MaRePID: Multi-Agent LLM Reasoning via Distributed PID Regulation

Anonymous authors
Paper under double-blind review

Abstract

Multi-agent large language model (LLM) systems increasingly solve complex tasks through multi-round reasoning, where one agent's corrections can influence the subsequent reasoning of others. These evolving interactions make it challenging to determine how feedback should guide subsequent corrections throughout the reasoning process. We introduce **MaRePID**, an inference-time framework for regulating multi-agent LLM reasoning through distributed proportional–integral–derivative (PID) control. MaRePID updates a bounded control state for each agent using evaluator-derived feedback and neighboring control summaries, then converts the updated state into textual directives through a predefined mapping. By incorporating current mismatch, its accumulation over time, and its recent change, MaRePID adjusts prompt guidance as reasoning progresses while accounting for interactions within the team. We additionally study prediction-triggered sensing to determine when new evaluator feedback is needed to avoid unnecessary queries. Across MultiAgentBench, AgentWebBench, MATH, and BigCodeBench, the full-sensing, always-update variant of MaRePID achieves higher task performance than the evaluated fixed and adaptive baselines, with an improvement of up to 18.3 points over the base multi-agent system on AgentWebBench. MaRePID enables inference-time regulation of multi-agent reasoning without modifying model parameters or the underlying communication protocol.

1 Introduction

Large Language Models (LLMs) have enabled multi-agent systems, where multiple agents collaborate through communication, debate, and iterative refinement to solve complex reasoning tasks. Recent studies demonstrate that interactions among LLM-based agents can improve reasoning through communicative collaboration (Qian et al., 2024), debate-based refinement (Du et al., 2024), and collective decision-making in multi-agent environments (Zhu et al., 2025). However, existing multi-agent frameworks typically rely on predefined coordination mechanisms, including fixed roles, communication structures, and interaction protocols (La Malfa et al., 2025). Such designs provide limited capability for agents to adapt their behaviors according to evolving feedback during inference. Therefore, achieving effective and adaptive coordination among LLM agents remains a fundamental challenge.

To improve multi-agent reasoning, existing approaches explore communication, reflection, and prompt optimization mechanisms. Communication and debate-based methods enhance collaboration by enabling agents to exchange information and critique intermediate outputs (Qian et al., 2024; Du et al., 2024). Reflection-based approaches allow models to analyze previous attempts and generate corrective feedback for future reasoning (Madaan et al., 2023; Shinn et al., 2023). Meanwhile, prompt optimization methods search for improved instruction configurations before deployment (Khattab et al., 2024; Opsahl-Ong et al., 2024; Agrawal et al., 2026). However, these approaches mainly provide static guidance or incorporate feedback as additional textual context, rather than explicitly regulating agent behaviors according to deviations observed during inference. Consequently, current multi-agent reasoning systems remain largely open-loop, lacking a principled mechanism for continuous feedback-driven adaptation.

We address multi-agent reasoning from the perspective of closed-loop control, where inference-time collaboration is formulated as a dynamic feedback regulation process. Under this formulation, agent

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.

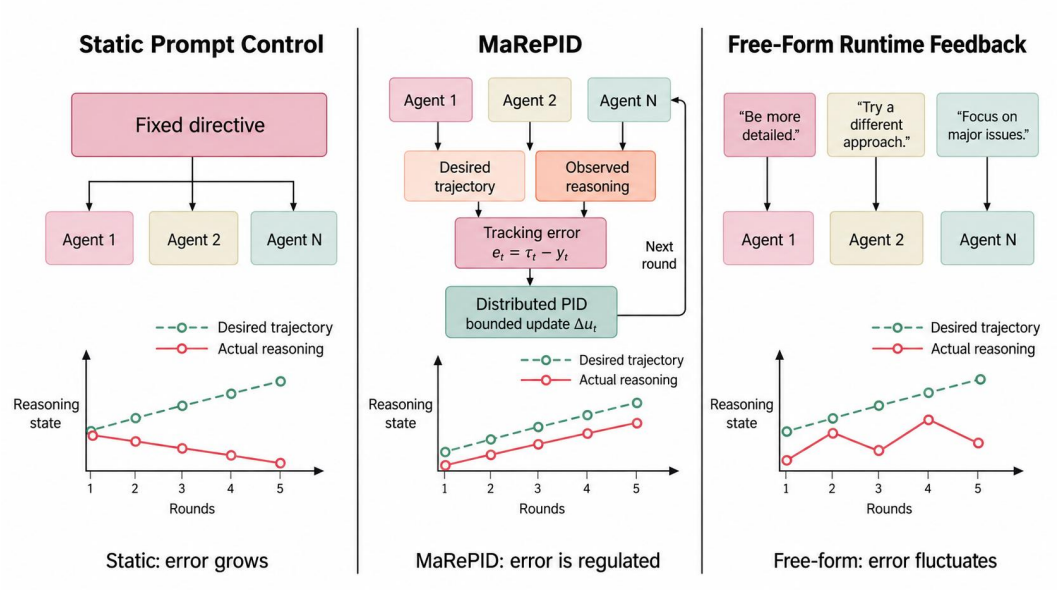

Figure 1: **Conceptual comparison of inference-time reasoning control.** Static prompting uses fixed directives, free-form feedback applies unstructured corrections, while MaRePID uses closed-loop PID regulation to track a desired reasoning trajectory across rounds.

reasoning behaviors are treated as system states, evaluation feedback generated during collaboration serves as measurement signals, and prompt directive updates act as control actions. This perspective connects multi-agent reasoning with classical feedback control principles, where system states are continuously monitored and adjusted toward desired objectives. Unlike conventional prompt optimization approaches that determine configurations before execution, a control-oriented formulation enables agents to adapt their behaviors throughout the reasoning trajectory.

Based on this perspective, we introduce **MaRePID**, a PID-inspired distributed feedback regulation framework for multi-agent reasoning. Rather than using PID as an optimization algorithm, MaRePID adapts proportional-integral-derivative principles to regulate prompt-level behaviors during inference. Each agent maintains controllable reasoning directives, receives evaluation feedback from the collaborative process, and updates its directives according to accumulated regulation errors. Furthermore, MaRePID introduces distributed feedback regulation, allowing each agent to incorporate both local feedback signals and information from neighboring agents. This design enables adaptive coordination among agents while preserving the black-box nature of underlying LLMs.

Our contributions are summarized as follows:

- We formulate multi-agent LLM reasoning as an inference-time feedback regulation problem, introducing a control-oriented perspective that connects agent behaviors, evaluation feedback, and prompt-level adjustments.
- We introduce **MaRePID**, a PID-inspired distributed feedback regulation framework that dynamically updates agent directives during inference without modifying model parameters.
- We conduct extensive experiments on multi-agent reasoning tasks, demonstrating the effectiveness of feedback regulation and analyzing the contribution of different control components.

2 Related Work

Multi-Agent LLM Systems. Recent research on multi-agent large language model (LLM) systems has explored how multiple agents can collaborate through role assignment, communication, and structured workflows. AutoGen (Wu et al., 2023) provides a flexible framework for constructing conversational agent systems with customizable interaction patterns. CAMEL (Li et al., 2023)

investigates role-playing based collaboration by assigning complementary roles to language agents. MetaGPT (Hong et al., 2024) and ChatDev (Qian et al., 2024) further organize multiple agents according to predefined software engineering roles and task workflows.

Beyond agent organization, recent studies investigate how interactions among agents improve reasoning performance. Debate-based approaches enhance reasoning through iterative discussion and critique among multiple agents (Du et al., 2024), while communicative agent frameworks study how information exchange protocols influence collective problem solving (Qian et al., 2024). Recent benchmarks further evaluate the capabilities and limitations of multi-agent LLM systems across diverse reasoning tasks (Zhu et al., 2025). However, existing multi-agent systems primarily rely on predefined roles, communication structures, and interaction protocols, where coordination behaviors are determined before or during execution rather than dynamically regulated according to evolving feedback (La Malfa et al., 2025).

Feedback-Based LLM Reasoning. A separate line of research improves LLM reasoning through iterative feedback and refinement. Chain-of-Thought prompting (Wei et al., 2022) and Self-Consistency (Wang et al., 2023) improve reasoning reliability by encouraging models to generate and aggregate intermediate reasoning trajectories. Beyond prompting, self-improvement methods introduce explicit feedback loops during inference. Self-Refine (Madaan et al., 2023) enables models to iteratively critique and revise their outputs, while Reflexion (Shinn et al., 2023) incorporates verbal feedback from previous attempts to guide future decisions. TextGrad (Yuksekgonul et al., 2025) further generalizes language-based optimization by propagating textual feedback through computation graphs.

Although these approaches demonstrate the effectiveness of feedback for improving individual reasoning trajectories, the feedback signals are typically used as additional textual information for refinement or optimization. They do not explicitly model deviations between current and desired reasoning states, nor do they provide a mechanism for continuously regulating agent behaviors throughout multi-agent interactions.

Control Perspectives for LLM Systems. Feedback control provides a general framework for regulating dynamic systems by adjusting control actions according to observed deviations between current and desired states (Åström & Murray, 2008). Classical proportional-integral-derivative (PID) controllers achieve feedback regulation by combining instantaneous errors, accumulated historical errors, and error trends to update control signals (Åström & Hägglund, 2001).This principle has been widely studied in engineering systems and distributed multi-agent control.

Recently, control-theoretic perspectives have been explored in the context of LLM systems. Controllability studies investigate how language model behaviors can be influenced through external interventions (Bhargava et al., 2023), while other works analyze LLM output controllability from a dynamical-system perspective (Soatto et al., 2023). Karn et al. (Karn, 2025) explore PID-inspired updates for controlling prompt variables in single-agent LLM settings. Meanwhile, distributed PID control has been extensively studied for coordinating conventional multi-agent dynamical systems (Liu, 2023).

However, existing control-oriented approaches mainly focus on controlling individual model behaviors or traditional dynamical systems, rather than regulating the emergent coordination dynamics of black-box multi-agent LLM reasoning. Applying feedback regulation to LLM-based collaboration introduces unique challenges, including semantic observations, noisy evaluations, and interaction-dependent coordination errors. MaRePID bridges this gap by formulating multi-agent reasoning as an inference-time feedback regulation problem, enabling distributed controllers to dynamically adjust agent behaviors based on evolving collaboration feedback.

