PLANNER-AWARE CONSEQUENCE FEEDBACK FOR COUPLING TASK ASSIGNMENT AND ROUTE PLANNING IN DYNAMIC MULTI-AGENT RESCUE

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic multi-agent rescue couples task assignment with route planning as hazards alter connectivity and traversal risk. We study how a learned dispatcher can use candidate routes returned by a fixed planner to inform target selection before commitment. We formulate Dynamic Multi-Rescuer Indoor Evacuation (DMRIE), a centralized cooperative decision problem on dynamic multi-floor building graphs, and propose Planner-Aware Consequence Feedback (PACF), a modular planner-to-dispatcher interface. For each legal rescuer–target pair, a fixed planner supplies currently feasible route candidates; PACF encodes their context and aggregates learned candidate scores into a target-logit correction. Mission-level rewards jointly train the dispatcher and PACF, while candidate generation, feasibility checking, and route selection follow fixed planner rules. Under a common execution protocol and building-disjoint splits, evaluations over three training seeds show that adding PACF increases mean Test rescue rates from 91.90% to 92.71% for the MAPDP-based dispatcher and from 90.51% to 93.48% for the MAPT-based dispatcher. Improvements vary across training seeds and task conditions. These results support the usefulness of planner-derived route context for learned target selection in the DMRIE benchmark.

1 INTRODUCTION

In indoor rescue, a small team must reach victims and deliver them to safe areas while fire and smoke change which connections remain traversable (Peña Queralta et al., 2020; Wong & Lee, 2023). A target that looks close can depend on a single deteriorating connector, whereas another may have several executable approaches. Choosing the target therefore determines more than the next destination: it commits a rescuer to a journey whose risk and reachability can change before the rescue is complete. Target choices should account for currently executable routes, while execution must replan as the building graph evolves (Deng et al., 2025; Liu & Zou, 2024).

Existing work recognizes the interaction between assignment and routing. Multi-agent path finding commonly plans routes for assigned goals, while integrated pickup-and-delivery and multi-goal methods optimize targets and paths together under explicit feasibility and cost objectives (Stern et al., 2019; Chen et al., 2021; Zhong et al., 2022). These methods establish that route feasibility belongs in the assignment decision. They leave a different question for learned dispatch: how should several routes returned by an external planner change the policy's preference among targets *before* it commits to one?

Learned dispatchers provide mechanisms for coordinating task choices. MAPDP scores agents' decisions in parallel (Zong et al., 2022); MAPT coordinates them autoregressively as requests arrive over time (Zou et al., 2026). The request dynamics studied there differ from a rescue setting in which the initial targets are known but hazards change graph connectivity and traversal risk. A planner can recompute legal paths as conditions change, but using it only after target selection leaves its candidate routes outside the target comparison. We study how a learned dispatcher can use candidate routes returned by a fixed planner to inform target selection before commitment.

We study this interface in Dynamic Multi-Rescuer Indoor Evacuation (DMRIE), a centralized cooperative task with known initial victim locations, a fixed rescuer team, and a changing multi-floor

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.

graph. Our method, Planner-Aware Consequence Feedback (PACF), queries a deterministic planner for currently feasible routes for each legal rescuer–target pair. As illustrated in Figure 1, PACF scores these candidates using route context and shared state representations, then aggregates the scores into a correction to the dispatcher's target logits. These latent scores inform target selection; route choice follows fixed planner costs. Movement executes one Space transition at a time and replans as the graph changes. PACF and the dispatcher learn jointly from mission-level rewards. We apply the same interface to a parallel MAPDP backbone and an autoregressive MAPT backbone.

Our contributions are:

- We formulate DMRIE as a task-assignment and route-planning problem with known initial targets and evolving graph passability, and instantiate it with 120 tasks on 30 multi-floor buildings for building-disjoint evaluation.
- We introduce PACF, a planner-to-dispatcher interface that converts currently feasible route alternatives into target-logit corrections learned from mission rewards, while retaining fixed candidate generation, feasibility checking, and route-selection rules.
- We evaluate PACF with matched controls on parallel and autoregressive dispatch backbones, report performance variation across training seeds and task conditions, and measure decision latency on a fixed evaluation subset.

2 PRELIMINARIES

2.1 PROBLEM FORMULATION

We study joint task assignment and route planning for a fixed team of M agents serving known targets in a dynamic environment. The environment is a graph $G = (V, E)$ whose vertices represent locations and whose edges carry traversal attributes. At dispatch event t, the currently executable graph is $G_t = (V, E_t)$, with $E_t \subseteq E$. Environmental changes can disable nodes or connections and alter traversal risk. We instantiate this problem as Dynamic Multi-Rescuer Indoor Evacuation (DMRIE): agents are rescuers, vertices are indoor *Spaces*, and the mission is to evacuate known victims to safe locations. The building, victim locations, safe areas, and team size are specified before an episode; Appendix B describes the instances and observations, and Section 3.3 summarizes the execution protocol.

Let $\mathcal{T}_i(s_t)$ be the legal target set for agent i in state s_t, and let $\mathcal{F}_{ij,t}$ contain its currently executable paths to target j. A movement decision selects a feasible target–route pair:

$$(j_{i,t}, r_{i,t}) \in \big\{(j, r) : j \in \mathcal{T}_i(s_t),\ r \in \mathcal{F}_{ij,t}\big\}. \tag{1}$$

Together with service and waiting actions, these pairs form the joint dispatch action $\widetilde{\mathbf{u}}_t$. Task assignment determines which agents serve which targets and in what sequence, while route planning determines executable paths to those targets. The two are coupled: a graph change can invalidate a path or make a target unreachable, and a detour can change an assignment's value through increased delay and exposure.

Following the cooperative joint-action formulation of Boutilier (1996), we model the underlying dynamics as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{U}, \mathsf{P}, r, \gamma)$. Its complete state contains the environment, all agents and tasks, and ongoing execution progress. The legal joint action $\mathbf{u}_n \in \mathcal{U}(s_n^{\text{sim}})$ specifies new commands or continuation of ongoing activities, $\mathsf{P}(s_{n+1}^{\text{sim}} \mid s_n^{\text{sim}}, \mathbf{u}_n)$ describes the transition, and r_n is the shared team reward. With n indexing simulation steps, the objective is

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{n=0}^{L-1} \gamma^n r_n\right], \qquad r_n = r(s_n^{\text{sim}}, \mathbf{u}_n, s_{n+1}^{\text{sim}}), \tag{2}$$

where L is the episode length and γ discounts each simulation step. Event t occurs at simulation step τ_t, with $\tau_0 = 0$ and $s_t = s_{\tau_t}^{\text{sim}}$. Activities continue between events. Appendix C.3 specifies how event transitions accumulate the rewards in this objective.

2.2 PPO UNDER CENTRALIZED TRAINING AND CENTRALIZED EXECUTION

We adopt *centralized training and centralized execution* (CTCE). A single coordinator receives the global observation $o_t = \Omega(s_t)$, containing graph-wide environment and task features, all agents'

states, and mission progress. It constructs the team decision by selecting one action for each available agent. Training uses a shared team reward and a centralized critic; execution retains the same global observation interface and central decision maker. The controlled units carry out the coordinator's commands.

Let ω collect the trainable policy parameters and c_t denote the coordinator's context, containing observation inputs, temporal features, and feasible-action information. We retain an agent-level policy $\pi^i_\omega(\widetilde{u}_{i,t} \mid x_{i,t})$ for each unit i. It selects an action from that unit's legal action set, with a target and route included for movement. Its context $x_{i,t}$ contains centrally available information, agent-specific features and masks, and the relevant coordination context. For MAPT, $x_{i,t}$ additionally includes preceding action types and targets, excluding route indices; MAPDP uses no current-action prefix. The team decision collects these individual actions as $\widetilde{\mathbf{u}}_t = (\widetilde{u}_{1,t}, \ldots, \widetilde{u}_{M,t})$, with probability

$$\pi_\omega(\widetilde{\mathbf{u}}_t \mid c_t) = \prod_{i=1}^{M} \pi^i_\omega(\widetilde{u}_{i,t} \mid x_{i,t}), \tag{3}$$

These policies are parameterized by the central model while retaining a separate action distribution for each agent. Non-ready agents contribute a probability-one placeholder action and continue their ongoing activities. Section 3 specifies the policy parameterization; Appendix C.1.3 details the conditional factors.

We optimize this joint policy using proximal policy optimization (PPO) (Schulman et al., 2017). Using Equation 3, PPO evaluates a stored joint action with recomputed network features and maximizes the clipped surrogate:

$$\rho_t(\omega) = \frac{\pi_\omega(\widetilde{\mathbf{u}}_t \mid c_t)}{\pi_{\omega_{\text{old}}}(\widetilde{\mathbf{u}}_t \mid c_t)} = \prod_{i=1}^{M} \frac{\pi^i_\omega(\widetilde{u}_{i,t} \mid x_{i,t})}{\pi^i_{\omega_{\text{old}}}(\widetilde{u}_{i,t} \mid x_{i,t})}, \tag{4}$$

$$L_{\text{clip}}(\omega) = \widehat{\mathbb{E}}_t\Big[\min\Big(\rho_t(\omega)\widehat{A}_t, \text{clip}(\rho_t(\omega), 1-\epsilon, 1+\epsilon)\widehat{A}_t\Big)\Big], \tag{5}$$

where $\widehat{A}_t$ is the team advantage estimated using the centralized critic and ϵ is the clipping threshold. In implementation, agent-level log-probabilities are summed to obtain the joint log-probability, and clipping is applied to the resulting joint ratio. Actor updates maximize this surrogate with entropy regularization, while the critic is fitted to return targets. Section 3.3 specifies how the dispatcher and PACF are trained within this framework; Appendix C.3 gives the advantage estimator and optimization settings.

3 METHOD

In this section, we present Planner-Aware Consequence Feedback (PACF), a planner-to-dispatcher interface for coupling task assignment with route planning. As Figure 1 shows, a fixed planner supplies currently feasible route candidates; PACF scores them using shared backbone features and aggregates the scores into a rescuer–target correction C_{ij} before commitment. The backbone and PACF learn jointly from mission rewards, while candidate generation, feasibility checking, and route selection follow fixed planner rules. We first define the decision interface, then explain how route context produces target feedback, and finally describe execution and learning. Algorithm 1 in Appendix C gives the complete procedure.

3.1 COUPLING TASK ASSIGNMENT AND ROUTE PLANNING

Targets with similar planner costs may differ in their next transitions, continuations, and available alternatives. For each legal rescuer–target pair, the planner supplies currently feasible routes $\mathcal{R}_{ij,t} \subseteq \mathcal{F}_{ij,t}$ and base costs $b_{ijk,t}$. PACF summarizes this route context as a target correction $C_{ij,t}$. We suppress t below and write $x_i = x_{i,t}$. Its masked target set $\mathcal{T}_i(x_i)$ also respects active commitments.

