# LIFT: LEARNING LONG-HORIZON TEAM FEASIBILITY FOR HETEROGENEOUS ROBOT EXPLORATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Heterogeneous robot exploration can depend on interactions whose value appears only later. Opening a door or operating an elevator may add little immediate coverage, yet change which joint plans become executable for the team. We call this changing set *future team feasibility* and present LIFT, a model-based approach for learning its evolution from partial observations. At each decision segment, a deterministic planner constructs a compact set of executable joint plans. LIFT represents the team state as a relational graph, encodes each plan by its physical semantics, and learns latent dynamics that predict future state, facility status, reward, and plan legality. These predictions are used in legality-masked imagined rollouts to evaluate long-horizon consequences. Feasibility-changing interactions are rare in ordinary exploration experience, so we augment replay with trajectories collected by a rule-based teacher and use temporary policy supervision during training. An event-aware history retains interaction information that remains relevant over longer decision horizons. In Habitat, temporary supervision changes the ranking of the enabling elevator action from 0/55 to 55/55 fixed states, and the preference persists after the supervision is removed. LIFT completes all 10 frozen evaluation episodes with 97.4% average coverage and achieves a 0.90 completion rate on a 50-episode extension, where it also records fewer collisions than PPO. Ablations show that teacher supervision is important for cross-floor interaction, while removing history degrades long-horizon prediction and performance on manipulation tasks.

## 1 INTRODUCTION

Long-horizon coordination becomes difficult when a useful action does not pay off immediately. A humanoid may open a door without increasing coverage, yet that action can create the only route for a wheeled robot to enter another region. Calling an elevator may contribute no local exploration gain, but can make an upper floor reachable to the team. In such cases, an action changes more than the environment state or immediate reward: it changes which joint plans the team can execute later. We refer to this evolving set of executable joint plans as *future team feasibility*.

Most exploration methods do not model this effect explicitly. Frontier-based and learned explorers choose among currently available targets using information gain, travel cost, or learned utility (Chiun et al., 2025; Cao et al., 2025). Heterogeneous and multi-floor methods add capability or topology constraints, but still reason primarily over choices that are feasible at the current decision step (Li et al., 2026c; Chen et al., 2025a). Cooperative MARL can in principle propagate delayed reward, yet the underlying robot–skill–target space contains many physically invalid combinations, and sparse team rewards make it difficult to assign credit to interactions whose benefit appears much later (Yu et al., 2022; Kuba et al., 2022; Bettini et al., 2023; Qu et al., 2025). World models provide a natural way to reason over longer horizons (Hafner et al., 2025; 2023; Hansen et al., 2024; Li et al., 2026b), but standard formulations do not explicitly represent how interactions such as opening a door or operating an elevator change the team's later executable plans.

This setting introduces three practical difficulties. First, the feasible joint-plan set changes over time, so a menu index does not identify a persistent physical action. Second, the transitions that alter feasibility are rare: ordinary exploration produces many navigation and coverage transitions, but relatively few door, elevator, or manipulation events. Third, the effect of an interaction may only



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

become relevant many decision segments later, requiring the model to retain information beyond a short recent-history window.

We propose LIFT, a model-based approach that learns how a current joint plan changes the team's future feasible choices. At each decision segment, a deterministic planner removes plans that violate capability, route, resource, or task constraints, and the remaining plans are represented by their physical semantics rather than by temporary menu indices. A relational latent dynamics model predicts how the team state, facility state, reward, and plan legality evolve after execution. The policy evaluates candidate plans through imagined rollouts under these predicted legality masks, so interactions with little immediate reward can receive value through the plans they enable later.

Rare feasibility-changing events require additional training support. We therefore add trajectories collected by a rule-based teacher and apply temporary policy supervision on those states during early training. An event-aware history retains interaction and task-transition information that remains useful over longer horizons. In Habitat, the learned policy acquires cross-floor enabling behavior that persists after teacher supervision is removed. It completes all 10 frozen evaluation episodes with 97.4% average coverage and reaches a 0.90 completion rate on the 50-episode extension.

The main contributions are:

- We formulate *future team feasibility* for heterogeneous robot exploration, describing how a current joint action changes the set of joint plans that will be executable later.
- We introduce a decision model that combines legality-checked joint plans, persistent action semantics, future-legality prediction, and legality-masked latent imagination for long-horizon plan evaluation.
- We develop a training procedure for rare feasibility-changing events using event-stratified teacher trajectories and temporary policy supervision, and examine the effects of representation learning and long-horizon history in controlled ablations.

## 2 Related Work

**Multi-robot exploration and heterogeneous planning.** Multi-robot exploration has progressed from frontier allocation based on geometric cost and information gain to learned coordination over graph-structured environment representations (Chiun et al., 2025; Cao et al., 2025). Recent work has also considered heterogeneous capabilities and multi-floor topology, including capability-constrained task assignment (Li et al., 2026c) and floor–stair reasoning (Chen et al., 2025a). In parallel, LLM-based systems such as COHERENT (Liu et al., 2024), EMOS (Chen et al., 2025b), CoELA (Zhang et al., 2024), and SMART-LLM (Kannan et al., 2024) use semantic knowledge to assign tasks across heterogeneous embodiments. These approaches provide increasingly expressive coordination mechanisms, but decisions are still made over actions or tasks that are feasible at the current step. Our setting focuses instead on interactions whose main effect is to change which joint plans become feasible later.

