PREDICT WHAT JOINT ACTIONS CAN CONTROL: GOAL-PROGRESS PREDICTION FOR POST-TRAINING RESIDUAL SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predictive models can guide action selection, but their usefulness depends on whether the prediction target captures differences that candidate actions can actually control. In cooperative multi-agent reinforcement learning, small coordinated interventions may have little influence on long-horizon outcomes, making it difficult to identify beneficial joint actions. We introduce Goal-Progress Estimation (GPE), a goal-progress predictive framework for post-training residual selection. Rather than reconstructing future states or predicting return to episode completion, GPE estimates short-horizon, candidate-dependent goal progress and uses these predictions to select coordinated residual actions around a frozen multi-agent policy. A conservative decision rule incorporates ensemble uncertainty and calibration, retaining the native policy as an exact fallback when predicted improvements are insufficiently reliable. On VMAS Transport, GPE improves offline candidate ranking and achieves substantially higher return than both native IPPO and a capacity-matched Direct Completion predictor. Guarded and greedy GPE achieve return gains of 1.228 and 1.802 over native IPPO, respectively, compared with 0.653 for Direct Completion. Further analyses highlight the role of prediction-target choice and conservative action selection, revealing a trade-off between return improvement and preservation of native-policy successes. Overall, our findings demonstrate the utility of short-horizon, action-controllable prediction for residual selection in cooperative multi-agent control.

1 INTRODUCTION

Predictive models can support action selection, but accurate future prediction does not necessarily identify which action is preferable. This distinction is particularly relevant in cooperative multi-agent reinforcement learning (MARL), where small coordinated interventions may have effects that are difficult to distinguish from the shared evolution of the system. Recent work on counterfactual quotient models formalizes the value of predicting action-dependent differences rather than complete futures (Chen et al., 2026). We investigate a complementary question: which prediction target is useful for selecting bounded joint-action corrections around an already-trained cooperative policy? We consider post-training residual selection, where the underlying MARL policy is frozen and an additional controller evaluates corrective actions at execution time. Residual reinforcement learning and Residual-MPPI demonstrate related approaches to augmenting existing controllers without replacing their original behavior (Johannink et al., 2019; Wang et al., 2025). In cooperative control, however, improvements may require simultaneous corrections to several agents. A selector must therefore compare complete-team interventions while preserving the original policy as a fallback. This motivates predicting a task-defined quantity that is directly useful for ranking such interventions. We therefore estimate the short-horizon goal-progress improvement of each candidate joint residual relative to the native policy action, and use this prediction to select among bounded complete-team corrections at execution time.

We introduce *Goal-Progress Estimation* (GPE), a short-horizon action-effect predictor for post-training residual selection. GPE estimates the relative package-to-goal progress produced by each complete-team residual over the same $K = 4$ steps for which that residual is committed. Each candidate is compared against the unmodified IPPO action, focusing supervision on short-horizon

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.

differences attributable to the intervention rather than the full continuation outcome. A centralized scorer uses joint geometric features to rank the candidates. Its guarded decision rule accepts a residual only when an uncertainty-penalized, calibrated score is positive; otherwise, it executes the native policy action. We isolate prediction-target choice through a matched comparison with Direct Completion. Both methods use the same $Q = 33$ candidate bank, input features, model architecture, training budget, and ensemble construction. Direct Completion predicts return following a short intervention and subsequent native-policy continuation, whereas GPE predicts relative four-step goal progress. On VMAS (A Vectorized Multi-Agent Simulator for Collective Robot Learning) Transport (Bettini et al., 2022), GPE provides a stronger offline candidate-ranking signal and larger paired return gains than Direct Completion. Greedy selection obtains the largest return improvement, whereas guarded selection incurs fewer observed native-success losses. Terminal-aware and backbone analyses define the limits of this evidence: they are separate protocols and do not establish broader generalisation. Our contributions are threefold:

- We develop a post-training selector that predicts short-horizon, candidate-relative goal progress to rank bounded complete-team residuals around a frozen MARL policy.
- We isolate prediction-target choice through matched Direct Completion controls, offline action-ranking diagnostics, and paired closed-loop evaluation.
- We examine the return–native-success trade-off through greedy, guarded, and terminal-aware selection, and report exploratory failures that delimit the current empirical scope.

2 Related Work

Residual control and baseline fallback. Residual reinforcement learning augments an existing controller with corrective actions (Johannink et al., 2019), while Residual-MPPI performs execution-time planning around a pretrained continuous-control policy (Wang et al., 2025). Safe Policy Improvement with Baseline Bootstrapping (SPIBB) also motivates reverting to baseline behavior when estimates are insufficiently supported (Laroche et al., 2019). GPE similarly retains a frozen policy as its default controller, but ranks bounded, complete-team residuals using a learned progress predictor. Its calibrated fallback is an empirical decision heuristic and does not inherit SPIBB's theoretical guarantees.

Counterfactual action effects and coordination. COMA employs an agent-specific counterfactual baseline for multi-agent credit assignment (Foerster et al., 2018); Mesnard et al. study counterfactual credit assignment using future-conditioned value functions (Mesnard et al., 2021). Deep Coordination Graphs factor joint-action values into pairwise payoffs for coordinated optimization (Boehmer et al., 2020). More directly related to our target motivation, Counterfactual Quotient Models learn action-dependent future effects while removing components shared across candidate actions (Chen et al., 2026). GPE shares the emphasis on relative effects but studies a different setting: task-defined progress prediction for post-training complete-team residual selection around a frozen cooperative policy, rather than general counterfactual representation learning or policy-gradient credit assignment.

Predictive control and multi-agent world models. TD-MPC2 uses a learned latent world model for local trajectory optimization (Hansen et al., 2024). Model-based MARL approaches such as MAMBA use learned models and imagined rollouts to support cooperative policy learning (Egorov & Shpilman, 2022). Wang and Meger combine joint state-action embeddings with world-model-based MARL (Wang & Meger, 2026). GPE instead freezes the underlying policy and predicts a task-specific scalar effect for execution-time candidate selection, without learning a general transition model or using imagined trajectories for policy optimization.

Predictive representations, model-predictive control, and target choice. SALE learns state–action representations for reinforcement learning (Fujimoto et al., 2023), while TIMAR studies transition-informed representation learning in cooperative MARL (Feng et al., 2025). Predictive-control methods instead use learned future models directly for action selection. PETS combines probabilistic dynamics ensembles with trajectory sampling for model-based control (Chua et al., 2018), while LOOP performs short-horizon online planning using learned dynamics and a terminal value function (Sikchi et al., 2021). TD-MPC and TD-MPC2 use task-oriented predictive models

for local trajectory optimization (Hansen et al., 2022; 2024). In multi-agent settings, WONDER uses predictive models of candidate UAV-trajectory effects for coordinated coverage optimization (Huang et al., 2026). GPE is related to these approaches through candidate-conditioned prediction, but it addresses a narrower prediction problem. It does not learn a target encoder, a general-purpose latent representation, or a full transition model, and it does not optimize imagined action sequences. Instead, GPE directly predicts a task-defined scalar: the short-horizon goal-progress improvement of each residual candidate relative to the native policy action. In the Transport task, this goal-distance target is nearly equivalent to the corresponding four-step reward advantage. Consequently, our claim concerns the utility of choosing an action-controllable short-horizon prediction target for residual selection, relative to predicting return to episode completion, rather than an inherent advantage of latent representation learning.

3 GOAL-PROGRESS ESTIMATION SELECTOR

Post-training controller GPE is not a replacement for RL training. We first train a standard IPPO policy on VMAS Transport and then freeze that policy. During all GPE evaluations, the IPPO weights are fixed: there is no further policy-gradient update, no online fine-tuning, and no change to the native actor. The frozen IPPO policy remains the controller that proposes the default action at every environment step. GPE is added after this training stage as a short-horizon action selector. At a replanning step, IPPO produces the native four-agent action. The post-training controller then constructs a small bank of complete-team residual actions around that native action, scores those candidates with the learned GPE scorer, and either keeps the native IPPO action or executes one selected residual for four primitive steps. After those four steps, the process is repeated from the new live observation. Thus the question we test is not whether GPE can train a policy from scratch, but whether a learned goal-progress scorer can improve the decisions of an already-trained IPPO policy without modifying that policy.

Control setting and joint candidates The control task has four cooperative agents working together to move one package to one shared goal. Each agent sees an 11-dimensional observation and chooses a two-dimensional continuous action between -1 and 1. The native IPPO policy provides the baseline action for all four agents at every step. We write the combined observations and actions of the four agents as one joint observation and one joint action for scoring and candidate construction. This concatenation is a notation for the centralized scorer; it does not replace the decentralized agent-wise IPPO actors. In particular,

$$o_t = \mathrm{concat}\Big(o_t^{(1)}, o_t^{(2)}, o_t^{(3)}, o_t^{(4)}\Big) \in \mathbb{R}^{44}, \qquad b_t = \mathrm{concat}\Big(b_t^{(1)}, b_t^{(2)}, b_t^{(3)}, b_t^{(4)}\Big) \in [-1,1]^8, \quad (1)$$

where $b_t^{(i)} = \pi_i(o_t^{(i)}) \in [-1,1]^2$ is the native IPPO action of agent i. The scorer therefore evaluates complete-team actions, while the frozen IPPO actors continue to produce the live baseline actions. At a replanning step, the primary configuration creates 33 complete-team candidates. Each candidate changes the actions of all four agents together, rather than choosing an action for each agent independently. Candidate zero is the unchanged IPPO action and is always available as the native-policy fallback. The other 32 candidates add small, bounded joint residuals to the four-agent action. More explicitly, for candidate q we write

$$\delta_t^{(q)} = \mathrm{concat}\Big(\delta_t^{(q,1)}, \delta_t^{(q,2)}, \delta_t^{(q,3)}, \delta_t^{(q,4)}\Big) \in \mathbb{R}^8, \qquad \delta_t^{(q,i)} \in \mathbb{R}^2, \quad (2)$$

and its component actions are

$$a_t^{(q,i)} = \mathrm{clip}\Big(b_t^{(i)} + \delta_t^{(q,i)}, -1, 1\Big), \qquad a_t^{(q)} = \mathrm{concat}\Big(a_t^{(q,1)}, a_t^{(q,2)}, a_t^{(q,3)}, a_t^{(q,4)}\Big). \quad (3)$$

Here $\mathrm{clip}(x, -1, 1)$ clips each coordinate to the environment's action bounds; for $q = 0$, $\delta_t^{(0)} = 0$ and therefore $a_t^{(0)} = b_t$. Each nonzero residual is assigned a target root-mean-square size of 0.08 across the eight action coordinates, with each coordinate bounded by 0.20. Each nonzero candidate is constructed by a deterministic pseudo-random draw: a stateless generator is keyed by the task, environment seed, timestep, and candidate index, then samples an eight-dimensional joint residual direction from a standard normal distribution, corresponding to the four agents' two-dimensional action corrections. The sampled residual is rescaled to a target RMS magnitude of 0.08 and clipped

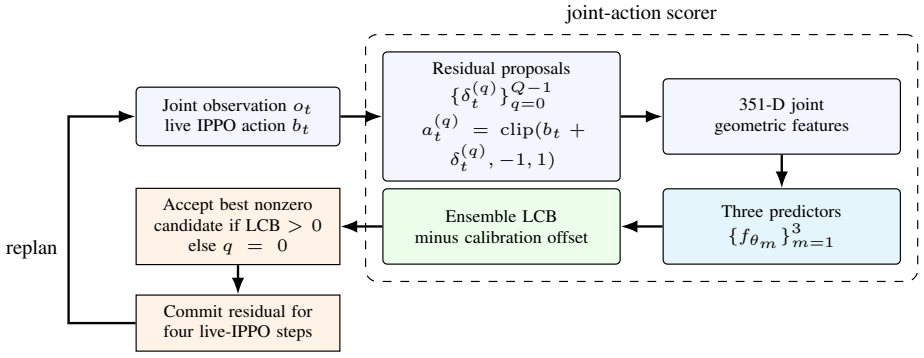

Figure 1: GPE is a receding-horizon residual selector around a fixed IPPO policy. The primary evaluation uses $Q = 33$ complete-team candidates; candidate zero is the unmodified live policy and remains in the bank as the exact fallback. The scorer is centralized; execution still uses each agent's live IPPO action plus the selected team residual.

coordinatewise to $[-0.20, 0.20]$. For a fixed task, environment seed, timestep, and candidate index, the residual direction is reproducible. Thus matched controller comparisons use the same residual directions at corresponding replanning times, while the resulting candidate actions remain centered on each controller's current native IPPO action. The primary $Q = 33$ candidate bank therefore consists of 32 bounded joint residual perturbations around the native IPPO action, together with candidate zero, which is the unmodified native action. Thus, at every replanning point, GPE chooses among the original IPPO action and 32 small coordinated team corrections around it. If candidate q is selected, the same residual is committed for $K = 4$ primitive steps:

$$a_{t+k} = \mathrm{clip}\Big(\pi(o_{t+k}) + \delta_t^{(q)}, -1, 1\Big), \qquad k \in \{0, \dots, K-1\}. \tag{4}$$

Thus IPPO still reacts to the live observation during the macro. Only the bounded team residual remains fixed.