Let z^0_{ij} be the backbone's base target logit. Conditional on a movement action, PACF combines the dispatcher's preference with the planner-derived feedback:

$$p_{\varphi,\theta}(j \mid \text{MOVE}, x_i) = \frac{\exp(z^0_{ij} - C_{ij}/\tau_A)}{\sum_{j' \in \mathcal{T}_i(x_i)} \exp(z^0_{ij'} - C_{ij'}/\tau_A)}. \tag{6}$$

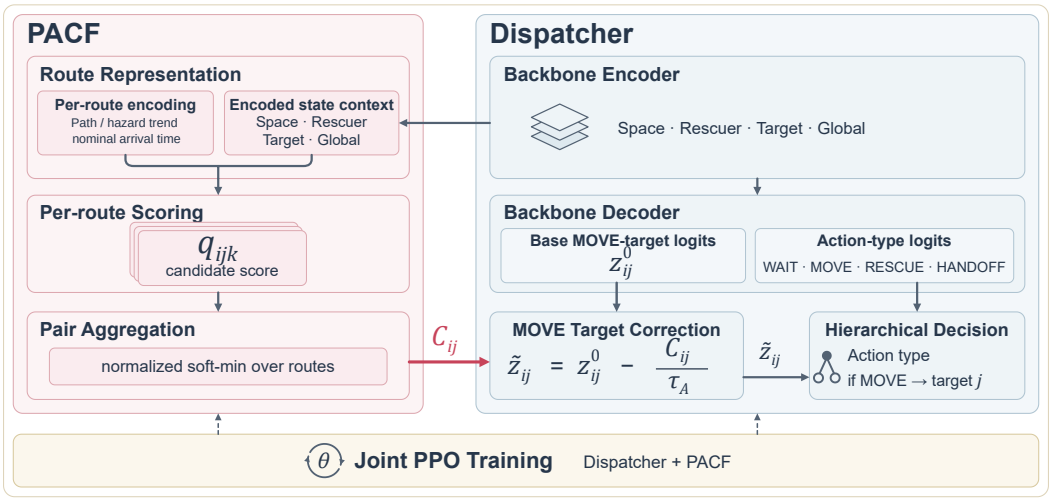

Figure 1: PACF's planner-to-dispatcher interface. A fixed planner supplies currently feasible route candidates. Learned scores are aggregated into C_{ij} to correct target logits before commitment; they do not select the executed route, which follows fixed planner costs. Joint PPO updates the dispatcher and PACF.

Here φ and θ are the backbone and PACF parameters, respectively, and $\tau_A = 1$. The backbone contributes task and team context, while C_{ij} conditions target preference on planner-derived route context. A lower correction increases a target's relative probability. Each agent retains its own action distribution, with the correction entering the conditional target factor; the action-type head remains jointly trainable.

After a target is selected, its route is chosen using the fixed planner-cost distribution

$$p_0(k \mid i, j, x_i) = \frac{\exp(-b_{ijk}/\tau_R)}{\sum_{k'=1}^{K_{ij}} \exp(-b_{ijk'}/\tau_R)}, \qquad \tau_R = 1. \tag{7}$$

Training samples from this distribution, and greedy evaluation selects a maximum-probability candidate. The selected route supplies the next Space transition, with passability checked during execution. Together, the two distributions connect route-informed task choice to an executable continuation. The paired controls set $C_{ij} = 0$ while retaining the same planner and route-selection rule.

This interface applies to both dispatch structures in our experiments. MAPDP (Zong et al., 2022) produces rescuer scores in parallel, whereas MAPT (Zou et al., 2026) conditions each decoder context on preceding dispatch decisions. In either case, PACF computes pair feedback from the shared observation before dispatch decoding, and each rescuer uses its own row to correct target logits.

3.2 Constructing planner-to-policy feedback

The left side of Figure 1 begins with routes feasible on the current graph. For each legal pair (i, j), the planner retains at most three candidates with distinct first Space transitions and minimum-fixed-cost continuations, using nominal duration and current risk. Blocked connections are excluded; feasibility is checked again during execution as hazards evolve.

PACF encodes each candidate's first transition, pooled path features, nominal travel times, and candidate-relative statistics, conditioned on shared rescuer, target, and mission embeddings. The resulting representation z_{ijk} uses current and preceding hazard observations and geometric travel estimates. Appendix C.1 specifies the encoder architecture and input features.

The Per-route Scoring block in Figure 1 maps z_{ijk} to a bounded score,

$$q_{ijk} = q_{\max} \tanh f_\theta(z_{ijk}), \qquad q_{\max} = 4, \tag{8}$$

and Pair Aggregation converts the scores of one target's alternatives into the correction used in Equation 6:

$$C_{ij} = -\tau_P \log \left(\frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \exp(-q_{ijk}/\tau_P) \right), \qquad \tau_P = 1. \tag{9}$$

The normalized soft minimum emphasizes lower scores without rewarding a target merely for having more equally scored candidates. Invalid pairs are masked. The candidate scores serve as latent features for target selection; they are not explicit route-return estimates and do not determine which route is executed. Conditional on the selected target, route selection uses only the fixed planner costs: training samples from Equation 7, while evaluation selects a minimum-cost candidate. The red C_{ij} connection in the figure therefore feeds route context into target comparison before commitment. Appendix A analyzes the aggregation and target correction.

3.3 Route Execution and Joint Learning

Agent-wise decisions and execution. At dispatch event t, the central model produces a policy $\pi^i_\omega(\cdot \mid x_{i,t})$ for each agent, with $\omega = (\varphi, \theta)$. Each ready agent samples its own action. For movement, this consists of an action type, a target selected using PACF feedback, and a candidate route selected using the fixed planner costs:

$$\pi^i_\omega((\text{MOVE}, j, k) \mid x_{i,t}) = p^i_\omega(\text{MOVE} \mid x_{i,t})\, p^i_\omega(j \mid \text{MOVE}, x_{i,t})\, p_0(k \mid i, j, x_{i,t}). \tag{10}$$

The selected route describes a path to the target, but the issued movement command executes only its next Space transition. Upon arrival at that next node, the coordinator observes the updated environment and the planner recomputes feasible continuations. An agent still committed to a valid target selects a new continuation toward that target; completion or invalidation of the commitment restores the corresponding task choices. Movement also checks passability and handles blocked transitions. At each decision boundary, non-ready agents retain probability-one placeholder actions and continue their ongoing activities. This repeated planning and next-transition execution adapts the route to graph changes throughout the journey.

Reward from rescue outcomes. The team receives reward for pickups and completed evacuations and a penalty for victim incapacitation. Evacuation credit depends on remaining health and mission progress and arrives only after safe-area queue completion. This delayed shared signal connects the earlier target choice, its route conditions, and the subsequent rescue outcome. Appendix C.3 gives the coefficients and event-level reward accumulation.

Updating the dispatcher and feedback. Rewards between dispatch events are accumulated into $\bar{r}_t$; the centralized critic uses these rewards, the next observation, and elapsed simulation steps to estimate team advantages $\widehat{A}_t$ and return targets (Appendix C.3). PPO combines the recorded agent-action probabilities using the joint ratio in Equation 4 and updates the policy with Equation 5. The critic is fitted to normalized return targets following Yu et al. (2022).

Training evaluates each recorded decision with its original legal routes and masks. Gradients from the corrected target probabilities pass through C_{ij}, candidate scores, and the route encoder into the shared backbone. In this way, rescue outcomes teach the dispatcher how route context should alter an earlier target choice. The planner itself remains fixed; its route factor p_0 cancels from the PPO ratio for a stored candidate set. The feedback scorer starts at zero output, so the initial target distribution equals the backbone's. Evaluation follows the same decision and execution flow with greedy choices and fixed parameters.

4 Experiments

We organize the experiments around three questions: **RQ1:** Does adding PACF improve mean rescue rate across dispatch backbones? **RQ2:** How do the observed gains vary across task conditions? **RQ3:** What additional decision cost does PACF incur?

Dataset. DMRIE contains 120 rescue tasks from 30 six-floor buildings, with three rescuers per task and building-disjoint Train/Validation/Test splits of 48/24/48 tasks. Medium, Hard, and Expert tasks contain 12, 18, and 24 victims, respectively. Appendix B defines the instances, observations, and dataset splits.

Baselines. We adapt two learned backbones to multi-agent task assignment: MAPDP (Zong et al., 2022) assigns targets through parallel pointer heads, whereas MAPT (Zou et al., 2026) selects agents' actions autoregressively. Both originate from pickup-and-delivery problems and are evaluated without and with PACF (Base and +PACF). We also compare nearest-target dispatch (Nearest), risk-cost Hungarian assignment (Hu), simulated annealing (SA), and genetic search (GA). All methods share the rescue execution protocol; Appendix D.1 summarizes the baseline planning rules and search budgets.

Evaluation. We train each learned condition with PPO using three random seeds, select checkpoints by Validation rescue rate, and evaluate them greedily on the Test set. Each run trains one policy on tasks from all three difficulty levels. Within each backbone, Base and +PACF share the planner, route-selection rule, training budget, and checkpoint-selection protocol. Rescue rate is the percentage of initial victims evacuated, pooling victim counts across tasks within each run; the appendix gives the full evaluation protocol.

4.1 Effect of PACF across dispatch backbones

Table 1: Test rescue rate (%). Learned methods report mean ± SD over three random seeds.

Level	Nearest	Hu	SA	GA	MAPDP		MAPT	
					Base	+PACF	Base	+PACF
Medium	**100.00**	**100.00**	98.44	**100.00**	97.22 ± 3.05	95.66 ± 3.05	92.71 ± 1.38	95.66 ± 0.30
Hard	96.18	95.49	95.49	85.07	97.45 ± 0.80	$\mathbf{97.57} \pm 0.92$	95.02 ± 1.40	$\mathbf{97.57} \pm 1.20$
Expert	79.17	81.77	51.04	71.35	85.07 ± 1.97	87.59 ± 2.62	86.02 ± 2.57	$\mathbf{89.32} \pm 4.54$
Overall	89.47	90.39	76.39	82.29	91.90 ± 1.04	92.71 ± 0.23	90.51 ± 1.22	$\mathbf{93.48} \pm 2.48$

To answer RQ1, we compare PACF with its matched Base under both parallel and autoregressive assignment, alongside the rule- and search-based methods. PACF raises mean overall Test rescue rate by 0.81 percentage points for MAPDP and 2.97 for MAPT, reaching 92.71% and 93.48%, respectively (Table 1). Both exceed the strongest heuristic, Hu (90.39%).

Table 2: PACF-minus-Base Test rescues by training seed (864 victims per seed).

Backbone	Seed 1	Seed 2	Seed 3
MAPDP	+7	+14	0
MAPT	−3	+52	+28

Table 2 shows nonnegative gains for all three MAPDP runs, while MAPT loses three rescues in seed 1 despite its larger mean gain. The matched comparisons show higher mean rescue rates with PACF on both backbones in the evaluated runs, with variation across seeds.

