LEARNING WHEN TO LISTEN: CLOSED-LOOP LLM ADVISORY FOR COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can serve as tactical advisors in cooperative multi-agent reinforcement learning (MARL), yet existing advisory channels fix the adoption decision outside the executing agents and provide no feedback on whether advice was followed or effective. We organize the interface around three dimensions: *who* decides adoption (D1), at what *semantic granularity* (D2), and what *feedback* the advisor receives (D3). We propose Closed-Loop Gated Advisory (CLoGA), a lightweight adapter for frozen recurrent executors that resolves all three: (i) semantic gating decomposes each directive into heads and lets each agent modulate adoption per dimension through learned, state-conditioned gates; (ii) a feedback channel returns per-head gate coefficients, relative reward, and situation drift to the frozen advisor for in-context revision without weight updates; and (iii) corrections are bounded residuals at the action readout rather than writes into recurrent memory. The system assumes a centralized advisory service at execution while preserving local recurrent backbones; this is an explicit channel assumption, not strictly communication-free decentralized execution. We establish conditional exact readout recovery and a readout-state deviation bound; both are pointwise results at a fixed history, not guarantees about joint-policy return. On SMAC and SMACv2 with MAPPO and HAPPO executors, CLoGA raises the weakest base by +0.25 and the imperfect-tier average by +0.09, while ungated injection of the same advice degrades three of four controlled maps; in a gate-type$\times$feedback comparison, per-head gating converts feedback into a $+0.024$ gain against $+0.006$ for an otherwise identical learned scalar gate, suggesting that granularity and observability interact.

1 INTRODUCTION

An LLM advisor can supply cooperative multi-agent reinforcement learning (MARL) agents with tactical guidance they would not otherwise discover (Cao et al., 2025; Sun et al., 2024), but a directive may be useful in one tactical dimension and mismatched in another. A formation suggestion can be relevant while its maneuver order is not. Treating the directive as an indivisible instruction risks disrupting a capable base policy, and none of the existing channels observes which part of its advice was followed. We study how an executor can selectively use such imperfect advice and expose its response to the advisor (Figure 1).

We organize the interface around three dimensions: *who* decides adoption (D1), its *semantic granularity* (D2), and *what feedback* the advisor receives (D3). Advisor-side weighting and scheduled overrides place adoption outside the executor; other interfaces regularize actions, augment observations, or select skills and query times (Table 1). Our central thesis is that these three dimensions jointly determine the value of an advisory channel, and that no existing method addresses all three. D1 matters because only the executor can judge whether advice fits its current situation; D2 matters because a directive can be right in one tactical dimension and wrong in another, so an indivisible adoption weight forces agents to accept a mismatched component to receive a useful one; D3 matters because an advisor that never observes adoption or outcome cannot correct itself across rounds. Table 1 summarizes the resulting gap. These considerations motivate executor-side filtering, but do not establish that one granularity is universally optimal.

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.

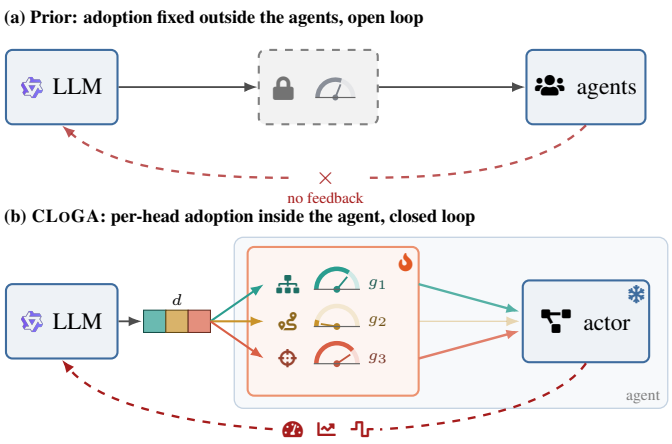


Figure 1: (a) An open-loop channel fixes a single adoption weight outside the executing agents with no feedback. (b) CLoGA places a learned, per-head gate inside each agent; adoption $\bar{g}_{1:3}$, reward Δr, and drift ρ return to the advisor.

A secondary finding sharpens this picture. Across the controlled suite, D2 and D3 behave as *interacting* rather than independent axes: the 2×2 comparison of §5.3 measures per-head gating converting feedback into a $+0.024$ gain against $+0.006$ for an otherwise identical learned scalar gate. We report this as an observation of the interface we test, not as a general law of advisory design.

Closed-Loop Gated Advisory (CLoGA) attaches a lightweight adapter to frozen recurrent actors. Learned per-head gates scale bounded semantic residuals at the action readout, while the uncorrected recurrent state is carried forward. A deterministic feedback channel returns gate coefficients, relative reward, and situation drift to the frozen LLM. The advisor accesses a centralized summary during execution; this is an explicit advisory-channel assumption, not strict communication-free decentralized execution.

Our contributions are (1) an executor-side, semantically partitioned advisory interface with conditional exact recovery and a readout-state deviation bound; (2) a feedback channel that makes per-head gate coefficients and outcomes visible to the advisor; and (3) an evaluation on SMAC and SMACv2 with MAPPO and HAPPO executors, including a 2×2 design that varies adoption granularity and feedback separately. The complete method raises the weakest base by +0.25 and the imperfect-tier average by +0.09, while ungated advice degrades three of four controlled maps.

2 Related Work

Cooperative MARL. Value factorization (Rashid et al., 2018) and policy-gradient methods (Yu et al., 2022; Kuba et al., 2022) provide backbones on SMAC (Samvelyan et al., 2019) and SMACv2 (Ellis et al., 2023). MAPPO uses PPO with centralized training and decentralized actors; HAPPO uses sequential agent-policy updates. We freeze these pretrained executors and train their advisory pathways and fresh critics, rather than changing the backbone learning algorithm.

LLM guidance and feedback. LLMs can design rewards, process information, or guide decisions (Cao et al., 2025; Sun et al., 2024; Zhuang et al., 2025). Feedback-aware LLM–RL is not new: Eureka iteratively improves reward code in context (Ma et al., 2024), and eSpark improves MARL action-pruning functions using policy feedback (Liu et al., 2025). Our focus is more specific: executor-side semantic residual gating and feedback about the executor's gate coefficients, rather than the first use of an LLM feedback loop.

Selective adoption. Uncertainty weighting (Shoaeinaeini & Harrison, 2025), scheduled override (Siedler & Gemp, 2025), observation augmentation (Jain & Grossmann, 2025), instruction regularization (Hu & Sadigh, 2023), and budgeted advising (Torrey & Taylor, 2013; da Silva et al., 2020) offer different ways to use external guidance. Skill routing (Hösch et al., 2026) and query

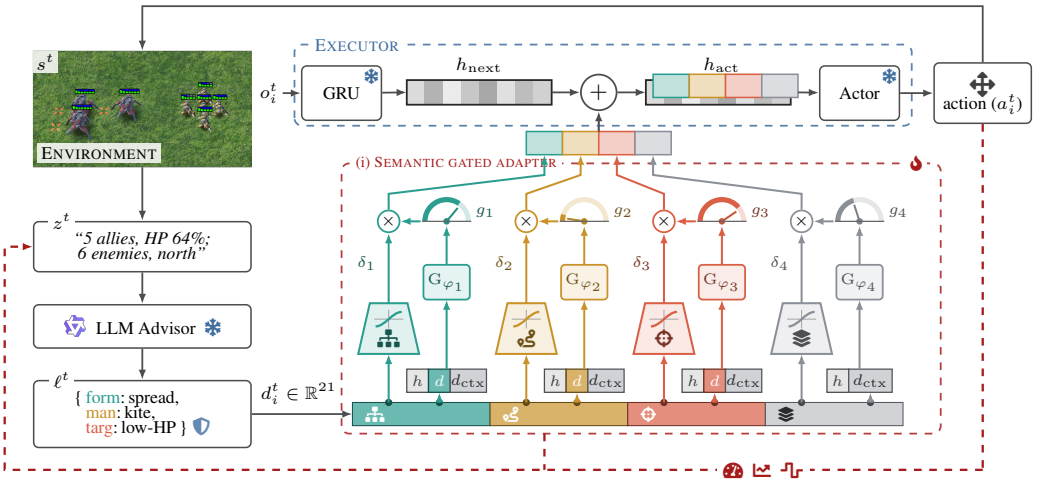


Figure 2: CLoGA overview (snowflake = frozen; flame = trained; blue/red dashed boxes = frozen executor / trained adapter). **Left**: every κ steps a deterministic summarizer renders the state into text and the frozen LLM (Qwen) returns a JSON directive, encoded per agent as $d_i^t \in \mathbb{R}^{21}$. **(i)**: the directive splits into four semantic lanes; each produces a bounded residual δ_k scaled by a learned gate $g_k = v_i^t \cdot \sigma(\mathrm{G}_k[h \mid d \mid d_{\text{ctx}}])$ conditioned on the agent state h_{next}, its directive slice, and the context slice (the context head reads $[h \mid d_{\text{ctx}}]$ only). The gated residuals write disjoint blocks of h_{act}; h_{next} carries forward unmodified. **(ii)**: the crimson path returns feedback with the next query—per-head adoption $\bar{g}_{1:3}$, reward trend Δr, and state drift ρ (the three icons).

Table 1: Adoption dimensions. D1: who decides adoption (executor-side, learned). D2: at what semantic granularity (per-dimension heads). D3: whether the advisor observes executor adoption coefficients and outcome. †: multi-agent.

Method	D1	D2	D3
Uncert.-aware (2025)	×	×	×
Obs.-augment (2025)	impl.	×	×
Sched. override† (2025)	×	×	×
Fixed KL† (2023)	×	×	×
Skill routing† (2026)	×	×	×
Query gating† (2025)	×	×	×
CLoGA† (ours)	✓	✓	✓

gating (Li et al., 2025) address consultation or routing decisions. Table 1 distinguishes these from the specific combination of executor-side semantic gates and adoption telemetry studied here; D3 does not assert that prior work lacks all forms of outcome feedback.

3 PRELIMINARIES

Task and base policy. The underlying cooperative task is a Dec-POMDP (Oliehoek & Amato, 2016), $\mathcal{M}=\langle \mathcal{N}, \mathcal{S}, \{\mathcal{O}_i\}, \{\mathcal{A}_i\}, P, R, \gamma \rangle$, with N agents, global state s^t, local observations o_i^t, actions $a_i^t \in \mathcal{A}_i$, transition law P, shared reward r^t, and discount γ. A frozen recurrent actor computes $h_{\text{next},i}^{\,t} = f_\theta(o_i^t, h_{\text{next},i}^{\,t-1}) \in \mathbb{R}^H$. Its unmodified action distribution is $\pi_{0,i}(a_i^t \mid h_{\text{next},i}^{\,t})$; π_0 denotes the pretrained joint *base policy*. Base competence is its evaluation win rate, not a separate training parameter.

Information boundary. We augment this task with an external advisory service. During rollout, including evaluation, a deterministic summarizer reads the current global state and a frozen LLM emits a shared directive. The encoder combines it with state-derived context and agent applicability. Actors retain local recurrent backbones, but their corrected action distributions also depend

on this centralized channel. A centralized critic is used for training, not action selection. The full system therefore assumes a centralized information/communication service at execution; it is not a solution under strictly local, communication-free Dec-POMDP policies. Advisory-arm comparisons share the summary interface, whereas improvement over the frozen base measures the whole added channel.

Evaluation domain. SMAC is cooperative StarCraft micromanagement: agents control allied units against a built-in opponent using local observations and available movement or combat actions. The team seeks to defeat the enemies within an episode limit; win rate is the fraction of evaluation episodes won. Map names identify combat configurations (e.g., `3m` is a three-Marine team). SMACv2 varies compositions and starting positions across episodes. The vocabulary below is designed for these combat tasks, not assumed to transfer unchanged to other domains.

4 METHOD

The backbone and action head remain frozen; only the advisory adapters, gates, and a fresh critic train. Figure 2 separates the transient action correction from recurrent memory and shows the feedback path.

4.1 ADVISORY CHANNEL

A rule-template summarizer renders the global state as text z^t. With periodic interval $\kappa = 8$ and the refresh conditions in Appendix A.1, the LLM returns a directive ℓ^t in a fixed, human-designed JSON vocabulary $\mathcal{V}$: formation (3 values), maneuver (4), target priority (4), focus fire, validity duration, confidence, and a logged reason. Formation concerns spreading or compacting units; maneuver concerns advance, retreat, kite, or hold; target priority selects an enemy criterion. A legality checker maps malformed or inconsistent output to a no-op. Between refreshes, the active directive is reused.