A task-defined controllable future target Let $s_{t+K}^{(q)}$ denote the environment state reached by applying residual $\delta_t^{(q)}$ through Equation 4. Define the scalar goal-distance potential $z(s) = \left\|p^{\mathrm{pkg}}(s) - p^{\mathrm{goal}}(s)\right\|_2$, where $p^{\mathrm{pkg}}(s) \in \mathbb{R}^2$ and $p^{\mathrm{goal}}(s) \in \mathbb{R}^2$ are the package and goal positions, and $\|\cdot\|_2$ is Euclidean norm. This is a fixed, task-defined scalar potential rather than a learned latent representation. The GPE target is the candidate minus factual change in this future potential:

$$y_{tq}^{\mathrm{gpe}} = -c\left[z\Big(s_{t+K}^{(q)}\Big) - z\Big(s_{t+K}^{(0)}\Big)\right], \qquad c = 100, \quad y_{t0}^{\mathrm{gpe}} = 0. \tag{5}$$

Here c is the Transport dense-reward scale. Positive y_{tq}^{gpe} means that candidate q moves the package closer to the goal than IPPO does over the same four steps. The exact four-step reward advantage is $y_{tq}^{\mathrm{chunk}} = \sum_{k=0}^{K-1}\left(r_{t+k}^{(q)} - r_{t+k}^{(0)}\right)$, where $r_{t+k}^{(q)}$ is the shared team reward at offset k in branch q. The GPE target in Eq. (5) and the exact four-step reward advantage above differ by a mean absolute value of 9.45×10^{-7}. We train on y^{gpe} and calibrate selection against the exact y^{chunk}. The predictor maps a current context and candidate action to a compact future target rather than reconstructing the raw observation. There is no learned target encoder, momentum/EMA target network, or general latent-space dynamics model. Our claim is therefore about the usefulness of a task-defined, action-controllable goal-progress target for residual selection, not about a learned representation. The predictor maps the current joint context and candidate residual to the candidate's four-step change in this fixed goal-distance potential relative to candidate zero. The matched Direct target is

$$y_{tq}^{\mathrm{dir}} = G_t^{(q)} - G_t^{(0)}, \qquad G_t^{(q)} = \sum_{j=t}^{\tau_q - 1} r_j^{(q)}, \tag{6}$$

where $G_t^{(q)}$ is undiscounted return from branch time t to its terminal step τ_q. Branch q uses its residual for the first K steps and native IPPO thereafter. Only this target differs between Direct and GPE.

Here t is the replanning time, q indexes a candidate joint action, and y_{tq}^{dir} is the scalar target supplied to the Direct Completion scorer. The factual candidate $q = 0$ has no residual, so $G_t^{(0)} = \sum_{j=t}^{\tau_0-1} r_j^{(0)}$ is the return of native IPPO from the same starting state. For $q > 0$, $G_t^{(q)}$ is measured on the branch that applies candidate q for $K = 4$ steps and then returns to native IPPO; $r_j^{(q)}$ is the shared team reward at step j, and τ_q is that branch's terminal step. Thus a positive y_{tq}^{dir} means that the intervention improves eventual episode return relative to candidate zero.

Under the Transport reward structure, the GPE target has a simple potential-difference interpretation. If the dense reward is $r_t = c[d(s_t) - d(s_{t+1})]$, then the candidate-relative GPE target is exactly equal to the corresponding strict K-step reward difference between candidate q and native candidate zero. Appendix G provides the derivation and a sufficient condition under bounded prediction error for preserving candidate rankings.

Joint geometric features The scorer uses a centralized feature vector that describes each candidate in the current package–goal geometry. It includes projections of each agent's action, residual, and velocity, together with smooth agent–package proximity features $h_\gamma(d) = \exp(-\gamma d)$ for five fixed distance scales $\gamma \in \{2, 5, 10, 20, 40\}$. Four agent blocks are augmented with coordinatewise mean, maximum, and minimum summaries and the normalized time feature, giving 351 inputs per candidate. These are inputs to the scorer only; the prediction target remains the scalar four-step progress relative to native IPPO. The complete channel definitions and normalization are given in Appendix F.

Scoring actions relative to IPPO and deciding when to intervene The scorer predicts each candidate's four-step progress relative to native IPPO. Candidate zero is anchored to a score of exactly zero, so positive and negative scores indicate predicted improvements and degradations, respectively. Three independently trained predictors provide ensemble means and disagreement estimates. Guarded selection penalizes disagreement and subtracts a held-out calibration offset, accepting a nonzero residual only when its calibrated score is positive; otherwise it executes the native action. The guarded score is therefore:

$$\underbrace{\mathrm{GuardedScore}}_{\text{guarded score}} = \underbrace{\mu}_{\text{mean predicted advantage}} - \underbrace{\lambda\sigma}_{\text{predictor disagreement penalty}} - \underbrace{\rho}_{\text{calibration correction}} \tag{7}$$

If the best candidate's guarded score is positive, the controller executes its correction. Otherwise, it uses native IPPO. Our greedy mode uses the same anchored predictor, but skips this uncertainty-and-calibration acceptance test and selects the highest-mean-scoring candidate in the full bank, including candidate zero. Because candidate zero has score exactly zero, greedy selection falls back to the native action whenever every nonzero candidate has a nonpositive mean score (with the lowest-index tie rule at zero). The exact normalization, ensemble construction, calibration, and fallback rule are given below and in Appendix E. For clarity, the quantities used in Algorithm 1 are, for each candidate q,

$$p_{mtq} = s_y f_{\theta_m}(x_{tq}), \qquad \mu_{tq} = \frac{1}{M}\sum_{m=1}^{M} p_{mtq}, \qquad \sigma_{tq} = \mathrm{sd}_{m=1}^{M}\, p_{mtq}, \qquad b_{tq} = \mu_{tq} - \lambda\sigma_{tq}, \tag{8}$$

and

$$\widehat{q}_t = \arg\max_{q\in\{0,\dots,Q-1\}} b_{tq}, \qquad q_t^\star = \begin{cases} \widehat{q}_t, & \widehat{q}_t \neq 0 \text{ and } b_{t\widehat{q}_t} - \rho > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here s_y converts the normalized network output back to target units before the ensemble mean and disagreement are computed; b_{tq} is the uncertainty- penalized base score and λ is its uncertainty weight. The candidate zero is assigned $b_{t0} = 0$ and is available as the fallback; it does not receive the calibration penalty $-\rho$. The greedy rule instead chooses $\arg\max_{q\in\{0,\dots,Q-1\}} \mu_{tq}$. The calibration offset ρ is estimated once from a held-out calibration split: for each context whose best base-scored candidate is nonzero, we record the prediction error $b_{t\widehat{q}_t} - y_{t\widehat{q}_t}^{\mathrm{chunk}}$ and set ρ to its empirical $(1-\alpha)$ quantile, with $\alpha = 0.20$. Deployment therefore accepts the proposed residual only when its calibrated score $b_{t\widehat{q}_t} - \rho$ is positive; otherwise it uses candidate zero. The argmax uses the lowest candidate index to break ties, so a tie at zero selects candidate zero. If fewer than 32 calibration contexts have a nonzero argmax, calibration raises a support error and the scorer is not promoted to deployment; it does not silently substitute an uncalibrated gate. The rationale for this predeclared minimum-support rule is given in Appendix E.

Algorithm 1 Receding-horizon GPE control

Require: IPPO policy π; predictors $\{f_{\theta_m}\}_{m=1}^{M}$; feature statistics m_ϕ, s_ϕ; target scale s_y; calibration correction ρ; candidate generator $\mathcal{G}$; horizon T; commitment K

Reset the environment and observe o_0
for $t = 0, K, 2K, \ldots, T-1$ **do**
  if the environment is terminal **then**
    break
  $b_t \leftarrow \pi(o_t)$; $\{\delta_t^{(q)}\}_{q=0}^{Q-1} \leftarrow \mathcal{G}(b_t, t)$ with $\delta_t^{(0)} = 0$
  for $q = 0, \ldots, Q-1$ **do**
    $x_{tq} \leftarrow (\phi(o_t, b_t, \delta_t^{(q)}, t/T) - m_\phi) \oslash s_\phi$
    Compute $p_{mtq} = s_y f_{\theta_m}(x_{tq})$, then μ_{tq}, σ_{tq}, and b_{tq} from Eq. (8)
  $\widehat{q}_t \leftarrow \arg\max_{q=0,\ldots,Q-1} b_{tq}$
  if $\widehat{q}_t \neq 0$ and $b_{t\widehat{q}_t} - \rho > 0$ **then** $q^* \leftarrow \widehat{q}_t$
  else $q^* \leftarrow 0$
  for $k = 0, \ldots, \min(K-1, T-1-t)$ **do**
    Execute $\mathrm{clip}(\pi(o_{t+k}) + \delta_t^{(q^*)}, -1, 1)$ and observe o_{t+k+1}
    if the environment is terminal **then**
      break

Scorer Variants and Decision Rules All scorer variants share the same candidate bank and network architecture but differ in their prediction targets. Direct predicts the change in full-episode return following a short residual intervention, after which IPPO resumes control. GPE instead predicts the relative package goal-progress over the committed four-step window. Here, "native IPPO" denotes the frozen policy without a selector, "candidate zero" its unchanged action, and "selection" the fraction of replanning points choosing a nonzero team residual. We also studied an exploratory terminal-aware extension that adds package–goal overlap information during scorer training. It is not part of the primary method or claim; its target, selection rule, and inconclusive diagnostic results are reported in Appendix H.1.

4 Experimental design

We study VMAS Transport with four agents, one package, and one goal, using a frozen BenchMARL IPPO policy, a strong baseline for this task (Bettini et al., 2022). We train the scorer on 768 recorded decision points, taking 12 from each of 64 environment seeds. At each point, candidate actions are compared from the same starting state. We reserve another 384 decision points for calibration and 384 for offline testing, using different environment seeds for each split so that no seeded episode contributes to more than one split. Direct Completion and GPE use the same inputs, architecture, candidate banks, and training settings, with different prediction targets. Each scorer contains three predictors trained for 5,000 AdamW updates (batch size 128, learning rate 3×10^{-4}, weight decay 10^{-4}, gradient cap 5). The offline screen checks ranking correlation, selected progress, intervention rate, and plan regret relative to Direct Completion. We train the scorer independently with six random seeds while keeping the IPPO policy fixed. Each trained scorer is evaluated on the same 5,000 episodes, with $Q = 33$ candidates and up to 2,000 environment steps per episode. All controllers receive the same episode assignments, and we compare returns and successes on matching episodes. The policy and scorer were trained using the original $T = 100$ episode protocol; all controllers are evaluated under extended $T = 2000$ deployment, while GPE still predicts four steps ahead. Native IPPO remains candidate zero. Across six independently trained scorer ensembles using a fixed IPPO backbone, the reported confidence intervals quantify paired episode-level uncertainty in mean return gain on the matched evaluation set. Variability across independently trained scorer ensembles is reported separately as the standard deviation of the six scorer-run mean gains in Table 1. A native-success loss occurs when native IPPO succeeds on an episode but the residual selector does not; lower is better, with zero indicating full retention of native successes.