4.2 Variation across task conditions

For RQ2, Table 1 shows that mean gains vary by difficulty. Pooled across the three seeds, MAPDP changes by −9, +1, and +29 rescues on Medium, Hard, and Expert, respectively; its overall increase is concentrated on Expert tasks. MAPT improves at every difficulty in the three-seed mean, although this does not hold for every seed. We also compare the four initial hazard placements: bottleneck (BP), cross-floor (CF), stair (SP), and victim-floor (VP) pressure.

Table 3: PACF-minus-Base change in evacuated victim instances, pooled across three training seeds. Each hazard-placement group contains 648 victim instances per method.

Hazard placement	MAPDP	MAPT
Bottleneck (BP)	+4	+2
Cross-floor (CF)	+17	0
Stair (SP)	−9	+63
Victim-floor (VP)	+9	+12
Overall	+21	+77

Stair-pressure tasks account for 63 of MAPT's 77 additional rescues (Table 3); the corresponding MAPDP count decreases by 9. The two backbones therefore show different responses to the same interface.

Medium tasks also expose a limitation. Pooled across seeds, each PACF backbone rescues $551/576$ victims, while Nearest and Hu reach 100% in their fixed evaluations. Of the 25 unrescued Medium victims per PACF backbone, 23 for MAPDP and 22 for MAPT occur under stair pressure (Table 12). These results characterize where improvements and errors occur without identifying their causal mechanism.

4.3 Computational overhead

For RQ3, with P rescuer–target pairs, $R \leq 3P$ valid routes, S route-edge occurrences, E_g graph edges, and network width d, feedback costs $O((S+R)d^2 + RE_g + P + E_g)$ at fixed depth. The RE_g term reflects dense overlap; this bound excludes planner search, which the three-route limit does not bound. We therefore measure complete decision latency.

We time the selected MAPT seed-1 checkpoints on three cross-floor tasks, one per difficulty, repeated three times per method (nine episodes each), using an exclusive NVIDIA L40, one CPU thread, and CUDA-synchronized FP32 inference (TF32 disabled). A single call returns the whole team's joint action. Timing covers candidate planning, network evaluation, route scoring, and decoding, while excluding model loading, initial static binding, warmup, and environment advancement. Mean and P95 latency pool all individual decisions per method; cumulative decision time is averaged over the nine episode totals.

Table 4: MAPT decision computation on the fixed cross-floor timing subset. Episode values are mean cumulative decision time.

Method	Mean (ms)	P95 (ms)	Decision time / episode (s)	Decisions / episode
MAPT Base	34.686	53.291	7.249	209.0
MAPT + PACF	35.585	56.552	5.314	149.3

Table 4 shows a 0.90 ms (2.6%) increase in mean decision latency and a 3.26 ms increase in P95 with PACF. Mean decisions per episode decrease from 209.0 to 149.3 (28.5%), while cumulative decision time decreases from 7.249 s to 5.314 s (26.7%). These episode totals depend on the policy trajectories and termination times; they do not imply lower per-call complexity or measure total simulation wall-clock time.

The matched Base retains candidate construction and route scoring while suppressing feedback. The measured latency difference is therefore relative to this shared implementation, rather than the full cost of PACF relative to a stripped-down dispatcher.

5 Related Work

Joint task assignment and path planning. Classical multi-agent path finding coordinates collision-free paths for assigned goals (Stern et al., 2019). Integrated pickup-and-delivery and

multi-goal methods jointly optimize assignments and paths (Chen et al., 2021; Zhong et al., 2022), while hierarchical methods address their coupling at different levels (Ho et al., 2024). Task-and-motion planning connects abstract commitments to executable motions (Kaelbling & Lozano-Perez, 2011; Motes et al., 2020), and action-graph formulations account for task-induced mobility changes (Okubo & Takahashi, 2023). These works already couple assignment and routing. PACF learns how executable route alternatives modify target preferences, with fixed feasibility checks and route selection.

Learned routing and multi-agent dispatch. Pointer networks and attention models learn constructive routing policies (Vinyals et al., 2015; Kool et al., 2019); multiple decoders diversify the resulting solutions (Xin et al., 2021). Dynamic pickup-and-delivery adds online requests and service constraints (Cai et al., 2023). Hierarchical reinforcement learning can partition incoming orders before routing optimization (Ma et al., 2021). MAPDP learns cooperative dispatch with parallel agent scoring (Zong et al., 2022), whereas MAPT generates joint decisions autoregressively (Zou et al., 2026). DMRIE fixes the initial targets and varies traversability and risk. PACF supplies candidate-route feedback to both backbones before target commitment, preserving their coordination structures.

Learning-augmented planning. Learning can guide conflict selection in conflict-based search (Huang et al., 2021), supply graph-transformer heuristics for bounded-suboptimal planning (Yu et al., 2023), or guide neighborhood search (Yan & Wu, 2024). Conversely, heuristic search can improve learned local MAPF policies (Veerapaneni et al., 2024). These methods combine learning and planning within path construction or repair. PACF instead modifies the upstream target distribution using returned routes, with fixed search and route-selection rules.

Dynamic rescue and indoor navigation. Multi-robot search and rescue combines planning, coordination, and perception (Peña Queralta et al., 2020). Disaster-response allocation considers task deadlines and robot constraints (Ghassemi & Chowdhury, 2022); safety-aware allocation can incorporate stochastic hazard dynamics (Tihanyi et al., 2023). Indoor systems use semantic connectivity graphs (Zhu et al., 2025) and BIM-based navigation with shared emergency information (Wong & Lee, 2023), while evacuation planners update routes using fire conditions (Deng et al., 2025; Liu & Zou, 2024). These studies motivate DMRIE's changing building graph. Our scope is centralized dispatch with known victim locations and observed hazards, without perception or future hazard prediction.

6 Conclusion

We formulated DMRIE to study task assignment and route planning as hazards change graph connectivity, and proposed PACF as a planner-to-dispatcher interface that uses candidate-route context to inform target selection before commitment. Mission rewards jointly train the dispatcher and feedback module, while planning and route selection follow fixed rules. Across three training seeds on building-disjoint Test tasks, PACF improves mean overall rescue rate for both MAPDP and MAPT, with a 2.6% increase in mean decision latency on the evaluated MAPT subset. Gains vary across seeds and task conditions, including a decline on Medium tasks for MAPDP. Each run trains one policy across all levels with fixed reward weights, leaving the effects of difficulty mixing and reward design unresolved. Future work will compare difficulty-conditioned and separately trained policies under matched budgets and test reward-weight sensitivity.

AI USE STATEMENT

We used generative AI to help restructure and edit this manuscript, inspect code and experiment records, and write scripts for aggregating results and producing figures and tables. All reported experimental values come from archived simulator outputs. This disclosure covers the assistance used in the present revision.

ETHICS STATEMENT

The study uses a synthetic rescue simulator. It omits perception errors, victim search, clinical triage, and many operational constraints of real incidents. The results have not been validated for deployment in human rescue operations. We report incapacitated and unresolved victims separately because an incomplete rescue does not necessarily imply incapacitation.

REPRODUCIBILITY STATEMENT

Appendix A gives the conditional derivations; Appendix B describes DMRIE instances and dataset splits. Appendix C gives PACF pseudocode, network and dispatch details, and training configuration. Appendix D contains the evaluation protocol and additional results by split, seed, and building. We plan to provide the implementation, frozen benchmark, selected checkpoints for the reported methods, configurations, task schedules, recorded results, and vector task diagrams as supplementary material through the submission system, together with commands for integrity checks, rebuilding statistics, and evaluating fixed checkpoints.

REFERENCES

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 195–210, 1996. URL https://www.cs.toronto.edu/~cebly/Papers/tark96.pdf.

Junchuang Cai, Qingling Zhu, Qiuzhen Lin, Lijia Ma, Jianqiang Li, and Zhong Ming. A survey of dynamic pickup and delivery problems. *Neurocomputing*, 554:126631, 2023. doi: 10.1016/j.neucom.2023.126631.

Zhe Chen, Javier Alonso-Mora, Xiaoshan Bai, Daniel D Harabor, and Peter J Stuckey. Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. *IEEE Robotics and Automation Letters*, 6(3):5816–5823, 2021. doi: 10.1109/lra.2021.3074883.

Hui Deng, Zihao Fan, Xinyi Wei, and Yichuan Deng. An ontology-based approach to dynamic indoor fire emergency evacuation path planning with BIM integration. *Journal of Building Engineering*, 106:112562, 2025. doi: 10.1016/j.jobe.2025.112562.

Payam Ghassemi and Souma Chowdhury. Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints. *Robotics and Autonomous Systems*, 147:103905, 2022. doi: 10.1016/j.robot.2021.103905.

Florence Ho, Ryota Higa, Takuro Kato, and Shinji Nakadai. A hierarchical approach for joint task allocation and path planning. *IEEE Robotics and Automation Letters*, 9(11):10137–10144, 2024. doi: 10.1109/lra.2024.3455787.

Taoan Huang, Sven Koenig, and Bistra Dilkina. Learning to resolve conflicts for multi-agent path finding with conflict-based search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11246–11253, 2021. doi: 10.1609/aaai.v35i13.17341.

Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477, 2011. doi: 10.1109/icra.2011.5980391.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Zhaohui Liu and Ruihong Zou. Dynamic evacuation path planning for subway station fire based on IACO. *Journal of Building Engineering*, 86:108828, 2024. doi: 10.1016/j.jobe.2024.108828.

Yi Ma, Xiaotian Hao, Jianye Hao, Jiawen Lu, Xing Liu, Xialiang Tong, Mingxuan Yuan, Zhigang Li, Jie Tang, and Zhaopeng Meng. A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23609–23620, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c6a01432c8138d46ba39957a8250e027-Paper.pdf.

James Motes, Read Sandstrom, Hannah Lee, Shawna Thomas, and Nancy M. Amato. Multi-robot task and motion planning with subtask dependencies. *IEEE Robotics and Automation Letters*, 5 (2):3338–3345, 2020. doi: 10.1109/lra.2020.2976329.

Takuma Okubo and Masaki Takahashi. Multi-agent action graph based task allocation and path planning considering changes in environment. *IEEE Access*, 11:21160–21175, 2023. doi: 10.1109/access.2023.3249757.

Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8: 191617–191643, 2020. doi: 10.1109/access.2020.3030190.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727.

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

Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Barták. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Proceedings of the International Symposium on Combinatorial Search*, 10(1):151–158, 2019. doi: 10.1609/socs.v10i1.18510.

Richard S. Sutton, Doina Precup, and Satinder P. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Daniel Tihanyi, Yimeng Lu, Orcun Karaca, and Maryam Kamgarpour. Multi-robot task allocation for safe planning against stochastic hazard dynamics. In *2023 European Control Conference (ECC)*, pp. 1–6, 2023. doi: 10.23919/ecc57647.2023.10178126.

Rishi Veerapaneni, Qian Wang, Kevin Ren, Arthur Jakobsson, Jiaoyang Li, and Maxim Likhachev. Improving learnt local MAPF policies with heuristic search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 597–606, 2024. doi: 10.1609/icaps.v34i1.31522.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf.