Why the interval is κ=8. The refresh period trades advisory freshness against advisor cost and feedback-window stability. Three measured facts constrain it. First, the advisor itself chooses durations averaging 8.0–10.2 steps (Appendix D.7)—the low end of its permitted $[4, 32]$ range—so a much larger κ would routinely refresh an expired directive. Second, freshness and drift (dims 13–15) are defined relative to κ_ℓ, so refreshing on the same order keeps staleness comparable across maps. Third, feedback reports a window *mean* and needs enough steps to be stable. Measured refresh rates are $\sim$3.0 per episode on `3m` and $\sim$11.3 on `3s5z_vs_3s6z`, with cache hits of 4.7–36.7% (Appendix A.8). This is a rationale, not a sensitivity result: a sweep over $\kappa \in \{4, 8, 16\}$ is the natural control and we have not run it.

For example, the `5m_vs_6m` transcript in Appendix E changes from `advance` with `nearest_enemy` to `kite` with `lowest_hp_enemy` and focus fire after engagement. The executor still selects primitive actions; the directive is guidance, not a binding sequence. Tables S1–S2 give the complete schema and 21-dimensional encoding.

A concrete loop. At step 0 of a `5m_vs_6m` run, both sides ranged and 13.98 apart, the advisor returns `maneuver=advance`, `target_priority=nearest_enemy`, `focus_fire=false`, `duration=16`. The executor encodes this per agent and the gates decide adoption per head. At step 8 the advisor receives *"mean adoption 0.51; situation drift 0.73; enemy survivors* -1*; center distance closed from 13.98 to 6.07. Per-head adoption: formation 0.43, maneuver 0.68, targeting 0.57"* and revises to `kite` with `lowest_hp_enemy` and focus fire at `duration=8`. Three fields change in one exchange, with no advisor weight update (Appendix E).

What the advisor recommends. Across 2 089 legal directives from the seed-1 runs, output is map-adaptive: `kite` dominates on `2c_vs_64zg` (91%, ranged Colossi against a melee swarm) and leads on every marine-mirror map, while `advance` dominates on melee-heavy `3s5z_vs_3s6z` (85%). Focus fire is near-universal where kill-securing is pivotal (`corridor` 100%, `MMM2` 92%) and more selective on open fights (`3m` 66%). Stated confidence is uniformly high (0.86–0.89) with little map variation—the advisor is *not* well calibrated about when its advice will be adopted, which is the premise for placing the gate in the executor (Appendix D.7).

Each $d_i^t \in \mathbb{R}^{21}$ contains tactical dimensions 0–10 and context dimensions 11–20. Metadata include advisor-reported confidence c_ℓ, freshness $\phi^t = 1 - (t - t_{\text{issue}})/\kappa_\ell$, and state drift $\rho^t = \min(1, \|s^t -$

$s^{t_{\text{issue}}}\|/\|s^{t_{\text{issue}}}\|$). Here t_{issue} is issuance time and κ_ℓ the directive's validity duration; drift uses the global-state representation. These are practical indicators of stated reliability, age, and state mismatch, not a proven sufficient or optimal metadata set. Applicability, role, and unit type make the encoding agent-specific.

4.2 Semantic Multi-Head Gated Adapter

The directive is partitioned into three tactical heads and one context head:

$$d_i^t = \big[d_{i,S_1}^t;\ d_{i,S_2}^t;\ d_{i,S_3}^t;\ d_{i,S_{\text{ctx}}}^t\big], \tag{1}$$

where S_1, S_2, and S_3 correspond to formation, maneuver, and targeting. Aligning gates with the vocabulary provides a head-named feedback interface. With $K = 4$, each head writes a disjoint block of size H/K. The validity mask $v_i^t = d_i^t[\text{legal}]d_i^t[\text{appl}] \in \{0, 1\}$ disables illegal or inapplicable advice. Head k has adapter parameters ω_k and gate parameters φ_k:

$$\delta_{k,i}^t = s \cdot \tanh\big(\text{Ad}_{\omega_k}(d_{i,S_k}^t)\big) \ \in \mathbb{R}^{H/K}, \quad \|\delta_{k,i}^t\|_\infty \le s, \tag{2}$$

where $s = 1$ is the residual amplitude limit. Tactical gates read the recurrent state, their own directive slice, and context:

$$g_{k,i}^t = v_i^t \cdot \sigma\Big(\text{G}_{\varphi_k}\big([{h_{\text{next}}}_{,i}^{\,t};\ d_{i,S_k}^t;\ d_{i,S_{\text{ctx}}}^t]\big)\Big) \ \in [0, 1]. \tag{3}$$

The context gate reads $[h_{\text{next}}; d_{S_{\text{ctx}}}]$ only. The frozen action head receives

$$ {h_{\text{act}}}_{,i}^{\,t} = {h_{\text{next}}}_{,i}^{\,t} + \Big\|_{k=1}^{K} \big(g_{k,i}^t \cdot \delta_{k,i}^t\big), \tag{4}$$

with $\|$ denoting block concatenation. Memory carries h_{next}, not h_{act}: the residual is not directly written into future recurrent memory. This is deliberate—writing residuals into the recurrence would let them propagate through the recurrent dynamics, so one directive's effect would depend on how long the memory retains it. Keeping the correction at the readout makes each directive's influence bounded at the step it applies. Actions can nevertheless change future observations, so cross-step effects remain. Semantic partitions describe residual inputs; they do not by themselves prove disentangled behavioral effects.

Proposition 1 (Conditional exact readout recovery)**.** *At a fixed agent history, if $v_i^t = 0$ or all $g_{k,i}^t = 0$, then ${h_{\text{act}}}_{,i}^{\,t} = {h_{\text{next}}}_{,i}^{\,t}$ and the action distribution equals the base policy evaluated at that history. The same holds at zero-residual initialization and for failures converted to a no-op.*

Proposition 2 (Bounded readout-state deviation)**.** *For the disjoint H/K-dimensional residual blocks of Eq.* (4)*, $\|{h_{\text{act}}}_{,i}^{\,t} - {h_{\text{next}}}_{,i}^{\,t}\|_2 \le s\sqrt{H/K}\,\|g_i^t\|_2$.*

Proofs are in Appendix B. These are pointwise structural results, not a joint-policy value bound or monotonic-return guarantee. With valid advice, a finite sigmoid need not reach zero; soft suppression differs from exact hard-mask recovery. We distinguish coefficients g_k from injection magnitudes $\|g_k\delta_k\|_2$ and action-distribution changes.

4.3 Closed-Loop Feedback

At each refresh, the advisor receives three deterministic signals. Let W_m be the set of agent–time pairs accumulated in advice window m; $|W_m|$ counts those pairs. Mean tactical gate coefficients are

$$\bar{g}_{k,m} = \frac{1}{|W_m|} \sum_{(i,t)\in W_m} g_{k,i}^t, \quad k \in \{1, 2, 3\}. \tag{5}$$

The context gate is excluded because it does not correspond to a tactical recommendation. Relative reward Δr_m compares the window's per-step reward rate with an episode-local EMA ($\alpha = 0.3$, 5% dead-band); ρ_m summarizes situation drift. A template renders, for example, *formation adopted ($\bar{g}_1 = 0.72$), maneuver ignored ($\bar{g}_2 = 0.05$), targeting adopted ($\bar{g}_3 = 0.81$); reward above recent average; drift 0.18*. These labels describe coefficients, not verified causal effects. Feedback is part of the summary and cache key, allowing in-context revision without LLM weight updates.

Table 2: Main results. Trained arms report last-5 win rate over 3 seeds (mean ± s.d.); Frozen MAPPO reports five evaluation points of the same checkpoint. (a) Baselines on four controlled maps; **bold**/underlined mark best/second. (b) Eight-map breadth of the complete method, grouped by frozen-base competence; Δ is relative to Frozen MAPPO (**bold**: $\Delta \geq 0.03$), and the bottom row gives CLoGA's mean tactical adoption $\bar{g}^{\mathrm{tac}}$ (per-head values in Figure 4b).

(a) Controlled baseline comparison

Method	3m	3s5z_vs_3s6z	MMM2	corridor	Mean
Frozen MAPPO	$0.740_{\pm.031}$	$\underline{0.900}_{\pm.022}$	$\underline{0.930}_{\pm.019}$	$0.980_{\pm.009}$	0.888
State-only	$0.762_{\pm.019}$	$0.891_{\pm.016}$	$0.917_{\pm.020}$	$0.975_{\pm.008}$	0.886
InstructPPO	$0.913_{\pm.025}$	$0.874_{\pm.028}$	$0.921_{\pm.019}$	$0.973_{\pm.010}$	0.920
Exact concat	$\underline{0.941}_{\pm.022}$	$0.868_{\pm.024}$	$0.914_{\pm.017}$	$\underline{0.981}_{\pm.008}$	$\underline{0.926}$
CLoGA (ours)	$\mathbf{0.993}_{\pm.008}$	$\mathbf{0.933}_{\pm.024}$	$\mathbf{0.958}_{\pm.014}$	$\mathbf{0.996}_{\pm.005}$	**0.970**

(b) Eight-map breadth and base-policy preservation
Imperfect tier (base < 0.95)

	3m	5m_vs_6m	3s5z_vs_3s6z	MMM2	*Avg*
Frozen MAPPO	$0.740_{\pm.031}$	$0.890_{\pm.026}$	$0.900_{\pm.022}$	$0.930_{\pm.019}$	0.865
CLoGA	$\mathbf{0.993}_{\pm.008}$	$\mathbf{0.947}_{\pm.021}$	$\mathbf{0.933}_{\pm.024}$	$\mathbf{0.958}_{\pm.014}$	**0.958**
Δ	**+0.25**	**+0.06**	**+0.03**	**+0.03**	**+0.09**
Adoption $\bar{g}^{\mathrm{tac}}$	0.37	0.52	0.40	0.32	0.40

(b) Eight-map breadth and base-policy preservation
Saturated tier (base ≥ 0.95)

	8m_vs_9m	corridor	2c_vs_64zg	2s_vs_1sc	*Avg*
Frozen MAPPO	$0.980_{\pm.011}$	$0.980_{\pm.009}$	$0.990_{\pm.007}$	$1.000_{\pm.000}$	0.988
CLoGA	$0.985_{\pm.007}$	$0.996_{\pm.005}$	$0.998_{\pm.004}$	$1.000_{\pm.000}$	0.995
Δ	+0.01	+0.02	+0.01	+0.00	+0.01
Adoption $\bar{g}^{\mathrm{tac}}$	0.58	0.75	0.84	0.39	0.64

4.4 Training

The $2K$ adapter/gate networks and a fresh critic train via PPO with Adam (lr 5×10^{-4}), clip 0.2, GAE($\gamma = 0.99$, $\lambda = 0.95$), entropy 0.01, four epochs, and 200k environment steps:

$$\mathcal{L} = \mathcal{L}_{\mathrm{clip}}(\theta_{\omega,\varphi}) + c_v\, \mathcal{L}_{\mathrm{value}} - c_e\, \mathcal{H}[\pi_g], \quad \beta_{\mathrm{gate}}{=}0. \tag{6}$$

Here $\theta_{\omega,\varphi}$ denotes advisory-path parameters, $c_v = 0.5$ the value-loss coefficient, and $c_e = 0.01$ the entropy coefficient. Final adapter layers are zero-initialized. Gates learn from task return, without adoption targets, a gate penalty, or KL regularization. HAPPO uses one adapter per actor. Tables S3–S4 give the per-head shapes and common settings; common settings are held fixed, not claimed to be optimal.

5 Experiments

We ask whether the advisory interface improves imperfect bases, which components contribute under the tested controls, and whether the same architecture and training protocol work with another executor and benchmark.

5.1 Setup

Eight SMAC maps use frozen MAPPO checkpoints (win rate 0.74–1.00), grouped into imperfect (base < 0.95) and saturated (base ≥ 0.95) tiers. Trained arms use seeds $\{1, 2, 3\}$ and 200k steps; every 5k steps, evaluation uses 32 episodes. The primary metric is each seed's final-five evaluation mean, summarized across seeds as mean ± s.d. Frozen MAPPO instead reports five evaluations of one checkpoint, not five independent seeds; best evaluation is a secondary selected-maximum statistic. The advisor is local Qwen3.5-9B (bfloat16, temperature 0), with hash caching and no-op fallback.