3 METHOD

3.1 OVERVIEW

MaRePID treats each reasoning round as a feedback step. The agents and their original communication protocol generate responses, while a separate controller changes the prompt directives applied at the next round. The controller operates on numerical states, which are mapped to text before being

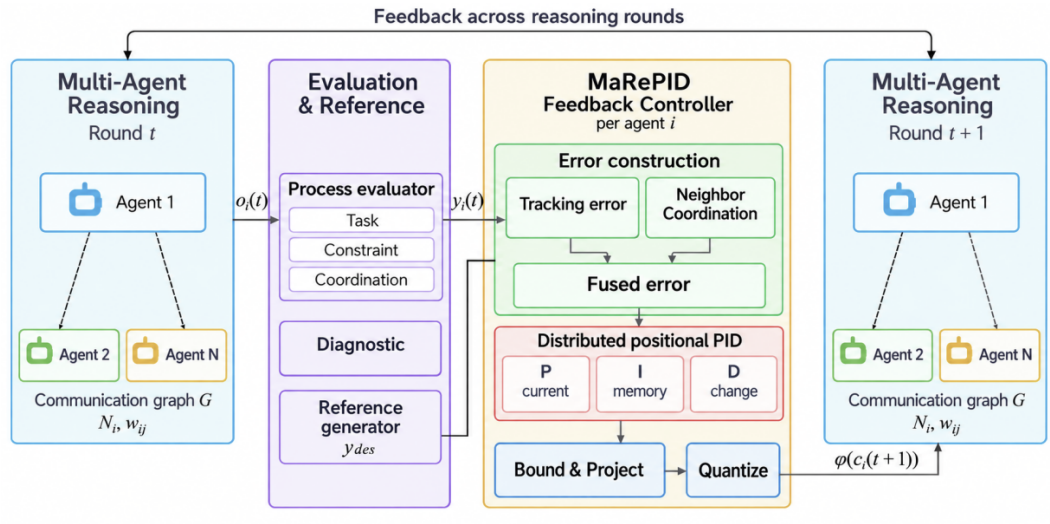

Figure 2: **MaRePID feedback loop.** Agents generate responses under quantized textual directives. Evaluator measurements and the latest neighboring control states form a control-space error. A bounded PID update determines the directives used in the next round.

passed to the frozen language models (Figure 2). This separation makes the error, the command, and the resulting prompt observable while treating the LLM response as a black-box process.

3.2 Feedback and Control Interface

Let $\mathcal{N} = \{1, \dots, N\}$ be the agents, $\mathcal{N}_i$ the neighbors of agent i, and R the number of reasoning rounds. The command $\mathbf{c}_i(t) \in \mathcal{C} = [0,1]^k$ is a normalized prompt-control vector, initialized at a task-specific baseline $\mathbf{c}_i(0)$. Agent i receives dialogue history $H_i(t)$ and the directive $\phi(\mathbf{c}_i(t))$. With prompt-composition map Γ_i and a frozen LLM policy π_i,

$$\begin{aligned} \mathbf{x}_i^{\mathrm{gen}}(t) &= \Gamma_i\big(H_i(t), \phi(\mathbf{c}_i(t))\big), \\ o_i(t) &\sim \pi_i(\cdot \mid \mathbf{x}_i^{\mathrm{gen}}(t)), \\ \mathbf{y}_i(t) &= \mathcal{E}_{\mathrm{proc}}(o_i(t)) \in [0,1]^m. \end{aligned} \tag{1}$$

Here $o_i(t)$ is the response and $\mathbf{y}_i(t)$ is the process evaluator's normalized measurement, with task quality, constraint satisfaction, and coordination as its three coordinates in the reported configurations. The evaluator is queried only after a response exists. If sensing is skipped, $\mathbf{y}_i(t)$ denotes the latest available measurement, held for the next update.

A reference generator produces a local numerical target $\mathbf{r}_i(t)$ from the available diagnostic state and neighbor summary. After graph-based reference alignment, the resulting estimate $\hat{\mathbf{r}}_i(t)$ defines the evaluator-space target used by the controller. We denote this interface by

$$\begin{aligned} \mathbf{r}_i(t) &= \mathcal{F}_{\mathrm{ref}}\big(\hat{\mathbf{s}}_{\mathrm{sys}}(t), \mathbf{h}_i(t), \mathrm{Prompt}_i; \theta_{\mathrm{ref}}\big), \\ \mathbf{y}_i^{\mathrm{des}}(t) &= \zeta_i(\hat{\mathbf{r}}_i(t)). \end{aligned} \tag{2}$$

Reference estimates also incorporate information along the directed communication graph:

$$\hat{\mathbf{r}}_i(t) = (1-\beta_r)\hat{\mathbf{r}}_i(t-1) + \gamma \sum_{j \in \mathcal{N}_i} w_{ij}\big(\hat{\mathbf{r}}_j(t-1) - \hat{\mathbf{r}}_i(t-1)\big) + \beta_r \mathbf{r}_i(t), \qquad t \geq 1. \tag{3}$$

We initialize $\hat{\mathbf{r}}_i(0) = \mathbf{r}_i(0)$. The reference interface and its task-specific calibration are documented in Table 9. In particular, the measurement and command need not have the same dimension ($m = 3$, $k = 4$). MaRePID therefore maps their discrepancy into control space with $\psi_i : \mathbb{R}^m \to \mathbb{R}^k$.

3.3 Distributed Error and Bounded PID Update

Let $\tilde{\mathbf{c}}_j(t)$ be the latest available command of neighbor j, $w_{ij} \geq 0$ its communication weight, and $a_i = \alpha$ if $\mathcal{N}_i \neq \varnothing$, or $a_i = 1$ otherwise. The controller forms

$$\mathbf{e}_i(t) = a_i \psi_i\big(\mathbf{y}_i^{\mathrm{des}}(t) - \mathbf{y}_i(t)\big) + (1 - a_i) \sum_{j \in \mathcal{N}_i} w_{ij}\big(\tilde{\mathbf{c}}_j(t) - \mathbf{c}_i(t)\big). \tag{4}$$

The first term is an evaluator-derived tracking correction. The second measures disagreement among control states. Both lie in $\mathbb{R}^k$. The already available commands on the right-hand side prevent a same-round circular definition. For the reported calibration, $\psi_i(z) = \mathbf{D}_i z$, with task-specific row-sparse $\mathbf{D}_i$ in Table 9. Define $\Delta\mathbf{e}_i(t) = \mathbf{e}_i(t) - \mathbf{e}_i(t-1)$ and initialize $\mathbf{e}_i(0) = \mathbf{0}$.

The bounded integral memory uses a conditional integration switch $\sigma_i(t)$ and a componentwise clip:

$$\begin{aligned} \mathbf{I}_i(t) &= \mathrm{clip}_{[-\mathbf{I}_{\max}, \mathbf{I}_{\max}]}\big(\mathbf{I}_i(t-1) + \sigma_i(t)\mathbf{e}_i(t)\big), \\ \sigma_i(t) &= \chi_i^{\mathrm{act}}(t-1)\big(1 - \mathrm{sat}_i(t-1)\big). \end{aligned} \tag{5}$$

The switch prevents integral accumulation immediately after an update was held or its offset saturated. MaRePID then computes a positional PID offset from the baseline command:

$$\begin{aligned} \mathbf{u}_i(t) &= \mathbf{K}_P \mathbf{e}_i(t) + \mathbf{K}_I \mathbf{I}_i(t) + \mathbf{K}_D \Delta\mathbf{e}_i(t), \\ \bar{\mathbf{u}}_i(t) &= \mathrm{sat}_{[\mathbf{u}_{\min}, \mathbf{u}_{\max}]}\big(\mathbf{u}_i(t)\big), \\ \mathbf{c}_i(t+1) &= \Pi_{\mathcal{C}}\big(\mathbf{c}_i(0) + \bar{\mathbf{u}}_i(t)\big). \end{aligned} \tag{6}$$

The last assignment applies when actuation is enabled. Otherwise $\mathbf{c}_i(t+1) = \mathbf{c}_i(t)$. Saturation and projection bound the requested command, rather than the LLM's realized behavior. Proportional action responds to the current discrepancy, integral action retains persistent discrepancies, and derivative action responds to the latest change.

For coordinate q, the command is quantized to a task-specific directive level:

$$\begin{aligned} s_i^{(q)}(t+1) &= \mathrm{round}\Big(s_{\min}^{(q)} + c_i^{(q)}(t+1)\big(s_{\max}^{(q)} - s_{\min}^{(q)}\big)\Big), \\ \mathbf{d}_i(t+1) &= \phi\big(\mathbf{c}_i(t+1)\big). \end{aligned} \tag{7}$$

The map ϕ verbalizes the resulting levels as prompt directives. Changes to $\mathbf{c}_i$ within one quantization cell can leave $\mathbf{d}_i$ unchanged. Directive ranges and coordinate meanings are given in Table 9.

3.4 Inference-Time Schedule and Optional Gates

Algorithm 1 MaRePID inference-time regulation (full sensing and always-on actuation)

1: Initialize $\mathbf{c}_i(1) \leftarrow \mathbf{c}_i(0)$, $\mathbf{I}_i(0) = \mathbf{0}$, and $\mathbf{e}_i(0) = \mathbf{0}$ for every agent i.
2: Generate $o_i(1)$ under $\phi(\mathbf{c}_i(1))$ and obtain $\mathbf{y}_i(1)$ for every agent.
3: **for** $t = 1, \ldots, R-1$ **do**
4: Update references using measurements available at round t.
5: **for** each agent i in parallel **do**
6: Form $\mathbf{e}_i(t)$ from its measurement and the latest available neighbor states using Eq. (4).
7: Update $\mathbf{I}_i(t)$ and $\mathbf{c}_i(t+1)$ using Eqs. (5)–(6).
8: **end for**
9: Generate $o_i(t+1)$ under $\phi(\mathbf{c}_i(t+1))$ and obtain $\mathbf{y}_i(t+1)$ for every agent.
10: **end for**
11: Return the final multi-agent response.

At round 1, each agent generates a response under its baseline directive and is evaluated. A round-t measurement then affects the command for round $t+1$, so an R-round run generates exactly R outputs per agent and has at most $R-1$ command updates. The main benchmark comparisons use full sensing and an actuation update at every subsequent round. Optional gates control the number of evaluator queries or command changes. Their sweeps are separate from the headline comparison.