Mun On Wong and Sanghoon Lee. Indoor navigation and information sharing for collaborative fire emergency response with BIM and multi-user networking. *Automation in Construction*, 148: 104781, 2023. doi: 10.1016/j.autcon.2023.104781.

Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):12042–12049, 2021. doi: 10.1609/aaai.v35i13.17430.

Zhongxia Yan and Cathy Wu. Neural neighborhood search for multi-agent path finding. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d41f8403e9bb5141bc2c81fad7658185-Paper-Conference.pdf.

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. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Chenning Yu, Qingbiao Li, Sicun Gao, and Amanda Prorok. Accelerating multi-agent planning using graph transformers with bounded suboptimality. In *2023 IEEE International Conference on Robotics and Automation*, pp. 3432–3439. IEEE, 2023. doi: 10.1109/icra48891.2023.10161018.

Xinyi Zhong, Jiaoyang Li, Sven Koenig, and Hang Ma. Optimal and bounded-suboptimal multi-goal task assignment and path finding. In *2022 International Conference on Robotics and Automation*, pp. 10731–10737. IEEE, 2022. doi: 10.1109/icra46639.2022.9812020.

Junxiang Zhu, Mun On Wong, Nicholas Nisbet, Jinying Xu, Tom Kelly, Sisi Zlatanova, and Ioannis Brilakis. Semantics-based connectivity graph for indoor pathfinding powered by IFC-Graph. *Automation in Construction*, 171:106019, 2025. doi: 10.1016/j.autcon.2025.106019.

Zefang Zong, Meng Zheng, Yong Li, and Depeng Jin. MAPDP: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 9980–9988, 2022. doi: 10.1609/aaai.v36i9.21236.

Zengyu Zou, Jingyuan Wang, Yixuan Huang, and Junjie Wu. Multi-agent pointer transformer: Seq-to-seq reinforcement learning for multi-vehicle dynamic pickup-delivery problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16593–16601, 2026. doi: 10.1609/aaai.v40i19.38700.

A Additional Derivations

We analyze PACF at a fixed observation and decoder context, with legal masks, candidate routes, and backbone logits held fixed. For MAPT, this also fixes the preceding dispatch prefix. The results characterize the feedback interface and give a sufficient condition for improving conditional expected return. They do not assume that PPO has learned accurate action values. The variational arguments use standard relative-entropy identities, related to policy optimization with relative-entropy constraints (Peters et al., 2010).

A.1 Variational meaning and stability of route aggregation

For one valid pair, write $q = (q_1, \dots, q_K)$, let u_K be uniform on its K candidates, and let Δ_K be the probability simplex. All statements below concern the computed scores on this fixed candidate set.

Proposition A.1 (Normalized route aggregation). *For $\tau_P > 0$, Equation 9 satisfies*

$$C(q) = \min_{w \in \Delta_K} \left\{ \sum_k w_k q_k + \tau_P \, \mathrm{KL}(w \| u_K) \right\}, \qquad \alpha_k(q) = \frac{e^{-q_k/\tau_P}}{\sum_\ell e^{-q_\ell/\tau_P}} \tag{11}$$

where α is the unique minimizer. Moreover,

$$\min_k q_k \le C(q) \le \frac{1}{K} \sum_k q_k, \qquad |C(q) - C(q')| \le \|q - q'\|_\infty. \tag{12}$$

Proof. Writing $Z_K = K^{-1} \sum_k e^{-q_k/\tau_P}$ and expanding the divergence gives

$$\sum_k w_k q_k + \tau_P \, \mathrm{KL}(w \| u_K) = -\tau_P \log Z_K + \tau_P \, \mathrm{KL}(w \| \alpha). \tag{13}$$

Nonnegativity of KL proves the minimum and its uniqueness. The lower bound follows from $\sum_k w_k q_k \ge \min_k q_k$; choosing $w = u_K$ gives the upper bound. Finally, $\partial C / \partial q_k = \alpha_k$, whose entries are nonnegative and sum to one. Integrating this gradient along the segment between q and q' proves the stated Lipschitz bound. □

Consequently, $|C(q)| \le q_{\max}$ for the bounded scorer in Equation 8. Equal candidate scores $q_k = c$ give $C(q) = c$ for every K, so equal scores produce the same correction regardless of candidate count. The weights α describe feedback aggregation; execution still uses the fixed planner-cost distribution p_0 in Equation 7. These weights coincide only in special cases.

A.2 Target correction as a KL-regularized decision

Fix an agent and suppress its index. On its nonempty legal target set $\mathcal{T}$, define the uncorrected distribution $p_B(j) = \mathrm{softmax}(z^0)_j$ and the PACF distribution p_C from Equation 6. Here p_B uses the *same* backbone parameters as p_C; it is not a separately trained Base checkpoint. Assume finite legal logits, so $p_B(j) > 0$.

Proposition A.2 (Regularized target reweighting). *For $\tau_A > 0$, PACF is the unique solution of*

$$p_C = \operatorname*{argmin}_{p \in \Delta_\mathcal{T}} \left\{ \mathbb{E}_p[C_j] + \tau_A \, \mathrm{KL}(p \| p_B) \right\}. \tag{14}$$

Its reduction in expected learned feedback cost is exactly

$$\mathbb{E}_{p_B}[C_j] - \mathbb{E}_{p_C}[C_j] = \tau_A [\mathrm{KL}(p_C \| p_B) + \mathrm{KL}(p_B \| p_C)] \ge 0. \tag{15}$$

Proof. Let $Z_A = \sum_j p_B(j) e^{-C_j/\tau_A}$. Equation 6 gives $p_C(j) = p_B(j) e^{-C_j/\tau_A} / Z_A$, hence

$$\mathbb{E}_p[C_j] + \tau_A \, \mathrm{KL}(p \| p_B) = -\tau_A \log Z_A + \tau_A \, \mathrm{KL}(p \| p_C). \tag{16}$$

This proves the optimizer. Taking expectations of $\log(p_C / p_B) = -C/\tau_A - \log Z_A$ under p_C and p_B and subtracting yields Equation 15. □

Thus the correction trades a lower expected target-feedback score against departure from the backbone preference. A constant C_j recovers p_B, including zero-initialized feedback. This variational objective describes PACF's forward computation; the implemented training objective remains PPO.

For a sampled target $j^\star$, the feedback path has the explicit derivative

$$\frac{\partial \log p_C(j^\star)}{\partial q_{jk}} = -\frac{\mathbf{1}\{j = j^\star\} - p_C(j)}{\tau_A}\,\alpha_{jk}. \tag{17}$$

This follows by differentiating the target softmax and using Proposition A.1. It specifies how a target-level policy gradient reaches candidate scores while the planner is fixed. It is a partial derivative through feedback; shared backbone parameters also receive their ordinary logit gradients. Singleton target sets have zero target-selection gradient, and clipping or saturated score encoders can limit parameter updates.

A.3 A SUFFICIENT CONDITION FOR CONDITIONAL RETURN IMPROVEMENT

Let Q_j be the expected discounted team return after choosing target j at the fixed movement context, with route selection following p_0. Hold the rules for all other current decisions and subsequent actions fixed in this definition; for MAPT, later decisions may depend on j through the same fixed conditional policy. This defines a comparison of one target distribution, including its downstream consequences.

Proposition A.3 (Improvement under calibrated feedback)**.** *Suppose there exist a target-independent offset a, a scale $\kappa > 0$, and $\varepsilon \geq 0$ such that*

$$\max_{j \in \mathcal{T}} |C_j - (a - \kappa Q_j)| \leq \varepsilon. \tag{18}$$

With $\mathrm{TV}(p, q) = \frac{1}{2}\sum_j |p(j) - q(j)|$ *and* $D_{\mathrm{sym}} = \mathrm{KL}(p_C \| p_B) + \mathrm{KL}(p_B \| p_C)$,

$$\mathbb{E}_{p_C}[Q_j] - \mathbb{E}_{p_B}[Q_j] \geq \frac{\tau_A D_{\mathrm{sym}} - 2\varepsilon\,\mathrm{TV}(p_C, p_B)}{\kappa}. \tag{19}$$

In particular, exact calibration gives nonnegative conditional return improvement.

Proof. Write $C_j = a - \kappa Q_j + e_j$, where $|e_j| \leq \varepsilon$. Proposition A.2 gives

$$\kappa(\mathbb{E}_{p_C}[Q_j] - \mathbb{E}_{p_B}[Q_j]) = \tau_A D_{\mathrm{sym}} + \sum_j (p_C(j) - p_B(j)) e_j. \tag{20}$$

The final term is at least $-\varepsilon \sum_j |p_C(j) - p_B(j)|$, proving the result. □

For approximate feedback, a sufficient condition is $\tau_A D_{\mathrm{sym}} \geq 2\varepsilon\,\mathrm{TV}(p_C, p_B)$. This identifies a regime where calibration error is small enough relative to the induced policy change. PACF is trained from team rewards, and our experiments do not estimate the calibration error ε. The proposition therefore provides a conditional justification, not a convergence guarantee for PPO or a guarantee for simultaneous changes to all agent policies. It applies to the stochastic conditional distribution; greedy Test decoding and the rescue-rate metric require their own empirical evaluation. Reward-weighted return and rescue rate are also distinct objectives.

B DMRIE AND DATASETS

B.1 DMRIE INSTANCE AND OBSERVATIONS

A DMRIE instance specifies a directed building graph $G = (V, E)$, a fixed rescuer team $\mathcal{R} = \{1, \ldots, M\}$, an initial victim set $\mathcal{P} = \{1, \ldots, N\}$, and safe Spaces $V_{\text{safe}} \subseteq V$. The building layout and initial entity placements are known before an episode. Fire, smoke, and portal conditions evolve during execution and determine the currently executable graph $G_t = (V, E_t)$. A failed Space disables its incident connections; a failed portal disables connections through it. Thus target reachability and usable routes can change while the entities and Space identities remain fixed.

The simulator state at decision event t is

$$s_t = (G_t, H_t, X_t^{\mathcal{R}}, X_t^{\mathcal{P}}, m_t), \tag{21}$$

where H_t contains hazards, $X_t^{\mathcal{R}}$ and $X_t^{\mathcal{P}}$ contain rescuer and victim states, and m_t contains mission and service progress. The central actor and critic receive the global feature observation $o_t = \Omega(s_t)$: graph connectivity, Space hazards and victim counts, all rescuer states, mission progress, and legal-action masks. PACF also uses the preceding observation to estimate observed hazard changes. A ready rescuer may wait, pick up victims in its current room, hand off carried victims at a safe Space, or choose a reachable victim room or safe Space as a movement target. Busy rescuers continue their current actions. Multiple rescuers may choose the same target.

Movement executes one Space transition at a time. The rescuer remains committed to its target, while the planner recomputes feasible continuations at decision boundaries, including after blocked transitions. Victims count as rescued after evacuation from a safe-area queue. All reported methods use the same execution and outcome rules.