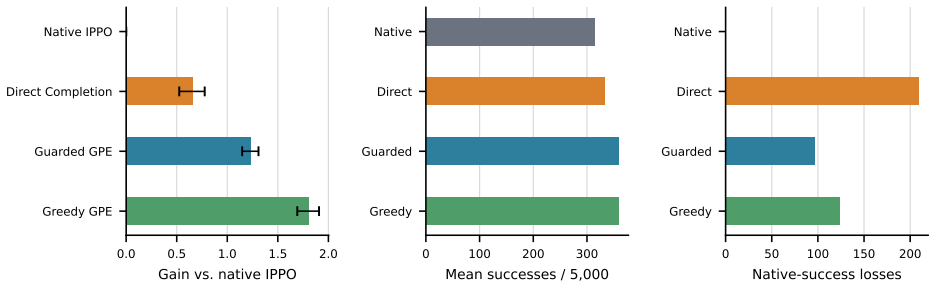

Figure 2: Primary matched confirmation at $Q = 33, T = 2000$, aggregated over six scorer seeds and 5,000 paired episodes per seed. Left: paired return gain over native IPPO (higher is better), with 95% bootstrap intervals. Middle: mean successful episodes out of 5,000 (higher is better). Right: native-success losses, counting episodes solved by native IPPO but lost by the selector.

Table 1: Primary matched Transport comparison ($Q = 33, T = 2000$): six scorer ensembles, each on 5,000 paired episodes. Gains and CIs are relative to native IPPO; New, Net extra, and losses aggregate 30,000 episodes. Bold marks the best selector value per outcome.

Controller	Mean return	Gain vs. IPPO	95% CI	Seed SD	Terminal successes	New successes	Net extra	Native losses
Native IPPO	6.11585	0.00000	[0,0]	0.00000	314.00	0	0	0
Direct Completion	6.76874	+0.65288	[0.52504,0.77682]	0.26340	333.50	326	+117	209
Guarded GPE	7.34390	+1.22805	[1.14724,1.30868]	0.09812	**359.17**	368	**+271**	**97**
Greedy GPE	**7.91757**	**+1.80172**	[1.69241,1.90740]	0.07366	359.00	**394**	+270	124

5 RESULTS

Direct Completion predictions have weak linear association with exact four-step candidate progress (Pearson 0.071), and its selected-progress interval crosses zero. GPE reaches 0.526 correlation, selects actions with positive exact progress, and reduces plan regret by 46.6%. This is the first place where the target change is visible; the complete offline screen is reported in Appendix C. The primary confirmation uses six scorer seeds, a common 5,000-episode bank, and $Q = 33, T = 2000$. Every method uses the same environment-seed assignments, frozen IPPO backbone, and the same keyed pseudo-random candidate-generation rule. Direct Completion improves over native IPPO, while both GPE decision rules improve more. Across the 30,000 paired controller episodes, native IPPO succeeds 1,884 times. Guarded GPE creates 368 successes on episodes that native IPPO does not solve and loses 97 native successes, for a net increase of 271 successes. Greedy GPE creates 394 new successes and loses 124 native successes, for a net increase of 270. Table 1 reports these matched crossover counts with the return results; protocol details are retained in Appendix E. The paired design estimates mean return differences on matched episode assignments, reducing variation from differences in evaluation samples.

5.1 ABLATION STUDIES

The main result compares GPE with a matched Direct Completion scorer. We also ran ablations to check which design choices matter after the GPE score has been trained: using the scorer greedily, adding the conservative gate, replacing GPE ranking with random rate-matched residuals, and changing the commitment horizon. These ablations separate two questions that are easy to mix together: whether the learned GPE score is useful, and whether the conservative gate is the best decision rule for return.

Selection-rule ablation. The guarded controller accepts a residual only when its calibrated lower confidence bound is positive; otherwise it executes candidate zero, the native IPPO action. For candidate q at decision time t, $\mathrm{LCB}_{tq}(\lambda) = \mu_{tq} - \lambda\sigma_{tq} - \rho$, where μ_{tq} is the ensemble mean, σ_{tq} its disagreement, λ the uncertainty weight, and ρ the calibration offset. The sealed setting uses $\lambda = 1$. Greedy GPE uses the same scorer but selects by μ_{tq} alone. Rate-matched random is a score-free control: at each four-step decision, it chooses uniformly from the 32 nonzero residuals with probability 0.45047, equal to Guarded GPE's observed mean selection rate on this primary

Table 2: Gate ablation at $Q = 33, T = 2000$ (six scorer runs; 5,000 paired episodes each). Gains are relative to IPPO; New, Net extra, and losses aggregate 30,000 episodes. Bold marks the best performance outcome.

Controller	Gain vs. IPPO	95% CI	Selection	Terminal successes	New	Net extra	Native losses
Rate-matched random	-0.64699	[-0.72743,-0.56612]	45.05%	301.00	48	-78	126
Guarded GPE	+1.22805	[1.14724,1.30868]	45.05%	**359.17**	368	**+271**	**97**
Greedy GPE	**+1.80172**	[1.69241,1.90740]	83.81%	359.00	**394**	+270	124

Table 3: Matched scorer-input ablation ($Q = 17, T = 100$). Only scorer inputs differ; all other components are fixed. Six scorer runs use 5,000 paired episodes each. All success quantities are aggregate counts over the 30,000 paired episodes; native IPPO has 66 terminal successes on this shared bank. Bold marks the best performance value.

Scorer input	Offline Pearson	Gain (mean ± seed SD)	Terminal successes (total)	New	Net extra	Native losses
Normalized geometry (ours)	**0.5224**	$+0.06942 \pm 0.00351$	**75**	**15**	**+9**	6
Raw normalized inputs	0.2508	$+0.02312 \pm 0.00136$	72	7	+6	**1**
Geometry without normalization	0.4594	$+0.06595 \pm 0.00496$	67	7	+1	6

protocol, and otherwise selects candidate zero. This holds the average number of interventions fixed while removing score-based choice. It is therefore a fairness control: it tests whether GPE's gain comes from informed residual selection rather than simply acting more frequently. As Table 2 shows, matching the intervention rate alone is not sufficient: score-free random residuals perform poorly, whereas both score-based rules improve return and terminal success. Greedy GPE yields the largest return gain; the guarded rule incurs fewer native-success losses because it can fall back to native IPPO when its score is uncertain. This is also visible in the matched crossover counts: guarded GPE creates 368 new successes and has $+271$ net extra successes, while greedy GPE creates 394 new successes and has $+270$ net extra successes. Random selection creates only 48 new successes and loses 126 native successes (-78 net).

Scorer-input ablation. We next test whether GPE benefits from the geometric input features described above, rather than simply any input vector of the same size. We hold the frozen IPPO controller, $Q = 17$ candidate bank, scalar $K = 4$ progress target, network, six scorer runs, guarded decision rule, and 5,000 matched episodes per run fixed. Only the scorer input changes: our normalized geometric features, raw normalized inputs, or geometric features without normalization. Table 3 therefore isolates what the feature design contributes. The normalized geometric input features are best on both prediction and control. It raises offline Pearson correlation to 0.522, compared with 0.251 for raw inputs, and yields a mean paired gain of $+0.06942$, compared with $+0.02312$. It also produces nine net extra terminal successes over 30,000 matched episodes. Removing normalization is closer in return ($+0.06595$), but gives only one net extra success. Thus the ablation supports the two design choices together: explicit geometry supplies a useful progress signal, and normalization makes that signal more reliable for selection.

Candidate-count ablation. Table 4 reports the candidate-budget comparison across three candidate-bank sizes. Each bank includes the native fallback (candidate zero) and bounded complete-team residuals; $Q = 17, 33, 65$ therefore provide 16, 32, and 64 non-native alternatives, respectively. The same sizes are used for Direct Completion and GPE, with a fresh calibration fit for each guarded-GPE bank. The table contains all results and tests whether a larger choice set improves ranking and closed-loop performance. Across the three bank sizes, the table shows the same pattern: GPE gains remain above Direct Completion, and greedy selection gives the largest returns while the guarded rule trades some return for a lower intervention rate. Increasing Q improves the available search space but also increases per-decision scoring cost; no additional interpretation is needed beyond the values shown in the table. Increasing the bank from $Q = 17$ to $Q = 33$ and $Q = 65$ improves greedy GPE's paired return gain from $+0.08312$ to $+0.09525$ and $+0.10398$, respectively. The matched Direct Completion gains remain smaller ($+0.06080$ at $Q = 33$ and $+0.06530$ at $Q = 65$), indicating that the improvement is not explained by searching more residuals alone. The $Q = 65$ guarded result is also positive (+0.08711) with one native-success loss, but remains below greedy GPE; the gate therefore preserves its conservative behavior as the candidate bank grows.

Table 4: Candidate-count ablation at the sealed $T = 100$ horizon, averaged over six scorer runs with 5,000 paired episodes per run. Q includes the native candidate. Return denotes mean episode return, and return gain is measured relative to native IPPO on matched episodes. Selection is the fraction of replanning decisions that choose a non-native residual. Successes reports the mean number of successful episodes per scorer run. New successes and net extra successes are aggregated across the six scorer runs (30,000 paired episodes per controller row), while native-success losses count episodes solved by native IPPO but not by the corresponding selector. Bold marks the best value within each Q block for a performance metric.

Controller	Return	Gain vs. IPPO	Selection	Successes	New	Extra	losses
Q=17 Direct Completion	0.72646	+0.01716	18.17%	19.00	0	0	**0**
Q=17 guarded GPE	0.77690	+0.06760	26.98%	20.00	7	+6	1
Q=17 greedy GPE	**0.79243**	**+0.08312**	61.36%	**20.50**	**9**	**+9**	**0**
Q=33 Direct Completion	0.77010	+0.06080	99.56%	19.17	1	+1	**0**
Q=33 guarded GPE	0.78972	+0.08041	28.70%	20.17	8	+7	1
Q=33 greedy GPE	**0.80455**	**+0.09525**	62.90%	**20.50**	**9**	**+9**	**0**
Q=65 Direct Completion	0.77460	+0.06530	99.82%	19.50	3	+3	**0**
Q=65 guarded GPE	0.79641	+0.08711	29.77%	20.33	9	+8	1
Q=65 greedy GPE	**0.81328**	**+0.10398**	64.12%	**20.67**	**11**	**+10**	1

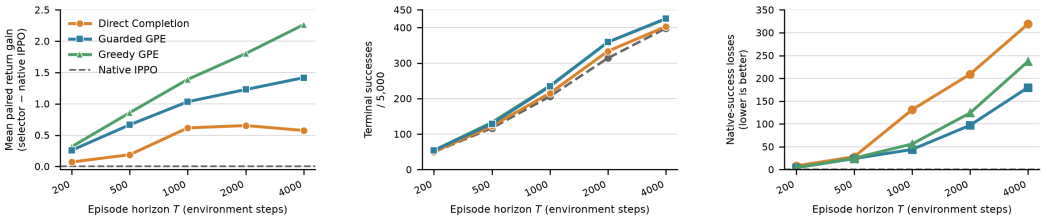

Figure 3: Transport horizon sensitivity with $Q = 33$, six scorer seeds, and 5,000 matched episodes per seed. Panels show paired return gain versus native IPPO, terminal successes, and native-success losses. Gray dashed: native IPPO; solid curves: selectors. Higher gain/success and lower loss are better.

Episode-horizon sensitivity (T). Here T is the maximum number of primitive environment steps allowed in one episode; it is not the number of evaluation episodes. We study the effect of this episode horizon by evaluating the same frozen policies and scorers at $T \in \{200, 500, 1000, 2000, 4000\}$. Only the episode limit and corresponding evaluation bookkeeping change; the candidate bank and selection rules stay fixed for each comparison, and the local prediction and commitment length remains $K = 4$. Figure 3 shows paired return gain, terminal successes, and native-success losses. Higher return and success are better, while lower loss is better. Appendix Table 10 reports the detailed numerical values underlying the uniform-$Q = 33$ horizon curve. TThe exploratory $Q = 17$ results at $T \in \{200, 500\}$ are reported in Table 9, while detailed $Q = 33$ horizon results and runtimes appear in Tables 10 and 18. Additional ablations are summarized in Appendix B.

6 Conclusion

We studied whether predicting short-horizon, candidate-relative goal progress can support post-training joint residual selection around a frozen cooperative policy. On VMAS Transport, GPE provides a stronger offline prediction signal and improves return more than Direct Completion. Greedy selection achieves the highest return, while guarded selection incurs fewer observed native-success losses. These results suggest that predicting the local effect of an intervention can provide a useful signal for choosing coordinated residual actions around a frozen IPPO policy. The findings also show that dense progress and terminal success must be evaluated separately. Future work should test this idea with additional policies and tasks, develop adaptive candidate banks, and study terminal-aware selection under the same deployment horizon.