For sensing, a diagnostic predictor uses information available at round t to obtain $\hat{\mathbf{s}}_{\mathrm{sys}}(t+1 \mid t)$ and a predicted reference displacement $\delta_i^{\mathrm{sen}}(t+1)$. A new evaluator query on the *already generated* output occurs when this displacement exceeds threshold $\delta_i^{\mathrm{sen},\star}$. Otherwise the previous measurement is reused. The actuation gate evaluates its event score $g_i(t)$ against deadband η_i, with a maximum-gap condition. When the gate is closed, the command is held. The deployed gate procedures, including their information timing, are specified in Appendix D.

Appendix C establishes the stability of the proposed nonlinear, quantized LLM-based closed-loop system by analyzing boundedness, deriving a local affine approximation, and characterizing the resulting stability conditions. The corresponding gated control procedure is presented in Appendix D.

4 Experimental Evaluation

We evaluate MaRePID from five perspectives. First, we measure its effect on task performance and agent coordination. Second, we test whether its gains depend on feedback regulation rather than changing directives over time. Third, we compare it with adaptive baselines under matched validation budgets. Fourth, we assess the contribution of each PID component. Finally, we examine its performance under changes in feedback quality and system conditions.

4.1 Experimental Setup

Tasks, teams, backbones and baselines. We evaluate MaRePID on four representative benchmark families: MultiAgentBench (MAB-Coding, MAB-Research, and MAB-Database), MATH, BigCodeBench, and AgentWebBench. These benchmarks cover collaborative reasoning, mathematical reasoning, code generation, and open-world information coordination.

We evaluate with Qwen3-8B, Qwen3-235B-A22B, and DeepSeek-V4. All methods use identical task splits, decoding configurations, and evaluation protocols.

Runtime configuration. Unless otherwise stated, each multi-agent run uses $N = 3$ agents for $R = 6$ reasoning rounds, yielding six responses and at most $R - 1 = 5$ controller updates per agent. Under full sensing, the evaluator is queried once per agent output, resulting in $NR = 18$ evaluator queries per task. MaRePID maintains a $k = 4$ dimensional prompt-control state and an $m = 3$ dimensional evaluator signal covering task quality, constraint satisfaction, and coordination. Detailed benchmark-specific configurations are provided in Appendix E.

The compared methods include single-agent CoT, self-consistency, No-PID, Static-tuned, Feedback-only, TextGrad, HiveMind, GEPA, and MIPROv2. Static-tuned searches the same directive space as MaRePID but keeps directives fixed during execution. Feedback-only receives evaluator signals without maintaining PID controller states.

Statistical testing. We use paired bootstrap testing over identical evaluation instances with 10,000 resamples. Confidence intervals are computed using percentile bootstrap and multiple comparisons are corrected using Holm correction.

4.2 Experiment 1: Main Performance Comparison

We first evaluate whether MaRePID improves multi-agent reasoning and coordination across diverse environments.

Table 1 summarizes performance across four benchmark families: MultiAgentBench, MATH, BigCodeBench, and AgentWebBench. Under the full-sensing, always-update setting, MaRePID achieves the highest reported scores among the compared methods across all evaluated columns. Relative to the No-PID multi-agent baseline, MaRePID improves the MAB-Coding task and coordination scores from $65.4/65.7$ to $74.6/71.9$, respectively. The contribution of individual PID components is further examined in the PID component ablation study.

Table 1: Main performance comparison across the evaluated benchmarks.

Method	MAB-Coding TS/CS	MAB-Research TS/CS	MAB-Database TS/CS	MATH Acc.	BigCodeBench Pass@1	AgentWebBench Overall
No-PID (base multi-agent system)	65.4 / 65.7	68.6 / 58.9	56.7 / 64.3	77.1	45.0	60.2
Single-agent CoT	60.6 / –	63.6 / –	51.9 / –	72.3	40.2	58.4
Single-agent self-consistency	62.8 / –	65.8 / –	54.1 / –	76.9	44.4	61.0
Static-tuned	69.4 / 69.2	72.9 / 65.9	61.1 / 68.7	80.0	49.4	67.1
Feedback-only	72.5 / 70.1	73.2 / 66.2	63.2 / 70.1	81.6	51.5	72.6
MIPROv2 (Opsahl-Ong et al., 2024)	72.6 / 71.1	74.2 / 66.4	63.3 / 71.2	80.8	51.1	72.4
GEPA (Agrawal et al., 2026)	73.7 / 70.0	74.9 / 67.1	64.1 / 70.9	81.6	54.0	72.8
TextGrad (Yuksekgonul et al., 2025)	73.5 / 70.2	75.8 / 68.6	64.0 / 71.1	82.0	53.5	71.9
HiveMind (Xia et al., 2026)	73.1 / 71.6	75.3 / 68.4	64.6 / 70.8	82.3	53.1	72.1
MaRePID	**74.6 / 71.9**	**76.0 / 69.8**	**65.0 / 72.8**	**83.1**	**54.6**	**78.5**

4.3 Experiment 2: PID Mechanism Isolation

We investigate whether the gains originate from closed-loop regulation rather than dynamic prompting alone.

We compare MaRePID against open-loop replay, shuffled feedback, and fixed dynamic schedules. In the controlled setting of Table 2, MaRePID scores 78.3, compared with 72.9 for open-loop replay, 74.0 for shuffled feedback, and 71.4 for a fixed dynamic schedule. The result supports feedback dependence in this setting. These variants also change the information available to the controller.

Table 2: PID mechanism isolation results. The score is computed under the controlled isolation setting.

Method	Overall Score
Fixed dynamic schedule	71.4
Open-loop replay	72.9
Shuffled feedback	74.0
Feedback-only	75.6
MaRePID	**78.3**

4.4 Experiment 3: Budget-Matched Comparison

Adaptive optimization approaches may benefit from additional search resources. We therefore compare methods under matched interaction and evaluation budgets.

At the displayed validation budget of 320 candidates and 19,200 runs, MaRePID's gain-screened profile achieves 77.5 task performance and 75.7 coordination scores. TextGrad and HiveMind achieve 77.4/73.1 and 76.9/74.8, respectively, under the same budget. This comparison evaluates the methods under matched validation resources.

As shown in Table 3, MaRePID achieves the highest task and coordination scores under the matched validation budget, reaching 77.5 and 75.7, respectively.

Table 3: Budget-matched comparison with adaptive optimization baselines.

Condition	Validation budget	Evaluator Calls	Task Score	Coordination Score
TextGrad (as reported)	24 trajectories (1,440 runs)	1,440	77.0	72.6
TextGrad (budget-matched)	320 trajectories (19,200 runs)	19,200	77.4	73.1
HiveMind (as reported)	32 trajectories (1,920 runs)	1,920	76.3	74.1
HiveMind (budget-matched)	320 trajectories (19,200 runs)	19,200	76.9	74.8
Feedback-only	none	–	74.9	72.5
MIPROv2 (budget-matched)	320 trials (19,200 runs)	19,200	75.5	73.9
MaRePID (gain screening)	320 tuples (19,200 runs)	19,200	77.5	75.7

4.5 Experiment 4: Ablation Study

We evaluate the contribution of individual PID components by removing the proportional, integral, and derivative terms.

In Table 4, the full PID controller achieves a task score of 77.5. Removing D, I, or P reduces the score to 75.9, 74.8, and 72.4, respectively. These ablation results indicate that the proportional,

integral, and derivative components provide complementary feedback signals for regulating multi-agent reasoning trajectories under the evaluated settings.

Table 4: Ablation of the PID controller components.

Variant	Task Score	Δ vs. full PID
Full PID (P+I+D)	**77.5**	–
P+I (no D)	75.9	-1.6
P+D (no I)	74.8	-2.7
I+D (no P)	72.4	-5.1
P only	73.5	-4.0
I only	71.8	-5.7
D only	70.2	-7.3
P+I with K_I doubled	75.3	-2.2

4.6 Experiment 5: Robustness Analysis

We evaluate robustness under evaluator changes, agent scaling, feedback noise, and delayed measurements. Appendix G summarizes the evaluation settings and robustness conditions. These experiments examine how MaRePID responds to variations in feedback quality, system scale, and measurement timing under different operating conditions.

Overall, MaRePID retains 77.0 task performance at a 64% sensing rate—only 0.5 points below full sensing—despite substantially fewer evaluator measurements.

4.7 Additional Analysis

The appendix provides the theoretical analysis of the closed-loop dynamics, including boundedness, local affine approximation, stability conditions, gated algorithms, and experimental configuration details. The theoretical characterization holds under the assumptions specified in the analysis.

Table 5 illustrates how MaRePID adaptively adjusts its control state across reasoning rounds. Table 6 shows that prediction-triggered sensing substantially reduces evaluator usage with only a modest performance drop, while outperforming random skipping under matched query budgets.

Table 5: Representative MaRePID controller trajectory across reasoning rounds.

Round t	1	2	3	4	5	6
$\tilde{\mathbf{c}}_i^{\mathrm{up}}(t)$ (upstream state, $j=1$)	0.520	0.790	0.621	0.796	0.693	0.805
$\mathbf{c}_i(t)$ (control used at round t)	0.520	0.682	0.670	0.663	0.753	0.679
$\mathbf{y}_i(t)$ (evaluator)	0.525	0.687	0.675	0.668	0.758	0.684
$\mathbf{e}_i(t)$ (fused error)	+0.195	+0.141	+0.085	+0.162	+0.031	+0.150
$\mathbf{I}_i(t)$ (integral)	0.195	0.336	0.421	0.583	0.615	0.765
$\Delta\mathbf{e}_i(t)$	+0.195	-0.054	-0.056	+0.077	-0.131	+0.119
$\mathbf{u}_i(t)$ (PID output)	0.162	0.150	0.143	0.233	0.159	–
$\mathbf{c}_i(t{+}1) = \mathrm{clip}_{\mathcal{C}}(\mathbf{c}_i(0) + \mathrm{sat}(\mathbf{u}_i(t)))$	0.682	0.670	0.663	0.753	0.679	–

Table 6: Prediction-triggered sensing trade-off.