B.2 DATASET COMPOSITION AND SPLITS

The dataset has 120 tasks on 30 procedurally generated six-floor buildings: ten buildings per difficulty level and four initial hazard-placement tasks per building. Table 5 lists the main instance sizes. Each task has three rescuers; Medium, Hard, and Expert have 12, 18, and 24 victims, respectively. Within a building, the four tasks share its graph, rescuer starts, victim placements, and safe Spaces, but differ in the placement of initial fires. The four recipes are cross-floor (CF), victim-floor (VP), bottleneck (BP), and stair (SP) pressure; they emphasize different fire locations without imposing disjoint geometric conditions.

Table 5: DMRIE dataset composition. Each six-floor building generates four tasks with three rescuers. Space ranges are measured from the frozen graphs.

Instance property	**Medium**	**Hard**	**Expert**
Executable Spaces per building	96	208–214	315–328
Victims per task	12	18	24
Occupied victim rooms	6	9	12
Initial fire sources	1	4	5
Safe Spaces	2	2	1
Buildings in Train / Validation / Test	4 / 2 / 4	4 / 2 / 4	4 / 2 / 4

Buildings are disjoint across Train, Validation, and Test; all four tasks from a building remain in one split. This gives 48, 24, and 48 tasks, respectively. The task definitions fix graph topology, placements, and initial hazards for evaluation. Some recipe labels share an identical initial configuration; they remain separate entries in the predefined task list and are not independent building samples.

Test task layouts and graph representation. Figure 2 displays all 48 Test tasks, grouped by difficulty. Each difficulty panel contains the four held-out buildings and their four pressure tasks. Every cell shows the complete six-floor initial layout. The panels use the frozen task definitions, including their shared placements and repeated initial configurations, without selecting examples by policy outcomes. The short building identifiers match Table 13. Figure 3 separately illustrates how a movement target is represented at the dispatch and execution levels.

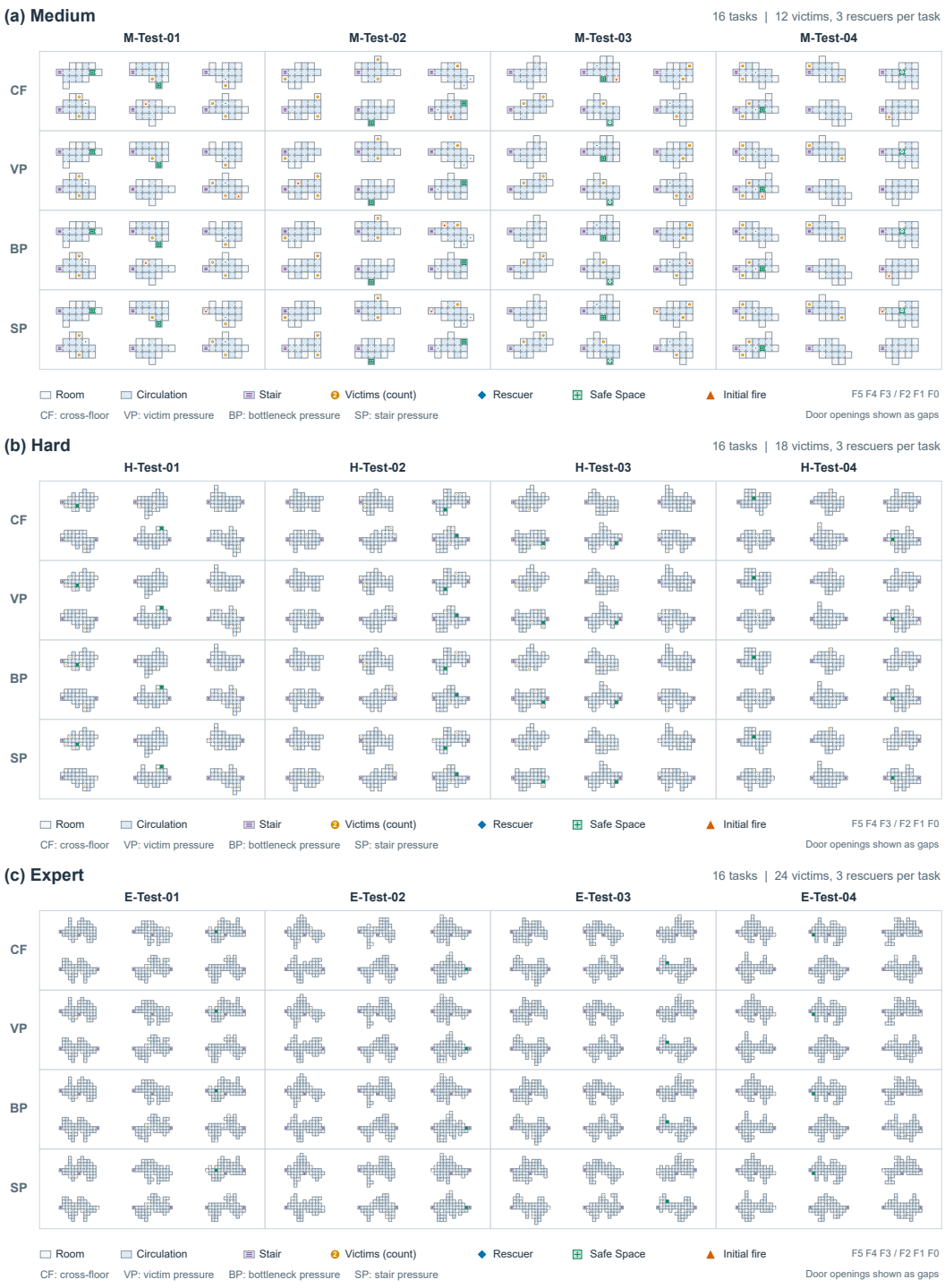

Figure 2: All 48 Test tasks in the controlled DMRIE benchmark. The three panels show Medium, Hard, and Expert; within each panel, columns are the four Test buildings and rows are cross-floor (CF), victim (VP), bottleneck (BP), and stair (SP) pressure. Every task cell contains all six floors, ordered by elevation as indicated in the panel. Room and circulation footprints, stair Spaces, safe Spaces, initial fire sources, victim counts, and rescuer starts are rendered from the frozen task definitions. The panels depict initial conditions, not policy trajectories or outcomes. Symbols and door widths are schematic; each difficulty panel uses a consistent geometric scale.

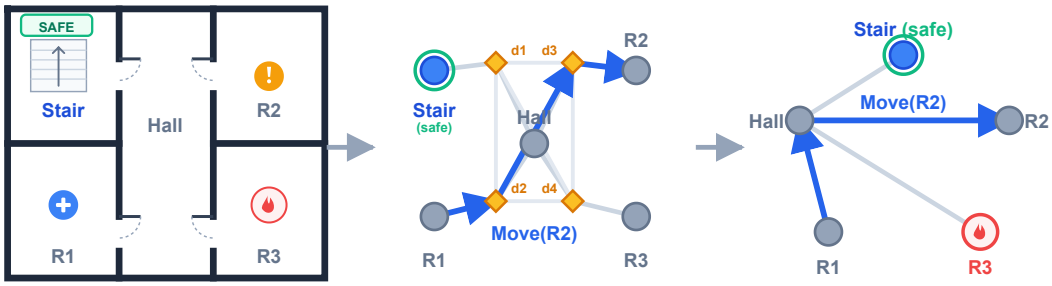

Figure 3: Schematic two-level building representation. (a) A physical layout contains rooms R1–R3, a corridor (Hall), and a stair Space marked as safe, with a rescuer in R1, a victim in R2, and fire in R3. (b) The execution-level walk graph expands the layout into Space anchors and door nodes; the highlighted movement to R2 follows R1 → d2 → d3 → R2. (c) The Space graph records shared-door adjacency, representing this movement as R1 → Hall → R2. Gray edges show connectivity and blue arrows show the illustrative movement. This diagram explains the representation and is not a subregion of the Test layouts in Figure 2.

C Implementation Details

Algorithm 1 summarizes the common decision and update sequence.

Algorithm 1 PACF: route-informed decisions and joint learning

Require: Training tasks, deterministic route planner, backbone φ, PACF θ, critic V
Ensure: Trained agent policies with planner-aware feedback
Initialize the models; set the feedback scorer's output to zero
for each training iteration **do**
  Initialize rollout buffer $\mathcal{B}$
  while the rollout collection budget is not exhausted **do**
    Start or continue an episode; obtain observation o_t, masks, and temporal inputs
    Encode o_t into shared Space, agent, and mission embeddings
    Generate $\{\mathcal{R}_{ij,t}, b_{ijk,t}\}$ for legal agent–target pairs
    Construct route representations $z_{ijk,t}$ using Equation 22
    Compute candidate scores $q_{ijk,t}$ and pair feedback $C_{ij,t}$
    for each unit i (parallel for MAPDP; sequential for MAPT) **do**
      Form context $x_{i,t}$; compute masked action-type probabilities and base target logits
      Sample action type $a_{i,t}$ ▷ Non-ready slots have a forced placeholder
      if $a_{i,t} =$ MOVE **then**
        Sample target $j_{i,t}$ using the feedback-corrected Equation 6
        Sample route $k_{i,t} \sim p_0(\cdot \mid i, j_{i,t}, x_{i,t})$
        Set $\widetilde{u}_{i,t} = (\text{MOVE}, j_{i,t}, k_{i,t})$
      else
        Set $\widetilde{u}_{i,t} = a_{i,t}$
      end if
      For MAPT, append this action type and target to the dispatch prefix
    end for
    Issue ready units' commands; each selected route supplies its next Space transition
    Advance to the next decision or episode boundary; record $\bar{r}_t$, d_t, and next observation
    Refresh the executable graph and commitments for the next decision
    Store policy inputs, masks, candidate tables, and actions in $\mathcal{B}$
    Store their joint log-probability, transition data, and boundary flags in $\mathcal{B}$
  end while
  Compute team advantages and returns using $\bar{r}_t$, d_t, and critic V
  Replay stored inputs and actions, recomputing embeddings and feedback
  Update φ, θ, and V with the CTCE PPO objective of Section 2.2
end for

C.1 Details of PACF in MAPT

MAPT reads the global Space, rescuer, and mission features, encodes the current graph, and autoregressively chooses an action for each ready rescuer. Each movement action has an action-type probability, a target probability corrected by PACF, and a route probability fixed by planner cost. The observation, planner, route encoder, and conditional target rule below are also used by MAPDP; the difference is how the backbone produces target logits.

C.1.1 Observation features and route encoding

The shared numeric observation contains seven Space features (fire level, smoke level, combined risk, passability, safe-Space indicator, waiting-victim count, and occupancy), four rescuer features (idle, walking, rescuing, and carrying count), and six global features (victim total and the fractions waiting, being rescued, evacuating, evacuated, and incapacitated). Directed edge attributes are length and a vertical-edge indicator. Legal-action masks, rescuer locations, active targets, and decision-readiness masks are supplied separately. The policies use no fitted feature normalization. The temporal and candidate-relative scalings below still apply.

