical next steps



- Improve local obstacle avoidance to reduce the collision burden.
- Use richer failure traces to understand when memory breaks down.

Thank you

Any questions?