Threshold $\delta^{\mathrm{sen},\star}$	Task Score	Realized sensing rate	Evaluator calls per task	Δ Task Score vs. full sensing	Predictor η_{pred}	Random skipping (matched queries)
0.00	77.5	100%	18.0	–	0.021	77.5
0.03	77.3	84%	15.2	-0.2	0.024	77.0
0.06	77.0	64%	11.5	-0.5	0.031	75.9
0.10	75.8	42%	7.5	-1.7	0.047	73.4
0.16	73.1	22%	3.9	-4.4	0.079	69.8
$\to\infty$	66.4	17%	3.0	-11.1	0.152	66.4

Realized sensing rate and predictor error η_{pred} (mean absolute error of the normalized one-step predictor on the diagnostic state) are measured from the logged runs under each threshold setting.
Random skipping uses the same evaluator query budget as the corresponding threshold setting, with query locations sampled uniformly over the $N \cdot R = 18$ opportunities.
The $\delta^{\mathrm{sen},\star} \to \infty$ row still performs the initial mandatory query of Algorithm 1 ($N = 3$ calls/task).

5 Discussion and Conclusion

MaRePID instantiates a bounded, inspectable prompt-level feedback interface for multi-agent LLMs. Its principal design choice is to convert process measurements and agent-specific targets into a quantized directive state with PID-based across-round regulation and neighbour coupling. Rather than introducing a new PID law, MaRePID adapts PID principles to regulate prompt-level directives during inference, while the mapping between directive strength and LLM behaviour remains task- and model-dependent. The aggregate comparisons show improvements on MAB-Coding from $65.4/65.7$ to $74.6/71.9$ in task and coordination scores relative to No-PID, while comparisons with adaptive text-based methods vary across benchmarks and backbones.

The theoretical analysis in Appendix C establishes boundedness and stability conditions for the nonlinear, clipped, and quantized LLM closed-loop system under the stated assumptions. The local affine approximation and corresponding Schur-stability analysis characterize the behavior of the regulated dynamics while accounting for quantization and bounded control updates. Sensing and actuation gates are optional extensions evaluated separately from the headline full-sensing, always-update comparison. The controlled mechanism-isolation experiments further verify the role of feedback-based regulation compared with dynamic scheduling alternatives. Further work may study learned and predictive controllers under broader settings, as well as automatic discovery of task-specific actuators.

AI use statement

Generative AI tools were used to assist manuscript editing, improve writing clarity, and enhance the presentation of the text.

Ethics statement

The work studies inference-time control of multi-agent language-model reasoning on public benchmarks. It involves no human subjects, personal data or model training, and we identify no application-specific ethical risks beyond those common to language models generally.

Reproducibility statement

The manuscript specifies the MaRePID control formulation, including the feedback interface, update equations, inference-time schedule, information timing, task-specific directive ranges, baseline configurations, controller hyperparameters, and statistical evaluation procedure. The appendix provides additional implementation details, notation, scenario configurations, templates, communication graph structures, per-backbone results, controller settings, and sensitivity analyses required to reproduce the reported evaluation setup.

References

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0e9e708b6f48e14fd0ac29e167413f76-Paper-Conference.pdf.

Aman Bhargava, Cameron Witkowski, Shi-Zhuo Looi, and Matt Thomson. What's the magic word? a control theory of llm prompting, 2023. URL https://arxiv.org/abs/2310.04444.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, zili wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6507b115562bb0a305f1958ccc87355a-Paper-Conference.pdf.

Rupesh Raj Karn. Linear feedback control systems for iterative prompt optimization in large language models. *arXiv preprint arXiv:2501.11979*, 2025. URL https://arxiv.org/abs/2501.11979.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Emanuele La Malfa, Gabriele La Malfa, Samuele Marro, Jie Zhang, Elizabeth Black, Michael Luck, Philip Torr, and Michael Wooldridge. Large language models miss the multi-agent mark. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4663. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/cc14cadd13ca5a13ba899cb49351c139-Paper-Position_Paper_Track.pdf.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023. doi: 10.52202/075280-2264. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf.

Guo-Ping Liu. Tracking control of multi-agent systems using a networked predictive PID tracking scheme. *IEEE/CAA Journal of Automatica Sinica*, 10(1):216–225, 2023. doi: 10.1109/JAS.2023.123030.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019.

Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377.

Stefano Soatto, Paulo Tabuada, Pratik Chaudhari, and Tian Yu Liu. Taming AI bots: Controllability of neural states in large language models. *arXiv preprint arXiv:2305.18449*, 2023. URL https://arxiv.org/abs/2305.18449.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Yihan Xia, Taotao Wang, Shengli Zhang, Zhangyuhua Weng, Bin Cao, and Soung Chang Liew. HiveMind: Contribution-guided online prompt optimization of LLM multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29767–29774, 2026. doi: 10.1609/aaai.v40i35.40222.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639:609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

Karl Johan Åström and Richard M. Murray. *Feedback Systems*. Princeton University Press, 2008. ISBN 9781400828739. doi: 10.1515/9781400828739. URL https://doi.org/10.1515/9781400828739.

K.J. Åström and T. Hägglund. The future of pid control. *Control Engineering Practice*, 9(11): 1163–1175, 2001. doi: 10.1016/S0967-0661(01)00062-4. URL https://www.sciencedirect.com/science/article/pii/S0967066101000624.

SUPPLEMENTARY MATERIAL

A NOTATION

Table 7 collects the main symbols. We write $N = |\mathcal{N}|$ for the number of agents and R for the reasoning horizon. The process evaluator is $\mathcal{E}_{\text{proc}}$, dialogue history is $H_i(t)$, and the maximum actuation gap is $T_i^{\max}$. The subscripts p, i, d in the gain parameters refer to proportional, integral and derivative action.

Table 7: Notation used in MaRePID.

Symbol	Meaning	Where it is used
$\mathcal{N} = \{1, \dots, n\}, \mathcal{N}_i; N = \lvert\mathcal{N}\rvert$	Agent set, neighbour set of i, and team size (written N when a count is meant)	Eq. (4)
$\mathcal{G}, w_{ij}, \mathcal{L}_w$	Communication graph, normalized weights, graph Laplacian	Eq. (4), Eq. (13)
t, R	Reasoning round and total number of rounds	Eq. (1)
$o_i(t), \mathbf{x}_i^{\text{gen}}(t), \Gamma_i, \pi_i$	Raw response, generation input, prompt-composition map, and frozen LLM policy	Eq. (1)
$\mathbf{c}_i(t) \in [0,1]^k$	Normalized prompt-control state (the actuated variable)	Section 3.2
$\mathcal{C} = [0,1]^k$	Bounded prompt-control space	Section 3.2
$\mathbf{c}_i(0)$	Baseline control position held by the positional loop	Eq. (6)
$\phi, s_{\min}^{(q)}, s_{\max}^{(q)}$	Control-to-directive map and its discrete level range	Eq. (7)
$\mathcal{E}_{\text{proc}}(\cdot)$	Process evaluator (the measurement map)	Eq. (1) ff.
$\mathbf{y}_i(t) \in [0,1]^m$	Normalized evaluator measurement	Eq. (4)
k, m	Dimensions of the prompt-control state and evaluator measurement, respectively	Section 3.2
$\mathcal{F}_{\text{sys}}, \hat{\mathbf{s}}_{\text{sys}}$	Diagnostic interface and task-level diagnostic state	Eq. (2)
$\mathcal{F}_{\text{ref}}, \theta_{\text{ref}}$	Reference generator and its parameters	Eq. (2)
$\mathbf{r}_i(t)$	Local numerical reference produced by $\mathcal{F}_{\text{ref}}$	Eq. (2)
$\tilde{\mathbf{r}}_i(t), \mathbf{y}_i^{\text{des}}(t) = \zeta_i(\tilde{\mathbf{r}}_i(t)), \gamma, \beta_r$	Graph-aligned reference estimate, its evaluator-space image, alignment gain, and innovation gain	Eqs. (2)–(3)
$\mathbf{h}_i(t), \mathbf{H}_{\text{sys}}(t)$	Neighbour summary and system-level history	Eq. (2)
ψ_i	Calibration map from evaluator space to control-space corrections	Eq. (4)
$\mathbf{q}_i(t)$	Instantaneous control-space pseudo-reference	Eq. (12)
α	Tracking–coordination tradeoff coefficient	Eq. (4)
$\tilde{\mathbf{c}}_j(t)$	Latest available neighbour control state	Eq. (4)
$\mathbf{e}_i(t), \Delta\mathbf{e}_i(t)$	Fused error and its round-to-round difference	Eq. (4)
$\mathbf{I}_i(t), \mathbf{I}_{\max}, \sigma_i(t)$	Integral memory, its clip, and the conditional-integration switch	Eq. (5)
$\mathbf{K}_P, \mathbf{K}_I, \mathbf{K}_D$	Positional PID gain matrices	Eq. (6)
$\beta_p, \beta_i, \beta_d$	Scalar validation-search gain parameters; the subscripts are the proportional/integral/derivative mnemonics, not agent indices	Table 9
$\mathbf{u}_i(t), \tilde{\mathbf{u}}_i(t), \mathbf{u}_{\min}, \mathbf{u}_{\max}$	PID output, saturated PID output, saturation bounds	Eq. (6)
$\mathcal{P}_\theta, \hat{\mathbf{s}}_{\text{sys}}(t+1 \mid t)$	One-step diagnostic predictor and its prediction	Appendix D
$\delta_i^{\text{sen}}(t), \delta_i^{\text{sen},\star}, \chi_i^{\text{sen}}(t)$	Predicted reference displacement, its threshold, and the sensing gate	Appendix D
$\chi_i^{\text{act}}(t), g_i(t), \eta_i, T_i^{\max}, \lambda_i, \mathbf{Q}_i, \mathbf{R}_i$	Actuation gate, event score, deadband, maximum gap, weights and weight matrices	Appendix D
t_i^{last}	Round of the most recent actuation	Appendix D
$H_i(t)$	Dialogue history, always written with a time argument; the maximum actuation gap is the separate scalar $T_i^{\max}$ above	Eq. (1)
$L_{\text{cal}}, \bar{L}_{\text{sur}}, L_{\psi,i}$	Gain-search scale, assumed local response bound, and calibration-map norm	Table 9, Assumptions C.2–C.3
g, h	Calibrated scalar response slope and effective fused-error slope	Theorem C.5
d_r, Δ_r, q_r	Numerical reference dimension, input-variation bound and filter decay rate	Assumption C.4, Lemma C.7
$\mathbf{A}_r, \mathbf{B}_r, \mathbf{G}_r$	Stacked reference-alignment dynamics, input map and frozen-reference equilibrium map	Lemma C.7
$\eta_{\ell,t}, B_\ell, w_t$	Residual input, its state-space injection map and total disturbance	Appendix C.3
N, R	Agents per task and rounds per task (at most $N \cdot R$ evaluator calls per task)	Section 4.1

B CLOSEST PRECEDENTS

Table 8 compares the actuator, command bounds, feedback signal, neighbour coupling, measurement schedule and analysis of MaRePID and related methods.