Each backbone uses hidden width 64, two encoder blocks, four attention heads, and relation width 32. MAPT uses two decoder blocks with autoregressive decoding and relation inputs. The two backbones share PACF's route-encoding and replay interface, not their complete encoder and decoder architecture.

For candidate r_{ijk}, the route encoder produces

$$z_{ijk} = \mathrm{RouteEnc}_\theta\left(r_{ijk}, \mathcal{R}_{ij} \mid h^{\varphi}_{ij}, \eta_{ijk}\right), \tag{22}$$

where h^{φ}_{ij} contains shared backbone embeddings of the rescuer, target, route Spaces, and mission. The temporal inputs η_{ijk} use current and preceding hazard observations and nominal travel timing. The construction below specifies these inputs and their aggregation.

For a Space v, the six-dimensional temporal vector is

$$\zeta_{v,t} = [F_{v,t}, S_{v,t}, \dot{F}_{v,t}, \dot{S}_{v,t}, m^{\mathrm{hist}}_t, \Delta^{\mathrm{obs}}_t], \tag{23}$$

where F and S are fire and smoke levels clipped to $[0,1]$. If the preceding observation occurred $\Delta t > 0$ seconds earlier, then

$$\dot{F}_{v,t} = \mathrm{clip}\left(\frac{60(F_{v,t} - F_{v,t-1})}{\Delta t}, -1, 1\right), \qquad \Delta^{\mathrm{obs}}_t = \min(\Delta t/60, 10), \tag{24}$$

and $\dot{S}$ is defined analogously. The history-valid indicator m^{hist}_t is one in this case. At initialization, trends, m^{hist}_t, and Δ^{obs}_t are zero. History is maintained per episode and reset at episode boundaries. These finite differences give the policy a memory of past observations; they are not simulator forecasts.

For the ℓ-th edge $e = (u, v)$ of a route with L_{ijk} edges, the 15-dimensional temporal input is

$$\eta_{ijke} = [\zeta_{u,t}; \zeta_{v,t}; \ell/L_{ijk}; (\sum_{m<\ell} d_{e_m})/60; d_e/60]. \tag{25}$$

The arrival and traversal times are nominal geometric quantities, rather than load- or congestion-corrected forecasts. For edge $e = (u, v)$, the input x_{ijke} combines shared backbone embeddings of the endpoints, agent i, target j, and mission with edge geometry. A learned encoder produces

$$z_{ijke} = x_{ijke} + T_\theta(\eta_{ijke}) + E_\theta([x_{ijke}; \eta_{ijke}]). \tag{26}$$

With hidden width 64 and two edge attributes, x_{ijke} and z_{ijke} have dimension 322. The temporal encoder T_θ has widths $15 \to 64 \to 322$, and the edge-token encoder E_θ has widths $337 \to 64 \to 322$; each uses GELU after its first linear layer. Route mean and maximum pooling operate over all valid edges.

The route encoder in Equation 22 retains the first edge explicitly and pools the complete route. Let z^{first}_{ijk}, $\bar{z}_{ijk}$, and $z^{\max}_{ijk}$ denote the first-edge, mean, and coordinatewise maximum tokens. Its candidate representation is

$$z_{ijk} = [z^{\mathrm{first}}_{ijk}; \bar{z}_{ijk}] + W_{\max} z^{\max}_{ijk} + W_d \frac{d_{ijk}}{60} + W_{\mathrm{rel}} \psi_{ijk}, \tag{27}$$

where $d_{ijk} = \sum_{e \in r_{ijk}} d_e$ is the nominal route duration and ψ_{ijk} describes the candidate relative to its alternatives, including cost, duration, and route overlap. In Equation 27, $W_{\max}$ maps $322 \to 644$, W_d maps $1 \to 644$, and W_{rel} maps $8 \to 644$; these projections include biases.

For one rescuer–target pair, let b_k and d_k be the candidate base cost and nominal duration. Let μ_b, μ_d and σ_b, σ_d be their means and population standard deviations, with each standard deviation lower-bounded by 10^{-3}. The eight inputs are

$$\begin{aligned}\psi_k = \big[&(b_k - b_{\min})/\sigma_b, (b_k - \mu_b)/\sigma_b, \operatorname{rank}(b_k)/\max(K-1,1),\\ &(d_k - d_{\min})/\sigma_d, (d_k - \mu_d)/\sigma_d, \mathbb{1}\{e_k^{\text{first}} = e_*^{\text{first}}\}, J(E_k, E_*), 1/K\big],\end{aligned} \tag{28}$$

where $*$ denotes the minimum-base-cost candidate, E_k and E_* are the directed edge sets of candidate k and candidate $*$, respectively, and $J(A,B) = |A \cap B|/|A \cup B|$. Ranks are zero-based, with candidate order breaking ties. This overlap feature describes alternatives; it is distinct from candidate filtering. The scalar scorer f_θ uses hidden widths 128 and 64 with GELU activations. Its output layer and the added feature projections are zero-initialized. Appendix C.3 specifies the stored output-head dimensions for checkpoint reproduction.

C.1.2 Candidate generation

Planning uses a fixed cost based on nominal duration and current risk:

$$c_e = \frac{d_e}{D}(1 + \beta \bar{h}_e), \qquad b_{ijk} = \sum_{e \in r_{ijk}} c_e, \qquad \beta = 2, \tag{29}$$

where d_e is the nominal duration of edge e, $D = \max_{e \in E} d_e$, and $\bar{h}_e$ is the mean current risk at its endpoints. The candidate set satisfies

$$\mathcal{R}_{ij} = \{r_{ijk}\}_{k=1}^{K_{ij}} \subseteq \mathcal{F}_{ij}, \qquad 1 \le K_{ij} \le 3 \quad \text{for valid pairs.} \tag{30}$$

The route graph uses directed Space connections and removes connections incident to blocked Spaces or blocked intermediate portals. For every distinct legal first Space, the planner searches for a minimum-fixed-cost continuation to the target that avoids revisiting the source. It ranks the resulting paths deterministically by base cost, path length, and edge identity, considers at most six raw alternatives, and rejects a candidate if its nominal duration exceeds 1.5 times the shortest nominal duration among those raw alternatives. At most three survive; a source with fewer feasible first transitions necessarily produces fewer candidates. Where the execution-expansion callback is used, expansion must also validate the path.

Candidate diversity is enforced by first-transition identity. Overlap is used as a representation feature, not as a candidate filter. Complete Space paths identify the executable candidates, and packed edge indices support neural computation and replay. Given the current graph and observation, candidate generation is deterministic. Training may sample an executed candidate from the fixed base-cost distribution in Equation 7; greedy evaluation takes its highest-probability candidate. This route choice uses planner base costs, not the learned Pair soft-min weights.

C.1.3 Conditional factorization and invariance

Proposition A.1 establishes the bounds and derivatives of Equation 9; Equation 17 gives the gradient from target probabilities to candidate scores. These aggregation weights use learned q_{ijk}, while the fixed route policy uses planner costs b_{ijk}.

For a fixed decoder context x_i, let a_i denote action type. Here context means conditioning inputs, including observation and temporal features, legal masks, and candidate tables; learned embeddings are recomputed under the evaluated parameters. For either backbone, the agent-level factor π_ω^i in Equation 3 expands for a movement action as follows, with the agent and parameter superscripts suppressed:

$$\pi(\text{MOVE}, j, k \mid x_i) = \pi_\varphi^{\text{type}}(\text{MOVE} \mid x_i) p_{\varphi,\theta}(j \mid \text{MOVE}, x_i) p_0(k \mid i, j, x_i). \tag{31}$$

Non-movement actions use only their legal action-type probability. PACF changes the second factor, using Equation 6. For any scalar a, replacing all valid C_{ij} by $C_{ij} + a$ leaves that factor unchanged

because $\exp(-a/\tau_A)$ cancels in the normalization. The first factor is also unchanged because its logits do not consume C_{ij}. The action-type head remains jointly trained. This invariance is an intervention at fixed parameters and decoder context; different target selections can change subsequent states and, in MAPT, the autoregressive prefix of later rescuers. Independently trained policies can therefore have different action-type behavior and action frequencies.

Equation 3 combines these individual action probabilities into the joint policy; we suppress the event index t here. Each decoder context x_i includes the shared global information and the applicable legal mask m_i. MAPDP's parallel pointer scores depend on the shared representation of all rescuers and Spaces. MAPT additionally conditions on preceding action types and targets; its prefix embeddings exclude route indices. Non-ready rescuers have a singleton placeholder action with probability one and continue their ongoing execution. Both dispatchers use the same hard target commitment and event-level execution rules, with no network-side continuation/replanning gate or exclusive non-safe-target reservation. Multiple rescuers may choose the same target when permitted by the common environment mask.

PPO recomputes the type probability of each stored action and, for movement, its conditional target and fixed route probabilities. Replay fixes the recorded inputs and realized action prefixes; the denominator uses the recorded old joint log-probability. A singleton legal distribution contributes zero log-probability. The fixed route factor remains in the executed-action likelihood but cancels from the old/new ratio when replay uses the same candidate table. No target log-partition correction is added to the MOVE type logit.

C.2 Details of PACF in MAPDP

MAPDP uses the observation fields, candidate planner, route representation, Pair aggregation, and fixed route distribution specified in Appendix C.1. Its backbone produces target scores for rescuers in parallel from a shared team representation. PACF computes C_{ij} for the current observation before these decisions and corrects each rescuer's masked target logits using the corresponding row of the feedback matrix. Thus every ready rescuer retains an individual action distribution, but all distributions use the same globally observed route table. The action-type probabilities do not consume C_{ij}, and a non-ready rescuer contributes a probability-one placeholder.

MAPDP Base and +PACF share the same backbone architecture, candidate and execution planners, legality masks, hard target commitment, and training schedule. Base sets $C_{ij} = 0$; +PACF learns it through the joint PPO ratio in Equation 4. Unlike MAPT's autoregressive decoder, MAPDP does not condition one rescuer's current target score on preceding rescuers' sampled dispatch actions. Both use the same factorization in Equation 31 and the same stored-candidate replay rule.

C.3 Training Costs and Configuration

Reward and decision events. The shared reward gives 0.10 once per pickup and -1 once per victim incapacitation. If the m-th victim completes evacuation with health ratio h_m among N initial victims, its reward is

$$r_m^{\text{evac}} = h_m + N\left[\left(\frac{m}{N}\right)^2 - \left(\frac{m-1}{N}\right)^2\right]. \tag{32}$$

Other proposed shaping terms have zero weight in the reported runs. The simulator advances in one-second steps, while dispatch decisions occur when rescuer activities complete. Event t occurs at simulation step τ_t; $d_t = \tau_{t+1} - \tau_t$ counts steps to the next decision or episode boundary. Rewards between events are accumulated as

$$\bar{r}_t = \sum_{\ell=0}^{d_t-1} \gamma^\ell r_{\tau_t+\ell}, \qquad J(\pi) = \mathbb{E}_\pi\left[\sum_t \gamma^{\tau_t}\bar{r}_t\right]. \tag{33}$$

This preserves the simulation-step objective in Equation 2; d_t is observed elapsed time, not a predicted route duration. The event-level accounting follows temporally extended action updates (Sutton et al., 1999).