**Multi-agent reinforcement learning.** Cooperative MARL provides a natural framework for long-horizon team objectives. MAPPO (Yu et al., 2022) and HAPPO (Kuba et al., 2022) learn centralized or sequentially updated policies for cooperative agents, while recent work extends MARL to heterogeneous teams (Bettini et al., 2023) and long-horizon credit assignment (Qu et al., 2025). In heterogeneous robot exploration, however, the underlying robot–skill–target space contains many combinations that are physically infeasible, and the actions that alter future reachability may be separated from their eventual reward by many decisions. LIFT therefore learns over a compact legality-checked joint-plan set and models how that set evolves after execution, rather than treating all robot–skill–target combinations as a fixed action space.

**World models and learning from demonstrations.** Latent world models support long-horizon control by predicting future latent states and rewards and optimizing policies through imagined rollouts (Hafner et al., 2023; 2025; Hansen et al., 2024). They have also been studied under sparse-reward settings (Li et al., 2026b). Our use of a world model differs mainly in what is represented and predicted: the latent state is built from a heterogeneous relational graph, joint plans are encoded by their physical semantics, and the model explicitly predicts future plan legality together

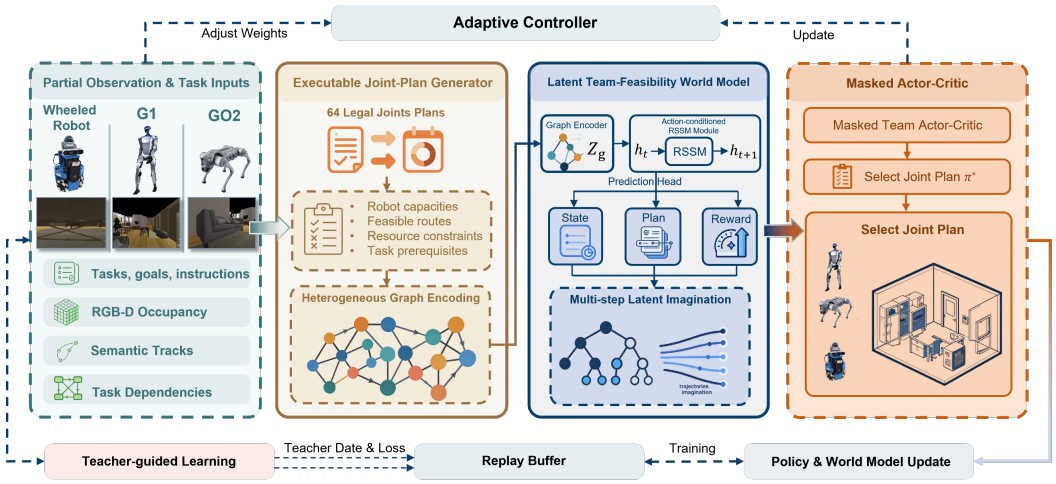



Figure 1: **Overview of LIFT.** Delivered observations are encoded as a heterogeneous relation graph, while a deterministic planner constructs a compact menu of executable joint plans. The selected plan enters an action-conditioned latent dynamics model through its physical semantics, and candidate plans are evaluated through legality-masked imagination before execution. Teacher trajectories provide additional training data for sparse interactions, while event-aware history retains information needed over longer decision horizons.

with environment and reward dynamics. Rare feasibility-changing events also motivate the use of directed experience. Prior work such as MoDem (Hansen et al., 2023) shows that demonstrations can accelerate model-based RL, while DAgger (Ross et al., 2011) and more recent hybrid BC–RL methods (Ankile et al., 2025) study how expert supervision can be combined with policy learning. LIFT uses teacher data only to cover rare interaction transitions and anneals policy supervision after those behaviors have been acquired.

## 3 METHOD

Figure 1 summarizes the decision and learning loop of LIFT. At each decision segment, the team constructs a legality-checked joint-plan menu from the delivered observation, evaluates these plans from the current latent state, executes one plan, and stores the resulting transition in replay. The model and policy are trained from this experience, with additional supervision on rare interaction events and selective retention of history over long episodes.

### 3.1 PROBLEM FORMULATION

We consider $N{=}3$ heterogeneous robots—a humanoid (G1), a quadruped (Go2), and a wheeled base—exploring an unknown multi-floor building. Time is divided into decision segments. At segment $t$, the team selects one joint plan, Habitat executes it as a bounded skill segment (Szot et al., 2021; Puig et al., 2023), and returns the next delivered observation $o_{t+1}$, team reward $r_t$, and termination signals. The policy does not observe the latent building state directly and acts only on the delivered observation $o_t$. Each robot chooses from a shared skill vocabulary, `[HOLD, GOTO, OPEN, CLOSE, PRESS_BUTTON, PICK, PLACE, CLIMB_STAIRS]`, subject to embodiment-specific constraints. G1 supports the full vocabulary, Go2 cannot manipulate objects, and the wheeled robot can neither manipulate nor climb stairs, although it may use an elevator operated by G1. At each decision segment, a deterministic planner constructs a compact set of complete joint plans and removes plans that violate capability, route, resource, or task constraints. We denote the resulting executable set by

$$\mathcal{M}_t = \mathcal{M}(o_t), \qquad |\mathcal{M}_t| \le 64.$$

Our interest is not only whether a plan is executable now, but how the executable set changes after the team acts. We use *future team feasibility* to denote the evolution of these executable joint-plan

sets over a horizon $H$:

$$\mathcal{F}_t^{(H)} = \big(\mathcal{M}_{t+1}, \mathcal{M}_{t+2}, \ldots, \mathcal{M}_{t+H}\big). \tag{1}$$

An enabling interaction is therefore an action whose immediate reward may be small, but whose execution changes one or more future feasible sets. Opening a door can make a previously unreachable region accessible to another robot; calling or operating an elevator can make cross-floor joint plans executable. The team seeks a policy that maximizes long-horizon return while selecting only executable joint plans:

$$\pi^\star = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right], \qquad a_t \sim \pi(\cdot \mid o_t), \quad a_t \in \mathcal{M}_t. \tag{2}$$

The difficulty is that the value of $a_t$ may depend on how it changes $\mathcal{F}_t^{(H)}$, rather than on its immediate reward.