Table 3: Controlled arms, including two scalar controls for the gate × feedback design. Comparisons share the Qwen advisor where applicable; feedback and trajectories may change the advice received.

Arm	Trainable / advice	Comparison target
Frozen MAPPO	none	checkpoint competence
State-only	adapter, no advice	extra capacity
w/ fixed gate	sem. adapters, g_k=0.5	fixed vs. learned adoption
Scalar / no FB	tied gate, no feedback	granularity without feedback
Scalar / full FB	tied gate + feedback	complete-interface comparison
w/o gate	sem. adapters, g_k=1	necessity of gating
w/ shuffled advice	sem. gate, unrelated advice	state–advice alignment
w/o feedback	sem. gate, no feedback	closing the loop
w/ shuffled feedback	sem. gate, unrelated feedback	correct feedback
InstructPPO	full actor, KL to prior	instruction regularizer
Exact concat	directive-logit projection	injection without gating
CLoGA	sem. gate + feedback	complete method

Table 4: Mechanism ablations on four controlled maps (3 seeds; last-5 win rate). **Bold**/underlining mark best and second in the measured block.

Arm	3m	3s5z_vs_3s6z	MMM2	corridor	Mean
Frozen MAPPO	0.740 ±.031	<u>0.900</u> ±.022	<u>0.930</u> ±.019	0.980 ±.009	0.888
Single-axis ablations					
w/ fixed gate	0.941 ±.015	0.872 ±.022	0.921 ±.017	0.978 ±.009	0.928
w/o gate	0.903 ±.028	0.821 ±.031	0.884 ±.026	0.952 ±.014	0.890
w/ shuffled advice	0.768 ±.024	0.842 ±.027	0.895 ±.021	0.961 ±.012	0.867
w/ shuffled feedback	0.949 ±.017	0.879 ±.023	0.917 ±.019	0.981 ±.008	0.932
Gate granularity × feedback (2 × 2)					
Scalar / no FB	0.945 ±.016	0.885 ±.020	0.924 ±.018	0.982 ±.008	0.934
Scalar / full FB	0.951 ±.015	0.891 ±.018	0.936 ±.017	0.984 ±.008	0.941
w/o feedback (per-head)	<u>0.966</u> ±.013	0.897 ±.019	<u>0.930</u> ±.016	<u>0.989</u> ±.006	<u>0.946</u>
CLoGA (per-head, full FB)	**0.993** ±.008	**0.933** ±.024	**0.958** ±.014	**0.996** ±.005	**0.970**

The controlled suite comprises `3m`, `3s5z_vs_3s6z`, `MMM2`, and `corridor`, spanning homogeneous, heterogeneous, asymmetric, and narrow-passage combat. Arms start from matched checkpoints and share the advisor and summarization interface when applicable (Table 3). Additional experiments train the same adapter architecture on five HAPPO/SMAC settings and three MAPPO/SMACv2 5vs5 races; these test architectural reuse with training, not zero-shot transfer of one adapter checkpoint. The learned scalar-gate control uses the same four-map suite and seed-level protocol, with the same budget and adapter/gate architecture as the per-head arms (see §5.3).

5.2 Main Results

Table 2(a) reports a controlled-suite mean of **0.970**, versus Exact concat's 0.926 and InstructPPO's 0.920. On the eight-map evaluation, the weakest base (`3m`, 0.740) reaches 0.993, closing about 97% of the gap to perfect win rate; the imperfect-tier mean rises from 0.865 to 0.958 (rounded gain +0.09). Saturated-tier means remain non-decreasing in these evaluations, with rounded gains +0.00–+0.02. Appendix Figure S1 visualizes the same map-level results.

Per-head coefficients differ by task (Figure 4a–b; trajectories in Appendix Figure S8). On `3m`, targeting has $g_3 = 0.58$ versus formation's $g_1 = 0.22$; on `5m_vs_6m`, maneuver leads at $g_2 = 0.62$. These are differences in *gate openness*, not ratios of behavioral influence: residual magnitude and the frozen action head also matter. Saturated-tier mean openness 0.64 exceeds the imperfect tier's 0.40 despite smaller gains, so openness alone measures neither benefit nor agreement with the base policy.

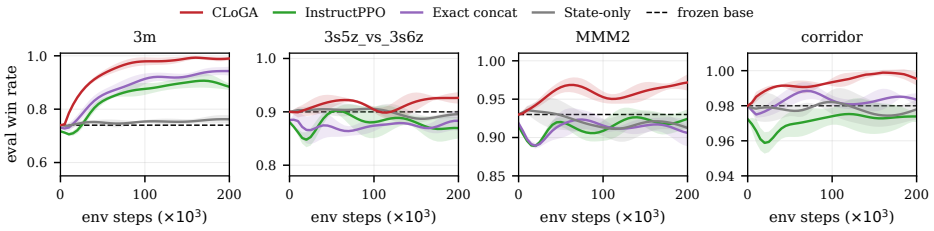


Figure 3: Evaluation curves on the four controlled maps (mean ± s.d., 3 seeds; dashed: frozen base). Zero-initialization matches the checkpoint initially; subsequent curve behavior is empirical, not a monotonicity guarantee.

5.3 Controlled Comparison

Capacity and learned gating. The State-only arm—which consumes the recurrent state but no directive, under a matched training budget—reaches 0.886 versus the frozen base's 0.888, offering little mean gain. Fixed gates reach 0.928, 3.8 points above ungated injection (0.890); the complete method adds 4.2 more. Ungated injection falls below the base on three maps, including `3s5z_vs_3s6z` (0.821 vs. 0.900) and `MMM2` (0.884 vs. 0.930). These comparisons rule out capacity and show that *some* learned, state-conditioned adoption is necessary, but they do not separate learned gating from per-head granularity: the fixed- and no-gate arms change both at once. The 2×2 design below isolates the second axis.

Gate granularity $\times$ feedback (2×2). The scalar (SC) control retains all four residual adapters and all four gate networks with their original inputs, but broadcasts a single state-conditioned coefficient to every residual block:

$$g^t_{\mathrm{sc},i} = v^t_i \cdot \frac{1}{K} \sum_{k=1}^{K} \sigma\big(\mathrm{G}_{\varphi_k}(\cdot)\big), \qquad {h_{\mathrm{act}}}^t_{,i} = {h_{\mathrm{next}}}^t_{,i} + \Big\|_{k=1}^{K} \big(g^t_{\mathrm{sc},i} \delta^t_{k,i}\big).$$

The aggregation averages the per-head gate *outputs* rather than their pre-sigmoid logits, so $g_{\mathrm{sc}} \in [\min_k g_k, \max_k g_k]$: a high gate on one head cannot raise adoption on the others, and a closed head drags the shared coefficient down. Structurally, SC and per-head (PH) differ only in whether the adoption coefficient is shared across heads; the residual adapters, gating inputs, hard mask, and no-op path are identical. Crossing SC/PH with full/no feedback gives four cells; Table 4 reports all four. Under no feedback, SC reaches 0.934 and PH 0.946—a granularity gap of Δ_{no}=0.012. Under full feedback, SC reaches 0.941 and PH 0.970, giving Δ_{full}=0.030. Feedback thus moves the per-head interface by $+0.024$ but the scalar interface by only $+0.006$. With a shared coefficient the advisor's feedback necessarily reports one number for three tactical heads and cannot address which dimension to revise—the asymmetry in the two feedback gains is consistent with the per-head telemetry being what makes the loop informative, though this comparison is descriptive rather than a fixed-advice counterfactual.

Two caveats constrain this reading. First, matched checkpoints and budgets do not hold the advice stream fixed: different policies induce different states, hence different summaries, directives, and feedback, in every cell, so the interaction is an interface-level comparison rather than a fixed-advice counterfactual. Second, with three seeds we treat the ordering as descriptive; the smallest per-map gaps lie within the seed spread.

Misaligned advice. Shuffling advice lowers the suite mean to 0.867: 10.3 points below the complete method and 2.1 below the frozen base. Mean tactical gate openness falls to about 0.12, but the remaining correction still permits degradation. This is attenuation, not automatic elimination of every harmful directive or exact recovery under a nonzero soft gate.

5.4 Mechanism Analysis

Figure 4a–b shows task-dependent gate profiles without a diversity objective: `corridor` ($\bar{g}^{\mathrm{tac}} = 0.75$) and `2c_vs_64zg` (0.84) have greater mean openness than `3m` (0.37). The profiles describe

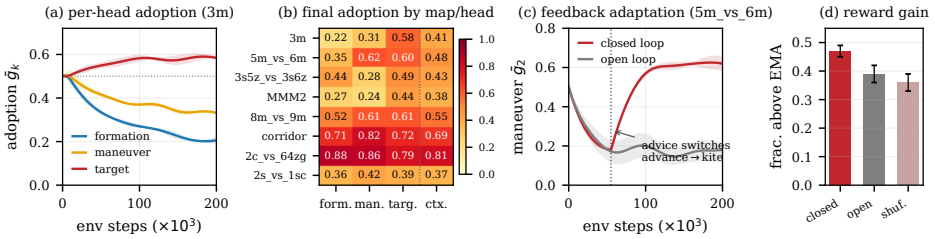


Figure 4: Adoption diagnostics (3 seeds). (a) Head differentiation from uniform init on `3m`. (b) Final adoption by map/head. (c) Advice/gate change with feedback on `5m_vs_6m`. (d) Fraction of advice windows above reward EMA.

Table 5: Generality (3 seeds; last-5 win rate). **Bold**: $\Delta \geq 0.04$. Adapter architecture, advisor, and protocol identical to Table 2.

Setting	Base	Last-5	Δ	Best	$\bar{g}^{\text{tac}}$
HAPPO executors (SMAC)					
5m_vs_6m	0.75 ±.03	**0.841** ±.026	**+0.09**	**0.906**	0.55
MMM2	0.85 ±.03	**0.903** ±.019	**+0.05**	**0.956**	0.36
3s5z	0.95 ±.02	0.958 ±.011	+0.01	0.988	0.42
8m_vs_9m	0.95 ±.02	0.952 ±.009	+0.00	0.981	0.51
corridor	0.95 ±.02	0.961 ±.012	+0.01	0.994	0.68
MAPPO executors (SMACv2, 5vs5)					
protoss	0.68 ±.03	**0.742** ±.028	**+0.06**	**0.813**	0.46
terran	0.61 ±.04	**0.679** ±.031	**+0.07**	**0.756**	0.49
zerg	0.45 ±.04	**0.492** ±.034	**+0.04**	**0.575**	0.41

the learned interface; head-specific behavioral effects would require intervention diagnostics beyond coefficients alone.

In panel (c), `5m_vs_6m` changes from *advance* with low maneuver openness ($\bar{g}_2 \approx 0.15$) to *kite* with openness 0.62 under feedback; the open-loop condition retains *advance*. Panel (d) reports 47% of windows above their reward EMA under full feedback, versus 39% open-loop and 36% with shuffled feedback. Appendix E gives a two-round transcript.

5.5 Generality: HAPPO Executors and SMACv2

With HAPPO executors, the same adapter, advisor, and protocol yield gains of +0.09 on `5m_vs_6m` and +0.05 on `MMM2`, with smaller gains on stronger bases (Table 5); the adapter acts on a compatible recurrent-state interface, so HAPPO's sequential policy updates require no change.

SMACv2's randomized 5vs5 compositions and starting positions yield gains of +0.04–+0.07 across the three races. Within-episode gate variance is about twice that on fixed-layout SMAC (Appendix Figure S7), consistent with varying coefficients rather than a single constant per map.

6 Discussion and Conclusion

CLoGA resolves the three adoption dimensions of §1: adoption is decided inside the executor (D1), decomposed by semantic head (D2), and reported back to the advisor (D3). The 2×2 comparison of Table 4 further indicates that D2 and D3 interact—per-head gating converts feedback into a $+0.024$ gain, an otherwise identical scalar gate only $+0.006$—a finding about the interface we test rather than a general law, resting on one task family, three seeds, and cells that do not hold the advice stream fixed.

The setting requires a centralized advisory service and a domain-designed vocabulary. Gate coefficients are not behavioral impact, and shuffled advice still degrades the frozen base on average, so suppression is attenuation, not elimination.

AI USE STATEMENT

In accordance with the ICLR 2027 AI Policy for Authors,[1] we disclose the following uses of generative AI in the research reported here and in the preparation of this manuscript.