ETHICS STATEMENT

The experiments use simulated cooperative robots and contain no human-subject or personal data. The main research-integrity risk is selective reporting after adaptive method development. We therefore separate development from confirmation, retain failed precursor summaries, and report both return gains and native-success losses for the confirmed Direct Completion, guarded GPE, and greedy GPE controllers.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, literature discovery, and improving the clarity of the presentation. AI tools were not used to generate experimental data or autonomously determine the reported results. All suggested literature, technical statements, experimental interpretations, and manuscript text were independently checked and verified by the authors. The authors take full responsibility for the methods, results, claims, and final content of the paper.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the environment, frozen IPPO backbone, candidate-bank construction, scorer architecture, training procedure, calibration protocol, evaluation horizons, and random-seed structure in the main paper and appendices. The fit, calibration, offline-test, development, and confirmation data banks are disjoint, and the same matched evaluation episodes are used when comparing controllers. We report the primary hyperparameters, candidate-bank sizes, scorer ensemble settings, and evaluation protocols required to reproduce the experiments. The source code used for the experiments is included in the supplementary material.

REFERENCES

Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. Vmas: A vectorized multi-agent simulator for collective robot learning, 2022. URL https://arxiv.org/abs/2207.03530.

Wendelin Boehmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 980–991. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/boehmer20a.html.

Junlin Chen, Ruijie Wang, and Jianxin Li. Counterfactual quotient models: Learning what actions change, not what the world does, 2026. URL https://arxiv.org/abs/2608.22092.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018. URL https://arxiv.org/abs/1805.12114.

Vladimir Egorov and Aleksei Shpilman. Scalable multi-agent model-based reinforcement learning, 2022. URL https://arxiv.org/abs/2205.15023.

Mingxiao Feng, Yaodong Yang, Wengang Zhou, and Houqiang Li. TIMAR: Transition-informed representation for sample-efficient multi-agent reinforcement learning. *Neural Networks*, 184: 107081, 2025. doi: 10.1016/j.neunet.2024.107081.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual Multi-Agent Policy Gradients. In *AAAI Conference on Artificial Intelligence*, pp. 2974–2982, 2018. doi: 10.1609/AAAI.V32I1.11794. URL https://mlanthology.org/aaai/2018/foerster2018aaai-counterfactual/.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang (Shane) Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 61573–61624. Curran Associates, Inc., 2023. doi: 10.52202/075280-2691. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c20ac0df6c213db6d3a930fe9c7296c8-Paper-Conference.pdf.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control, 2022. URL https://arxiv.org/abs/2203.04955.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024. URL https://arxiv.org/abs/2310.16828.

Jiahao Huang, Rongpeng Li, Zhifeng Zhao, Guoru Ding, and Honggang Zhang. WONDER: A radio world model-based negotiation framework for multi-agent UAV coverage optimization. *arXiv preprint arXiv:2608.16955*, 2026. URL https://arxiv.org/abs/2608.16955.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation*, pp. 6023–6029, 2019. doi: 10.1109/ICRA.2019.8794127.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652–3661. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/laroche19a.html.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S Stepleton, Nicolas Heess, Arthur Guez, Eric Moulines, Marcus Hutter, Lars Buesing, and Remi Munos. Counterfactual credit assignment in model-free reinforcement learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7654–7664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/mesnard21a.html.

Harshit Sikchi, Wenxuan Zhou, and David Held. Learning off-policy with online planning, 2021. URL https://arxiv.org/abs/2008.10066.

Pengcheng Wang, Chenran Li, Catherine Weaver, Kenta Kawamoto, Masayoshi Tomizuka, Chen Tang, and Wei Zhan. Residual-mppi: Online policy customization for continuous control, 2025. URL https://arxiv.org/abs/2407.00898.

Zhizun Wang and David Meger. Multi-agent model-based reinforcement learning with joint state-action learned embeddings, 2026. URL https://arxiv.org/abs/2602.12520.

A CANDIDATE-BANK CONSTRUCTION

Figure 4 summarizes how the complete-team candidate bank is formed at each replanning point. Candidate zero is the exact native action; the other candidates are coordinated perturbations applied to all four agents together.

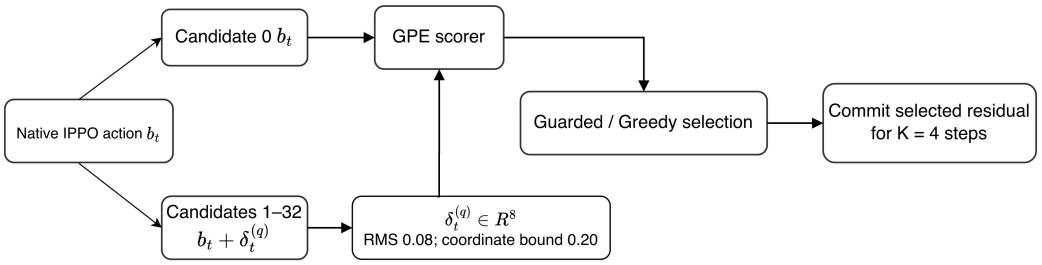

Figure 4: Construction of the complete-team candidate bank used by GPE at each replanning point. Candidate zero is the unmodified native IPPO joint action, while the remaining candidates add bounded eight-dimensional residuals that jointly perturb the actions of all four agents. The resulting candidate joint actions are scored by GPE, after which the guarded or greedy decision rule selects either a residual candidate or the native-policy fallback. The selected residual is then committed for $K = 4$ primitive steps before replanning.

B ADDITIONAL EXPERIMENTS

This section presents additional ablations and exploratory experiments that examine the robustness and scope of GPE beyond the primary evaluation. Unless otherwise specified, comparisons use a frozen policy and paired evaluation episodes to measure the effects of residual selection. We report both return improvements and native-success losses to distinguish overall performance from the preservation of episodes successfully completed by native IPPO.

Scorer-input ablation details. Table 3 in the main paper reports the fully matched six-seed comparison. Each row retains the same scalar $K = 4$ goal-progress target and changes only the scorer input features. The offline Pearson value is linear association with true four-step progress; the paired return gain and terminal crossover counts provide the complementary closed-loop test of whether that prediction signal is useful for selection.

LCB sensitivity. We test whether the guarded controller is conservative only because its uncertainty multiplier is $\lambda = 1$. Using the same six scorers and 5,000 paired episodes, we recalibrate the gate for $\lambda \in \{0.25, 0.5, 1.0, 2.0\}$ at $Q = 33, T = 2000$ (Table 5; Figure 5). The return intervals overlap: $\lambda = 0.5$ has the largest mean gain, while $\lambda = 2$ is slightly lower and does not reduce native-success losses. Thus the calibrated positive-LCB rule matters more than any single multiplier. The $\lambda = 1$ sensitivity row uses a fresh recalibration, so its selection rate and interval need not exactly match the sealed primary row even though the reported mean gain and loss count coincide.

Table 5: LCB sensitivity for guarded GPE at $Q = 33, T = 2000$, averaged over six scorer seeds and 5,000 paired episodes per seed. Each row recalibrates the gate for its displayed λ. Return gain and success are higher-is-better; native-success losses are better when lower. New and net extra successes aggregate the 30,000 paired controller episodes.

Controller	Gain vs. IPPO	95% CI	Success rate	Selection	New	Net extra	Native losses
LCB, $\lambda = 0.25$	+1.22922	[1.15066,1.30892]	7.173%	43.26%	361	+268	**93**
LCB, $\lambda = 0.5$	**+1.24827**	[1.16942,1.32477]	**7.197%**	42.85%	370	**+275**	95
LCB, $\lambda = 1.0$	+1.22805	[1.15008,1.30708]	7.183%	43.80%	368	+271	97
LCB, $\lambda = 2.0$	+1.14874	[1.07200,1.22705]	7.097%	43.13%	339	+245	94

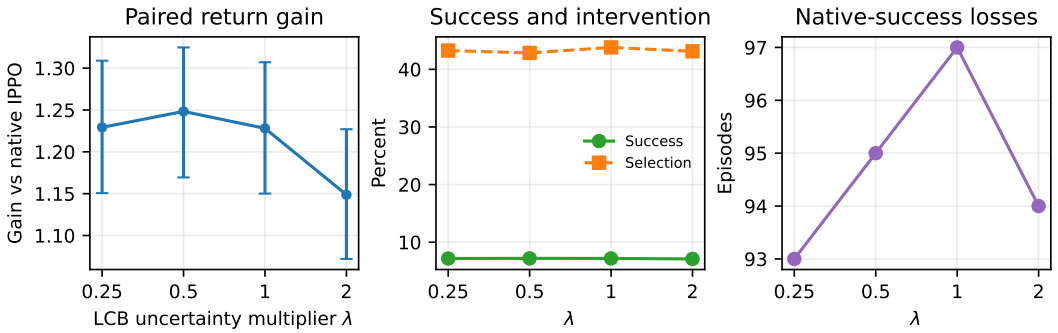

Figure 5: Primary LCB sensitivity at $Q = 33, T = 2000$, aggregated over six scorer seeds and 5,000 paired episodes per seed. Left: paired return gain over native IPPO with 95% bootstrap intervals. Middle: terminal success rate and fraction of replanning decisions that use a nonzero residual. Right: native-success losses, where zero is best. The four multipliers have overlapping return intervals; changing λ alone does not remove the guarded–greedy performance trade-off.

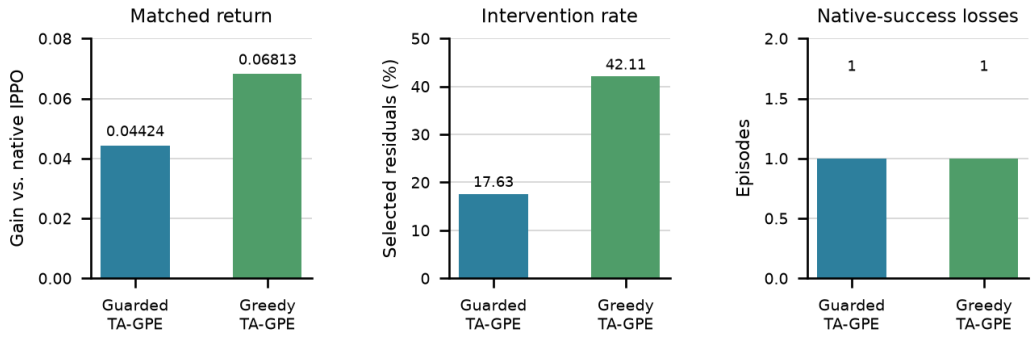

Figure 6: Terminal-aware diagnostics under the separate $Q = 33, T = 100$ confirmation, aggregated over six scorer seeds with 5,000 paired episodes per seed. Higher return is better; intervention rate is descriptive; lower native-success loss is better.

Horizon and terminal-aware checks. Evaluations at $T \in \{200, 500, 1000, 2000, 4000\}$ use the same $Q = 33$ candidate bank and frozen scorer, changing only the episode limit; return generally increases with horizon, while the guarded rule has fewer native-success losses than greedy selection. The corresponding curve and complete numerical results are given in Figure 3 and Table 10, respectively. The terminal-aware extension adds overlap information to the progress target. In the completed Q=33 confirmation, guarded and greedy TA–GPE gain $+0.04424$ and $+0.06813$ over IPPO, respectively, with one native-success loss each across 30,000 paired episodes. The two loss counts therefore answer different experimental questions and cannot show that terminal-aware GPE reduces losses relative to standard GPE. Figure 6 and Tables 6 summarize the terminal-aware confirmation.

Commitment-length ablation (K). We also tested longer local prediction and commitment lengths under a separate exploratory protocol. Here K is both the number of primitive steps for which the residual is held fixed and the local prediction horizon. Table 7 contains the complete three-seed development results. Although $K = 8$ and $K = 16$ pass their offline screens and improve over native IPPO, neither outperforms its length-matched scalar K-step reward control: the contrast interval includes zero for $K = 8$ and includes negative values for $K = 16$. We therefore did not promote either setting to confirmation. Exact selected progress is accumulated over a different number of steps for each row, so it is not an across-row measure of the preferred deployment setting; the closed-loop return columns are more informative for that comparison.