### 3.2 Joint-plan decision architecture

LIFT separates hard feasibility checks from learned plan evaluation. Given the delivered observation $o_t$, a deterministic planner first constructs a compact set of executable joint plans. The policy then compares only these plans, rather than learning over the full robot–skill–target combination space.

**Legal joint-plan interface.** At decision segment $t$, the planner generates a menu $\mathcal{M}_t = \mathcal{M}(o_t)$ of at most 64 complete three-robot joint plans. Plans that violate embodiment capabilities, route feasibility, interaction preconditions, task dependencies, or resource constraints are removed before policy evaluation. Each remaining plan is associated with a physical semantics vector $\phi(a)$ describing its robot, skill, target, task binding, and route-related attributes. The menu is regenerated at every decision segment, so its slot indices do not have persistent meaning. LIFT therefore conditions the policy and dynamics model on $\phi(a)$ rather than on the position of a plan in the current menu. This gives the same physical action a consistent representation even when the surrounding candidate set changes.

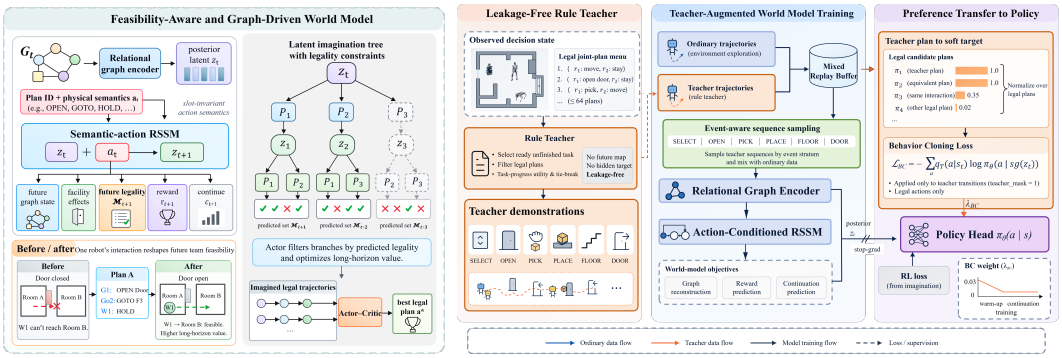
**Policy-visible information.** All decisions are made from information available at the current decision segment. The policy receives per-robot observations and states, together with the fused occupancy map, frontiers, task graph, semantic tracks, recent history, and traversed topology. Exact object poses, future maps, future joint-plan menus, and other simulator-only states are not available to either the policy or the teacher at decision time. The observation noise and delivery model used in the experiments are described in §4.1.

**Decision loop.** The delivered observation is encoded as the heterogeneous relation graph described in §3.3. From the resulting latent state, the actor scores the plans in $\mathcal{M}_t$ using their semantics vectors and selects one executable joint plan. During evaluation we use deterministic argmax selection. The selected plan is executed in Habitat as a bounded skill segment, after which the next delivered observation, reward, execution outcome, and termination signals are written to replay. Replay transitions are used to update the world model, actor, and critic. The world model provides the long-horizon plan evaluation described in §3.3, while rare interaction transitions receive the additional training treatment in §3.4.

### 3.3 Learning future team feasibility

The policy must evaluate a joint plan before observing the states that it may enable. We therefore learn a latent dynamics model that predicts not only the next team state and reward, but also how the set of executable plans changes after the selected joint plan is carried out (Figure 2a).

**Relational state representation.** The delivered observation is represented as a typed relation graph $G_t$. Its nodes describe robots, frontiers, tasks, regions, portals, semantic entities, recent history, and traversed topology, while edges encode relations such as assignment, reachability, task dependency, observation, communication, and traversal. A three-layer relational graph-attention




(a) Learning future team feasibility. (b) Learning enabling interactions from sparse experience.

Figure 2: **Learning and using future team feasibility.** **(a)** The relation graph and joint-plan semantics condition the latent dynamics model, which predicts future state, reward, and plan legality and evaluates candidate plans through legality-masked imagination. **(b)** Rare interaction trajectories are mixed with ordinary replay through event-aware sampling, while temporary policy supervision transfers enabling behavior and is annealed during training.

encoder (Veličković et al., 2018; Battaglia et al., 2018) maps $G_t$ to graph features $Z_t$, which condition the posterior state of a DreamerV3-style RSSM (Hafner et al., 2023). The latent transition is conditioned on the semantics of the selected joint plan,

$$p(s_{t+1} \mid s_t, \phi(a_t)),$$

rather than on its position in the current menu.

**Predicting future feasibility.** The latent state is trained to retain information needed for both environment prediction and plan evaluation. From $s_{t+1}$, the model reconstructs the task-level graph and predicts facility state, team reward, continuation, and the legality of candidate joint plans:

$$\mathcal{L}_{\text{wm}} = \underbrace{\mathcal{L}_{\text{node}}}_{\text{node state}} + \underbrace{\mathcal{L}_{\text{mask}}}_{\text{node validity}} + \underbrace{\mathcal{L}_{\text{legal}}}_{\text{plan legality}} + \underbrace{\mathcal{L}_{\text{facil}}}_{\text{facility state}} + \underbrace{\mathcal{L}_{\text{rew}}}_{\text{team reward}} + \underbrace{\mathcal{L}_{\text{cont}}}_{\text{continuation}} . \tag{3}$$

Among these predictions, $\hat{\ell}_{t+1}$ provides an explicit estimate of which joint plans are expected to remain or become executable after the current action. At observed states, plan semantics and legality are supplied by the deterministic planner. During imagination, the planner is not queried; the model predicts the semantics of the next candidate slots together with their legality mask, and the actor evaluates only the candidates retained by this predicted mask.