Research ideation and execution. Large language model assistants were used as a working aid throughout the research reported here: to discuss the framing of the advice-interface question, to help design and implement components of the study—including the gating variants used in the granularity-axis comparison and their integration into the training code—and to review implementation correctness. Every such suggestion was reviewed, independently re-derived where applicable, and validated by the authors through the unit tests and the experiments reported in this paper, which were defined, configured, launched, and interpreted by the authors. The authors are responsible for the research design, all experimental results, and all conclusions.

Retrieval and discovery. LLM-based tools were used to search for and survey related work, and to locate primary sources for claims about prior methods. All substantive literature review, and the verification of every cited work against its original publication, were carried out by the authors; no reference is included on the basis of an AI suggestion alone.

Manuscript preparation and writing. LLM assistance was used to adapt the manuscript to the ICLR 2027 LaTeX template, to check formatting, and to edit the prose for clarity and concision, including narrowing the scope of the theoretical claims to what the presented evidence supports. The technical content, the reported measurements, and the figure data are those of the authors.

No generative AI system was used to generate synthetic datasets, to prove mathematical claims, or to produce figures or tables. All authors have reviewed and take responsibility for every AI-assisted element of this work.

REPRODUCIBILITY STATEMENT

The implementation descriptions, directive schema, encoding, network architecture, and evaluation protocol are given in Appendix A. Extended derivations are in Appendix B, baseline descriptions in Appendix C, and the reproducibility description in Appendix H. Every run logs per-episode advice traces and the advice cache, from which the reported advisor statistics and cost accounting are computed; code, configurations, base checkpoints, and figure-generation scripts will be released upon publication.

REFERENCES

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2025.

Felipe Leno da Silva, Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. Uncertainty-aware action advising for deep reinforcement learning agents. In *AAAI Conference on Artificial Intelligence*, pp. 5792–5799, 2020.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Jannik Hösch, Alessandro Sestini, Florian Fuchs, Amir Baghi, Joakim Bergdahl, Konrad Tollmar, Jean-Philippe Barrette-LaPierre, and Linus Gisslén. Hierarchical control in multi-agent games: LLM-based planning and RL execution. *arXiv preprint arXiv:2606.20014*, 2026.

Hengyuan Hu and Dorsa Sadigh. Language instructed reinforcement learning for human-AI coordination. In *International Conference on Machine Learning (ICML)*, pp. 13584–13598, 2023.

[1] https://iclr.cc/Conferences/2027/AIPolicyForAuthors

Vaibhav Jain and Gerrit Grossmann. Guiding exploration in reinforcement learning through LLM-augmented observations. *arXiv preprint arXiv:2510.08779*, 2025.

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

Zhengyang Li, Sawyer Campos, and Nana Wang. Language-driven coordination and learning in multi-agent simulation environments. *arXiv preprint arXiv:2506.04251*, 2025.

Zhihao Liu, Xianliang Yang, Zichuan Liu, Yifan Xia, Wei Jiang, Yuanyu Zhang, Lijuan Li, Guoliang Fan, Lei Song, and Jiang Bian. Knowing what not to do: Leverage language model insights for action space pruning in multi-agent reinforcement learning. *Transactions on Machine Learning Research*, 2025. URL `https://openreview.net/forum?id=T49vPTkIt5`.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer, 2016.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2019.

Maryam Shoaeinaeini and Brent Harrison. Guiding reinforcement learning using uncertainty-aware large language models. In *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pp. 363–371, 2025.

Philipp Dominic Siedler and Ian Gemp. LLM-mediated guidance of MARL systems. *arXiv preprint arXiv:2503.13553*, 2025.

Chuanneng Sun, Songjun Huang, and Dario Pompili. LLM-based multi-agent reinforcement learning: Current and future directions. *arXiv preprint arXiv:2405.11106*, 2024.

Lisa Torrey and Matthew E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1053–1060, 2013.

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

Yuan Zhuang, Yi Shen, Zhili Zhang, Yuxiao Chen, and Fei Miao. YOLO-MARL: You only LLM once for multi-agent reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11283–11290, 2025.

SUPPLEMENTARY MATERIAL

This document supplements the main paper with (A) complete implementation details of the advisory channel—summarizer, advisor prompt, directive encoding, adapter architecture with per-head shapes, the closed-loop feedback protocol, and cost accounting; (B) extended derivations of the structural propositions; (C) implementation details of every baseline and ablation arm; (D) additional quantitative results, including full learning curves, ablation curves, adoption–gain calibration, deviation diagnostics, and measured statistics of the advisor's output; (E) a verbatim advisor–executor transcript from the training logs; (F) visualizations of agent behavior on SMAC and SMACv2; and (G) limitations. Section and float numbers are prefixed (A–H, S1, ...) to distinguish them from the main paper.

A IMPLEMENTATION DETAILS

A.1 ADVISORY-CHANNEL PIPELINE

Algorithm S1 summarizes one environment step of CLOGA at rollout time. The pretrained backbone f_θ and action head ϕ are frozen; only the semantic adapters $\{\mathrm{Ad}_{\omega_k}\}$, gates $\{\mathrm{G}_{\varphi_k}\}$, and a fresh critic receive gradients. The corrected state h_{act} is used *only* for the current action distribution; the recurrent memory always carries h_{next} forward, so advice can never accumulate across steps.

Algorithm S1 CLOGA rollout step (executor side)

if no directive, directive expired, drift exceeds threshold, or $t \bmod \kappa = 0$: {refresh}
  $z^t \leftarrow \mathrm{Summarize}(s^t,\ \ell^{\mathrm{prev}},\ \mathrm{Feedback}(\bar{g}, \Delta r, \rho))$
  $\ell^t \leftarrow \mathrm{LLM}(z^t)$ via cache on $\mathrm{sha256}(z^t)$; parse + legality check; no-op on failure
  reset window accumulators for $(\bar{g}, \Delta r, \rho)$
$d_i^t \leftarrow \mathrm{Encode}(\ell^t, s^t) \in \mathbb{R}^{21}$ for each agent i {Table S2}
for each agent i (frozen backbone) **do**
  $h_{\mathrm{next},i}^{\ t} \leftarrow f_\theta(o_i^t, h_{\mathrm{next},i}^{\ t-1})$
  $a \leftarrow d_i^t[\mathrm{legal}] \cdot d_i^t[\mathrm{appl}]$ {hard mask}
  for head $k = 1..K$ **do**
    $\delta_k \leftarrow s \cdot \tanh(\mathrm{Ad}_{\omega_k}(d_{i,S_k}^t))$
    $g_k \leftarrow a \cdot \sigma(\mathrm{G}_{\varphi_k}([h_{\mathrm{next},i}^{\ t}; d_{i,S_k}^t; d_{i,S_{\mathrm{ctx}}}^t]))$
  end for
  $h_{\mathrm{act},i}^{\ t} \leftarrow h_{\mathrm{next},i}^{\ t} + \|_k\,(g_k \cdot \delta_k)$; $a_i^t \sim \pi_g(\cdot \mid h_{\mathrm{act},i}^{\ t})$ via frozen head ϕ
  carry $h_{\mathrm{next},i}^{\ t}$ (not $h_{\mathrm{act},i}^{\ t}$) into the recurrent memory
end for
accumulate $(g_{k,i}^t,\ r^t,\ \mathrm{drift})$ into the current advice window

A.2 STATE SUMMARIZER

The summarizer is a rule template over the global state; it involves no learned components and no LLM call. Determinism is a hard requirement: the sha256 hash of the summary text is the key of the advice cache, so all numbers are formatted with exactly two decimals and every sentence is produced by a fixed template. The summary covers, in fixed order: (1) timestep, alive counts, and remaining steps with a qualitative time-pressure phrase (SMAC episodes are lost on timeout); (2) unit compositions of both teams with combat type and attack range per unit type; (3) an explicit high-value-target line when the enemy fields healers or splash dealers (e.g. Medivac, Colossus); (4) a composition-dependent tactical hint (all-ranged → kiting applies; all-melee → close distance; mixed → coordinate front/back lines); (5) health summaries including the weakest ally's exposure (distance to the enemy center) and, for Protoss units, shield state—a full-shield and a shield-broken Stalker at equal effective HP differ sharply in killability; (6) the lowest-HP enemy with an in-/out-of-range judgment relative to the allies' longest attack range; (7) formation dispersions and center-to-center distance; (8) the current focus-fire target, recent casualty events, and the previous directive with its remaining validity and stated reason. When closed-loop feedback is enabled, the feedback

Table S1: Directive JSON schema. A legality checker extracts the first JSON object from the raw response, validates every field against this vocabulary, and demotes unparsable or inconsistent output to a no-op (which zeroes the `legal` bit of the encoding, Table S2).

Field	Values
`formation`	`disperse` / `compact` / `maintain`
`maneuver`	`advance` / `retreat` / `kite` / `hold`
`target_priority`	`lowest_hp_enemy` / `nearest_enemy` / `highest_threat` / `current_target`
`focus_fire`	`true` / `false`
`duration`	integer in $[4, 32]$ (validity in env. steps)
`confidence`	float in $[0, 1]$
`reason`	one-sentence free text (logged; echoed in the next summary for self-consistency)

paragraph of §A.6 is appended verbatim—it is thereby part of the cache key, so identical situations with different adoption histories are distinct queries.

The deployed system renders summaries and prompts in Chinese (the advisor is a Chinese-capable Qwen model); we show a literal English translation of an example summary:

> *[Step 34/120] 5/5 allies alive, 6/6 enemies alive. 86 steps remain (time is ample).*
> *Ally composition: 5 Marines [ranged, range 6.00]. Enemy composition: 6 Marines [ranged, range 6.00].*
> *Tactical hint: all allied units are ranged; kiting applies.*
> *Ally health is high (mean ratio 0.84); unit 2 is lowest (0.51, distance 4.32 to the enemy center, exposed).*
> *Enemy mean health 0.92; unit 4 is lowest (0.55), 5.10 from our center, in range (nearest 4.87, can be attacked now).*
> *Ally formation is moderate (pairwise 3.12), enemy formation is dense (pairwise 1.85), center distance 7.90.*
> *Current focus target: enemy 4 (health 0.55, in range). Recent events: ally 3 damaged.*
> *Previous directive: keep formation (maintain), attack lowest-HP enemy, advance, focus fire, 4 steps remaining; stated reason: numbers are even, push before reinforcements.*

A.3 Advisor Prompt and Legality Checking

The system prompt fixes the advisor's role ("high-level tactical staff; you provide direction, not micro-control"), a six-level decision framework applied in priority order (emergency survival → kill-securing focus fire → formation repair → composition-appropriate maneuver → endgame harvest → default push), a short unit-tactics glossary, and the output contract: *exactly one JSON object*, no markdown and no prose, with the seven fields of Table S1. A final section instructs the model how to consume the closed-loop feedback paragraph: switch *only* the ignored head while keeping accepted heads, treat below-average window reward as a signal to change tactics, and shorten `duration` when situation drift is above its recent baseline.

Example raw response. The following is a verbatim advisor response from the seed-1 `5m_vs_6m` training run (only `reason` translated):

```
{"formation": "disperse",
 "target_priority": "nearest_enemy",
 "maneuver": "advance",
 "focus_fire": false,
 "duration": 16,
 "confidence": 0.9,
 "reason": "Both sides are ranged and far
  apart; disperse first to reduce focus-
  fire risk, then push into range."}
```

With temperature 0 and the strict output contract, format violations are rare in practice: across the 2 089 refreshes sampled from the eight seed-1 SMAC runs (§D.7), *zero* responses were demoted

Table S2: The 21-dimensional per-agent directive d_i^t and its partition into the $K{=}4$ semantic heads. Dimensions 0–10 are one-hot tactical semantics; 11–20 are contextual signals. `maintain`, `hold`-as-default, etc. are encoded by all-zero slices, distinguishable from "no directive" via the `legal` bit.