Table 6: Terminal-aware confirmation at $Q = 33, T = 100$. New, Net extra, and losses aggregate 30,000 paired episodes.

Controller	Gain vs. IPPO	Selection	Successes / 5,000	New	Net extra	Native losses
Guarded TA–GPE	+0.04424	17.63%	8.83	0	-1	1
Greedy TA–GPE	**+0.06813**	42.11%	8.83	0	-1	1

Table 7: Exploratory commitment-length ablation using three scorer seeds. K is the local prediction and residual-commitment length. Pearson, exact selected progress, and selection are offline quantities; Return–IPPO and Return–control are paired closed-loop gains. Bold marks the best value in each directly comparable column.

K	Pearson	Exact selected	Selection	Return–IPPO	Return–control
4	0.526	0.000990	25.52%	**0.08198**	**0.06751**
8	**0.535**	0.003083	24.57%	0.06303	0.00314
16	0.357	**0.005666**	23.52%	0.05170	−0.00924

B.1 Short-horizon backbone comparison and success retention

We also test whether post-training residual selection extends to MAPPO and ISAC. For each frozen policy, we train matched Direct Completion and GPE scorers on policy-specific counterfactual candidates and compare the selected controller with its native backbone. Table 8 and Figure 7 summarize the available $Q = 17, T = 100$ results on 800 matched episodes. IPPO and MAPPO use guarded selection; ISAC uses greedy selection. Replication differs across backbones, so these are within-backbone tests of residual prediction rather than a ranking of MARL algorithms.

Table 8: Exploratory backbone comparison at $Q = 17, T = 100$ on 800 matched episodes per run. Native return reports the absolute episode mean ± SD of each frozen controller; gains are relative to that controller on matched episodes. IPPO and MAPPO use guarded selection; ISAC uses greedy selection. The deliberately unequal replication counts are one fixed IPPO policy × six scorer runs, six MAPPO policies × six scorer runs, and three ISAC policies × three scorer runs. Thus cross-backbone values are not direct comparisons. Uncertainty intervals and safety outcomes appear in Figure 7.

Backbone	Native mean ± SD	Direct gain	GPE gain
IPPO	0.63245 ± 1.28851	$+0.01716$	$+0.06980$
MAPPO	0.20411 ± 0.82317	$+0.02693$	$+0.02687$
ISAC	0.00027026 ± 0.00750	$+0.00009086$	$+0.00008013$

We also applied the same residual-selection pipeline to MAPPO and ISAC as additional frozen backbones. Table 8 shows small positive mean return gains for both Direct Completion and GPE on these exploratory evaluations. The substantially larger improvement occurs with IPPO: GPE exceeds Direct Completion by 0.05264 in mean return. On MAPPO and ISAC, GPE does not exceed Direct Completion. These experiments establish the scope of the result: the strongest evidence is for IPPO, while broader backbone generalization requires further matched study.

IPPO success retention. For the IPPO subset, we evaluated GPE using six scorer seeds on a common bank of 800 paired episodes at $Q = 17, T = 100$. All six seeds achieved positive return gains over native IPPO, ranging from $+0.06693$ to $+0.07267$. However, native IPPO succeeded on three episodes, whereas GPE succeeded on only two for every scorer seed, with all six seeds losing the same native-policy success. This result illustrates that consistent improvements in average return do not necessarily translate into preservation of native-policy successes. Figure 7 places these results alongside MAPPO and ISAC; the backbone stress-test paragraph discusses their different decision rules, replication counts, and available diagnostic evidence.

We also study the earlier $Q = 17$ candidate bank at $T = 200$ and $T = 500$. At $T = 200$, native IPPO reaches 49 successes per 5,000 episodes. Guarded and greedy GPE gain $+0.21619$ and $+0.27023$,

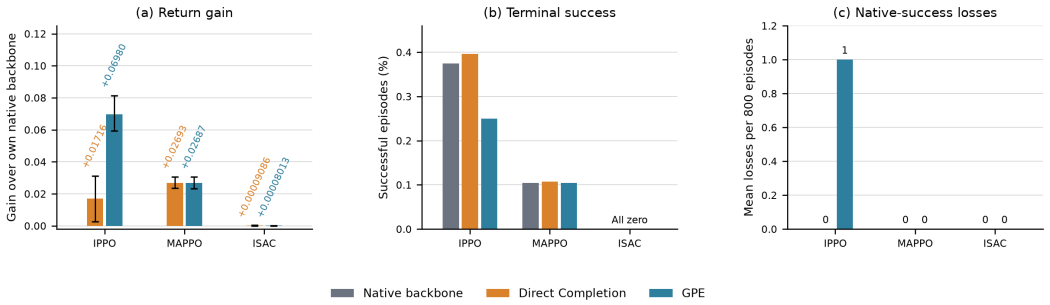

Figure 7: Backbone overview at $Q = 17, T = 100$ on the same 800 episode assignments. IPPO and MAPPO use guarded selection; ISAC uses greedy selection (hatched gain/loss bars). Table 8 gives replication counts. Left: return gain over each native policy with 95% intervals from jointly resampled episode assignments, conditional on the trained models. Middle: terminal-success rate, the fraction of rollouts ending with the package inside the goal. Right: mean native-success losses per 800-episode run. Higher gain and terminal success, and fewer losses, are preferable; ISAC has no native successes to preserve. Decision-rule and replication differences limit comparisons across backbones.

Table 9: Exploratory Transport evaluations for $T \in \{200, 500\}$ with $Q = 17$. Each row averages six scorer seeds over 5,000 matched episodes per seed, using frozen $T = 100$ policies and scorer checkpoints. Gain is paired against native IPPO on identical episodes. Higher gain and success are better, while lower native-success losses are better. New and net extra successes aggregate the 30,000 paired controller episodes for each row. Selection denotes the fraction of replanning decisions using a non-native residual.

T	Controller	Mean return	Gain vs. IPPO	Successes	New	Extra	losses	Selection
200	Native IPPO	1.61400	0.00000	49.00	0	0	0	0.00%
	Direct Completion	1.67546	+0.06146	48.83	12	-1	13	24.81%
	Guarded GPE	1.83019	+0.21619	**54.33**	**36**	**+32**	**4**	37.27%
	Greedy GPE	**1.88423**	**+0.27023**	53.67	32	+28	**4**	73.05%
500	Native IPPO	3.70318	0.00000	117.00	0	0	0	0.00%
	Direct Completion	3.85640	+0.15323	121.00	53	+24	29	29.31%
	Guarded GPE	4.24189	+0.53871	129.00	94	+72	**22**	42.35%
	Greedy GPE	**4.41566**	**+0.71248**	**129.67**	**109**	**+76**	33	79.94%

respectively; across 30,000 matched controller episodes, they create 36 and 32 new successes, for net increases of 32 and 28. Guarded GPE has the highest mean success count (54.33), while greedy GPE has the highest return. At $T = 500$, native IPPO reaches 117 successes. Direct Completion reaches 121 with gain $+0.15323$, while guarded and greedy GPE reach 129 and 129.67, with gains $+0.53871$ and $+0.71248$. Their new-success counts are 94 and 109, which yield net increases of 72 and 76 after 22 and 33 native-success losses. Thus the longer-horizon results show the same return–loss trade-off: greedy selection is more aggressive, while the gate preserves more native successes. These exploratory $Q = 17$ results are listed in Table 9. We also study performance across episode horizons using a uniform $Q = 33$ candidate bank. Table 10 reports the results, and Table 18 reports the corresponding end-to-end runtimes. Overall, the larger candidate bank yields stronger return gains, while terminal-success differences between the two candidate budgets are smaller and not uniformly ordered.

Table 10: Numeric values underlying the uniform $Q = 33$ horizon curve. Each entry uses six scorer runs and 5,000 matched episodes per run. Return is mean ± SD across scorer-run means; terminal successes are mean counts per 5,000 episodes. New, Net extra, and losses aggregate the 30,000 paired episodes.

T	Controller	Mean return ± SD	Terminal successes	New	Net extra	Native losses
200	Native IPPO	1.61400 ± 0.00000	49.00	0	0	0
	Direct Completion	1.68460 ± 0.01180	50.17	15	+7	8
	Guarded GPE	1.87302 ± 0.00937	**54.00**	**34**	**+30**	**4**
	Greedy GPE	1.92949 ± 0.01525	**54.00**	**34**	**+30**	**4**
500	Native IPPO	3.70318 ± 0.00000	117.00	0	0	0
	Direct Completion	3.89057 ± 0.04597	122.50	61	+33	28
	Guarded GPE	4.36808 ± 0.03716	129.50	99	+75	**24**
	Greedy GPE	4.56116 ± 0.02217	**133.00**	**121**	**+96**	25
1000	Native IPPO	5.71623 ± 0.00000	206.00	0	0	0
	Direct Completion	6.33149 ± 0.11808	215.17	186	+55	131
	Guarded GPE	6.74938 ± 0.05658	234.33	214	+170	**44**
	Greedy GPE	7.10580 ± 0.05354	**236.00**	**236**	**+180**	56
2000	Native IPPO	6.11585 ± 0.00000	314.00	0	0	0
	Direct Completion	6.76874 ± 0.26340	333.50	326	+117	209
	Guarded GPE	7.34390 ± 0.09812	**359.17**	368	**+271**	**97**
	Greedy GPE	7.91757 ± 0.07366	359.00	**394**	+270	124
4000	Native IPPO	1.85498 ± 0.00000	398.00	0	0	0
	Direct Completion	2.42935 ± 0.39816	402.67	347	+28	319
	Guarded GPE	3.27202 ± 0.15197	**425.33**	344	**+164**	**180**
	Greedy GPE	4.11871 ± 0.13156	424.50	**396**	+159	237

C OFFLINE DIAGNOSTICS

We evaluate the predictive and action-selection quality of Direct Completion and GPE on 384 held-out decision contexts, averaging results across three independently trained scorer seeds. Table 11 reports the complete offline eligibility metrics, while Figure 8 visualizes the key differences in prediction correlation, exact selected progress, and plan regret.

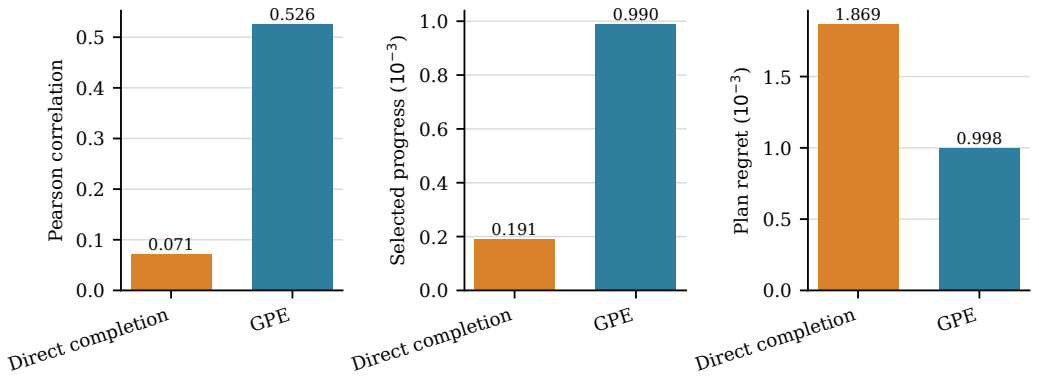

Figure 8: Offline diagnostics on 384 held-out contexts. Direct Completion and GPE share the same candidate bank and model capacity but use different prediction targets. GPE has higher candidate-level correlation, larger exact selected progress, and lower plan regret.

Table 11: Sealed offline eligibility screen, averaged over the original three scorer seeds on 384 held-out contexts. Pearson correlation measures linear association between predicted and true candidate progress ($+1$ is perfect positive linear association, 0 is no linear association, and negative values indicate inverse association; it is not a direct within-context ranking- accuracy measure). Exact selected progress is the true four-step progress of the selected candidate relative to native IPPO; positive values are better. The 95% lower bound should be positive, selection should exceed 5%, and plan regret is lower-is-better.

Method	Pearson	Exact progress	95% lower	Select	Regret
Direct completion	0.071	0.000191	-0.000042	20.05%	0.001869
GPE	**0.526**	**0.000990**	**0.000724**	25.52%	**0.000998**

D NOTATION

Table 12: Symbols used in the method and algorithm.