**Feasibility-aware imagination.** Training the dynamics model alone does not determine which feasible plan should be selected. During policy optimization, we roll the RSSM forward for $H{=}15$ imagined decision segments from posterior states sampled from replay. At each imagined state, the actor scores candidate plans and the predicted legality mask restricts sampling to plans that the model expects to be executable. The critic estimates discounted return along these imagined trajectories, and the actor is trained with $\lambda$-returns. This allows an interaction with little immediate reward to receive value through the plans it enables later. For example, opening a door can be preferred because the imagined rollout predicts that a teammate gains access to a previously unreachable region, even when the opening action itself contributes no coverage.

### 3.4 Learning enabling interactions from sparse experience

Enabling interactions are uncommon in ordinary exploration trajectories. Most collected experience consists of navigation, frontier visitation, and coverage updates, whereas door operation, elevator use, and manipulation occur only occasionally. This creates a practical problem for long-horizon learning: the model may learn that an interaction is valid without the policy ever assigning it sufficient value. In an early training run, for example, the model recognized all 79 observed cross-floor

opportunities as legal, yet the actor selected none of them. Simply appending 26 teacher episodes to replay produced little change. We therefore use directed interaction data together with temporary policy supervision. The training procedure is shown in Figure 2b.

**Directed interaction data.** We use a deterministic rule-based teacher to collect trajectories containing interactions that are rarely encountered during ordinary exploration. The teacher receives the same policy-visible state and legal joint-plan menu as the learner. At each decision step, it identifies the earliest ready unfinished task and ranks the available plans according to task progress. The resulting trajectories provide additional examples of door operation, elevator use, floor transitions, and manipulation.

**Event-aware replay.** Teacher and ordinary trajectories are stored in the same replay buffer. Teacher rows carry a binary `teacher_mask`, which is used only during training and is never exposed to the policy. Rather than sampling teacher data in proportion to its small share of the buffer, we stratify teacher sequences by interaction type and mix them with ordinary sequences in each update. The same world-model objectives in Eq. equation 3 are applied to both sources, so rare interaction effects are learned together with the dynamics of ordinary exploration.

**Temporary policy supervision.** Modeling an interaction correctly does not guarantee that the actor will choose it. For teacher states, we therefore construct a soft target over the current legal menu. The demonstrated plan and functionally equivalent plans receive the largest weight, plans that realize the same interaction through a different binding receive a smaller weight, and the remaining legal plans retain a small nonzero probability:

$$q_T(a \mid s_t) = \frac{w(a)}{\sum_{a' \in \mathcal{M}_t} w(a')}, \qquad \mathcal{L}_{\mathrm{bc}} = -\sum_{a \in \mathcal{M}_t} q_T(a \mid s_t) \log \pi_\theta\big(a \mid \mathrm{sg}(z_t)\big). \tag{4}$$

We use $w(a) = 1$ for the demonstrated plan and its functional equivalents, $0.35$ for alternative bindings that achieve the same interaction, and $0.02$ for the other legal plans. The loss is applied only to teacher rows and only to the policy head. The stop-gradient operator prevents this supervision from directly modifying the latent representation learned by the world model. The supervision coefficient is set according to the relative gradient magnitudes of the RL and BC objectives:

$$\lambda^\star \approx \frac{\|\nabla_{\theta_{\mathrm{pol}}} \mathcal{L}_{\mathrm{rl}}\|}{\|\nabla_{\theta_{\mathrm{pol}}} \mathcal{L}_{\mathrm{bc}}^{(1)}\|}, \tag{5}$$

which gives $\lambda^\star \approx 0.03$ in our training runs. The coefficient is then annealed after the warm-start stage, so subsequent policy updates rely on RL rather than continued teacher guidance.

**Event-aware history.** Some enabling interactions depend on information observed many decision segments earlier. We therefore retain a bounded set of history nodes in the policy-visible graph instead of using a fixed recency window. Interaction events, failures, floor changes, task boundaries, and newly traversed topology are retained preferentially, while the per-robot history budget varies between 8 and 32 records according to recent event density and route uncertainty. Unselected history nodes are masked so that the graph shape remains fixed. Additional implementation details are provided in Appendix A.

## 4 Experiments

We evaluate LIFT in Habitat under the same observation, action-menu, reward, and skill-execution interface used by all baselines. We first compare closed-loop task performance, and then use controlled ablations and open-loop prediction tests to examine where the gains come from.

### 4.1 Setup

**Environments and observations.** We use 50 Habitat scenes drawn from ReplicaCAD, MP3D+YCB, and custom multi-floor environments containing doors and elevators. Forty-five scenes are used for training and five are held out for evaluation; all teacher trajectories are collected

Table 1: **Closed-loop comparison on the frozen 10-episode split** (mean ± std; greedy decoding). All methods use the same observation boundary, legal joint-plan interface, reward, and skill executor. LIFT and PPO complete all ten episodes; LIFT obtains slightly higher coverage, MFC, and return, with fewer collisions, while PPO requires fewer decisions and less simulated time.