#	Content	Range	Head
0	focus fire flag	$\{0,1\}$	target
1	formation: disperse	$\{0,1\}$	formation
2	formation: compact	$\{0,1\}$	formation
3	maneuver: advance	$\{0,1\}$	maneuver
4	maneuver: retreat	$\{0,1\}$	maneuver
5	maneuver: kite	$\{0,1\}$	maneuver
6	maneuver: hold	$\{0,1\}$	maneuver
7	priority: lowest-HP enemy	$\{0,1\}$	target
8	priority: nearest enemy	$\{0,1\}$	target
9	priority: highest threat	$\{0,1\}$	target
10	priority: current target	$\{0,1\}$	target
11	advice strength	$[0,1]$	context
12	advisor confidence c_ℓ	$[0,1]$	context
13	validity time remaining	$[0,1]$	context
14	freshness ϕ^t	$[0,1]$	context
15	observation drift ρ^t	$[0,1]$	context
16	applicability (alive)	$\{0,1\}$	context
17	role weight	$[0,1]$	context
18	unit type (normalized id)	$[0,1]$	context
19	legality bit	$\{0,1\}$	context
20	urgency (health + casualties)	$[0,1]$	context

for illegality. The legality layer nevertheless remains load-bearing for the recovery property of Proposition 1—it is what converts any conceivable failure (malformed JSON, out-of-vocabulary value, inconsistent fields, service outage) into a no-op whose encoding zeroes the residual at that step.

Serving and caching. The advisor is Qwen3.5-9B served locally in bfloat16 with temperature 0 and at most 160 new tokens; "thinking" segments are disabled at the chat-template level (they would exhaust the token budget before the JSON). Responses are cached by summary hash, so within a run each distinct situation is answered once; repeated queries are exact cache hits, which both removes LLM sampling as a seed-to-seed confound and amortizes inference cost. Two failure-handling rules matter for the recovery property of Proposition 1: (i) any network, parsing, or legality failure yields a no-op directive after bounded retries—failed requests are bounded by the retry policy; (ii) a no-op produced by a *transient* failure is never written to the cache, so an outage does not permanently bias a situation toward the base policy.

A.4 Directive Encoding

Table S2 lists the full layout. One global directive yields *per-agent* vectors: applicability (dim 16) zeroes dead agents, role weight and unit type (dims 17–18) differ across heterogeneous compositions, and drift/freshness (dims 14–15) evolve within the validity window. Urgency (dim 20) is $1 - \overline{\mathrm{hp}}_{\mathrm{ally}}$, increased by 0.1 per allied death event in the current window (capped at 1). The hard mask of Eq. (3) in the main paper multiplies dims 19 and 16, so illegal advice, advice dropout, and dead agents give a zero residual at that step—this does not rely on the network *learning* to ignore anything.

A.5 Adapter and Gate Networks

Every head k owns a two-layer adapter $\mathrm{Ad}_{\omega_k} : \mathbb{R}^{|S_k|} \to \mathbb{R}^{H/K}$ and a two-layer gate G_{φ_k}, both with hidden width 64 and ReLU, with $H{=}64$, $K{=}4$, $H/K{=}16$, and residual bound $s{=}1$. Tactical gates read $[h_{\mathrm{next}}; d_{S_k}; d_{S_{\mathrm{ctx}}}]$; the context gate reads $[h_{\mathrm{next}}; d_{S_{\mathrm{ctx}}}]$ only. The final linear layer of every adapter is zero-initialized, so $\delta \equiv 0$ and the initial action distribution matches the frozen checkpoint

Table S3: Per-head widths of the advisory pathway. The scalar-gate control keeps all four adapters and all four gate networks, so its architecture is identical to the one above except that the gate outputs are averaged and broadcast (Appendix C).

Head	$\lvert S_k\rvert$	Gate input	Residual block
formation	2	$H + \lvert S_k\rvert + \lvert S_{\mathrm{ctx}}\rvert$	H/K
maneuver	4	$H + \lvert S_k\rvert + \lvert S_{\mathrm{ctx}}\rvert$	H/K
target	5	$H + \lvert S_k\rvert + \lvert S_{\mathrm{ctx}}\rvert$	H/K
context	10	$H + \lvert S_{\mathrm{ctx}}\rvert$	H/K
total	21	—	H

exactly (the concat-split in the exact-concat baseline needs a `contiguous()` call to reproduce this on CUDA kernels). Table S3 gives the per-head shapes.

Beyond the gate values g_k (adoption *coefficients*), the implementation logs per head the realized injection magnitude $\|g_k \cdot \delta_k\|_2$ (adoption *effect*): a wide-open gate on a near-zero residual barely changes the action distribution, so the two statistics are reported separately in all diagnostics.

A.6 Closed-Loop Feedback Protocol

At each refresh, if the previous directive was legal and non-noop and at least one step was observed under it, a deterministic feedback paragraph is prepended to the next query (English translation; the anchor phrase at the start is matched verbatim by the system prompt):

> *Closed-loop feedback (execution of your previous directive): mean adoption 0.31 (below recent baseline 0.42); per-window reward rate $+0.012$ (below recent average $+0.019$); situation drift 0.34 (above recent baseline; consider a shorter duration); meanwhile allies -1, enemies -0, center distance closed from 7.90 to 5.12. Per-head adoption: formation 0.58 (adopted), maneuver 0.15 (ignored), targeting 0.66 (adopted). Adjust accordingly: if adoption is below recent levels the directive does not fit the local situation—change tactical direction; …*

Three design choices are worth recording. *Relative, not absolute.* Adoption, reward rate, and drift are each compared against an episode-local exponential moving average (all α=0.3; dead-bands ± 0.03 for adoption, $\pm 5\%$ for reward, ± 0.05 for drift) rather than fixed thresholds: SMAC's shaped reward scale varies strongly with battle intensity, and raw magnitudes carry no "was this better than usual" signal. The EMA is updated only in the branch that actually emits feedback, so windows without effective advice never pollute the baseline. *Head-targeted correction.* The per-head sentence names the head in the advisor's own vocabulary; the context head is excluded because it modulates meta-information the advisor did not produce. *Causal drift detail.* The abstract drift value is grounded by a survivor-count and center-distance delta ("meanwhile …"), letting the advisor reason about *what happened* during the window instead of reverse-engineering it from the next summary.

A.7 Training Protocol and Compute

Table S4 lists all hyperparameters; none are tuned per map. Base checkpoints are trained with the public MAPPO (on-policy) and HAPPO (HARL) codebases and loaded as frozen executors; the adapter attaches to the recurrent hidden state and is therefore agnostic to how the backbone was trained (for HAPPO, one adapter instance per actor). Each run uses one NVIDIA A100 for both the executor and the co-located advisor; a full 200k-step run takes 18.9 hours. Every trained arm runs seeds $\{1, 2, 3\}$; the frozen baseline reports five evaluation points of the same checkpoint. The primary metric is the mean of the final five evaluations (win rate over 32 evaluation episodes each).

A.8 Advisor Cost Accounting

The advisor is consulted only at *refreshes*—when no directive is active, the active directive expires, drift exceeds its threshold, or the periodic κ-interval fires—so query volume is governed by the

Table S4: Hyperparameters (identical across all maps, executor families, and benchmarks; no per-map tuning).

Optimizer / lr (actor & critic)	Adam / 5×10^{-4}
PPO clip / epochs / entropy	0.2 / 4 / 0.01
GAE γ / λ / value-loss coef.	0.99 / 0.95 / 0.5
Rollout episodes per update	8
Total env. steps / eval. points	200k / every 5k
Hidden dim H / heads K / widths	64 / 4 / 64
Residual bound s / gate penalty β_{gate}	1.0 / 0
Advice interval κ / duration range	8 / $[4, 32]$
Advice dropout / gate init	0 / $\sigma(0)$=0.5, $\delta \equiv 0$
Feedback EMAs α (gate, reward, drift)	0.3 / 0.3 / 0.3
Advisor / precision / temperature	Qwen3.5-9B / bf16 / 0
Max new tokens / retries	160 / 2 (exp. backoff)

durations the advisor itself chooses. Measured over the trace-sampled episodes of the eight seed-1 SMAC runs, refreshes range from $\sim$3.0 per episode (`3m`, short episodes, mean chosen duration 8.0) to $\sim$11.3 per episode (`3s5z_vs_3s6z`, long attritional battles), i.e. one LLM call per 8–20 environment steps before caching. Caching then removes the recurring cost of repeated situations: the measured fraction of refreshes served from cache ranges from 4.7% (`3s5z_vs_3s6z`, whose long battles rarely revisit identical summaries) to 36.7% (`2s_vs_1sc`, whose short scripted openings recur across episodes). Because the cache key is the summary hash and feedback is part of the summary, closed-loop runs pay a systematically lower hit rate than open-loop runs—feedback individualizes otherwise identical situations; this is the intended trade of cost for corrective signal.

B Extended Derivations

For completeness we restate the two structural propositions of the main paper with full derivations, and record a locality corollary used implicitly by the per-head analyses.

Setting. Fix an agent i and step t (indices dropped). The readout computes $h_{\text{act}} = h_{\text{next}} + \|_{k=1}^{K}(g_k\, \delta_k)$ with $\delta_k = s\tanh(\text{Ad}_{\omega_k}(d_{S_k})) \in \mathbb{R}^{H/K}$, $g_k = v \cdot \sigma(\cdot) \in [0,1]$, $v = d[\text{legal}] \cdot d[\text{appl}] \in \{0,1\}$, where head k writes only its own H/K-dimensional block.

Proposition 1 (exact recovery). *If $v = 0$ or $g_k = 0$ for all k, then $h_{\text{act}} = h_{\text{next}}$ and $\pi_g = \pi_0$; the same holds at initialization and for LLM failures mapped to a no-op, comparing policies at the same history.*

Proof. Three cases. (i) $v = 0$: since $g_k = v \cdot \sigma(\cdot)$ and $\sigma(\cdot) \in (0,1)$ is bounded, $g_k = 0$ exactly for every k; each block $g_k\delta_k$ is the zero vector, so the concatenated residual is zero and $h_{\text{act}} = h_{\text{next}}$. The frozen action head then reproduces $\pi_0(\cdot \mid h_{\text{next}})$ identically. (ii) $g_k = 0$ for all k with $v = 1$: identical conclusion by the same substitution. (iii) Initialization: the last linear layer of every adapter is zero-initialized, so $\text{Ad}_{\omega_k}(\cdot) \equiv 0$ and $\delta_k = s\tanh(0) = 0$ regardless of g; the residual vanishes for *every* input. Any LLM failure (outage, malformed output, illegal fields) is demoted to a no-op directive whose encoding sets $d[\text{legal}] = 0$, reducing to case (i). All three cases are exact identities, not approximations: no property of the learned weights beyond the zero initialization is used, and in cases (i)–(ii) none at all. □

Corollary (per-head locality). *Setting gate $k = 0$ removes exactly the readout residual of head k and leaves the residuals of all other heads unchanged.*

Proof. Head $j \neq k$'s block $g_j\delta_j$ depends on $(h_{\text{next}}, d_{S_j}, d_{S_{\text{ctx}}})$ only—not on g_k or δ_k—and the blocks occupy disjoint coordinates of h_{act}. Setting $g_k = 0$ therefore zeroes block k and no other entry of the residual. □

This corollary concerns readout-block locality at fixed inputs. It does not imply independent behavioral effects: the frozen action head combines all hidden coordinates. Gate coefficients provide

head-labeled interface telemetry, while realized injection and action-level interventions are needed to measure behavioral influence.

Proposition 2 (bounded readout-state deviation). *For a fixed agent and time,* $\|h_{\mathrm{act}} - h_{\mathrm{next}}\|_2 \le s\sqrt{H/K}\,\|g\|_2$.

Proof. Each coordinate of δ_k lies in $[-s, s]$ by the $\tanh$ bound, so $\|\delta_k\|_2 \le s\sqrt{H/K}$. Since the K blocks occupy disjoint coordinates,

$$\|h_{\mathrm{act}} - h_{\mathrm{next}}\|_2^2 = \sum_{k=1}^{K} g_k^2 \|\delta_k\|_2^2 \le s^2 \frac{H}{K} \sum_{k=1}^{K} g_k^2 = s^2 \frac{H}{K} \|g\|_2^2.$$

Taking nonnegative square roots proves the claim. No assumption about a Lipschitz action head or a state-occupancy distribution is required for this readout-state bound. □

Scope. This is a pointwise bound between corrected and uncorrected readouts at the same history. It does not bound joint-policy return or guarantee non-degradation under learned soft gates. Proposition 1 likewise compares action distributions at a fixed history; histories can diverge after earlier actions. For valid advice, a finite sigmoid output is generally positive, so exact hard-mask recovery and approximate soft suppression are distinct.

Why the corrected readout is not carried. Feeding h_{act} back into the recurrence would directly inject residuals into memory; their propagation would depend on the recurrent dynamics. Carrying h_{next} avoids that direct write and keeps the current residual subject to the displayed bound. It does not eliminate indirect effects through actions, environment transitions, and future observations. The main-method configuration excludes the write-memory variant (Appendix H).