Symbol	Definition
N, T, K, Q, M	Number of agents (4), active episode horizon ($T = 100$ for training/data and $T = 2000$ for the primary evaluation), local prediction/commitment length (4), primary candidate count (33), and predictors per ensemble (3); $Q = 17$ is the ablation setting.
i, t, k, q, m	Agent, replanning step, within-macro offset, candidate, and ensemble-member indices.
o_t^i, o_t	Agent i's 11-dimensional observation and the stacked joint observation.
π, b_t	Frozen deterministic IPPO policy and its stacked live action $b_t = \pi(o_t)$.
$\delta_t^{(q)}, a_t^{(q)}$	Joint residual matrix and clipped candidate action for candidate q.
$s_{t+K}^{(q)}$	Environment state after executing candidate q for K live-IPPO steps.
$p^{\mathrm{pkg}}, p^{\mathrm{goal}}$	Two-dimensional package and goal positions.
$z, y^{\mathrm{gpe}}, y^{\mathrm{chunk}}$	Goal-distance scalar, progress target, and exact four-step reward advantage.
$G_t^{(q)}, y^{\mathrm{dir}}$	Branch-to-completion return and its candidate minus factual advantage.
ϕ, x, m_ϕ, s_ϕ	Raw 351-dimensional feature, normalized feature, training mean, and training scale.
g_θ, f_θ, s_y	Unanchored network, factual-anchored predictor, and positive target scale.
μ, σ, λ, b	Ensemble mean, standard deviation, uncertainty multiplier (1), and lower base score.
α, e, ρ	Calibration error level (0.20), chosen-candidate overprediction residual, and its quantile correction.

E PROTOCOL AND PREDICTOR-TRAINING DETAILS

Predictor training and calibrated selection. Let $\phi_{tq} \in \mathbb{R}^{351}$ be the feature vector for context t and candidate q. We standardize it coordinatewise with training mean $m_\phi \in \mathbb{R}^{351}$ and positive scale $s_\phi \in \mathbb{R}^{351}$:

$$x_{tq} = (\phi_{tq} - m_\phi) \oslash s_\phi, \tag{10}$$

where $\oslash$ is elementwise division and each entry of s_ϕ is the larger of the empirical standard deviation and 10^{-4}.

A multilayer perceptron $g_\theta : \mathbb{R}^{351} \to \mathbb{R}$ produces an anchored prediction

$$f_\theta(x_{tq}) = g_\theta(x_{tq}) - g_\theta(x_{t0}), \qquad f_\theta(x_{t0}) = 0. \tag{11}$$

The implementation overwrites the factual output with exact zero after subtraction. Hidden widths are 256, 256, and 128 with SiLU activations; the first layer also has layer normalization.

For target y_{tq}, let $s_y > 0$ be its training standard deviation with floor 10^{-4}, and let $\widetilde{y}_{tq} = y_{tq}/s_y$. On a minibatch $\mathcal{B}$, the loss is

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|(Q-1)} \sum_{t \in \mathcal{B}} \sum_{q=1}^{Q-1} w_{tq} \left(f_\theta(x_{tq}) - \widetilde{y}_{tq} \right)^2 + \beta \, \mathrm{CE}(f_\theta(x_t), q_t^*) \,, \tag{12}$$

Table 13: Primary Transport protocol ($Q = 33, T_{\text{eval}} = 2000$). Fit, calibration, offline-test, development, and confirmation banks are disjoint; the confirmation bank is shared across controllers and scorer runs.

Parameter	Primary setting
Fit bank	64 environments, 12 snapshots each (768 contexts)
Calibration bank	32 environments, 12 snapshots each (384 contexts)
Offline-test bank	32 environments, 12 snapshots each (384 contexts)
Development bank	320 held-out environments
Confirmation bank	5,000 matched episodes per scorer run
Candidate bank	$Q = 33$: 32 residual candidates plus the exact native action
Training/data horizon	$T_{\text{train}} = T_{\text{data}} = 100$ steps
Evaluation horizon	$T_{\text{eval}} = 2000$ steps per episode
Scorer-training runs	6 independent runs using a fixed IPPO backbone
Ensemble members per run	3 neural networks
Candidate residuals	Target RMS 0.08; bound 0.20
Commitment	4 steps; live IPPO action recomputed each step
Network	351, 256, 256, 128, 1; LayerNorm; SiLU
Optimization	AdamW; 5,000 updates; batch 128; learning rate 3×10^{-4}
Regularization	Weight decay 10^{-4}; gradient cap 5; rank weight 0.1
Calibrated gate	$\alpha = 0.20$; $\lambda = 1$; empirical correction ρ; minimum 32 calibration selections
Primary return intervals	95% paired bootstrap intervals; 5,000 draws

where $w_{tq} = 3$ if $|y_{tq}| > 10^{-5}$ and $w_{tq} = 1$ otherwise, $\beta = 0.1$, $x_t = (x_{t0}, \ldots, x_{t,Q-1})$, $q_t^* = \arg\max_{q \in \mathcal{Q}} y_{tq}$, and CE is multiclass cross entropy treating the Q predictions as logits.

Each training seed fits $M = 3$ predictors. At test context t, predictor m yields an unnormalized score $p_{mtq} = s_y f_{\theta_m}(x_{tq})$. Define

$$\mu_{tq} = \frac{1}{M} \sum_{m=1}^{M} p_{mtq}, \qquad \sigma_{tq} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (p_{mtq} - \mu_{tq})^2}, \qquad b_{tq} = \mu_{tq} - \lambda \sigma_{tq}, \tag{13}$$

where μ_{tq} is ensemble mean, σ_{tq} is population standard deviation, and $\lambda = 1$ in the sealed guarded setting. For the factual candidate, anchoring gives $p_{mt0} = \mu_{t0} = \sigma_{t0} = b_{t0} = 0$. On held-out calibration contexts, let $\widehat{q}_t = \arg\max_{q \in \{0,\ldots,Q-1\}} b_{tq}$, with $b_{t0} = 0$. Only contexts with $\widehat{q}_t \neq 0$ contribute a calibration residual. For those contexts, the overprediction residual is $e_t = b_{t,\widehat{q}_t} - y_{t,\widehat{q}_t}^{\text{chunk}}$. The scalar ρ is the finite-sample $(1-\alpha)$ empirical quantile of these e_t, with $\alpha = 0.20$. At deployment, candidate $\widehat{q}_t$ is accepted only if its calibrated score $b_{t\widehat{q}_t} - \rho$ is positive; otherwise candidate zero is used. Thus the factual candidate is an explicit zero-score fallback rather than receiving the calibration offset $-\rho$. This is the same b_{tq} and acceptance rule used in Eq. (9) and Algorithm 1. The minimum-support requirement of 32 counts these selected nonzero calibration contexts, not all 384 calibration contexts; contexts whose best base score is candidate zero are excluded from the residual quantile. Ties in the argmax use the lowest candidate index, so a tie at zero selects candidate zero. If fewer than 32 contexts qualify, the implementation rejects the calibration fit with a support error and does not deploy that scorer; it does not silently use an uncalibrated gate. This gate therefore requires at least 32 nonfactual calibration choices. The threshold is a protocol-level engineering safeguard, not a theoretical consequence of using $Q = 33$ candidates. We use it because the empirical $(1 - \alpha)$ residual quantile becomes unstable when it is estimated from only a handful of selected contexts; at 32 contexts, the $\alpha = 0.20$ tail is supported by several observations rather than a single extreme sample. This threshold was fixed before confirmation, counts contexts rather than candidate actions, and is not itself a formal safety guarantee.

F Complete feature definition

For current observation o_t, let $u_t^g \in \mathbb{R}^2$ be the unit vector from package to goal. For each agent i, let $u_t^i \in \mathbb{R}^2$ be the normalized package-to-agent displacement stored in the Transport observation, d_t^i its norm, and $v_t^i \in \mathbb{R}^2$ the agent velocity minus package velocity. All unit-vector denominators have floor 10^{-6}.

For two planar vectors $x = (x_1, x_2)$ and $y = (y_1, y_2)$, define $\langle x, y\rangle = x_1y_1 + x_2y_2$ and $x \times y = x_1y_2 - x_2y_1$. For each $w \in \{a_t^{(q,i)}, \delta_t^{(q,i)}, v_t^i\}$, the feature map includes

$$\langle w, u_t^g\rangle, \quad \langle w, u_t^i\rangle, \quad w \times u_t^g, \quad w \times u_t^i, \quad \|w\|_2, \tag{14}$$

where $a_t^{(q,i)}$ and $\delta_t^{(q,i)}$ are agent i's two action coordinates in candidate q. For each fixed distance-decay coefficient $\gamma \in \{2, 5, 10, 20, 40\}$, define the proximity feature $h_{t\gamma}^i = \exp(-\gamma d_t^i)$. Larger coefficients emphasize shorter distances; γ here is unrelated to reward discounting. The map also includes

$$h_{t\gamma}^i, \quad h_{t\gamma}^i\langle\delta_t^{(q,i)}, u_t^g\rangle, \quad h_{t\gamma}^i\langle\delta_t^{(q,i)}, u_t^i\rangle, \quad h_{t\gamma}^i(\delta_t^{(q,i)} \times u_t^i). \tag{15}$$

Equation 14 contributes 15 channels and Equation 15 contributes 20. Concatenating these with the raw 11-dimensional observation, two action coordinates, and two residual coordinates yields agent block $\psi_{tq}^i \in \mathbb{R}^{50}$. The final map is

$$\phi_{tq} = \left[\psi_{tq}^1; \ldots; \psi_{tq}^4; \mathrm{mean}_i\, \psi_{tq}^i; \max_i \psi_{tq}^i; \min_i \psi_{tq}^i; t/T\right] \in \mathbb{R}^{351}, \tag{16}$$

where semicolons denote concatenation and mean, maximum, and minimum operate coordinatewise over the four agents. The stored dataset contains eight observation steps for compatibility with the precursor experiments, but Equation 16 uses only the current step.

G THEORY OF THE LOCAL GOAL-PROGRESS TARGET

This section states the limited guarantees supplied by the target definition. They concern short-horizon potential differences and prediction error; they do not assert that the geometric feature map is sufficient for global task success.

Potential-difference interpretation. Let

$$d(s) = \left\|p^{\mathrm{pkg}}(s) - p^{\mathrm{goal}}(s)\right\|_2. \tag{17}$$

For a candidate q and the native candidate 0, the GPE target is

$$y_{tq}^{\mathrm{gpe}} = -c\left[d\left(s_{t+K}^{(q)}\right) - d\left(s_{t+K}^{(0)}\right)\right]. \tag{18}$$

Consequently, $y_{tq}^{\mathrm{gpe}} > 0$ if and only if candidate q leaves the package closer to the goal than the native action after the same K steps.

Relation to the shaped reward. Suppose the dense reward is the potential difference

$$r_t = c\,[d(s_t) - d(s_{t+1})]. \tag{19}$$

Both branches start from the same state, so summing over the commitment gives

$$\sum_{k=0}^{K-1}\left(r_{t+k}^{(q)} - r_{t+k}^{(0)}\right) = c\left[d(s_t^{(q)}) - d(s_{t+K}^{(q)}) - d(s_t^{(0)}) + d(s_{t+K}^{(0)})\right] \tag{20}$$

$$= -c\left[d(s_{t+K}^{(q)}) - d(s_{t+K}^{(0)})\right] = y_{tq}^{\mathrm{gpe}}. \tag{21}$$

Thus, under the potential-based reward assumption, GPE and the strict K-step scalar reward target are mathematically the same quantity. The empirical comparison remains useful because the implementation, numerical scaling, and feature representation can still affect prediction and selection.

When prediction preserves a candidate ranking. Let y_{tq} be the true local target and $\widehat{y}_{tq}$ its prediction. If for all candidates q,

$$|\widehat{y}_{tq} - y_{tq}| \le \varepsilon, \tag{22}$$

then the ordering of candidates q and r is preserved whenever

$$y_{tq} - y_{tr} > 2\varepsilon. \tag{23}$$

Indeed, subtracting the two error bounds gives $\widehat{y}_{tq} - \widehat{y}_{tr} \ge (y_{tq} - y_{tr}) - 2\varepsilon > 0$. Therefore the scorer need not predict the complete future accurately; it must resolve candidate differences larger than its local prediction error.