| Method | CR ↑ | CF ↑ | MFC ↑ | Return ↑ | DS ↓ | Time (s) ↓ | Collisions ↓ | Energy (Wh) ↓ |
|---|---|---|---|---|---|---|---|---|
| LIFT (Ours) | **1.00** | **0.974** ± .021 | **0.971** ± .022 | **24.6** ± 7.3 | 13.4 ± 12.6 | 351 ± 355 | **65.8** ± 59.5 | 2.56 ± 2.54 |
| PPO | **1.00** | 0.969 ± .020 | 0.968 ± .019 | 24.2 ± 7.2 | 10.6 ± 10.2 | 292 ± 271 | 106.6 ± 121.5 | 2.97 ± 3.05 |
| MAPPO | 0.70 ± 0.48 | 0.923 ± .150 | 0.873 ± .308 | −1.4 ± 36.0 | 28.8 ± 28.6 | 817 ± 818 | 246.7 ± 248.3 | 7.32 ± 7.51 |
| D3QN | 0.80 ± 0.42 | 0.931 ± .153 | 0.881 ± .310 | 2.7 ± 32.0 | 27.3 ± 27.5 | 800 ± 807 | 255.3 ± 281.3 | 7.46 ± 7.55 |
| DecPPO | 0.80 ± 0.42 | 0.928 ± .152 | 0.878 ± .309 | 1.3 ± 34.0 | 27.6 ± 24.2 | 856 ± 733 | 285.8 ± 245.0 | 8.76 ± 7.48 |
| A2C | 0.50 ± 0.53 | 0.876 ± .200 | 0.776 ± .410 | −12.2 ± 39.6 | 34.7 ± 31.0 | 936 ± 854 | 226.2 ± 178.1 | 7.37 ± 6.00 |
| REINFORCE | 0.40 ± 0.52 | 0.883 ± .203 | 0.783 ± .413 | −19.0 ± 32.8 | 45.3 ± 24.7 | 1051 ± 607 | 231.4 ± 179.1 | 8.40 ± 5.72 |
| Priority (rules) | 0.90 ± 0.32 | 0.969 ± .023 | 0.969 ± .023 | 21.0 ± 14.4 | **8.3** ± 9.4 | **217** ± 256 | 98.5 ± 129.0 | **2.36** ± 2.77 |

Table 2: **Closed-loop evaluation. (a)** Successful episodes by task category on the frozen 10-episode split; each interactive category contains two episodes and General contains four. **(b)** Full results on the 50-episode extension (10 elevator, 10 door, 10 cup, and 20 general episodes; mean ± standard deviation).

(a) Completion by task category.

| Method | Elevator | Door | Cup | General |
|---|---|---|---|---|
| LIFT (Ours) | 2/2 | 2/2 | 2/2 | 4/4 |
| PPO | 2/2 | 2/2 | 2/2 | 4/4 |
| MAPPO | 1/2 | 2/2 | 0/2 | 4/4 |
| D3QN | 1/2 | 1/2 | 2/2 | 4/4 |
| DecPPO | 1/2 | 2/2 | 1/2 | 4/4 |
| A2C | 0/2 | 1/2 | 1/2 | 3/4 |
| REINFORCE | 0/2 | 0/2 | 0/2 | 4/4 |
| Priority | 2/2 | 1/2 | 2/2 | 4/4 |

(b) 50-episode extension.

| Metric | LIFT (Ours) | PPO |
|---|---|---|
| CR ↑ | **0.90 ± 0.30** | 0.88 ± 0.33 |
| CF ↑ | 0.965 ± .079 | **0.967 ± .071** |
| MFC ↑ | 0.964 ± .079 | **0.967 ± .071** |
| Return ↑ | 16.3 ± 20.4 | **17.5 ± 21.0** |
| DS ↓ | 18.9 ± 19.7 | **15.2 ± 18.3** |
| Time (s) ↓ | 480 ± 523 | **349 ± 365** |
| Collisions ↓ | **91.8 ± 96.6** | 124.8 ± 162.3 |
| Energy (Wh) ↓ | 3.49 ± 3.72 | **3.39 ± 3.91** |

from the training scenes. Semantic tracks are subject to visibility gating, a 15% miss rate, positional noise, and up to two decision segments of delivery latency. All methods receive the same delivered observations and use the same legality-checked joint-plan menu, team reward, and skill executor. This keeps action feasibility fixed across methods and isolates differences in plan evaluation and long-horizon decision making.

**Evaluation protocol.** We evaluate prediction and closed-loop control separately. For open-loop evaluation, the model is initialized from an 8-segment observed prefix and rolled forward for 1, 5, 10, and 20 decision segments using recorded joint plans but no future observations. We report plan-legality F1, graph-mask F1, facility-state accuracy, and reward prediction error. For closed-loop evaluation, each policy selects joint plans greedily from the current legal menu and executes them in Habitat. The main comparison uses a frozen 10-episode split containing two elevator, two door, two cup, and four general exploration episodes, with a maximum of 64 decision segments per episode. We additionally evaluate LIFT and PPO on a 50-episode extension with disjoint seeds; the compact comparison is reported in Table 2b, with full metrics in Appendix B. Closed-loop metrics include completion rate, coverage fraction, minimum-floor coverage (MFC), discounted return, decision steps, simulated time, collisions, and energy consumption.

**Baselines.** We compare against PPO (Schulman et al., 2017), MAPPO (Yu et al., 2022), A2C (Mnih et al., 2016), D3QN (Mnih et al., 2015), a decentralized graph-attention PPO baseline (DecPPO), and REINFORCE (Williams, 1992). Our implementations follow the corresponding exploration or task-allocation settings used in ALIS-WC (Li et al., 2026a), CQLite (Latif & Parasuraman, 2024), MARVEL (Chiun et al., 2025), and Dai et al. (2025), where applicable. We also include a deterministic task-priority planner, denoted Priority, as a non-learning reference.