Table 8: Closest precedents. The table records the actuated variable, whether it is bounded by construction, the signal that forms the feedback error, whether agents are coupled through the error, whether the method decides *when* to measure, and what kind of analysis is offered. "–" indicates the property is not part of the method.

Method	**Actuated variable**	**Bounded**	**Feedback error formed from**	**Neighbour coupling**	**Decides when to query**	**Analysis**
Karn (2025)	prompt text	no	measured score	no (per-agent)	no	control framing
Liu (2023)	physical plant input	yes (plant)	neighbour-fused tracking error	yes	no	closed-loop stability
TextGrad	free-text variable	no	textual critique	no	no	–
HiveMind	free-text prompt	no	attribution weight	via attribution	no	–
GEPA / MIPROv2	free-text prompt	no	validation trace	no	offline search	–
TMPC	subgoal buffer text	no	rollout value	no	no	predictive planning
SupervisorAgent	supervision call	no	filter score	no	yes (supervision)	–
MaRePID	**quantized directive**	**yes**	**measured score + neighbour state**	**yes**	**yes (measurement)**	**local dynamics and bounds**

Two features shape the analysis relative to Liu (2023): the actuator is a quantized textual directive, and the response is produced by a black-box LLM stack. Appendix C therefore combines exact bounds for the implemented clips with a local affine response model. The process evaluator is also a separate paid call, making measurement timing part of the control design in Section 3.4. Relative to Karn (2025), the bounded numeric actuator supports a state-space description of the update and the scalar stability condition of Appendix C.

C Analysis of the Feedback Dynamics

C.1 Notation and update timing

Let $\mathbf{c}_i^0 = \mathbf{c}_i(0)$ denote the fixed anchor and let $\mathbf{c}_i(t)$ be the command used in round t. A measurement at round t determines the command for round $t+1$, giving $R-1$ command updates over an R-round trajectory. We first study the corresponding local affine dynamics, then establish bounds for the clipped controller and a tracking estimate for the reference filter.

Write $\mathcal{W} = (w_{ij})$, $d_i = \sum_j w_{ij}$, $D_w = \mathrm{diag}(d_i)$ and $\mathcal{L}_w = D_w - \mathcal{W}$. Let $a_i = 1$ for an agent with an empty upstream neighbour set and $a_i = \alpha$ otherwise. Under synchronous communication, Eq. (4) becomes

$$\mathbf{e}_i(t) = a_i \mathbf{D}_i \big(\mathbf{y}_i^{\mathrm{des}}(t) - \mathbf{y}_i(t)\big) - (1-a_i) \sum_j (\mathcal{L}_w)_{ij} \mathbf{c}_j(t).$$

The gain family in Table 9 uses the same local mixing weight a_i. For the selected parameters, the scalar gains are $(0.50, 0.25, 0.08)$ at non-root agents and $(0.30, 0.09, 0.048)$ at roots.

C.2 Assumptions

Assumption C.1 (Bounded interface)**.** The initial command belongs to $\mathcal{C} = [0,1]^k$. Executed commands are projected onto $\mathcal{C}$, the positional offset is saturated componentwise, and the integral state is clipped. Measurements, including reused measurements, and desired targets belong to $[0,1]^m$. The neighbour weights are nonnegative, with $d_i = 1$ for nonempty neighbour sets and $d_i = 0$ at roots.

Assumption C.2 (Local response model)**.** For fixed dialogue history and external context, let $f = (f_1^\top, \ldots, f_N^\top)^\top$ be a differentiable model of the stacked response on a neighbourhood $\mathcal{D}$, with Jacobian G. Whenever a sensitivity bound is used, assume that a finite constant $\bar{L}_{\mathrm{sur}}$ satisfies

$$\|f(c) - f(c')\| \le \bar{L}_{\mathrm{sur}} \|c - c'\|, \qquad c, c' \in \mathcal{D}.$$

The measured response is represented by this model plus a residual accounting for quantization, model approximation and measurement variability.

Table 9: Hyperparameters and controller configuration used in the reported experiments.

Quantity	Value used in every reported run	Reference
Control coordinates k	$k = 4$; four coordinates per benchmark	Section 3.3
Evaluator coordinates m	$m = 3$ (task quality, constraint satisfaction, coordination)	Section 3.2
Calibration map ψ_i	row-sparse $\mathbf{D}_i \in \mathbb{R}^{4\times 3}$, one nonzero per row (rows follow the listed directive order; columns are task quality, constraint satisfaction, coordination). Coding/BigCodeBench: $[0.82, 0, 0]$; $[0, 0.71, 0]$; $[0.90, 0, 0]$; $[0, 0, 0.65]$ Research: $[0.88, 0, 0]$; $[0.79, 0, 0]$; $[0, 0.66, 0]$; $[0, 0, 0.74]$ Database: $[0, 0.69, 0]$; $[0.85, 0, 0]$; $[0.77, 0, 0]$; $[0, 0, 0.71]$ MATH: $[0.91, 0, 0]$; $[0.83, 0, 0]$; $[0, 0.76, 0]$; $[0, 0, 0.62]$	Assumption C.3, Eq. (4)
Reference image ζ_i	identity after clipping to $[0, 1]$	Eq. (2)
Alignment gain γ	0.20 per round	Eq. (3)
Innovation gain β_r	0.35 per round	Eq. (3)
Initial control $\mathbf{c}_i(0)$	Research 0.55; Coding/BigCodeBench 0.52; Database 0.48; MATH 0.50 (all four coordinates)	Eq. (6)
Increment saturation $\mathbf{u}_{\max}$	$0.5 \cdot \mathbf{1}$, symmetric	Eq. (6)
Integral clip $\mathbf{I}_{\max}$	$5 \cdot \mathbf{1}$	Eq. (5)
Event deadband η_i	0.00 (always-update) for all benchmark results; sweep in Appendix D	Appendix D
Event weights $\lambda_i, \mathbf{Q}_i, \mathbf{R}_i$	$\lambda_i = 1.0$, $\mathbf{Q}_i = \mathbf{R}_i = \mathbf{I}$	Appendix D
Maximum gap $T_i^{\max}$	1 (no gap) for all benchmark results	Appendix D
Sensing threshold $\delta^{\mathrm{sen},\star}$	0.00 (full sensing) for all benchmark results; sweep in Appendix D	Appendix D
Directive ranges $s_{\min}, s_{\max}$	Coding/BigCodeBench: LOC $[20, 220]$, cyclomatic complexity $[1, 20]$, reasoning depth $[1, 10]$, consistency $[0, 10]$	Eq. (7)
	Database: query complexity $[1, 10]$, evidence completeness $[1, 10]$, diagnostic depth $[1, 10]$, collaboration efficiency $[0, 10]$	Eq. (7)
	MATH: reasoning steps $[3, 15]$, domain-knowledge range $[1, 10]$, symbol accuracy $[1, 10]$, consensus threshold $[0, 1]$	Eq. (7)
	Research: search breadth $[1, 10]$, search depth $[1, 10]$, authority bias $[0, 1]$, information integration $[0, 1]$	Eq. (7)
Quantized space size $\lvert\mathcal{C}_{\mathrm{disc}}\rvert$	Coding 442,200; Database 11,000; Research 400; MATH 2,600	Eq. (7), Eq. (7)

Assumption C.3 (Evaluator-to-control calibration)**.** The calibration map is $\psi_i(z) = \mathbf{D}_i z$, where $\mathbf{D}_i \in \mathbb{R}^{k\times m}$ has the row-sparse coefficients of Table 9 and finite norm $L_{\psi,i} = \|\mathbf{D}_i\|$. Here $k = 4$

and $m = 3$. The scalar response slope used below includes the corresponding calibration coefficient and its sign.

The gain-family scale L_{cal} and the entries of $\mathbf{D}_i$ serve different purposes: L_{cal} sets a common search scale, whereas $\mathbf{D}_i$ maps each measured deviation to its control coordinate. Both are retained explicitly in the local equations.

Assumption C.4 (Reference filter)**.** The deployed communication graphs are directed chains, with unit weight on each upstream edge and a zero row at the root. In the reference-alignment equation, write $r_i(t) \in \mathbb{R}^{d_r}$ for the numerical representation of the reference; evaluator-space targets have $d_r = m$. Assume

$$\gamma \geq 0, \qquad 0 < \beta_r < 1, \qquad d_i \leq d_{\max}, \qquad \beta_r + \gamma d_{\max} \leq 1,$$

and let the stacked input satisfy $\|r(t+1) - r(t)\|_\infty \leq \Delta_r$.

The parameters $\beta_r = 0.35$, $\gamma = 0.20$ and $d_{\max} = 1$ satisfy these inequalities. The root has self-weight $1 - \beta_r = 0.65$; each remaining agent has self-weight $1 - \beta_r - \gamma = 0.45$ and upstream weight 0.20.

C.3 Local network dynamics and perturbations

For the affine analysis, fix the numerical target, dialogue history and external context, and consider full sensing and actuation in a neighbourhood where all clipping and saturation operations are inactive. Let $n_c = Nk$,

$$\mathsf{D} = \mathrm{blkdiag}(\mathbf{D}_i), \qquad \mathsf{S} = \mathrm{diag}(a_i) \otimes I_k,$$

and let $G \in \mathbb{R}^{Nm \times Nk}$ be the Jacobian of the stacked response model, with G_i its ith block row. The local fused-error matrix is

$$H = \mathsf{S}\mathsf{D}G + (I_{n_c} - \mathsf{S})(\mathcal{L}_w \otimes I_k).$$

Let K_P, K_I and K_D denote the block-diagonal gain matrices, and put $\mathsf{K}_\Sigma = \mathsf{K}_P + \mathsf{K}_I + \mathsf{K}_D$. About an equilibrium with zero fused error, let x_t and j_t be the command and integral deviations. Writing $e_t = -Hx_t + v_t$ and

$$\xi_t = (x_t^\top, e_{t-1}^\top, j_{t-1}^\top)^\top,$$

gives

$$\xi_{t+1} = A_{\mathrm{cl}}\xi_t + Bv_t, \qquad A_{\mathrm{cl}} = \begin{pmatrix} -\mathsf{K}_\Sigma H & -\mathsf{K}_D & \mathsf{K}_I \\ -H & 0 & 0 \\ -H & 0 & I_{n_c} \end{pmatrix}, \qquad B = \begin{pmatrix} \mathsf{K}_\Sigma \\ I_{n_c} \\ I_{n_c} \end{pmatrix}. \tag{8}$$

The stored error e_{t-1} is part of the controller state, with initialization $e_0 = 0$. The order of matrix multiplication in Eq. (8) retains the calibration and cross-coordinate response couplings.