These statements are local and conditional. They do not prove that a residual will reach the terminal goal, nor do they provide a closed-loop safety guarantee after the selector changes the visited-state distribution. Those properties are evaluated empirically in the matched confirmation experiments.

G.1 TERMINAL-AWARE TARGET DEFINITIONS

For package position p, goal position g, package velocity v_p, and agent positions a_i, the six observable TA-1 channels are

$$d_{pg} = \|g - p\|_2, \qquad v_{\text{rad}} = v_p^\top \frac{g - p}{\|g - p\|_2 + \epsilon}, \tag{24}$$

$$v_{\text{speed}} = \|v_p\|_2, \qquad d_{\text{mean}} = \frac{1}{4}\sum_{i=1}^{4}\|a_i - p\|_2, \tag{25}$$

$$d_{\max} = \max_i \|a_i - p\|_2, \qquad c = \frac{1}{4}\sum_{i=1}^{4}\exp\left[-\left(\frac{\|a_i - p\|_2}{d_c}\right)^2\right]. \tag{26}$$

Here c is the contact-support summary and d_c is its distance scale. TA-1 adds the signed terminal-overlap margin

$$m_{\text{term}} = r_{\text{goal}} - d_{\text{box}}(g, \text{oriented package}), \tag{27}$$

where positive values indicate package–goal overlap, zero indicates contact, and negative values indicate separation. The margin is used as a training signal; the deployed scorer receives only the observable channels. TA-2 would predict candidate-caused transitions out of the terminal set, but no such transitions were present in the enriched bank and that target was not trained.

Algorithm 2 Terminal-aware GPE selection

Require: Frozen IPPO policy π; terminal-aware ensemble; feature statistics; calibration correction ρ_{TA}; risk limit $r_{\max}$; candidate generator $\mathcal{G}$; horizon T; commitment K

Reset the environment and observe o_0
for $t = 0, K, 2K, \ldots, T-1$ **do**
  if the environment is terminal **then**
    break
  $b_t \leftarrow \pi(o_t)$; generate $\{\delta_t^{(q)}\}_{q=0}^{Q-1} \leftarrow \mathcal{G}(b_t, t)$ with $\delta_t^{(0)} = 0$
  for $q = 0, \ldots, Q-1$ **do**
    Predict $\Delta s_{tq}^{(m)}$ for each ensemble member m
    $P_{tq}^{(m)} \leftarrow -100\,\Delta d_{pg,tq}^{(m)}$; compute $B_{tq} = \text{mean}_m\, P_{tq}^{(m)} - \lambda\,\text{sd}_m\, P_{tq}^{(m)} - \rho_{\text{TA}}$
    $R_{tq} \leftarrow \text{mean}_m \max(\Delta v_{\text{rad,tq}}^{(m)}, \Delta v_{\text{speed,tq}}^{(m)}, -\Delta m_{\text{term,tq}}^{(m)}) + \text{sd}_m[\max(\cdot)]$
  $\mathcal{A}_t \leftarrow \{0\} \cup \{q > 0 : B_{tq} > 0 \ \wedge\ R_{tq} \le r_{\max}\}$
  $q^* \leftarrow \arg\max_{q \in \mathcal{A}_t} B_{tq}$; candidate zero is selected if $\mathcal{A}_t = \{0\}$
  for $k = 0, \ldots, \min(K-1, T-1-t)$ **do**
    Execute $\text{clip}(\pi(o_{t+k}) + \delta_t^{(q^*)}, -1, 1)$ and observe o_{t+k+1}
    if the environment is terminal **then**
      break

H ADDITIONAL NUMERICAL RESULTS

H.1 TERMINAL-AWARE EXTENSION AND SAFETY DIAGNOSTICS

The standard progress target provides dense reward information but does not explicitly represent the terminal package–goal overlap event. TA-1 therefore adds the signed overlap margin as a training signal while retaining the same candidate bank, $K = 4$ commitment, scorer architecture, and greedy/guarded deployment rules. The full target and selection procedure are given in Appendix G.1. A scorer trained only on near-goal snapshots failed its offline promotion gate, so the reported TA-1 fit bank mixes ordinary and near-goal contexts.

The six-seed confirmation uses a separate $Q = 33, T = 100$ protocol with 5,000 paired episodes per scorer seed and calibrated risk limit 0.01. It is not matched to the primary $T = 2000$ protocol. Both decision rules have positive return gain but create no new terminal successes and lose one native success across 30,000 episodes. These results are therefore supplementary evidence about the terminal-aware target, not a quantitative safety comparison with standard GPE.

Terminal-aware offline screen. Table 14 reports the six-run offline screen for the mixed-distribution TA-1 repair at $Q = 33, T = 100$. Each run uses a calibrated risk limit of 0.01. Pearson measures linear association between predicted and measured candidate progress; it is not a direct action-ranking measure. Exact selected progress measures the true four-step improvement over native IPPO after selection. Positive bootstrap lower bounds support positive mean progress. Selection measures intervention frequency, not performance.

Table 14: Six-run TA-1 offline screen at $Q = 33, T = 100$. Bootstrap lower is the lower confidence endpoint for mean selected progress.

Run	Pearson	Exact selected	Selection	Bootstrap lower
1	0.287	0.000787	13.28%	0.000539
2	0.277	0.000671	13.54%	0.000442
3	0.293	0.000685	14.06%	0.000460
4	0.274	0.000709	11.20%	0.000479
5	0.298	0.000669	13.02%	0.000444
6	0.277	0.000686	13.28%	0.000455
Mean	0.284	0.000701	13.06%	N/A (run-wise CI)

Run-wise terminal-aware confirmation. Table 15 expands the aggregate $Q = 33, T = 100$ result. Each run evaluates both rules on the same 5,000 episodes; native IPPO solves nine episodes in that bank. Both rules have positive mean return gain in every run, but each loses one native success in Run 2. Losses count episodes solved by native IPPO but not by the selector. The six runs reuse the same episode assignments and are not six disjoint environment banks.

Table 15: Run-wise $Q = 33, T = 100$ terminal-aware confirmation on 5,000 paired full episodes per seed. Gains are relative to native IPPO; success counts are out of 5,000.

Run	Guarded gain	Greedy gain	Guarded succ.	Greedy succ.	Guarded loss	Greedy loss
1	+0.04471	+0.06680	9	9	0	0
2	+0.04422	+0.06702	8	8	1	1
3	+0.04694	+0.07035	9	9	0	0
4	+0.04255	+0.06886	9	9	0	0
5	+0.04268	+0.06837	9	9	0	0
6	+0.04437	+0.06739	9	9	0	0
Mean	+0.04424	+0.06813	8.83	8.83	0.17	0.17
Sum over runs	N/A	N/A	53	53	1	1

Success-enriched near-goal confirmation. The near-goal stress test complements the full-episode terminal-aware result by conditioning on decision snapshots in which native IPPO is already close to

success. Table 16 gives the run-wise Direct Completion and GPE results. These $Q = 17$ snapshot experiment is distinct from the $Q = 33, T = 100$ terminal-aware full-episode confirmation above.

Table 16: Six-run $Q = 17$ success-enriched near-goal evaluation. This is a snapshot-level safety stress test with 19,968 near-goal decision snapshots (1,664 environment seeds and 12 snapshots per seed), not a 5,000-episode full-horizon evaluation. Return gains are versus native IPPO; intervals are paired 95% bootstrap CIs over the 19,968 snapshots. A positive interval favors the selector for return. Zero native-success losses is best, and a near-zero upper confidence bound on the loss rate supports preservation.

Scorer	Gain	95% CI	Native losses	Retention
Direct run 1	0.001502	[0.001451, 0.001553]	1	99.992%
Direct run 2	0.001428	[0.001382, 0.001473]	0	100.000%
Direct run 3	0.001105	[0.001057, 0.001151]	1	99.992%
Direct run 4	0.001595	[0.001546, 0.001646]	1	99.992%
Direct run 5	0.001050	[0.001005, 0.001095]	0	100.000%
Direct run 6	0.000928	[0.000887, 0.000970]	1	99.992%
GPE run 1	**0.002154**	**[0.002100, 0.002210]**	**0**	**100.000%**
GPE run 2	**0.002040**	**[0.001988, 0.002093]**	**0**	**100.000%**
GPE run 3	**0.002366**	**[0.002310, 0.002421]**	**0**	**100.000%**
GPE run 4	**0.001912**	**[0.001858, 0.001965]**	**0**	**100.000%**
GPE run 5	**0.002433**	**[0.002377, 0.002489]**	**0**	**100.000%**
GPE run 6	**0.002505**	**[0.002450, 0.002561]**	**0**	**100.000%**

I HORIZON NOTATION AND PROTOCOL DISTINCTION

Here T denotes the maximum number of primitive environment steps in one episode. It is distinct from the GPE prediction and commitment horizon $K = 4$: GPE predicts the local four-step effect and replans after each commitment. The two protocols used in the study are:

Protocol	T_{train} (IPPO)	T_{data} (GPE)	T_{eval}
Original primary	100	100	2000
Horizon-matched attempt†	2000	2000	2000

Thus the original result uses $(T_{\text{train}}, T_{\text{data}}, T_{\text{eval}}) = (100, 100, 2000)$, whereas the attempted horizon-matched protocol was designed to use $(2000, 2000, 2000)$. The latter produced the six IPPO checkpoints, but its scorers did not satisfy the offline promotion rule across all six policies, so no guarded six-policy confirmation was run. Neither protocol makes GPE predict 2,000 steps ahead; its prediction horizon remains $K = 4$. To test the fully matched protocol, we trained six fresh IPPO policies with $T = 2000$. All six runs produced checkpoints, and the $Q = 33$ Direct Completion and GPE pipelines collected fit, calibration, and offline-test data for each policy. Five policies initially trained all twelve scorer models (two targets and six scorer runs); the sixth lacked calibration support. Here, support is the number of calibration contexts whose best base-scored candidate is nonzero. At least 32 such contexts are required to estimate the upper-tail calibration quantile reliably. Enlarging this policy's calibration bank to 24,576 contexts supplied the needed support and allowed its scorers to train. However, the offline promotion condition was not met across all six policies, so we did not run a guarded development or 5,000-episode confirmation. This attempt therefore provides neither positive nor negative closed-loop evidence for GPE with a $T = 2000$-trained policy. The training budget remained about 10 million environment frames in both protocols: this corresponds to roughly 100,000 episodes at $T_{\text{train}} = 100$ and roughly 5,000 episodes at $T_{\text{train}} = 2000$, distinct from the held-out 5,000-episode evaluation bank.

Normalized time feature under extended deployment. The scorer includes one normalized time input, t/T. During the original training/data protocol it is $t/100$; during the $T = 2000$ evaluation it is recomputed as $t/2000$. It therefore remains in $[0, 1]$ in both settings: for example, 0.5 denotes the midpoint of the active episode rather than a fixed number of primitive steps. This does not add a new input or reveal the numeric horizon to the scorer, but its semantic meaning changes (step 50 during

training versus step 1,000 during evaluation). Recomputing the feature avoids out-of-range values such as $t/100 = 10$ at evaluation step 1,000. All evaluated controllers use the same active horizon and time normalization, making their comparison matched under extended deployment. It does not, however, remove the broader distribution shift caused by deploying policies and scorers trained at $T = 100$ for 2,000-step episodes.

†The horizon-matched attempt is retained as a negative screening result and should not be interpreted as closed-loop evidence for or against GPE with a $T = 2000$-trained policy.

J Computational Overhead

Online scorer overhead. Using CPU timing at $Q = 33$, the complete scorer pipeline takes 3.36 ms per decision for greedy GPE and 3.42 ms for guarded GPE, including candidate generation, feature construction, ensemble prediction, and selection. With one decision every four steps, this adds about 1.7 s over a 2,000-step episode. Table 17 reports the corresponding mean episode times: 10.92 s for native IPPO, 12.66 s for greedy GPE, and 12.72 s for guarded GPE. Thus, the measured deployment overhead is 15.96% and 16.49%, respectively; training, loading, and calibration are excluded.

Table 17: CPU deployment overhead at $T = 2000, Q = 33$. Values are mean wall-clock time per episode over six matched assignments; overhead is relative to native IPPO and lower is better.