**Training details.** LIFT uses a 12M-parameter latent dynamics model with an imagination horizon of $H{=}15$. The checkpoint used for the main baseline comparison is trained for 6,000 steps. For the component analysis in Section 4.3, all ablation variants are trained for 20,000 steps under the same

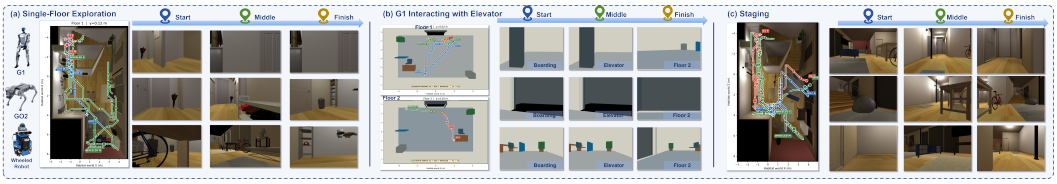



Figure 3: **Representative Habitat rollouts of LIFT.** Each panel shows the fused top-down map with executed trajectories and egocentric observations from the same episode. *(a)* Single-floor exploration in ReplicaCAD. *(b)* Elevator interaction from Floor 1 to Floor 2. *(c)* Exploration in a cluttered multi-room scene.

ablation protocol; these runs are used only for within-ablation comparisons and are not mixed with the main baseline results. All experiments are run on a single NVIDIA RTX 4090 workstation. Additional implementation and training details are provided in the appendix.

### 4.2 Closed-loop comparison

Table 1 reports closed-loop performance on the frozen 10-episode split. Both LIFT and PPO complete all ten episodes, so completion alone does not separate the two methods. LIFT achieves marginally higher coverage, minimum-floor coverage, and return, and records 65.8 collisions on average compared with 106.6 for PPO. PPO, however, reaches completion with fewer high-level decisions and shorter simulated time, averaging 10.6 decisions and 292 s per episode versus 13.4 decisions and 351 s for LIFT.

The category breakdown in Table 2a shows that most failures are concentrated in tasks that require interaction. Most methods solve the general exploration episodes, whereas elevator and cup tasks are less reliable. MAPPO, for example, fails both cup episodes, and REINFORCE completes all four general episodes but none of the interactive ones.

We further compare LIFT and PPO on the 50-episode extension in Table 2b. Their completion rates remain close, at 0.90 and 0.88 respectively. LIFT records fewer collisions, while PPO is slightly better in coverage and return. Full results are provided in Appendix B.

Representative rollouts are shown in Figure 3. In the elevator episode, the humanoid completes the interaction sequence needed to move the team to the second floor, after which the newly available Floor-2 plans are executed. The other examples show single-floor division of labor and exploration in a cluttered multi-room scene.

### 4.3 Ablation and model analysis

We evaluate the effect of teacher supervision, event-aware history, and end-to-end training of the relational encoder. Each variant is trained for 20,000 steps and evaluated on the same frozen holdouts containing 60 general and 60 cup episodes. For the teacher-supervision ablation, we additionally use a fixed set of 12 elevator episodes to test cross-floor interaction. Figure 4 reports closed-loop completion together with open-loop graph and reward prediction results.

**Teacher supervision.** Removing the behavior-cloning loss on teacher states reduces general-task completion from 0.97 to 0.72 and average return from 23.5 to 11.8. On the separate 12-episode elevator diagnostic, the ablated policy completes none of the episodes, whereas the full model completes all of them. Reward prediction also degrades at longer horizons: at horizon 20, the error is 1.63 instead of 1.13 on the general holdout and 2.26 instead of 1.79 on the cup holdout. In the fixed-state elevator test, the enabling action is ranked first in 0/55 states before temporary supervision and 55/55 afterward.

**Event-aware history.** Removing the history mechanism has little effect on general-task completion (0.97 for both variants), but changes the long-horizon prediction behavior. Graph-mask F1 decreases from 0.70 to 0.56 on the general holdout and from 0.72 to 0.59 on the cup holdout as the

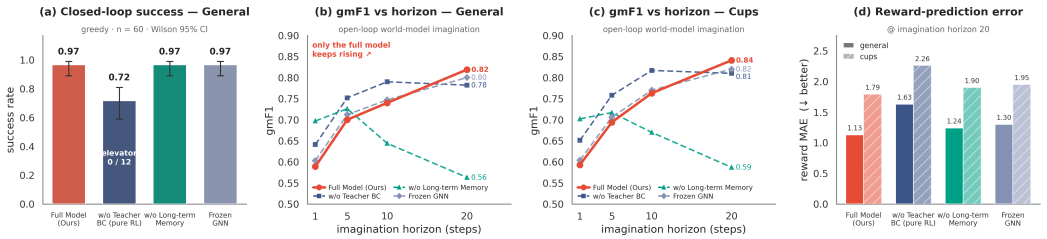



Figure 4: **Ablation on the enlarged frozen holdout.** All variants use identical evaluation episodes. *(a)* Closed-loop completion after removing teacher policy supervision, event-aware history, or end-to-end encoder training. *(b,c)* Graph-mask F1 over increasing open-loop rollout horizons on the general and cup holdouts. *(d)* Reward-prediction error at horizon 20. Removing teacher supervision reduces general completion from 0.97 to 0.72, while removing history or freezing the encoder reduces cup completion from 0.68 to 0.45.

rollout horizon increases, while the full model reaches 0.82 and 0.84, respectively. The difference is also visible in closed-loop cup tasks, where completion drops from 0.68 to 0.45 without history.

**Relational encoder training.** Freezing the graph encoder leaves general-task completion largely unchanged but reduces cup completion to 0.45. The frozen variant also has lower graph-mask F1 and higher reward-prediction error across rollout horizons. The difference is largest on the cluttered manipulation episodes.

The deterministic Priority planner reaches completion rates of 0.95 on the general holdout and 0.60 on the cup holdout. The full model reaches 0.97 and 0.68, respectively. The remaining failures are concentrated in the most cluttered cup scenes.