Residual inputs. Generation and evaluator noise originate in evaluator space, directive quantization in control space, and reference variation in the target coordinates. For example, a measurement residual ε_t contributes $-\mathsf{S}\mathsf{D}\varepsilon_t$ to v_t, while a desired-target perturbation $\delta y_t^{\mathrm{des}}$ contributes $\mathsf{S}\mathsf{D}\delta y_t^{\mathrm{des}}$. More generally, write the state-space disturbance as

$$w_t = \sum_\ell B_\ell \eta_{\ell,t},$$

where B_ℓ maps the corresponding residual into the augmented state. On a skipped evaluator query, the controller receives the reused-measurement residual; prediction error enters through the sensing decision. The evaluator standard deviation over 400 repeated scorings is 0.021, which characterizes the observed measurement variability. Uniform perturbation estimates below use an explicit bound on w_t.

Perturbation estimate. For $\xi_{t+1} = A\xi_t + w_t$, repeated substitution gives

$$\xi_t = A^t\xi_0 + \sum_{s=0}^{t-1} A^{t-1-s} w_s.$$

Suppose $\|A^j\| \le Mq^j$ for all $j \ge 0$, where $M \ge 1$ and $0 < q < 1$. Then

$$\|\xi_t\| \le Mq^t\|\xi_0\| + \frac{M(1-q^t)}{1-q} \max_{0\le s<t} \|w_s\|.$$

For a time-varying linear model, the common quadratic inequality $A_t^\top P A_t \preceq q^2 P$, with $P \succ 0$, gives the corresponding estimate in the P-norm. These conditions specify the stability hypothesis needed for a geometric disturbance bound.

Null directions and accessible states. At a root, $\mathbf{D}_i \in \mathbb{R}^{4\times 3}$, so the corresponding block row of H has rank at most three. Consequently H is singular on the full command space. For $Hv = 0$ and invertible K_I,

$$A_{\mathrm{cl}} \begin{pmatrix} v \\ 0 \\ \mathsf{K}_I^{-1} v \end{pmatrix} = \begin{pmatrix} v \\ 0 \\ \mathsf{K}_I^{-1} v \end{pmatrix}.$$

Thus the unrestricted affine model contains a unit eigenvalue. With zero initial integral, isotropic local gains and inactive clipping, the root's control offsets and integral states lie in $\operatorname{im} \mathbf{D}_i$. The dynamics on this accessible subspace depend on the calibrated response $\mathbf{D}_i G_i$ and the network couplings.

C.4 Scalar stability condition

Consider a scalar response $y = b_y + g_{\mathrm{plant}} c$ with row calibration κ, and set $g = \kappa g_{\mathrm{plant}}$. Holding the target and upstream controls fixed gives

$$e_t = b - hc_t, \qquad h = \alpha g + 1 - \alpha$$

at a non-root agent, and $h = g$ at a root. For full sensing and actuation with inactive bounds, the scalar recurrence is

$$I_t = I_{t-1} + e_t, \qquad c_{t+1} = c^0 + K_P e_t + K_I I_t + K_D(e_t - e_{t-1}). \tag{9}$$

For $h > 0$ and $K_I > 0$, an equilibrium of Eq. (9) is

$$e_* = 0, \qquad c_* = b/h, \qquad I_* = (c_* - c^0)/K_I.$$

Assume that c_*, $c_* - c^0$ and I_* lie in the interiors of the command, offset and integral intervals, respectively. With $x_t = c_t - c_*$ and $j_t = I_t - I_*$, the state equation is

$$\xi_{t+1} = A(h)\xi_t, \qquad \xi_t = \begin{pmatrix} x_t \\ e_{t-1} \\ j_{t-1} \end{pmatrix}, \qquad A(h) = \begin{pmatrix} -h(K_P + K_I + K_D) & -K_D & K_I \\ -h & 0 & 0 \\ -h & 0 & 1 \end{pmatrix}. \tag{10}$$

Indeed, $j_t = j_{t-1} - hx_t$ and $x_{t+1} = -h(K_P + K_I + K_D)x_t - K_D e_{t-1} + K_I j_{t-1}$. Taking $\det(zI - A(h))$ yields the characteristic polynomial in Theorem C.5.

Theorem C.5 (Scalar Schur stability)**.** *Consider Eq.* (9) *with fixed target and context, full sensing and actuation, and inactive clipping and saturation. Suppose the interior equilibrium above exists, $h > 0$, $K_P, K_D \ge 0$ and $K_I > 0$. The affine equilibrium is asymptotically stable if and only if*

$$h(2K_P + K_I + 4K_D) < 2.$$

Proof. Set $a = hK_P \ge 0$, $b = hK_I > 0$ and $d = hK_D \ge 0$. Write

$$p(z) = z^3 + A_1 z^2 + A_2 z + A_3, \qquad A_1 = a + b + d - 1, \quad A_2 = -a - 2d, \quad A_3 = d.$$

The strict cubic Jury conditions are

$$|A_3| < 1, \qquad 1 + A_1 + A_2 + A_3 > 0,$$
$$1 - A_1 + A_2 - A_3 > 0, \qquad 1 - A_2 + A_1 A_3 - A_3^2 > 0.$$

Substitution reduces them to

$$d < 1, \qquad b > 0, \qquad 2 - 2a - b - 4d > 0, \qquad 1 + a + d + ad + bd > 0.$$

The last inequality holds for $a, d \ge 0$ and $b > 0$. Moreover, $2a + b + 4d < 2$ implies $d < 1/2$. The conditions are therefore equivalent to $h(2K_P + K_I + 4K_D) < 2$. □

At the boundary $h(2K_P + K_I + 4K_D) = 2$, the polynomial has the root $z = -1$. With $K_D = 0$, it factors as

$$p(z) = z\big(z^2 + [h(K_P + K_I) - 1]z - hK_P\big),$$

and the stability condition reduces to $h(2K_P + K_I) < 2$.

C.5 LOCAL SENSITIVITY ESTIMATES

On held-out validation prompts, we perturb one normalized control coordinate by $\Delta = 0.1$ and compute

$$\widehat{L}_{\mathrm{sur}} = \frac{\|\bar{\mathbf{y}}' - \bar{\mathbf{y}}\|_2}{\|\Delta \mathbf{c}\|_2}.$$

This statistic summarizes the magnitude of the directional finite-difference response. Table 10 reports its median, interquartile range and 90th percentile over the sampled controls. These sample statistics set the scale of the gain search. Assumption C.2 separately specifies the local response regularity used in the analysis.

Table 10: Directional finite-difference sensitivity on held-out validation tasks (Qwen3-8B). The median of the four group medians is 0.97, rounded to $L_{\mathrm{cal}} = 1.0$ for the gain family.

Control group	Median $\hat{L}_{\mathrm{sur}}$	IQR	90th percentile
Research actuators	0.94	[0.71, 1.22]	1.55
Coding / BigCodeBench actuators	1.00	[0.82, 1.31]	1.68
Database actuators	0.87	[0.66, 1.14]	1.47
MATH actuators	1.21	[0.95, 1.52]	1.83

C.6 GAIN SELECTION AND SCALAR SENSITIVITY ANALYSIS

The validation gain search uses $\beta_p \in \{0.18, 0.24, 0.30, 0.42, 0.54\}$, $\beta_i \in \{0.30, 0.45, 0.60, 0.90\}$, $\beta_d \in \{0.08, 0.16, 0.24, 0.32\}$ and $\alpha \in \{0.4, 0.6, 0.8, 1.0\}$, giving 320 tuples. Each tuple is converted to gain values using the configuration summarized in Table 9, with $L_{\mathrm{cal}} = 1$, and evaluated on 60 held-out validation tasks per setting.

Validation screening. For a tuple θ and the 90th-percentile sensitivity L_s of setting s, put $\hat{g}_s = \alpha L_s + 1 - \alpha$ and define the heuristic screening scores

$$\varrho_s(\theta) = \left| 1 - \frac{\hat{g}_s K_I(\theta)}{1 + \hat{g}_s K_P(\theta)} \right|, \qquad \kappa_s(\theta) = \frac{K_D(\theta)}{K_P(\theta)} \hat{g}_s. \tag{11}$$

A tuple passes the screening when $\varrho_s \le 0.90$ and $\kappa_s \le 0.5$. These thresholds favour sufficient integral action and moderate derivative gain relative to the sampled response scale. The resulting candidate set is used for validation-based selection; the scalar stability region is given separately by Theorem C.5.

The screen accepts 220, 227, 227, 216 and 218 tuples in Research, Coding, BigCodeBench, Database and MATH, respectively. Across settings, 213 tuples pass in at least four cases and 199 pass in all five. Of the 121 remaining tuples, 81 fail only the first threshold, 17 only the second and 23 both. The maximum ϱ_s over the grid is 0.969 in Database, with 0.965 in Coding.

The selected tuple, $(\beta_p, \beta_i, \beta_d, \alpha) = (0.30, 0.60, 0.16, 0.6)$, attains the highest validation task score among the 199 candidates. The test scores of the five highest-ranked validation tuples differ by at most 0.4 points.