C Baseline and Ablation Implementations

Arms use the protocol of §A.7 and matched checkpoints; advisory arms share the advisor and summarization interface where applicable. Changing gates or feedback can also change subsequent advice and trajectories, so these are not fixed-advice counterfactuals.

State-only. The readout is replaced by four residual blocks that consume only h_{next} (no directive input), with block and hidden widths chosen to match the main advisory pathway's budget. Zero-initialized like the main method. Isolates *extra trainable capacity*.

w/ fixed gate. The learned gate output is overwritten by a constant $g_k{=}0.5$ (the value the learned gates start from); adapters still train. Isolates *fixed vs. learned adoption*.

w/o gate. As above with $g_k{=}1$: the bounded residuals are injected at full strength. Isolates *the necessity of gating*; the same advisory interface and adapter architecture, but not necessarily identical advice streams.

w/ shuffled advice. Each refresh receives the directive generated for an unrelated state (a random permutation over the run's advice stream); encoding, gating, and training are unchanged. Isolates *state–advice alignment* from generic language conditioning.

w/o feedback. `closed_loop=false`: the advisor receives summaries without the feedback paragraph. Isolates *closing the loop*.

w/ shuffled feedback. Feedback paragraphs are exchanged between unrelated windows, preserving format, length, and marginal statistics while destroying correctness. Isolates *feedback content* from feedback presence.

InstructPPO. An InstructRL-style (Hu & Sadigh, 2023) instruction regularizer: a deterministic rule maps the current directive and snapshot to a prior distribution π_{prior} over *available* actions (movement scored by alignment with the advised direction, attack actions boosted for the advised target; inverse temperature $\beta{=}2.0$, component weights $1.0/0.5/0.5/1.5$ for move/stop/attack/target), and the full actor is trained with an added $\mathrm{KL}(\pi \,\|\, \pi_{\mathrm{prior}})$ term. When advice is absent or illegal the prior degrades to uniform over available actions, so the regularizer injects no tactical bias. This arm trains

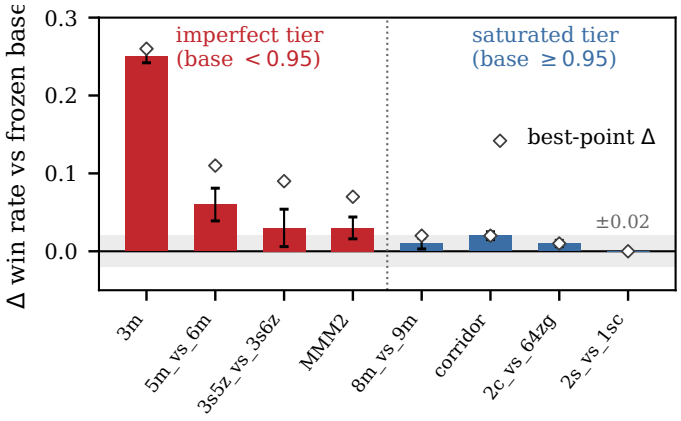


Figure S1: Δ over frozen base on 8 maps (bars: last-5 $\pm$ s.d.; diamonds: best). All imperfect bases improve; saturated maps stay within ± 0.02.

the entire actor rather than the advisory pathway alone; it tests the fixed-strength adoption paradigm, not a capacity match.

Exact concat. The action head is split into the frozen base head on h_{next} plus a zero-initialized, bias-free projection of the 21-dim directive onto action logits. Initialization equals the checkpoint bitwise; the directive path trains. This is the strongest "inject without gating" variant we found that preserves the zero-initialized starting point.

Scalar gate. The scalar arm underlies the gate-type axis of the 2×2 design. It keeps all four residual adapters and all four gate networks, with the same inputs, initialization, hard mask, and no-op path as the main method; only the aggregation of the gate outputs differs: $g^t_{\text{sc},i} = v^t_i \cdot \frac{1}{K} \sum_k \sigma\big(\mathrm{G}_{\varphi_k}(\cdot)\big)$, broadcast to all four residual blocks. The average is taken over the sigmoid outputs, not the pre-sigmoid logits, so $g_{\text{sc}} \in [\min_k g_k, \max_k g_k]$: one head's high gate cannot raise adoption on the others, and one closed head pulls the shared coefficient down. Because the four adapters and four gate networks are retained, this arm isolates *whether the adoption coefficient is shared across heads* rather than confounding the comparison with a different parameterization. In closed-loop runs the feedback paragraph necessarily reports a single coefficient for the three tactical heads (the per-head sentence is replaced by the ignored-field hint of Appendix A.6), which is the interface difference the 2×2 comparison is designed to expose.

D Additional Quantitative Results

D.1 Full Learning Curves

Figure S2 shows the eight-map learning curves corresponding to Table 2(b); Figure S3 gives the HAPPO and SMACv2 settings of Table 5. The complete-method curves are empirical diagnostics. Zero initialization establishes equality at initialization only, and subsequent behavior is not a consequence of Proposition 1 or Proposition 2. In particular, a readout-state bound does not guarantee that every training iterate outperforms the base.

D.2 Ablation Learning Curves

Figure S4 plots training curves for the mechanism-ablation arms of Table 4 on the weakest (`3m`) and the heterogeneous (`3s5z_vs_3s6z`) controlled maps. The trajectories show *when*, not just where, each arm ends: *w/o gate* passes through a pronounced transient dip below the frozen base early in training—ungated injection perturbs a working policy before the adapters adapt—and on `3s5z_vs_3s6z` it never recovers to the baseline. *Shuffled advice* declines gently rather than sharply, consistent with the gate progressively closing on misaligned input (Figure S7b) instead of the policy being actively corrupted. The two feedback ablations track the full method for the first

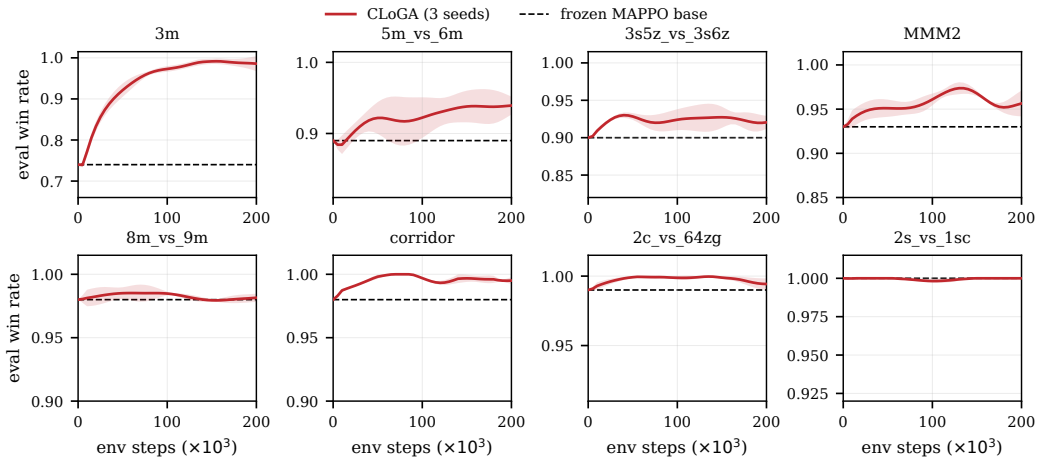


Figure S2: CLoGA evaluation win rate on all eight SMAC maps (MAPPO executors; mean ± s.d. over 3 seeds; dashed line: frozen base). Top row: imperfect tier; bottom row: saturated tier. Note the per-panel y-ranges.

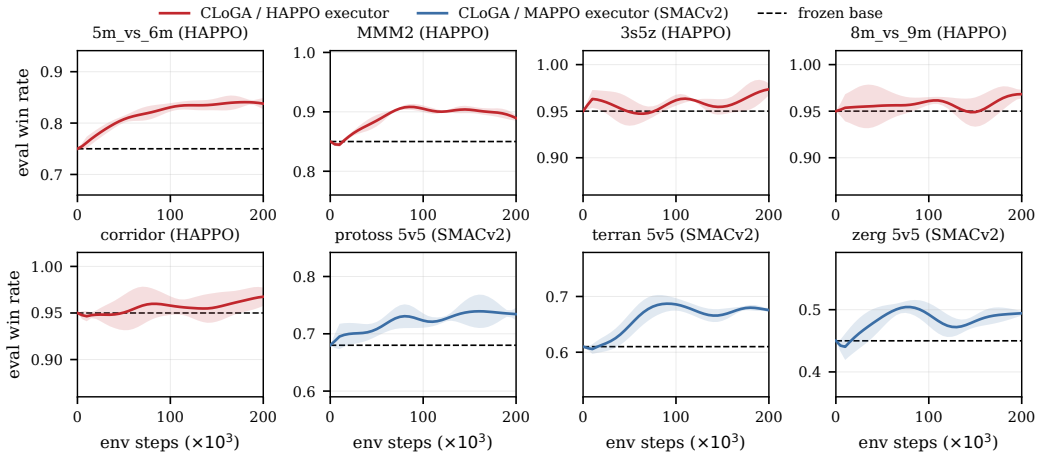


Figure S3: Generality: identical adapter, advisor, and protocol on HAPPO executors (five SMAC maps, red) and on SMACv2 5vs5 with procedurally randomized compositions (MAPPO executors, blue).

∼50k steps and separate later, matching the expectation that feedback matters once the executor's adoption pattern has stabilized enough to be informative to the advisor.

D.3 Adoption–Gain Calibration

Figure S5 collects the values from Tables 2(b) and 5. Panel (a) shows the reported map-level gains, concentrated among weaker bases and non-negative for these evaluated complete-method settings. Panel (b) shows a non-monotone association between mean gate openness and base competence. This descriptive association does not establish that broad adoption is risk-free or that the gate coefficients measure realized behavioral benefit.

D.4 Deviation Diagnostics for the Fail-Safe Bounds

Proposition 2 bounds the current readout-state deviation by $s\sqrt{H/K}\,\|g\|_2$. Figure S6 compares that bound with measured deviation and separately plots $\mathrm{KL}(\pi_g \,\|\, \pi_0)$ on 3m. The displayed deviation is below the bound and the plotted KL grows smoothly from zero. Smooth learning dynamics and

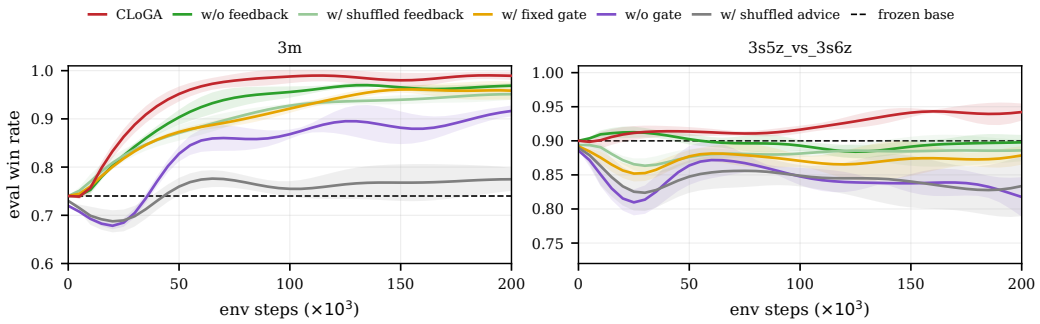


Figure S4: Mechanism-ablation training curves on `3m` and `3s5z_vs_3s6z` (mean ± s.d., 3 seeds; dashed: frozen base). Endpoints correspond to Table 4. Note the early transient of *w/o gate* below the base on both maps.

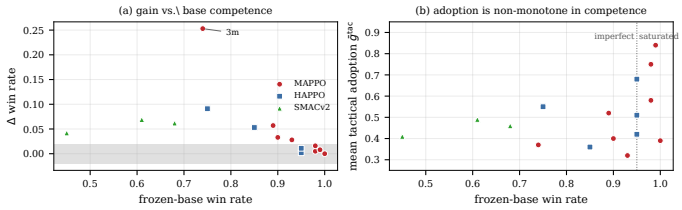


Figure S5: Every evaluated setting (8 MAPPO maps, 5 HAPPO maps, 3 SMACv2 races). (a) Gain vs. base competence; the shaded band is ±0.02. (b) Mean tactical adoption vs. base competence.

return improvements are empirical observations, not consequences of the pointwise bound; no joint-policy value guarantee is claimed.