## 5 Conclusion

This paper studies heterogeneous robot exploration in settings where an interaction can matter mainly because it changes what the team will be able to do later. LIFT makes this effect explicit by reasoning over legality-checked joint plans and predicting how their execution changes future plan availability. Sparse interaction events are supported during training by directed trajectories and temporary policy supervision, while event-aware history retains information needed over longer decision horizons. In Habitat, LIFT completes all 10 frozen evaluation episodes with 97.4% average coverage and remains competitive with PPO on the larger 50-episode extension while recording fewer collisions. The ablations further show that removing teacher supervision or long-horizon history degrades performance on the interaction-heavy settings targeted by the method. The present evaluation is limited to simulation, and the policy reasons over a planner-generated menu of at most 64 joint plans. Extending the same formulation to physical heterogeneous teams and larger, less structured action spaces is left for future work.

## Use of Generative AI

We used a large language model (LLM) during the preparation of this paper and its accompanying software. For the manuscript, the model assisted with local editing in the Abstract, Introduction, Related Work, Method, Experiments, and Conclusion sections. This assistance included grammar correction, sentence revision, terminology consistency, LaTeX syntax, figure captions, equations, citations, and cross-references. The model was not used to generate experimental evidence or to replace the authors' review of the scientific claims.

The model also assisted with software implementation based on specifications provided by the authors. In particular, it was used for code completion, code revision, local refactoring, and debugging in parts of the legality-checked joint-plan interface, the graph-driven world model, teacher-data replay and policy-supervision components, event-adaptive history memory, training and monitoring utilities, and evaluation scripts. It also helped diagnose local problems involving tensor shapes, re-

play schemas, action masks, checkpoint loading, and data-flow contracts, and assisted in preparing unit and integration test cases. The authors made the final decisions about the research question, the team-feasibility formulation, the LIFT architecture, the observation boundary, the data split, the experimental protocol, and the interpretation of results.

We did not use the model to generate synthetic data, formulate the scientific hypotheses or claims, design the experiments, clean or analyze the data, interpret the results, or translate the manuscript. All text and code suggestions used in the final submission were reviewed by the authors. Code used for the reported experiments was checked against the relevant tests and experiment records. The authors take responsibility for the final manuscript, source code, claims, analyses, reported results, and other artifacts.

## REFERENCES

Lars Ankile et al. Residual off-policy RL for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Matteo Bettini, Ajay Shankar, and Amanda Prorok. Heterogeneous multi-robot reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1485–1494, 2023.

Yuhong Cao, Yizhuo Wang, Jingsong Liang, Shuhao Liao, Yifeng Zhang, Peizhuo Li, and Guillaume Sartoretti. HEADER: Hierarchical robot exploration via attention-based deep reinforcement learning with expert-guided reward. *arXiv preprint arXiv:2510.15679*, 2025.

Junhao Chen et al. LITE: A learning-integrated topological explorer for multi-floor indoor environments. *arXiv preprint arXiv:2507.21517*, 2025a.

Junting Chen et al. EMOS: Embodiment-aware heterogeneous multi-robot operating system with LLM agents. In *International Conference on Learning Representations (ICLR)*, 2025b.

Jimmy Chiun, Shizhe Zhang, Yizhuo Wang, Yuhong Cao, and Guillaume Sartoretti. MARVEL: Multi-agent reinforcement learning for constrained field-of-view multi-robot exploration in large-scale environments. *arXiv preprint arXiv:2502.20217*, 2025.

Weidai Dai, Utsav Rai, Jimmy Chiun, Yuhong Cao, and Guillaume Sartoretti. Heterogeneous multi-robot task allocation and scheduling via reinforcement learning. *IEEE Robotics and Automation Letters (RA-L)*, 10(3):2654–2661, 2025.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. MoDem: Accelerating visual model-based reinforcement learning with demonstrations. In *International Conference on Learning Representations (ICLR)*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Machine Learning (ICML)*, 2024.

Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. SMART-LLM: Smart multi-agent robot task planning using large language models. In *International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Ehsan Latif and Ramviyas Parasuraman. CQLite: Communication-efficient multi-robot exploration using coverage-biased distributed Q-learning. *IEEE Robotics and Automation Letters (RA-L)*, 2024.

Xuyang Li, Lei-Lei Li, Jianwu Fang, Boyuan Chen, and Jianru Xue. Event-driven sleep-wake scheduling for heterogeneous robots under LTL constraints. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026a. doi: 10.15607/RSS.2026.XXII.039.

Yao-Hui Li et al. From scalar rewards to potential trends: Shaping potential landscapes for model-based reinforcement learning. *arXiv preprint arXiv:2602.03201*, 2026b.

Yuxiang Li, Kun Chen, Jiancheng Wang, Shihao Fang, Haoyao Chen, and Yunhui Liu. Decentralized heterogeneous multi-robot collaborative exploration for indoor and outdoor 3D environments. *arXiv preprint arXiv:2604.23693*, 2026c.

Kehui Liu et al. COHERENT: Collaboration of heterogeneous multi-robot system with large language models. *arXiv preprint arXiv:2409.15146*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

Yun Qu, Yuhang Jiang, Boyuan Wang, Yixiu Mao, Cheems Wang, Chang Liu, and Xiangyang Ji. Latent reward: LLM-empowered credit assignment in episodic reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 20095–20103, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Brian Yamauchi. Frontier-based exploration using multiple robots. In *International Conference on Autonomous Agents (Agents)*, pp. 47–53, 1998.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2022.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

## A Implementation Details

**Plan representation.** The joint-plan menu is rebuilt at every decision segment. Its slot indices therefore change over time and are not used as action identities in the dynamics model. Instead, the RSSM receives the physical semantics of the selected plan, including the robot, skill, target, task binding, and route attributes. The selected plan is gathered from the candidate tensor with a one-hot selector. A separate head predicts the semantics of candidate plans at future latent states together with their legality.