Comparison with the scalar stability region. To compare the screening scores with the local dynamics, consider scalar models with calibrated slopes set to $g = L_s$. For the non-root gains $(K_P, K_I, K_D) = (0.50, 0.25, 0.08)$, their effective slopes are $h_s = 0.6 L_s + 0.4$. Evaluating the spectral radius of the matrix in Eq. (10) gives

Scalar setting	L_s	h_s	$\rho(A(h_s))$
Research	1.55	1.330	1.043160
Coding / BigCodeBench	1.68	1.408	1.103377
Database	1.47	1.282	1.006233
MATH	1.83	1.498	1.173162

For these gains, Theorem C.5 requires $h < 2/1.57 \approx 1.273885$. Each scalar model in the table lies outside this region. Applying the same construction across the grid, 43 of the 199 screened tuples satisfy the scalar condition in every setting, while 156 violate it in at least one setting. Thus validation screening and scalar stability impose different restrictions on the gain family. The comparison uses the specified scalar substitution $g = L_s$; identifying the network matrix H additionally requires response signs, calibration and cross-coordinate sensitivities.

C.7 Projection, saturation and boundedness

Proposition C.6 (Command and integral bounds)**.** *Suppose $\mathbf{c}_i(1) = \mathbf{c}_i^0 \in [0,1]^k$, offsets are saturated in $[-\mathbf{u}_{\max}, \mathbf{u}_{\max}]$ and the integral is clipped to $[-\mathbf{I}_{\max}, \mathbf{I}_{\max}]$. Both the actuation and hold branches preserve*

$$\mathbf{c}_i(t) \in \mathcal{B}_i := \prod_q \big[\max(0, c_i^{0,q} - u_{\max,q}),\ \min(1, c_i^{0,q} + u_{\max,q})\big], \qquad |I_i^q(t)| \le I_{\max,q}.$$

In particular, $\|\mathbf{c}_i(t) - \mathbf{c}_i(s)\|_2 \le \sqrt{k}$ for any two rounds. Under Assumption C.1,

$$|e_i^q(t)| \le E_{iq} := a_i \sum_{\ell=1}^{m} |D_{i,q\ell}| + (1 - a_i) d_i.$$

For nonnegative diagonal gains, the unsaturated PID output satisfies

$$|u_i^q(t)| \le (K_{P,i}^q + 2K_{D,i}^q) E_{iq} + K_{I,i}^q I_{\max,q}.$$

Proof. Offset saturation and position clipping place each executed command in $\mathcal{B}_i$. The initial anchor belongs to $\mathcal{B}_i$, and the hold branch preserves the previous command, so induction establishes invariance. Integral clipping gives the second bound. Measurement differences and neighbour-command differences have absolute value at most one, yielding the bound on e_i^q by the triangle inequality. Finally, $|e_i^q(t) - e_i^q(t-1)| \le 2E_{iq}$, including the initialization $e_i^q(0) = 0$, gives the output bound. □

The bounds hold for arbitrary sensing and actuation decisions satisfying the bounded interface. Projection also satisfies $\|\Pi_{\mathcal{C}}(v) - \Pi_{\mathcal{C}}(w)\|_2 \le \|v - w\|_2$ for any pair of pre-projection commands.

Reachable-command envelope. The box $\mathcal{B}_i$ is an outer bound on the commands reachable from the fixed anchor. With $u_{\max,q} = 0.5$, it is $[0.02, 1.00]^k$ for Coding/BigCodeBench and $[0.00, 0.98]^k$ for Database. Finite horizons, shared evaluator coordinates and integral memory further constrain reachability. The static-control sweep covers the full admissible box $\mathcal{C}$, whereas a dynamic trajectory remains in the envelope associated with its anchor.

Quantization error. Let $M_q = s_{\max}^{(q)} - s_{\min}^{(q)} > 0$ and let $Q_q(c_q)$ be the normalized value recovered from the nearest-integer directive. Then

$$|Q_q(c_q) - c_q| \le \frac{1}{2M_q}.$$

Under Assumption C.2, whenever both inputs belong to $\mathcal{D}$,

$$\|f(Q(c)) - f(c)\| \le \bar{L}_{\mathrm{sur}} \|Q(c) - c\|$$

for the corresponding stacked response model f. The total response discrepancy also includes the residual of the local model. Within a quantization cell, a continuous state change can leave the directive unchanged; such rounds account for 8.6% of the Qwen3-8B MAB-Coding trajectories.

C.8 Graph representation and reference tracking

At a fixed round, define the instantaneous pseudo-reference by

$$\mathbf{q}_i(t) = \mathbf{c}_i(t) + \psi_i(\mathbf{y}_i^{\mathrm{des}}(t) - \mathbf{y}_i(t)). \tag{12}$$

Freeze $\mathbf{q}(t)$ for the following graph interpretation. For a symmetric positive-semidefinite graph Laplacian and a common mixing weight $0 < \alpha \leq 1$, define

$$J_t(c) = \frac{\alpha}{2}\|c - \mathbf{q}(t)\|_2^2 + \frac{1-\alpha}{2} c^\top (\mathcal{L}_w \otimes I_k) c, \qquad c \in \mathcal{C}^N. \tag{13}$$

The first term measures displacement from the current evaluator-induced correction, and the second penalizes differences across graph edges. Proposition C.8 gives the corresponding instantaneous gradient representation. Reference alignment on the directed deployment graph is described by the following lemma.

Lemma C.7 (Directed reference tracking)**.** *Under Assumption C.4, stack the numerical reference estimates in $\hat{R}_t$. Equation* (3) *becomes*

$$\hat{R}_{t+1} = A_r \hat{R}_t + B_r r_{t+1}, \qquad A_r = \big[(1-\beta_r) I_N - \gamma(D_w - \mathcal{W})\big] \otimes I_{d_r}, \qquad B_r = \beta_r I_{N d_r}.$$

Let $q_r = 1 - \beta_r$, $G_r = (I - A_r)^{-1} B_r$ and $R_*^t = G_r r_t$. Then

$$\|\hat{R}_t - R_*^t\|_\infty \leq q_r^t \|\hat{R}_0 - R_*^0\|_\infty + \frac{q_r(1-q_r^t)}{1-q_r}\Delta_r.$$

For constant input r, the estimates converge to $G_r r$ at rate q_r.

For one coordinate on a three-agent chain, the parameters give

$$A_r = \begin{pmatrix} 0.65 & 0 & 0 \\ 0.20 & 0.45 & 0 \\ 0 & 0.20 & 0.45 \end{pmatrix}, \qquad \|A_r\|_\infty = 0.65.$$

The asymptotic tracking bound is $(13/7)\Delta_r$. The equilibrium $G_r r$ averages the reference inputs according to the directed graph; heterogeneous inputs can produce different equilibrium values across agents.

Proposition C.8 (Instantaneous gradient representation)**.** *Fix a round and its pseudo-reference. Suppose $0 < \alpha \leq 1$, the graph Laplacian is symmetric positive semidefinite and the synchronous feedback uses the common mixing weight $a_i = \alpha$. Then J_t is α-strongly convex and has a unique minimizer c^* on $\mathcal{C}^N$. At the current command,*

$$e(t) = -\nabla J_t(c(t)).$$

The minimizer is characterized by $0 \in \nabla J_t(c^*) + \mathcal{N}_{\mathcal{C}^N}(c^*)$; for an interior minimizer, this reduces to $\nabla J_t(c^*) = 0$.

C.9 Proof of Lemma C.7

The neighbour difference sum equals $-(D_w - \mathcal{W})\hat{R}$, including the zero neighbour row at each root. Assumption C.4 implies

$$A_r \geq 0, \qquad A_r \mathbf{1} = q_r \mathbf{1}, \qquad \|A_r\|_\infty = q_r < 1.$$

The Neumann series gives $G_r \geq 0$ and $G_r \mathbf{1} = \mathbf{1}$, hence $\|G_r\|_\infty = 1$. Since $(I - A_r)G_r = B_r$, the error $z_t = \hat{R}_t - G_r r_t$ satisfies

$$z_{t+1} = A_r z_t - A_r G_r (r_{t+1} - r_t).$$

Iterating this relation and using the input-variation bound yields

$$\|z_t\|_\infty \leq q_r^t \|z_0\|_\infty + \Delta_r \sum_{j=1}^{t} q_r^j.$$

The geometric sum equals $q_r(1 - q_r^t)/(1 - q_r)$, proving the estimate. □

C.10 Proof of Proposition C.8

With $\mathbf{q}(t)$ fixed,

$$\nabla^2 J_t = \alpha I + (1-\alpha)(\mathcal{L}_w \otimes I_k) \succeq \alpha I \succ 0.$$

Strong convexity and compactness of the nonempty convex set $\mathcal{C}^N$ give a unique minimizer. At the current command, Eq. (12) gives $\mathbf{q}(t) - c(t) = \psi(\mathbf{y}^{\mathrm{des}}(t) - \mathbf{y}(t))$. Substitution into the gradient of J_t yields $e(t) = -\nabla J_t(c(t))$. The normal-cone condition follows from first-order optimality for a differentiable convex function on a convex set; in the interior the normal cone is zero. □

D CONTROLLER ALGORITHMS AND THE SINGLE-AGENT UPDATE

Figure 3 gives the system-level loop and Figure 4 traces one round of the local loop: the evaluator measures the output, the reference generator produces a target, and the normalized control state is updated by the positional PID law, bounded, quantized into a directive and used for the next round. Algorithm 2 is the gated extension and Algorithm 3 is the sensing sub-routine it calls.

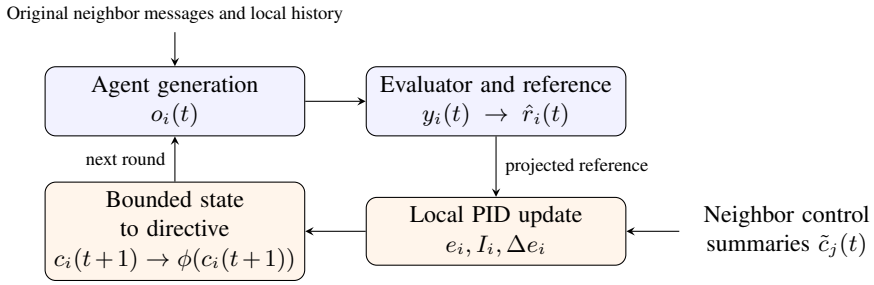

Figure 3: **MaRePID feedback loop.** Agent outputs feed both the original communication context and the evaluator. Evaluator-derived references and neighbouring control summaries determine a local error; a bounded positional PID law produces the control state that is quantized into the next directive. Headline benchmark results use full sensing and an update at every round; sensing-gate sweeps are reported separately.