PPO updates. With $V(o_t)$ denoting the denormalized global critic value, the temporal-difference residual is $\delta_t = \bar{r}_t + \gamma^{d_t} V(o_{t+1}) - V(o_t)$ and the advantage is $\widehat{A}_t = \delta_t + \gamma^{d_t}\lambda\widehat{A}_{t+1}$ within an episode.

We use $\gamma = 0.9991338667254804$ per simulation step and $\lambda = 0.95$ per decision; termination has zero bootstrap and time-limit truncation uses the final critic value. Advantage recursion stops at either episode boundary. The actor optimizes the joint PPO ratio in Equation 5 with clipping threshold 0.2, value coefficient 0.5, entropy coefficient 0.01, learning rate 5×10^{-5}, gradient-norm cap 0.5, two PPO passes, and minibatches of 64. The critic uses ValueNorm-normalized return targets. Replay retains the collected legal masks, route candidates, route choices, and temporal features. PACF receives task-return gradients through target probabilities; it is not trained with route-ranking labels.

Table 6: Training and checkpoint-selection budget for the four reported learning methods. Episode counts include both termination and truncation.

Setting	Value
Training runs per method	3
Epochs per run	64
Task entries per epoch	32
Rollout workers per entry	4
Collected episodes per epoch	128
Collected episodes per run	8,192
Episode horizon (simulation steps)	1,200
PPO passes per update	2
Minibatch size	64
Validation epochs	8, 16, 32, 48, 64

Budget and hardware. Each of the three seeds trains for 64 epochs, with 32 task entries scheduled from the 48 Train tasks and four episodes per entry per epoch, for 8,192 episodes per run. The same schedule and budget apply to both matched backbone pairs. A run uses a single NVIDIA L40 GPU on a machine with a 32-core CPU.

Stored network head. The route scorer stores a $644 \to 128 \to 64 \to 2$ head. Its first output is the scalar f_θ used by PACF; the second output does not affect action probabilities. The two-output checkpoint shape is retained for reproduction. The remaining encoder dimensions and input channels are given in Appendix C.1.

D Additional Results

D.1 Evaluation protocol and result tables

Each learned condition has three independent runs. We check greedy Validation rescue rate at epochs 8, 16, 32, 48, and 64, breaking ties by fewer incapacitations and then earlier epoch. Selection uses environment seeds 410001, 410002, and 410003 for runs 1–3, incremented by task index. The selected checkpoints are evaluated on all 48 Test tasks with seed $520001 + i$ for zero-based task index i; the MAPT fixed-checkpoint split audit uses this schedule on each split. Test outcomes were inspected during development.

Nearest, Hu, SA, and GA issue target commands executed on shortest-nominal-duration routes. SA and GA each use 60 objective evaluations per decision. Within each learned backbone, Base and PACF share candidate planning, route selection, execution rules, reward, and training budget; Base sets $C_{ij} = 0$.

Rescue rate pools evacuated victims within each run and reports the mean and sample SD across three runs. Heuristics use one fixed-seed evaluation. A full success evacuates every victim in a task; unresolved victims are neither evacuated nor recorded as incapacitated. The three runs and the four tasks from one building are not independent building samples.

Table 7: Effect of PACF on each dispatch backbone (Test rescue rate in %, mean ± sample SD across three training seeds). Each pair shares the candidate planner, fixed route-selection rule, execution constraints, training budget, and checkpoint-selection protocol. Bold marks the highest mean within each backbone and column, including ties.

Backbone	PACF	Medium	Hard	Expert	Overall
MAPDP	w/o	**97.22** ± 3.05	97.45 ± 0.80	85.07 ± 1.97	91.90 ± 1.04
	w/	95.66 ± 3.05	**97.57** ± 0.92	**87.59** ± 2.62	**92.71** ± 0.23
MAPT	w/o	92.71 ± 1.38	95.02 ± 1.40	86.02 ± 2.57	90.51 ± 1.22
	w/	**95.66** ± 0.30	**97.57** ± 1.20	**89.32** ± 4.54	**93.48** ± 2.48

Table 8: Additional Test metrics for the learned methods (mean ± SD over three seeds). Casualties and unresolved victims are split totals; full success is a task percentage. Steps and decisions are episode means, not standalone quality measures. Bold compares outcome means within each backbone pair.

Method	Casualties ↓	Unresolved ↓	Full success (%) ↑	Simulation steps	Joint decisions
MAPDP Base	58.0 ± 4.4	12.0 ± 10.4	63.9 ± 5.2	354.2 ± 23.0	182.2 ± 7.0
MAPDP + PACF	**55.7** ± 0.6	**7.3** ± 1.5	63.9 ± 4.8	344.7 ± 11.3	188.7 ± 10.2
MAPT Base	65.7 ± 2.5	16.3 ± 8.4	55.6 ± 4.3	442.6 ± 15.8	251.8 ± 19.3
MAPT + PACF	**48.7** ± 15.9	**7.7** ± 5.5	**63.9** ± 1.2	344.5 ± 2.2	191.7 ± 3.5

D.2 Selected checkpoints and fixed-split evaluations

Table 9: Rescue rate (%) for each training seed. Validation is the score used to select the listed epoch during training; Test is the subsequent fixed-checkpoint evaluation. These Validation scores use the selection schedule, not the separate fixed-split audit in Table 10.

Method	Seed	Selected epoch	Selection Validation	Test
MAPDP Base	1	48	95.14	91.90
	2	48	96.99	90.86
	3	48	95.14	92.94
MAPDP + PACF	1	32	94.68	92.71
	2	48	96.53	92.48
	3	16	95.37	92.94
MAPT Base	1	16	92.82	90.97
	2	16	92.13	89.12
	3	8	91.90	91.44
MAPT + PACF	1	48	96.53	90.62
	2	64	96.06	95.14
	3	48	96.76	94.68

Table 9 lists every selected epoch, its checkpoint-selection Validation score, and the formal Test outcome for the four reported learning conditions. Selection follows the same rule and budget for both matched backbone pairs. Each condition retains all three training seeds. All Test evaluations use the same 48 tasks and environment seed $520001 + i$ for task index i in the ordered task list. The training-seed labels identify independently trained runs, not these common evaluation seeds.

Table 10 gives the additional fixed-checkpoint Train/Validation/Test evaluations for the MAPT matched pair. Each run holds its own selected checkpoint fixed across splits, using the common task-indexed evaluation seed schedule. These are distinct from the Validation evaluations used to select the checkpoints. The fixed-split audit covers only the MAPT pair; epoch-64 stochastic training samples are not evaluations of a fixed Train set and are not included in this table.

Table 10: MAPT rescue rate across splits (%, mean ± sample SD over three runs). Each run uses the same Validation-selected checkpoint on all splits. These are evaluations on different task sets, not a training trajectory. Bold marks the highest mean within each split and column, including ties. This fixed-checkpoint audit is available for the MAPT pair; it is distinct from selection-time Validation scores.

Method	Medium	Hard	Expert	Overall
Train				
MAPT Base	**96.35** ± 0.00	91.67 ± 1.25	73.70 ± 2.23	84.72 ± 0.92
MAPT + PACF	**96.35** ± 1.56	**98.73** ± 1.00	**81.94** ± 4.93	**90.74** ± 1.71
Validation				
MAPT Base	98.61 ± 1.20	97.45 ± 2.23	85.24 ± 1.83	92.28 ± 0.48
MAPT + PACF	**100.00** ± 0.00	**100.00** ± 0.00	**92.01** ± 0.80	**96.45** ± 0.35
Test				
MAPT Base	92.71 ± 1.38	95.02 ± 1.40	86.02 ± 2.57	90.51 ± 1.22
MAPT + PACF	**95.66** ± 0.30	**97.57** ± 1.20	**89.32** ± 4.54	**93.48** ± 2.48

Table 11: MAPT fixed-checkpoint Train and Validation audit by building (rescue rate in %, mean ± sample SD over three runs). Each row pools four recipe entries; these scores are distinct from selection-time Validation.

Building	MAPT Base	MAPT + PACF
M-Train-01	85.42 ± 0.00	85.42 ± 6.25
M-Train-02	100.00 ± 0.00	100.00 ± 0.00
M-Train-03	100.00 ± 0.00	100.00 ± 0.00
M-Train-04	100.00 ± 0.00	100.00 ± 0.00
H-Train-01	84.26 ± 1.60	**100.00** ± 0.00
H-Train-02	100.00 ± 0.00	100.00 ± 0.00
H-Train-03	93.52 ± 2.12	**98.15** ± 0.80
H-Train-04	88.89 ± 4.81	**96.76** ± 4.46
E-Train-01	54.17 ± 2.76	**67.01** ± 10.01
E-Train-02	85.42 ± 1.04	**92.36** ± 0.60
E-Train-03	81.94 ± 0.60	**93.40** ± 4.34
E-Train-04	73.26 ± 6.01	**75.00** ± 5.41
M-Val-01	97.22 ± 2.41	**100.00** ± 0.00
M-Val-02	100.00 ± 0.00	100.00 ± 0.00
H-Val-01	94.91 ± 4.46	**100.00** ± 0.00
H-Val-02	100.00 ± 0.00	100.00 ± 0.00
E-Val-01	94.10 ± 4.70	**95.83** ± 2.76
E-Val-02	76.39 ± 8.36	**88.19** ± 1.59

The MAPDP Test differences for seeds 1–3 are $+0.81$, $+1.62$, and 0.00 pp, corresponding to $+7$, $+14$, and 0 rescues. MAPT differences are -0.35, $+6.02$, and $+3.24$ pp, or -3, $+52$, and $+28$ rescues. Figure 4 shows the individual paired MAPT differences across splits.

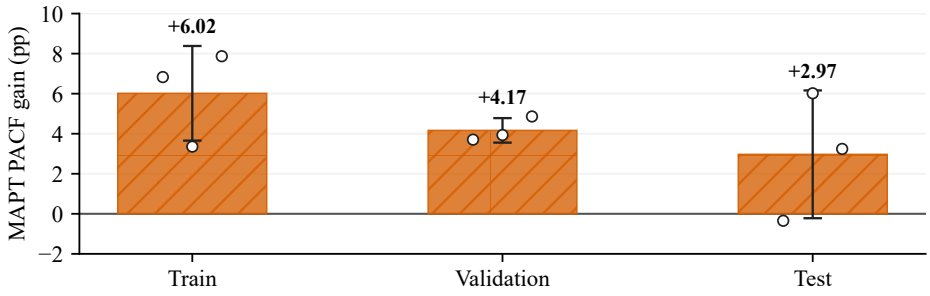

Figure 4: Fixed-checkpoint MAPT PACF-minus-Base rescue-rate differences across building-disjoint splits (pp). Bars show the paired three-seed mean, whiskers the sample SD, and dots individual paired differences. Checkpoints were selected on Validation; these are split evaluations, not a training curve.