D.5 Per-State Gating and Misaligned Advice

Figure S7(a) reports within-episode variance of tactical gate coefficients, approximately twice as large on SMACv2 as on the plotted SMAC settings. This is consistent with varying state-conditioned coefficients rather than one map-level constant, but does not isolate the contribution of individual inputs. Under shuffled advice, the mean coefficient falls from 0.5 to about 0.12 (panel b); Table 4 nevertheless shows a mean of 0.867 versus the frozen 0.888. Suppression is therefore incomplete, with residual performance degradation.

D.6 Per-Head Adoption: Remaining Maps and Generality Settings

Figure S8 shows per-head gate evolution on four representative maps; Figure S9 completes the picture on the remaining four. The same taxonomy recurs: single-dimension maps produce high head variance (`8m_vs_9m`: maneuver and targeting lead formation), extreme-adoption maps open all heads (`2c_vs_64zg`: everything above 0.79—kiting Colossi against a Zergling swarm is precisely the regime the advisory vocabulary describes), and the already-saturated `2s_vs_1sc` settles all heads mildly below the 0.5 initialization. Figure S10 extends the final-adoption heatmap to the generality settings. The HAPPO profiles mirror their MAPPO counterparts on shared maps (maneuver-led on `5m_vs_6m`, targeting-led on `MMM2`) even though the executors were trained by a different algorithm—consistent with adoption profiles being a property of the *task's* coordination gap rather than of the executor family. SMACv2 rows are flatter across heads, as expected when the relevant tactical dimension changes from episode to episode with the drawn composition.

D.7 What the Advisor Actually Recommends

The preceding analyses concern the *executor's* side of the channel; Figure S11 profiles the *advisor's* side, aggregated from the advice traces of the seed-1 training runs (2 089 legal directives

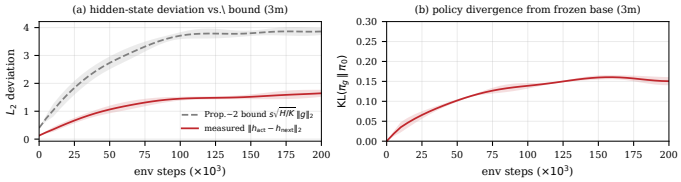


Figure S6: Fail-safe diagnostics on `3m` (3 seeds). (a) Measured $\|h_{\text{act}} - h_{\text{next}}\|_2$ vs. the Proposition-2 bound. (b) $\mathrm{KL}(\pi_g \,\|\, \pi_0)$, logged at every step.

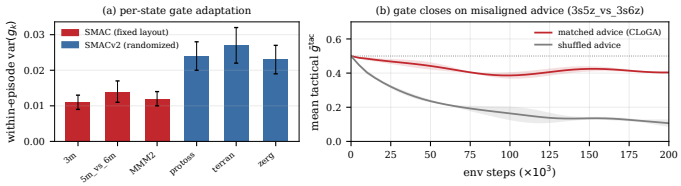


Figure S7: (a) Within-episode variance of tactical gates on SMAC vs. SMACv2. (b) Mean tactical gate under matched vs. shuffled advice.

across the eight SMAC maps; measured data, not synthesized). The advisor's output is strongly map-adaptive and tracks unit-composition tactics: `kite` dominates on the Colossi-vs-Zerglings map `2c_vs_64zg` (91% of directives—ranged units against a melee swarm) and leads on all marine-mirror maps, whereas `advance` dominates on the melee-heavy `3s5z_vs_3s6z` (85%) and on `corridor` (77%). Focus fire is recommended nearly always where kill-securing is pivotal (`corridor` 100%, `2c_vs_64zg` 98%, `MMM2` 92%) and more selectively on open marine fights (`3m` 66%). Chosen durations average 8.0–10.2 steps—the advisor voluntarily operates near the lower end of the permitted $[4, 32]$ range, matching the fast-changing engagements—and stated confidence is uniformly high (0.86–0.89) with little map variation, confirming that executor-side gating, not advisor self-assessment, carries the adoption decision: the advisor is *not* well calibrated about when its advice will be adopted, which is precisely the premise of placing the gate in the executor (D1).

E An Advisor–Executor Transcript

The following two consecutive rounds are taken verbatim from the advice trace of the seed-1 `5m_vs_6m` training run (episode 0; summaries condensed to the relevant lines, all text translated from Chinese, numbers unaltered). They show the full loop—summary, directive, execution, feedback, revision—in one concrete instance.

Round 1 (step 0). Summary (excerpt): *"[Step 0/70] 5/5 allies alive, 6/6 enemies alive; 70 steps remain (time is ample). Allies: 5 Marines [ranged, range 6.00]; enemies: 6 Marines [ranged, range 6.00]. Tactical hint: all allied units are ranged; kiting applies. Lowest-HP enemy at 13.86 from our center, out of range (nearest 13.17 > range 6.00). Center distance 13.98. No focus target; no active directive."* Directive: `formation=disperse`, `target_priority=nearest_enemy`, `maneuver=advance`, `focus_fire=false`, `duration=16`, `confidence=0.9`; reason: *"Both sides are ranged and far apart; disperse first to reduce focus-fire risk, then push as a group into range."*

Round 2 (step 8). Feedback (verbatim): *"Closed-loop feedback (execution of your previous directive): mean adoption 0.51 (mostly adopted; first window of this episode), per-step reward rate $+0.316$ (first window, no baseline yet), situation drift 0.73 (sharp change during validity); meanwhile enemy survivors -1, center distance closed from 13.98 to 6.07. Per-head adoption: formation 0.43 (partially adopted), maneuver 0.68 (adopted), targeting 0.57 (adopted)."* Revised directive: `formation=disperse`, `target_priority=lowest_hp_enemy`, `maneuver=kite`, `focus_fire=true`, `duration=8`, `confidence=0.85`; reason: *"Both sides are ranged and the distance has closed to 6.07; disperse immediately to reduce focus-fire risk, and focus the lowest-HP enemy to build a numbers advantage."*

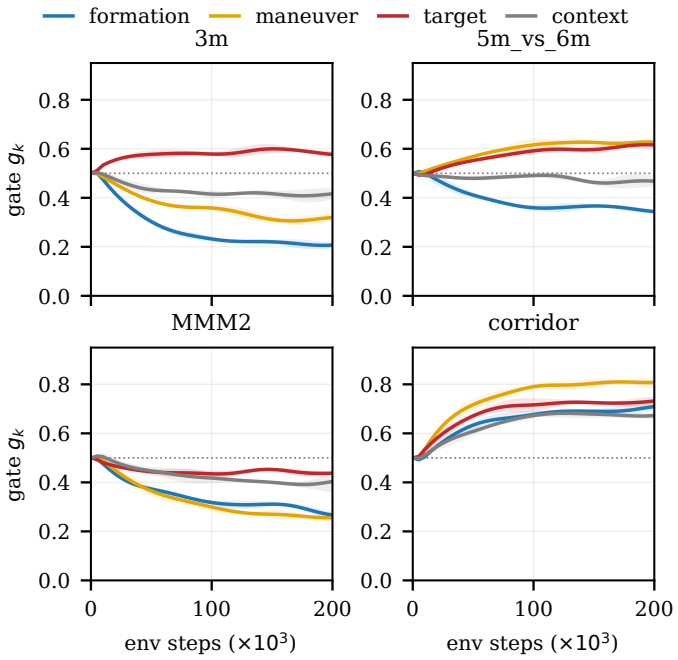


Figure S8: Per-head gate evolution. Gates differentiate from the uniform 0.5 initialization according to task structure.

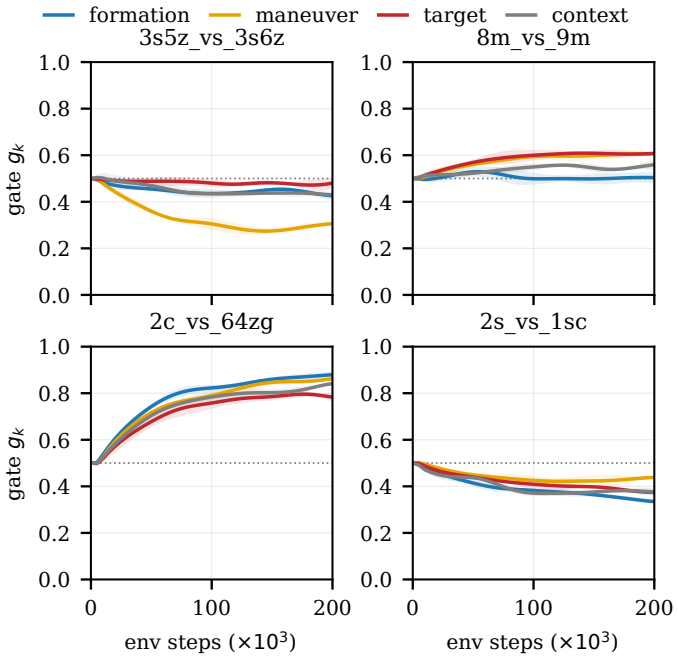


Figure S9: Per-head gate evolution on the four SMAC maps not shown in Figure S8 (mean $\pm$ s.d., 3 seeds; dotted line: uniform initialization $\sigma(0)$=0.5).

Three directive changes occur in this exchange: maneuver changes from `advance` to `kite` as engagement distance shrinks (13.98 to 6.07), duration changes from 16 to 8, and targeting changes from `nearest_enemy` without focus fire to `lowest_hp_enemy` with focus fire. The revised reason repeats the reported distance 6.07. This illustrates consistency between the feedback context and

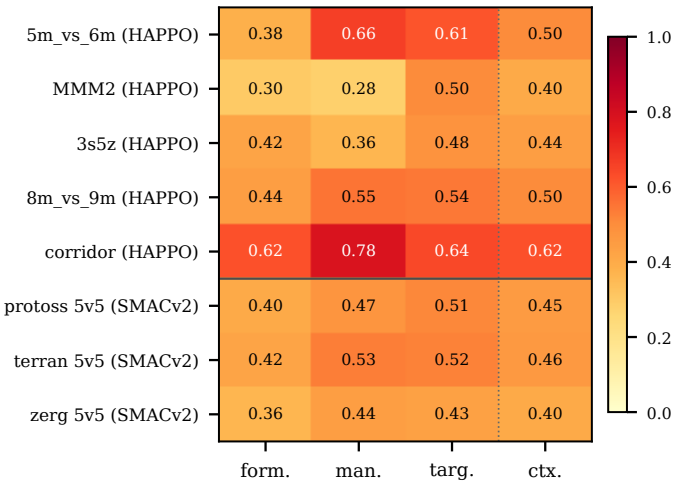


Figure S10: Final adoption by head for the generality settings (top: HAPPO executors on SMAC; bottom: MAPPO executors on SMACv2). Tactical-head means match the $\bar{g}^{\mathrm{tac}}$ column of Table 5.

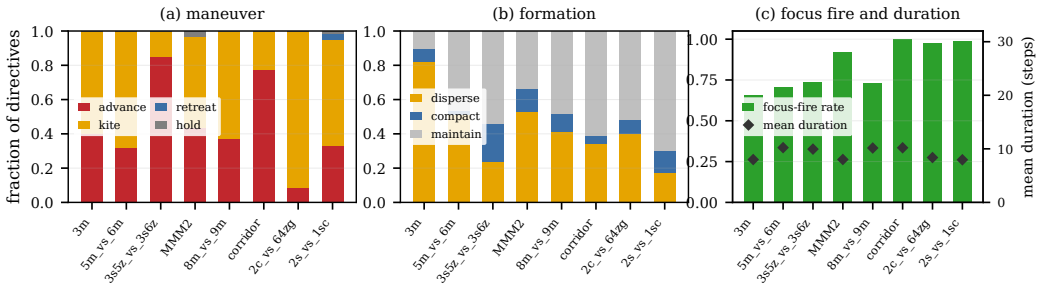


Figure S11: Measured advisor-output statistics per map, aggregated from the advice traces of the seed-1 training runs (2 089 legal directives; sampled training episodes). (a) Maneuver distribution. (b) Formation distribution. (c) Focus-fire rate (bars, left axis) and mean chosen duration (diamonds, right axis).

the revised directive, without an advisor weight update. Because the new state summary and multiple feedback fields change together, this transcript alone does not identify which field caused each revision.

F Agent Behavior on SMAC and SMACv2

Figure S12 shows a raw in-game frame; Figures S13 and S14 visualize the two behavioral episodes analyzed quantitatively in the main paper.

