Our experimental framework creates a controlled setting for systematically assessing the development of navigational capabilities in Large Language Model agents via ongoing contrastive alignment. The simulation architecture prioritizes reproducibility, scalability, and accurate evaluation of agent behavioral changes over many training iterations.

Environmental Architecture and State Representation.

The simulation environment consists of discrete grid worlds with dimensions $N \times N$ where $N \in \{20, 50, 100\}$, accommodating varying complexity. Each cell $c_{i,j}$ assumes states from $S = \{empty, obstacle, goal, agent\}$. The state s_t encodes:

- Local spatial information inside an $k \times k$ observation window centered on the agent
- Global goal coordinates (x_g, y_g)
- Temporal information (current step and remaining time)

Dynamic environment reconfiguration occurs every 100 steps, introducing new obstacles and goal relocations. Environment complexity uses navigational difficulty:

$$D = \frac{L_{direct}}{L_{optimal}} \times \rho_{obstacles} \times \frac{1}{T_{max}}$$

Agent Configuration and Action Space.

Agents observe a 5×5 local grid, global goal coordinates, and step count. The action space $A = \{North, South, East, West\}$. Invalid moves consume a time step without changing position. Episodic memory uses a sliding window buffer of the last 50 observations or actions. Initial placements are determined by stratified randomization across quadrants to guarantee varied starting conditions.

Contrastive Alignment Framework.

Contrastive learning utilizes trajectory pairs: positive samples are successful navigation sequences; negative samples are failed or inefficient paths. The contrastive loss:

$$L_{contrastive} = - E_{(s_t, a_t^+, a_t^-)}\left[log \frac{exp\,(f(s_t, a_t^+)/\tau)}{exp\,(f(s_t, a_t^+)/\tau) + exp\,(f(s_t, a_t^-)/\tau)}\right]$$

Red-Blue Agent Interaction Protocol.

We implement a competitive multi-agent scenario: red agents introduce navigational challenges via dynamic obstacles and strategic positioning; blue agents adapt to maintain navigation performance. A turn-based protocol ensures red agents observe blue movements before introducing disruptions, with constraints to avoid trivial blocking.

Training Dynamics and Optimization.

We use Proximal Policy Optimization with learning rate $\alpha = 3 \times 10^{-4}$, gradient clipping at 0.5, and clipping parameter $\epsilon = 0.2$. Batch size is 128 episodes; experience replay buffer capacity is 10,000 transitions with prioritized sampling by TD error. The PPO objective:

$$L_{policy} = E_{s,a\sim\pi_{old}}[min\,(r(\theta)\,A(s,a),\ clip(r(\theta), 1 - \epsilon, 1 + \epsilon)\,A(s,a))]$$

Computational Infrastructure.

Implementation uses TensorFlow 2.9 with Horovod-based distributed training on NVIDIA V100 GPUs. We employ synchronous data-parallel training across 8-16 GPU nodes. Each agent process is constrained to 4 GB of GPU RAM, with trash collection occurring every 1,000 steps. Reproducibility is ensured by deterministic seeding of TensorFlow, NumPy, and environment RNGs. All configurations and hyperparameters are version-controlled.

Episode Structure and Evaluation Metrics.

Episodes last up to $T = 500$ time steps, terminating upon goal achievement. Metrics include:

- Navigational efficiency: $\eta = \frac{L_{optimal}}{L_{actual}}$
- Success rate across environments
- Adaptation speed (convergence time to effective policies)
- Robustness under environmental perturbations

Thus, comprehensive post-hoc analysis of emergent navigational strategies is enabled using detailed logs of state transitions, actions, rewards, and interactions through contrastive alignment.