**Legality masking in imagination.** At execution time, legality comes from the deterministic planner. During imagination, the corresponding mask is produced by the world model. If all candidates fall below the legality threshold, the highest-scoring candidate is left available to avoid an empty action set. The mask used when sampling an imagined action is stored with that action and reused when its log-probability is evaluated.

**Graph reconstruction.** Graph nodes contain continuous, binary, and categorical attributes, which are trained with different reconstruction losses. Continuous fields use masked symlog-MSE, binary fields use Bernoulli loss, and categorical fields use softmax cross-entropy. We also predict the age of delayed semantic observations and provide this quantity to the model when observations arrive asynchronously.

**Teacher mask and BC loss.** The `teacher_mask` is part of the replay record but not the policy observation. It is used only to identify rows on which the behavior-cloning loss in Section 3.4 is applied. The loss uses the existing policy head and is normalized by the number of teacher rows in the batch. We log the number of sampled teacher rows during training to verify that the event-aware sampler provides the intended coverage.

**History selection.** The route ledger stores the complete sequence of executed segments, but only a bounded subset is included in the policy graph. Records associated with interactions, failures, floor changes, task boundaries, newly traversed topology, and changes in task progress receive higher priority. The per-robot budget ranges from 8 to 32 records and changes with recent event density and route uncertainty. Interaction, failure, floor-change, and task-boundary records are retained before lower-priority entries. Records that are not selected are masked in the policy input rather than removed from the fixed graph tensor.

## B Full Results on the 50-Episode Extension

The 50-episode extension contains 10 elevator, 10 door, 10 cup, and 20 general episodes. Table 3 reports the full set of metrics corresponding to the compact comparison in Table 2b.

Table 3: Closed-loop results on the 50-episode extension (mean $\pm$ standard deviation across episodes).

| Method | CR ↑ | CF ↑ | MFC ↑ | Return ↑ | DS ↓ | Time (s) ↓ | Collisions ↓ | Energy (Wh) ↓ |
|---|---|---|---|---|---|---|---|---|
| LIFT (Ours) | **0.90** $\pm$ 0.30 | 0.965 $\pm$ .079 | 0.964 $\pm$ .079 | 16.3 $\pm$ 20.4 | 18.9 $\pm$ 19.7 | 480 $\pm$ 523 | **91.8** $\pm$ 96.6 | 3.49 $\pm$ 3.72 |
| PPO | 0.88 $\pm$ 0.33 | **0.967** $\pm$ .071 | **0.967** $\pm$ .071 | **17.5** $\pm$ 21.0 | **15.2** $\pm$ 18.3 | **349** $\pm$ 365 | 124.8 $\pm$ 162.3 | **3.39** $\pm$ 3.91 |

## C Additional Seed Check

We ran PPO and DecPPO with a second seed using a 2.5-hour training budget. The main runs used approximately 7 hours, so the values in Table 4 should not be compared directly with Table 1. Under this reduced budget, PPO again has higher completion and return than DecPPO.

Table 4: PPO and DecPPO with a second seed and a 2.5-hour training budget (mean $\pm$ standard deviation on the frozen 10-episode split).

| Method | CR ↑ | CF ↑ | MFC ↑ | Return ↑ | DS ↓ | Time (s) ↓ | Collisions ↓ | Energy (Wh) ↓ |
|---|---|---|---|---|---|---|---|---|
| PPO | $0.90 \pm 0.32$ | $0.929 \pm .152$ | $0.879 \pm .310$ | $11.7 \pm 23.0$ | $21.8 \pm 21.3$ | $560 \pm 504$ | $190.7 \pm 141.7$ | $4.94 \pm 3.70$ |
| DecPPO | $0.60 \pm 0.52$ | $0.880 \pm .203$ | $0.780 \pm .412$ | $-6.3 \pm 39.5$ | $30.1 \pm 29.9$ | $846 \pm 839$ | $230.3 \pm 222.1$ | $7.38 \pm 7.21$ |

## D METRIC DEFINITIONS

Metrics are computed per episode and averaged over the corresponding evaluation split. We report the sample standard deviation.

**Completion and coverage.** Completion rate is $\mathrm{CR} = N_{\mathrm{succ}}/N_{\mathrm{total}}$. An episode counts as successful when all task units are completed and each task-relevant floor reaches at least 95% coverage of its reachable area before the episode terminates. Coverage fraction is $\mathrm{CF} = |A_{\mathrm{explored}} \cap A_{\mathrm{free}}|/|A_{\mathrm{free}}|$, where $A_{\mathrm{explored}}$ denotes observed free space and $A_{\mathrm{free}}$ denotes traversable free space on the Habitat occupancy grid (Yamauchi, 1998; Chiun et al., 2025). For multi-floor scenes, minimum floor coverage is $\mathrm{MFC} = \min_f \mathrm{CF}_f$.

**Return and execution cost.** Return is the discounted team reward, $R = \sum_t \gamma^t r_t$. Decision steps (DS) count high-level joint-plan decisions rather than low-level controller updates. Time is the simulated episode duration in Habitat. Collisions include contacts between robots and the environment as well as inter-robot contacts. Energy is $E = \sum_i \int P_i(t)\,\mathrm{d}t$ and is reported in Wh.

**Additional logged metrics.** The communication-disconnect ratio is $\mathrm{CDR} = T_{\mathrm{disconnected}}/T_{\mathrm{total}}$. The invalid-action rate is $\mathrm{IAR} = N_{\mathrm{illegal}}/N_{\mathrm{actions}}$. For methods that act only through the legality-checked joint-plan menu, $\mathrm{IAR} = 0$, so it is not included in the comparison tables.