The fixed MAPT checkpoints gain $+6.02$, $+4.17$, and $+2.97$ pp on Train, Validation, and Test, respectively. These splits contain different buildings, and Validation informed checkpoint selection; the differences do not directly measure learning speed or overfitting.

D.3 Building and difficulty–pressure results

Table 12: Test rescue rate (%) by difficulty and hazard placement; learned methods report mean ± SD over three random seeds.

Task	Nearest	Hu	SA	GA	MAPDP		MAPT	
					Base	+PACF	Base	+PACF
Expert								
BP	89.58	94.79	64.58	85.42	93.75 ± 1.04	93.40 ± 0.60	**95.49** ± 1.20	94.44 ± 1.59
CF	75.00	80.21	70.83	73.96	82.99 ± 2.62	88.89 ± 3.66	**89.24** ± 0.60	88.89 ± 7.96
SP	79.17	73.96	14.58	57.29	79.86 ± 0.60	80.56 ± 2.62	72.22 ± 9.34	**84.38** ± 5.21
VP	72.92	78.12	54.17	68.75	83.68 ± 4.92	87.50 ± 3.76	87.15 ± 2.17	**89.58** ± 5.21
Hard								
BP	88.89	88.89	94.44	94.44	94.91 ± 2.89	**97.22** ± 2.41	94.44 ± 0.00	**97.22** ± 0.00
CF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00** ± 0.00	**100.00** ± 0.00	99.54 ± 0.80	**100.00** ± 0.00
SP	95.83	93.06	87.50	59.72	96.76 ± 4.46	95.83 ± 4.17	92.59 ± 5.61	**97.22** ± 4.81
VP	**100.00**	**100.00**	**100.00**	86.11	98.15 ± 1.60	97.22 ± 3.67	93.52 ± 0.80	95.83 ± 0.00
Medium								
BP	**100.00**	**100.00**	**100.00**	**100.00**	98.61 ± 1.20	98.61 ± 1.20	98.61 ± 1.20	97.92 ± 0.00
CF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00** ± 0.00	**100.00** ± 0.00	**100.00** ± 0.00	**100.00** ± 0.00
SP	**100.00**	**100.00**	93.75	**100.00**	90.28 ± 11.47	84.03 ± 11.47	72.22 ± 6.70	84.72 ± 1.20
VP	**100.00**	**100.00**	**100.00**	**100.00**	**100.00** ± 0.00	**100.00** ± 0.00	**100.00** ± 0.00	**100.00** ± 0.00

Table 13: Rescue rate for each Test building (%, mean ± sample SD over three runs), pooling its four hazard-placement recipes. Bold marks the higher mean within each matched dispatcher pair; ties remain unbolded. Identifiers encode difficulty, split, and a within-group index.

Building	MAPDP Base	MAPDP + PACF	MAPT Base	MAPT + PACF
M-Test-01	95.83 ± 5.51	**96.53** ± 6.01	86.81 ± 6.70	**90.97** ± 1.20
M-Test-02	**95.14** ± 5.24	87.50 ± 4.17	**97.22** ± 1.20	95.83 ± 0.00
M-Test-03	97.92 ± 2.08	**98.61** ± 2.41	90.28 ± 4.81	**95.83** ± 0.00
M-Test-04	100.00 ± 0.00	100.00 ± 0.00	96.53 ± 6.01	**100.00** ± 0.00
H-Test-01	95.83 ± 3.67	**98.15** ± 3.21	93.98 ± 0.80	**100.00** ± 0.00
H-Test-02	94.91 ± 2.89	**95.37** ± 4.46	**94.91** ± 4.01	94.44 ± 4.81
H-Test-03	**100.00** ± 0.00	96.76 ± 1.60	92.13 ± 0.80	**95.83** ± 0.00
H-Test-04	99.07 ± 1.60	**100.00** ± 0.00	99.07 ± 1.60	**100.00** ± 0.00
E-Test-01	**96.53** ± 1.20	96.18 ± 0.60	95.83 ± 2.76	**97.92** ± 2.76
E-Test-02	**94.79** ± 3.61	94.10 ± 1.20	91.32 ± 1.20	**95.49** ± 0.60
E-Test-03	67.71 ± 11.60	**75.69** ± 4.70	70.49 ± 2.17	**79.17** ± 13.62
E-Test-04	81.25 ± 3.61	**84.38** ± 6.51	**86.46** ± 8.14	84.72 ± 6.28

Table 14: Test outcomes by hazard placement, pooled over three training seeds (36 episodes and 648 victims per method–recipe row).

Method	Recipe	Rescued	Casualties	Unresolved	Full success
MAPDP Base	BP	617/648	31	0	22/36
	CF	599/648	41	8	28/36
	SP	569/648	59	20	17/36
	VP	597/648	43	8	25/36
MAPDP + PACF	BP	621/648	27	0	21/36
	CF	616/648	29	3	29/36
	SP	560/648	72	16	18/36
	VP	606/648	39	3	24/36
MAPT Base	BP	621/648	27	0	19/36
	CF	616/648	22	10	29/36
	SP	512/648	107	29	11/36
	VP	597/648	41	10	21/36
MAPT + PACF	BP	623/648	25	0	21/36
	CF	616/648	29	3	29/36
	SP	575/648	56	17	17/36
	VP	609/648	36	3	25/36

Table 12 expands Table 1 into Expert, Hard, and Medium blocks for the same eight methods. Each row pools four Test buildings per run under one hazard-placement configuration, as defined in Appendix B. Table 13 instead aggregates the four pressure tasks separately for every building. Table 14 reports the complete pooled outcomes by hazard placement; Table 3 in the main text summarizes the PACF-minus-Base rescue-count differences. Under CF, MAPT rescue counts are identical with and without PACF, while casualties change from 22 to 29 and unresolved victims from 10 to 3; rescue rate alone does not capture this distinction.

D.4 Zero-shot MAPT team-size sensitivity

This evaluation fixes a 12-task Test subset before varying the team size: within each difficulty, four distinct tasks cover CF, VP, BP, and SP. The three selected training checkpoints for each MAPT condition, task configurations, and corresponding environment seeds are fixed. The two-rescuer variants retain two original starts; four-rescuer variants add one predetermined legal start. Victim counts, occupied rooms, per-room victim loads, initial fires, and safe Spaces do not change. For each method and team size there are $12 \times 3 = 36$ episodes and 648 victim instances. The three-rescuer records are reused from the complete Test evaluations; the two- and four-rescuer settings contribute 144 new episodes without retraining.

Table 15: MAPT zero-shot team-size sensitivity on the fixed 12-task subset. Each row pools 36 episodes from three fixed training seeds; all rows contain 648 victim instances. Rescue counts require completion of safe-Space evacuation.

Method	Rescuers	Evacuated	Rate (%)	Casualties	Unresolved	Full success
MAPT + PACF	2	625/648	96.45	19	4	26/36
MAPT + PACF	3	635/648	97.99	13	0	31/36
MAPT + PACF	4	639/648	98.61	9	0	34/36
MAPT Base	2	628/648	96.91	19	1	26/36
MAPT Base	3	615/648	94.91	26	7	27/36
MAPT Base	4	618/648	95.37	30	0	28/36

The paired PACF-minus-control rescue-rate differences for training seeds 1–3 are $(+0.93, +0.46, -2.78)$ percentage points with two rescuers, $(+3.24, +6.02, 0.00)$ with three, and $(+3.24, +7.87, -1.39)$ with four. Their means are -0.46, $+3.09$, and $+3.24$ percentage points, respectively, with sample SDs of 2.02, 3.01, and 4.63 points. Thus PACF yields -3, $+20$, and $+21$ additional evacuated victims in the three team-size settings, respectively. This subset is closer to full rescue than the complete Test split and does not substitute for its overall comparison. Rescue service is room-level: one action collects the eligible victims in a room, while escort size and health affect movement speed; outcome rates count people only after safe-queue evacuation. This team-size evaluation covers MAPT; no MAPDP team-size result is reported. These evaluations were run with competing workloads and provide no comparable wall-clock timing result.

D.5 Epoch-indexed Validation evaluations

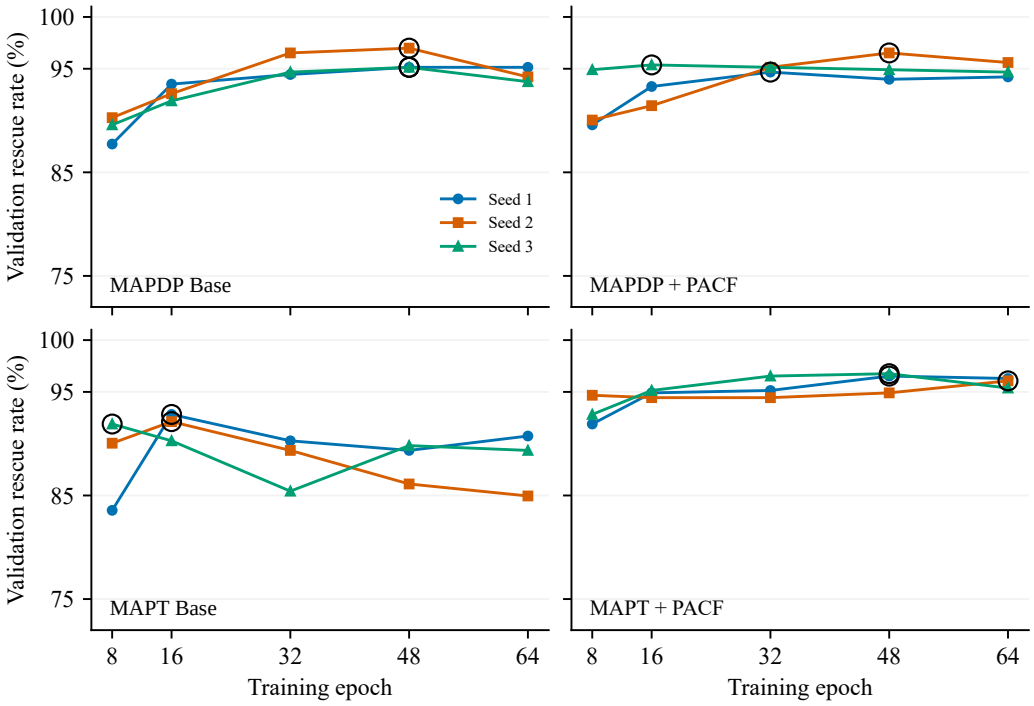

Figure 5: Greedy Validation evaluations during training for the four reported methods, each with three training seeds. Points occur at epochs 8, 16, 32, 48, and 64; connecting segments link the recorded evaluations. Black rings mark the selected checkpoints. Checkpoint-selection and fixed-checkpoint evaluations use different environment seed schedules (Appendix D.1), so their Validation scores can differ for the same checkpoint.