Focus fire on `3m` (Figure S13). The weakest base policy's dominant failure is fire spreading: damage is distributed, no enemy dies early, and the numbers advantage never materializes. The targeting head—the head with the highest learned adoption on this map—injects exactly this correction, and the keyframes show the mechanism: all attack arrows collapse onto the advised target, and re-target jointly after each kill.

Maneuver correction on `5m_vs_6m` (Figure S14). The visualization accompanies the aggregate maneuver diagnostic in Figure 4(c), with gate coefficients 0.15 and 0.62. Appendix E provides a separate concrete two-round interaction example. These should not be conflated: its reported gate coefficients differ, and neither depiction alone establishes that feedback, rather than co-varying state information, caused the change.

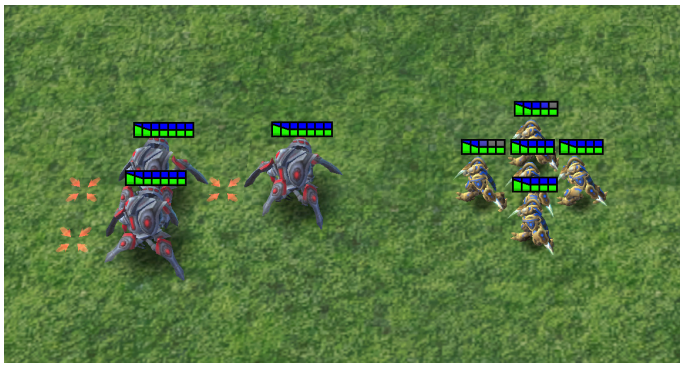


Figure S12: In-game frame from a SMAC episode (allied Colossi vs. Zerg composition). Health/shield bars are rendered above units; the orange markers show the current move-attack order of the leftmost group.

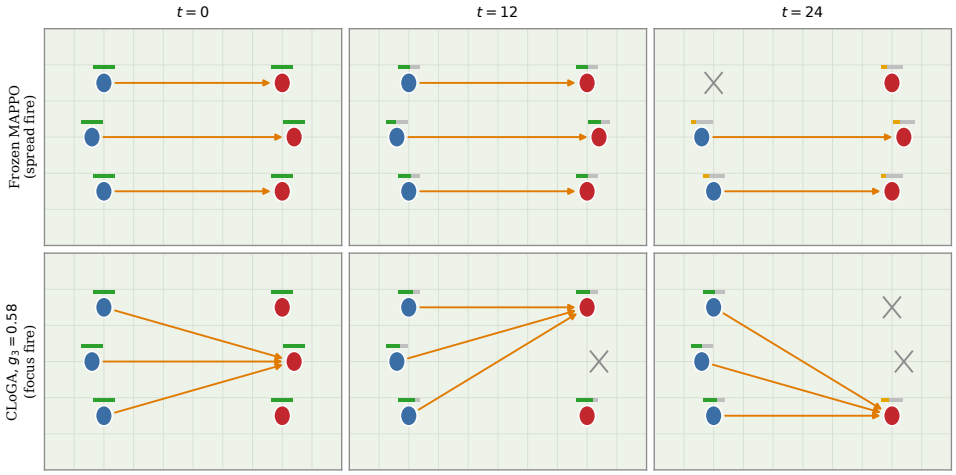


Figure S13: Top-down keyframes on `3m` (blue: allies; red: enemies; bars: health; arrows: attack targets; crosses: deaths). **Top**: the frozen base spreads fire across all three enemies; no kill is secured before an ally falls. **Bottom**: with the targeting head open (g_3=0.58), all three marines concentrate on the advised lowest-HP target and re-focus after each kill, converting even trades into a 3-vs-1 endgame. Formation advice is simultaneously ignored (g_1=0.22)—per-head gating adopts one dimension without the other.

Choke-point usage on `corridor` (Figure S15). `corridor` pits six Zealots against a Zergling swarm whose numbers only tell in open ground. The map records the highest maneuver adoption of the entire suite (g_2=0.82, Figure 4(b)), and the behavioral reading is direct: the advised fall-back-and-hold converts the engagement into a sequence of narrow-front fights where at most two or three Zerglings reach the line at once. This is also the map where the advisor is at its most stereotyped (100% focus fire, 100% lowest-HP priority, §D.7)—when one tactic dominates a map, high adoption of a repetitive directive is the *correct* learned response, and the gate obliges.

SMACv2: composition-conditioned adoption (Figure S16). On randomized compositions no single behavior pattern characterizes a map. Figure S16 contrasts two draws faced by the same trained `protoss 5v5` policy: an all-ranged draw where the advised *kite* is adopted at g_2=0.71 and the units fight a retreating line, and a melee-heavy draw where kiting is useless, the advisor recommends *advance*, and adoption redistributes toward closing distance. The aggregate signature of this per-draw adaptation is the doubled within-episode gate variance of Figure S7(a).

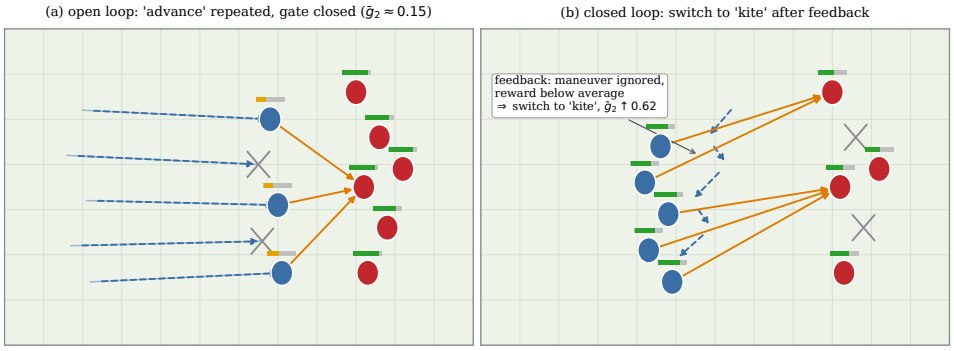


Figure S14: The feedback-driven strategy correction of Figure 4(c), seen at the behavior level on `5m_vs_6m`. (a) Open loop: the advisor keeps recommending *advance* into a 5-vs-6 engagement; the executor's maneuver gate stays closed and the fallback engagement trades poorly. (b) Closed loop: after feedback reports the maneuver head ignored and window reward below average, the advisor switches to *kite*; adoption rises to 0.62 and the marines fight a hit-and-run retreat southward, picking off pursuers without losses.

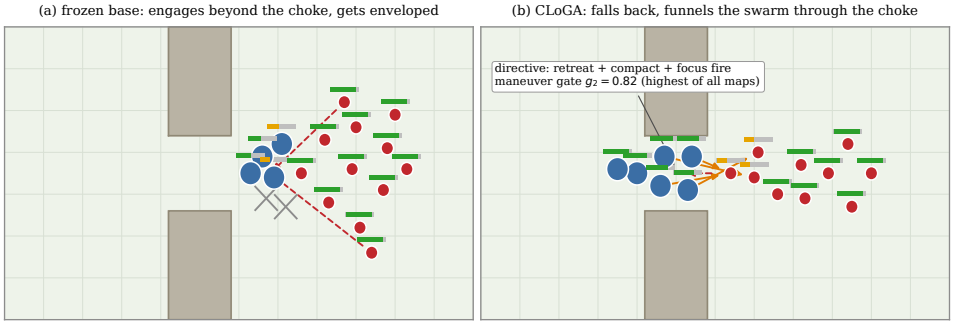


Figure S15: Choke-point behavior on `corridor` (gray blocks: walls; small red: Zerglings; blue: Zealots). (a) The frozen base engages beyond the choke and is enveloped by the swarm. (b) Under CLoGA, the advised *retreat* + *compact* + *focus fire* is adopted through the suite's highest maneuver gate; the swarm funnels through the choke and is destroyed at the mouth.

G Limitations and Failure Modes

Gains are bounded by advice relevance, not by the mechanism. CLoGA is a filter, not a generator: it can use guidance expressible in the directive vocabulary, but it cannot create tactics that vocabulary cannot express, and it does not guarantee rejection of every harmful suggestion. The saturated tier (gains $+0.00$–$+0.02$) is the benign face of this bound; the SMACv2 `zerg` race (base $0.45 \rightarrow 0.49$) is the sharper one—when the base policy's deficit is micro-execution rather than tactical direction, better adoption of correct-but-coarse advice helps only marginally.

The vocabulary is domain-designed. The three tactical fields and their values encode StarCraft combat knowledge. Porting to another cooperative domain requires re-designing the vocabulary and summarizer (though not the adapter, whose structure only assumes *some* semantic partition). What the framework removes is the need to hand-tune *adoption*; it does not remove the need to design *interfaces*.

One directive, per-agent adoption, no per-agent advice. The advisor issues a single global directive; heterogeneity across agents enters only through applicability, role, and unit type in the encoding. Situations where different sub-groups need *contradictory* guidance (e.g. a flank and a holding force) are expressible only as whichever compromise the vocabulary admits; the per-head gates can then implement partial, agent-varying adoption of that compromise, but not two distinct plans. Per-group directives with per-group feedback are the natural extension.

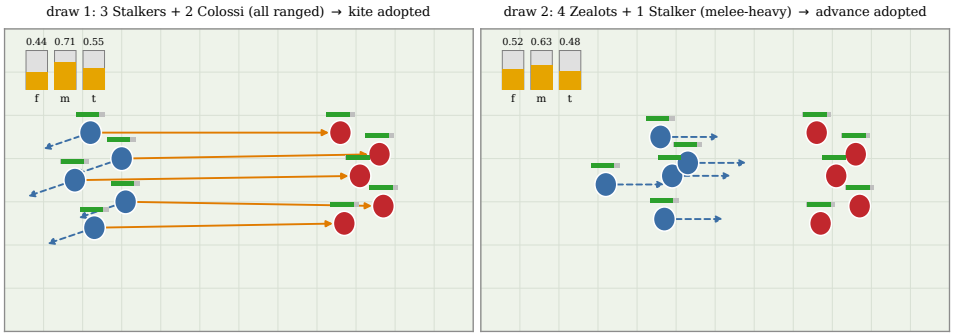


Figure S16: Same trained policy, two SMACv2 `protoss 5v5` draws (inset bars: per-head gates $g_1/g_2/g_3$ for the current window). Left: all-ranged draw—*kite* adopted (g_2=0.71), units fire while giving ground. Right: melee-heavy draw—*advance* adopted, units close distance. The gate reweights heads per draw; a map-level constant could satisfy at most one of the two.

Feedback is windowed and aggregate. Adoption is reported as a window mean over agents and steps; a directive adopted intensely by two agents and ignored by three reads as "partially adopted". The EMA baselines make the signal relative and robust, but also mean the first window of every episode carries weaker semantics ("no baseline yet"), and a pathologically oscillating adoption pattern could alias into a stable-looking mean. We observed no such case, but the aggregation is a real information bottleneck chosen for cacheability.

Latency couples the advisor to the rollout. Uncached refreshes block the environment step that triggers them. Local serving, caching, and $\kappa = 8$ reduce query frequency but do not by themselves establish a small wall-clock overhead. The current cost analysis (Appendix A.8) reports refresh and cache statistics, not measured uncached latency or the fraction of wall-clock time spent on the advisor. A remote or larger advisor may motivate asynchronous refresh, which would change the freshness semantics of dimensions 13–15.

H Reproducibility

All experiments use the configuration files released with the code (one YAML per arm; the main-method configuration is validated at launch so that `fusion=gated`, `gate_type=semantic`, frozen backbone and head, closed loop, and checkpoint-copied head initialization cannot silently drift). Determinism levers: temperature-0 advisor with hash-keyed caching; deterministic summarizer and feedback templates (two-decimal formatting, fixed sentence order); fixed seeds $\{1, 2, 3\}$; legality checking with no-op fallback. The evaluation protocol (32-episode evaluations every 5k steps; last-5 mean) and the base-checkpoint matching rule (each advisory arm is compared against *its own* loaded checkpoint, matched by checkpoint file, not by map name) follow the main paper. Every run additionally logs per-episode advice traces (summary, raw LLM response, parsed directive, cache-hit flag, feedback) and the advice cache itself; the measured statistics of §D.7, the cost accounting of §A.8, and the transcript of §E are all computed or excerpted directly from these logs. Code, configurations, base checkpoints, advice traces, and the figure-generation scripts for this supplement will be released upon publication.