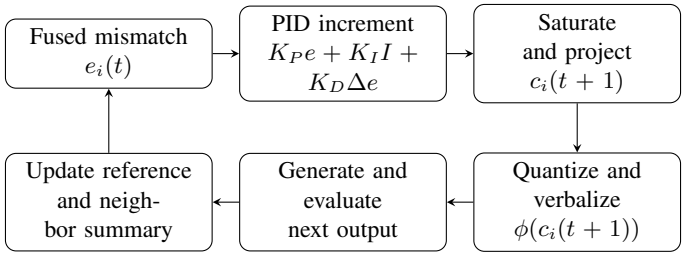

Figure 4: Single-agent MaRePID update. The evaluator measures the output, the reference generator produces a target, and the normalized control state is updated by a positional PID law, bounded, quantized into a directive, and used for the next round.

Algorithm 2 MaRePID: inference-time control with sensing and actuation gates

Require: Baselines $\mathbf{c}_i(0)$; reference estimates $\hat{\mathbf{r}}_i(t)$; weights w_{ij}; gains $\mathbf{K}_P, \mathbf{K}_I, \mathbf{K}_D$; gate parameters $(\delta_i^{\mathrm{sen},\star}, \eta_i, T_i^{\max}, \lambda_i, \mathbf{Q}_i, \mathbf{R}_i)$; bounds; horizon R

Ensure: Controls $\mathbf{c}_i(t)$ and generated outputs $o_i(t)$ for $t = 1, \ldots, R$: exactly R generated outputs and at most R evaluator queries per agent

Initialize $\mathbf{c}_i(1) \leftarrow \mathbf{c}_i(0)$, $\hat{\mathbf{r}}_i(0) \leftarrow \mathbf{r}_i(0)$, $\mathbf{I}_i(0) = \mathbf{0}$, $\mathbf{e}_i(0) = \mathbf{0}$, $\mathrm{sat}_i(0) = 0$, $\chi_i^{\mathrm{act}}(0) = 1$, $t_i^{\mathrm{last}} = 1 - T_i^{\max}$.

Round 1 is free of both gates. Generate $o_i(1)$ under $\phi(\mathbf{c}_i(1))$ and query the evaluator to obtain $\mathbf{y}_i(1) = \mathcal{E}_{\mathrm{proc}}(o_i(1))$; set $\chi_i^{\mathrm{sen}}(1) \leftarrow 1$. Round 1 is never skipped, and it is the only round that is both generated and measured unconditionally.

for $t = 1, \ldots, R-1$ **do**

  for each agent $i \in \mathcal{N}$ in parallel **do**

    Form $\mathbf{r}_i(t)$ from the latest available measurement $\mathbf{y}_i(t)$, update $\hat{\mathbf{r}}_i(t)$ by Eq. (3), and set $\mathbf{y}_i^{\mathrm{des}}(t) = \zeta_i(\hat{\mathbf{r}}_i(t))$..

    Receive neighbour states $\tilde{\mathbf{c}}_j(t)$ and form $\mathbf{e}_i(t)$ by Eq. (4), using $\mathbf{e}_i(t) = \psi_i(\mathbf{y}_i^{\mathrm{des}}(t) - \mathbf{y}_i(t))$ if $\mathcal{N}_i = \varnothing$.

    Compute $\Delta\mathbf{e}_i(t)$ and the actuation decision $\chi_i^{\mathrm{act}}(t)$ from the configured event score, deadband and maximum-gap condition.

    Update the integral memory by Eq. (5) with $\sigma_i(t) = \chi_i^{\mathrm{act}}(t-1)(1 - \mathrm{sat}_i(t-1))$.

    if $\chi_i^{\mathrm{act}}(t) = 1$ **then**

      $\mathbf{u}_i(t) \leftarrow$ Eq. (6); $\bar{\mathbf{u}}_i(t) \leftarrow \mathrm{sat}_{[\mathbf{u}_{\min}, \mathbf{u}_{\max}]}(\mathbf{u}_i(t))$; $\mathbf{c}_i(t+1) \leftarrow \mathrm{clip}_{\mathcal{C}}(\mathbf{c}_i(0) + \bar{\mathbf{u}}_i(t))$; $\mathrm{sat}_i(t) \leftarrow \mathbf{1}[\bar{\mathbf{u}}_i(t) \neq \mathbf{u}_i(t)]$; $t_i^{\mathrm{last}} \leftarrow t$.

    else

      $\mathbf{c}_i(t+1) \leftarrow \mathbf{c}_i(t)$ (hold); $\mathrm{sat}_i(t) \leftarrow 0$.

    end if

    Generate $o_i(t+1)$ under $\phi(\mathbf{c}_i(t+1))$.

    $(\chi_i^{\mathrm{sen}}(t+1), \mathbf{y}_i(t+1)) \leftarrow$ Algorithm 3, called *after* $o_i(t+1)$ exists, on the latest available measurement $\mathbf{y}_i(t)$; if $\chi_i^{\mathrm{sen}}(t+1) = 0$ then $\mathbf{y}_i(t+1) \leftarrow \mathbf{y}_i(t)$.

  end for

end for

The $N \cdot R$ evaluator-call budget includes the mandatory first-round queries: with $N = 3$ agents and $R = 6$ rounds, each agent generates exactly $R = 6$ outputs and the full-sensing baseline performs exactly $N \cdot R = 18$ evaluator queries, of which the $N = 3$ first-round queries are the first three. No separate round-0 generation or query is billed.

Algorithm 3 MaRePID: prediction-triggered sensing

Require: The round-$(t{+}1)$ output $o_i(t{+}1)$, already generated; latest available measurement $\mathbf{y}_i(t)$; control $\mathbf{c}_i(t)$; neighbour summary $\mathbf{h}(t)$; current reference $\mathbf{r}_i(t)$; threshold $\delta_i^{\mathrm{sen},\star}$

Ensure: Sensing decision $\chi_i^{\mathrm{sen}}(t{+}1)$, the measurement $\mathbf{y}_i(t{+}1)$ and the reference $\mathbf{r}_i(t{+}1)$ used at round $t{+}1$

1: Predict the next diagnostic state from round-t information only: $\hat{\mathbf{s}}_{\mathrm{sys}}(t+1 \mid t) \leftarrow \mathcal{P}_\theta(\{\mathbf{y}_i(t)\}, \{\mathbf{c}_i(t)\}, \mathbf{h}(t), t)$. If $\chi_i^{\mathrm{sen}}(t) = 0$ the reused $\mathbf{y}_i(t-1)$ enters this feature vector.

2: Form the predicted local reference $\mathbf{r}_i^{\mathrm{pred}}(t+1 \mid t) \leftarrow \mathcal{F}_{\mathrm{ref}}(\hat{\mathbf{s}}_{\mathrm{sys}}(t+1 \mid t), \mathbf{h}_i(t), \mathrm{Prompt}_i)$ and the predicted reference displacement $\delta_i^{\mathrm{sen}}(t+1)$ used by the configured sensing threshold.

3: **if** $\delta_i^{\mathrm{sen}}(t+1) \geq \delta_i^{\mathrm{sen},\star}$ **then**

4: Query the evaluator on the output already in hand, $\mathbf{y}_i(t+1) \leftarrow \mathcal{E}_{\mathrm{proc}}(o_i(t+1))$; set $\mathbf{r}_i(t+1) \leftarrow \mathbf{r}_i^{\mathrm{pred}}(t+1 \mid t)$; $\chi_i^{\mathrm{sen}}(t+1) \leftarrow 1$.

5: **else**

6: Skip the query: keep $\mathbf{y}_i(t+1) \leftarrow \mathbf{y}_i(t)$ and $\mathbf{r}_i(t+1) \leftarrow \mathbf{r}_i(t)$; $\chi_i^{\mathrm{sen}}(t+1) \leftarrow 0$.

7: **end if**

E Experimental Details

E.1 Benchmark Configuration

We evaluate the same benchmark configurations described in the main paper. All methods use identical task splits, decoding settings, and evaluation protocols.

E.2 Baseline Implementation

Static baselines, feedback baselines, and adaptive optimization baselines use the configurations described in Section 4. Validation budgets and inference-time budgets are reported separately.

E.3 Statistical Testing

We use paired bootstrap testing over identical evaluation instances with 10,000 resamples. Confidence intervals use percentile bootstrap estimates and multiple comparisons are corrected using Holm correction.

F PID Mechanism Isolation

To distinguish feedback regulation from dynamic prompting, we evaluate three controlled variants.

Open-loop replay. We first execute MaRePID on validation trajectories and record the generated directive sequence. During evaluation, the same sequence is replayed while removing evaluator feedback.

Shuffled feedback. Evaluator signals are randomly permuted across rounds. The feedback format and frequency remain unchanged, but temporal correspondence is removed.

Fixed dynamic schedule. A validation-generated directive trajectory is applied without runtime feedback adaptation.

The measured scores are reported in Table 2.

G Additional Ablation and Robustness

G.1 PID Component Ablation

We evaluate the contribution of individual PID components by removing the proportional, integral, and derivative terms.

Table 11: PID component ablation variants.

Variant	Purpose
Full MaRePID	Reference
w/o P	Remove immediate correction
w/o I	Remove accumulated correction
w/o D	Remove predictive correction
w/o feedback	Remove closed-loop signal
w/o evaluator	Remove measurement module

G.2 Evaluator Robustness

We evaluate robustness under weaker and noisy evaluators to measure sensitivity to feedback quality.

Table 12: Robustness evaluation settings.

Setting	Evaluation axis
Evaluator	Weak / noisy evaluator
Agent scale	Different team sizes
Noise	Feedback perturbation

G.3 Agent Scaling and Feedback Noise

Additional robustness experiments vary the number of agents and introduce controlled feedback perturbations.

H Cost Analysis

Table 13 reports evaluator calls, relative token cost, and performance for a separate sensing comparison. The full-sensing configuration in this cost analysis uses 20 evaluator calls, corresponding to its specific evaluation setting, while Algorithm 1 uses the $N = 3$, $R = 6$ benchmark configuration with 18 full-sensing calls.

Table 13: Cost-performance trade-off under different sensing strategies.

Setting	Evaluator Calls	Token Cost	Performance
Full sensing	20	1.00x	78.5
Adaptive sensing	14	0.72x	77.9
Random sensing	14	0.71x	74.8