Controller	Mean wall-clock time/episode	Overhead vs. IPPO
Native IPPO	10.92 s	—
Greedy GPE	12.66 s	+15.96%
Guarded GPE	12.72 s	+16.49%

Table 18: End-to-end elapsed wall-clock time (hours) for each complete $Q = 33$ Transport horizon evaluation: six scorer runs, 5,000 matched episodes per run, and native IPPO, Direct Completion, greedy GPE, and guarded GPE controllers. Values are campaign elapsed time, not per-seed time or summed compute time.

Setting	Wall-clock time (h)
T=200, Q=33	0.68
T=500, Q=33	1.69
T=1000, Q=33	5.01
T=2000, Q=33	8.41

K Qualitative Transport examples

These examples illustrate individual matched rollouts; aggregate evidence is reported in the main results Table. The first figure is a success-conversion example: all controllers begin from the same state, native IPPO fails, and both GPE decision rules reach the goal. The second figure shows the complementary decision-rule behavior in five matched rollouts: greedy GPE accepts residuals that reach success while the more conservative guarded rule remains with, or close to, native behavior. These examples explain how the rules can differ, but do not establish that greedy is generally preferable; the aggregate return and native-success-loss comparisons in Table 2 provide that evidence. In the first figure, the upper panels show the final package and agent trajectories, the lower-left panel plots package-centre distance to the goal (lower is better), and the lower-right zoom verifies terminal body overlap. The GPE trajectories reduce this distance until the package enters the goal, whereas native IPPO moves away from the goal and does not satisfy the terminal condition. In the second figure, each top panel shows one matched trajectory and each lower panel shows its package–goal distance over time; the success marker identifies the step at which greedy GPE reaches the terminal condition.

Policy-intervention traces. Figure 11 replay the same matched success-conversion episode from the primary $Q = 33, T = 2000$ protocol. They make explicit that GPE does not replace or retrain

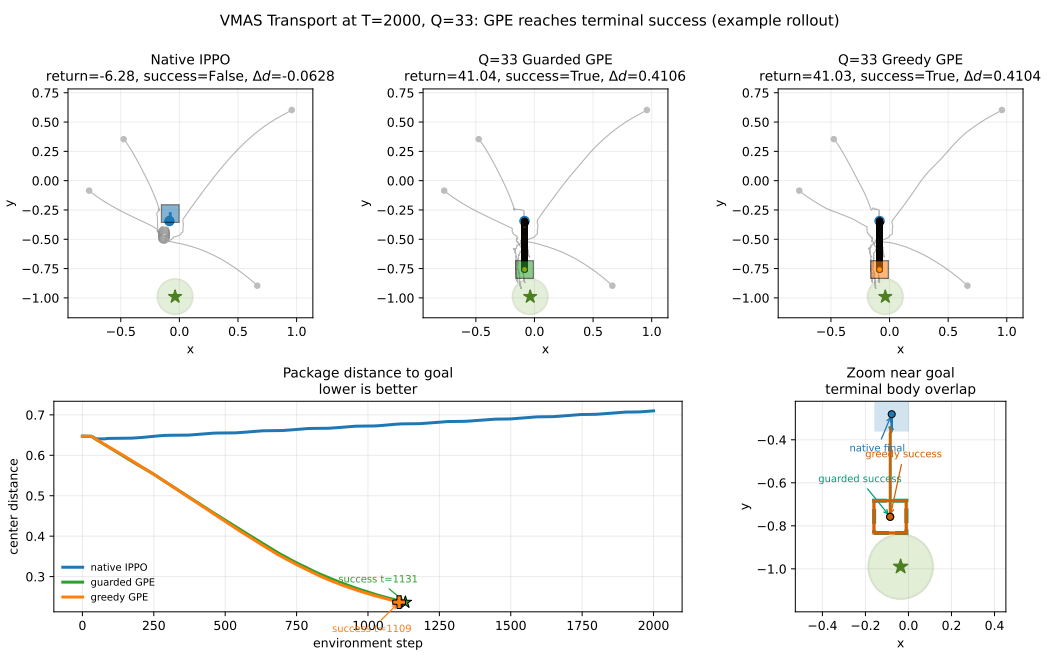

Figure 9: Qualitative $T = 2000$, $Q = 33$ Transport success conversion. All controllers begin from the same state. Native IPPO (left) fails; guarded GPE (middle) and greedy GPE (right) both reach terminal success. The lower-left panel plots package-centre distance to the goal (lower is better), and the lower-right zoom verifies terminal body overlap.

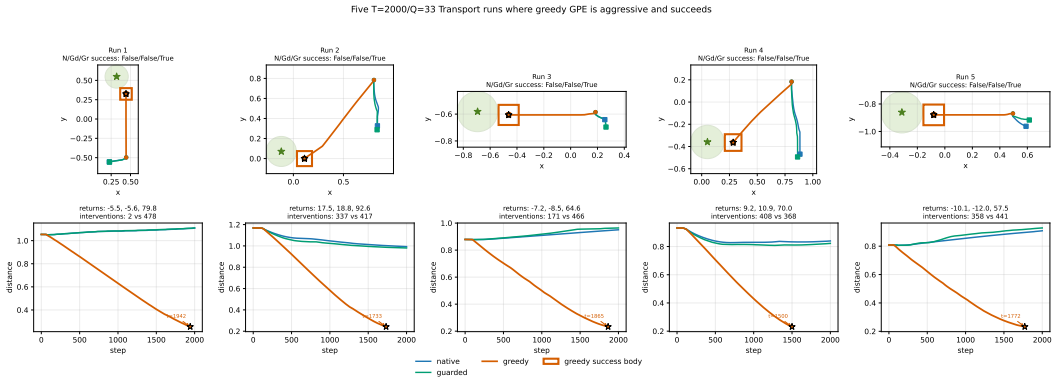

Figure 10: Five matched $T = 2000$, $Q = 33$ Transport rollouts in which greedy GPE reaches terminal success while guarded GPE remains conservative. The figure illustrates the decision-rule trade-off; aggregate results are reported in the main paper.

IPPO. At every primitive step, IPPO supplies the native joint action; every four steps the selector may add a bounded joint residual, which is then committed for the next four live steps. In both figures, the upper-left panel shows the package path, and the upper-right panel shows package–goal distance. The blue native trajectory moves away from the goal and fails, whereas the green guarded and orange greedy trajectories reduce distance until terminal success (dashed vertical markers). The lower-left panel gives the magnitude of the selected eight-dimensional joint residual: greedy intervenes almost continuously in this example, while the guarded rule has more zero-residual periods. The lower-right panel decomposes the selected residual into its four two-dimensional agent corrections. A value of zero means that the corresponding agent follows its native IPPO action exactly. These are illustrative controller traces; the aggregate six-seed comparisons establish the reported return–native-success-retention trade-off.

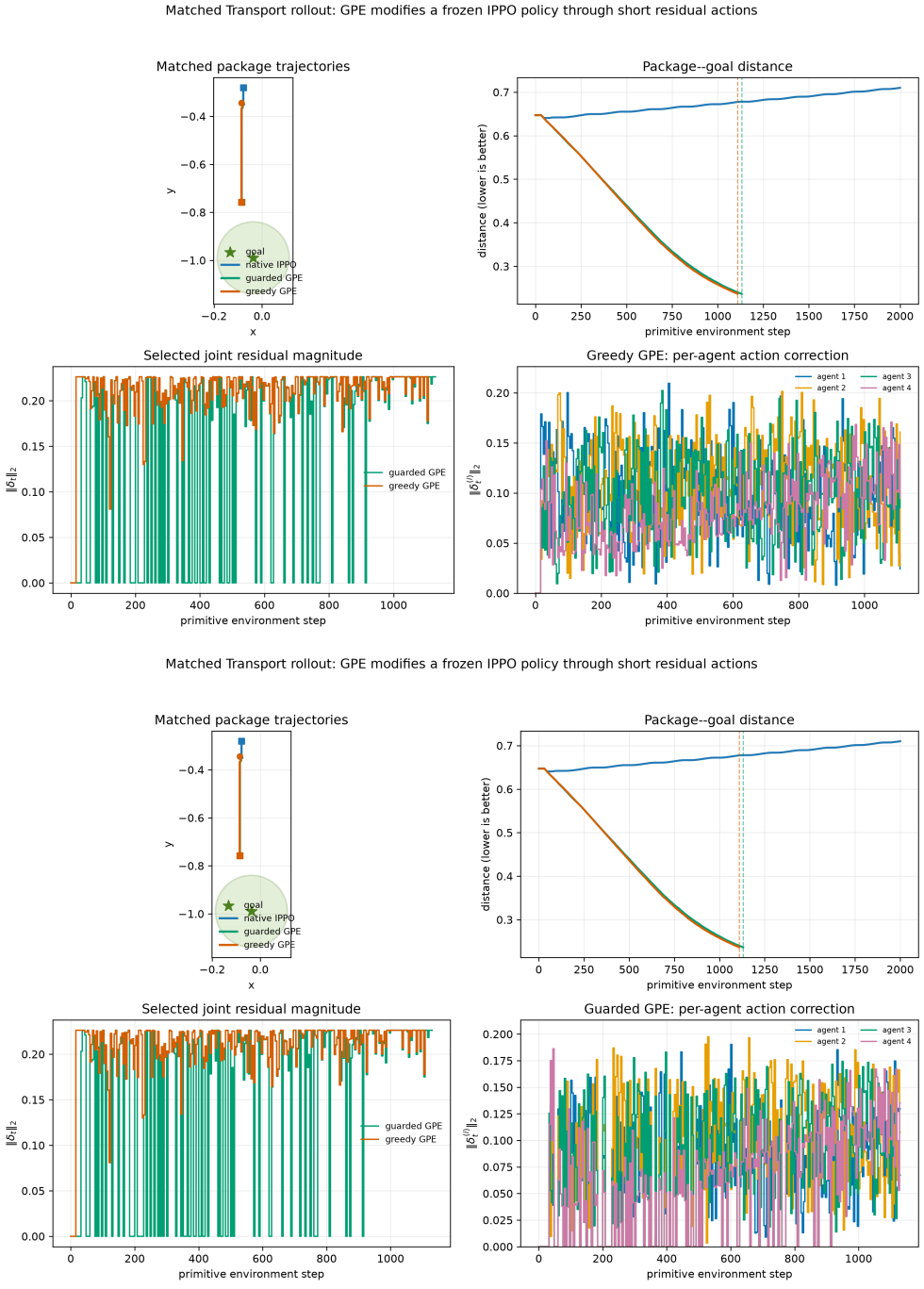

Figure 11: Policy-intervention traces for the same matched $Q = 33, T = 2000$ Transport rollout. Top: greedy GPE. Bottom: guarded GPE. Native IPPO moves away from the goal, while both GPE rules apply bounded four-agent residual corrections and reach terminal success. The lower-left panels show joint residual magnitude; the lower-right panels show the four per-agent correction magnitudes.

L LIMITATIONS AND FUTURE WORK

GPE requires a task-specific prediction target that captures progress over the residual commitment horizon. In Transport, package–goal distance provides a simple and informative quantity, but an equally suitable target may not be available in tasks with sparse rewards, multiple competing objectives, or delayed effects. The present evidence therefore establishes the method only for the evaluated IPPO Transport setting.

The short-horizon target also does not fully capture the consequences of an intervention. A residual that improves four-step goal progress may change later coordination or prevent an eventual task success. In particular, the Transport terminal condition depends on package-goal overlap and orientation, which are not completely described by centre-distance progress. Although the terminal-aware experiments investigate this failure mode, they use a separate evaluation protocol and do not establish a quantitative safety improvement over standard GPE.

A further limitation is that the guarded selector relies on predictions calibrated using a fixed offline distribution. Selecting residuals changes the states subsequently visited by the controller, so calibration does not provide a formal guarantee of positive progress, policy improvement, or native-success preservation during closed-loop deployment. The exact native-action fallback prevents action modification when the gate rejects a candidate, but it cannot undo the downstream effects of previously accepted interventions.

Finally, GPE requires matched counterfactual candidate rollouts, which may be expensive outside simulation. The fixed $Q = 33$ candidate bank is also not adaptive and may become costly as the number of agents grows. Developing data-efficient and adaptive candidate-generation methods is an important direction for future work. Future work should test this idea with additional policies and tasks, develop adaptive candidate banks, and study terminal-aware selection under the same deployment horizon