Overcoming Attention Distraction: Training-Free Latent Communication for Multi-Agent Systems

Zhengjie Zhou[1*†] **Yahao Liu**[1*] **Ziyue Feng**[2] **Tengfei Liu**[1†] **Weiqiang Wang**[1]

[1]Ant Group [2]University of Chicago
{zhouzhengjie.zzj, liuyahao.lyh, aaron.ltf, weiqiang.wwq}@antgroup.com
zoeyfeng@uchicago.edu

Abstract

Scaling multi-agent systems driven by large language models is transitioning from lossy text-based communication to continuous latent interaction. However, existing latent interaction approaches relying on concatenation-based Softmax cross-attention suffer from an Attention Distraction Dilemma: due to the inherent anisotropy of latent spaces and the semantic misalignment induced by agent role heterogeneity, non-linear Softmax structurally amplifies irrelevant tokens, diluting the collaborative signal as interactions deepen. To address this issue, we propose a training-free additive mechanism that models upstream representations as a semantic manifold. By shifting from concatenation-based Softmax competition to linear subspace projection, this perspective inherently bypasses scalar noise accumulation. Specifically, we introduce two components: a Resonance Filter that extracts principal semantic directional alignments to overcome latent anisotropy, and an SNR-Adaptive Spectral Gain that dynamically regulates this projection to counteract role heterogeneity. We analyze this approach using the Asymptotic SNR Tipping Point Theorem, illustrating the scaling advantages of the mechanism beyond a critical noise threshold. Experiments indicate that the proposed approach outperforms baselines in 43 of 54 evaluated configurations, maintaining semantic reasoning in long-context collaborations.

1 Introduction

Multi-agent systems (MAS) driven by large language models (LLMs) show significant potential in solving complex tasks through collaboration [1, 42, 30], spanning mathematical reasoning [44, 71], open-domain QA [25, 34], and decision-making [14, 33]. In these systems, natural language typically serves as both internal thought representation and inter-agent communication medium [20]. Collaboration is generally achieved by concatenating upstream textual outputs into downstream prompts. However, explicit text concatenation quantizes continuous vectors into discrete tokens, causing semantic information loss [45, 54]. Furthermore, this concatenation escalates system overhead as task complexity and agent count increase [46, 66, 24].

Recent studies explore latent interaction in MAS [6, 54, 74]. Unlike textual exchange, latent interaction transmits continuous representations from upstream to downstream agents, preserving semantic details and bypassing decoding-encoding overhead. Several approaches investigate reinforcement learning protocols, evolving communication strategies through global rewards [37, 31, 55, 65]. However, because MAS typically require zero-shot adaptability across diverse tasks, modern systems favor training-free configurations. Within this training-free scope, existing approaches generally fall into two categories: (i) dense communication, which shares raw hidden states or KV caches for end-to-end alignment [11, 74, 63, 50]; and (ii) purified communication, which extracts compact latent

*Equal contribution †Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

thoughts via feature decoupling [54, 69, 45, 68, 32, 28]. Despite this progress, both categories rely on Softmax interaction, presenting critical structural limitations: **(1)** due to the anisotropic nature of representations in LLMs, background tokens often exhibit high similarity to the query [12, 18]; and **(2)** distinct roles assigned via prompting create a natural mismatch between the task-specific subspaces of upstream and downstream agents, impeding precise Query-Key alignment [58, 5]. Together, these combined factors cause noise to accumulate in the Softmax denominator. Consequently, downstream agents face an **Attention Distraction Dilemma** (detailed in Section 2). As illustrated in Fig. 1, growing background noise dilutes the collaborative signal as interaction deepens [10, 72].

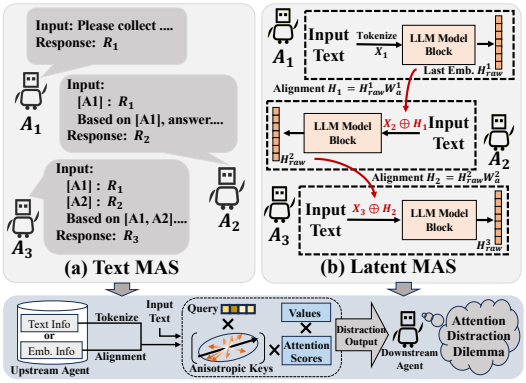


Figure 1: Illustration of the Attention Distraction Dilemma. The distribution shift between agents causes attention mechanisms to degrade, allowing noise to dominate as interaction deepens.

To address this dilemma, we propose a training-free additive mechanism that models upstream representations as a semantic manifold. By shifting the interaction paradigm from concatenation-based Softmax competition to linear subspace projection, this approach establishes a sequence-independent directional alignment that structurally prevents scalar noise accumulation. To actualize this, we construct a **Resonance Filter** that utilizes the principal covariance directions of the upstream manifold. This filter counteracts the anisotropic nature of representations in LLMs by aligning the downstream intent with the principal semantic subspace, inducing constructive interference for core logical components. Furthermore, to address the task-specific mismatch induced by the role heterogeneity of agents, we introduce an **SNR-Adaptive Spectral Gain**. Acting as an alignment-aware safety valve, this coefficient dynamically modulates the projection strength based on the geometric coherence between agents, dampening the injection when subspaces severely diverge. Finally, we formalize this mechanism via the ***Asymptotic Signal-to-Noise Ratio (SNR) Tipping Point Theorem***. This theorem identifies a critical background threshold dictated by the geometric properties of the representation space, proving that beyond this noise threshold, our additive mechanism guarantees a higher information lower bound than standard cross-attention.

In summary, the main contributions are:

- We formalize the Attention Distraction Dilemma within MAS and show that the non-linear rectification of standard Softmax cross-attention leads to scalar noise accumulation.
- We propose a training-free additive mechanism to resolve the Attention Distraction Dilemma. By conceptualizing upstream representations as a semantic manifold, we shift the interaction paradigm from concatenation-based Softmax competition to linear subspace projection, structurally preventing scalar noise accumulation and preserving collaborative signals.
- We present the Asymptotic SNR Tipping Point Theorem, illustrating the scaling advantage of the additive mechanism in high-noise regimes and providing guidance for large-scale MAS.
- Comprehensive experiments show that the proposed mechanism outperforms baselines in 43 of 54 experimental settings across multiple reasoning tasks.

2 Problem Formulation

This section outlines the multi-agent system (MAS) setup and introduces the Attention Distraction Dilemma to contextualize the subsequent analysis.

Single Agent in a Transformer. A single agent typically employs the autoregressive Transformer architecture [59, 4, 47]. Given natural language input, the model generates a token sequence $T \in \{1, \ldots, V\}^{L_x}$, with vocabulary size V and sequence length L_x. For the i-th agent, an input embedding layer $W_{\text{in}} \in \mathbb{R}^{V \times D}$ maps these tokens to a representation matrix $X^{(i)} \in \mathbb{R}^{L_x \times D}$, where D is the hidden dimension. After processing through N Transformer layers for contextual feature fusion, an output layer $W_{\text{out}} \in \mathbb{R}^{V \times D}$ projects the hidden states back to the vocabulary space to produce the generation probability distribution.

Multi-Agent Systems. Multi-agent systems accommodate heterogeneity by integrating agents with distinct architectures. This analysis focuses on the interaction between the upstream $(i-1)$-th agent and the downstream i-th agent. We assume an approximate alignment mapping W_a projects the raw upstream hidden representations $H_{\text{raw}}^{(i-1)}$ into the downstream input space. This yields the aligned trajectory $H^{(i-1)} = H_{\text{raw}}^{(i-1)} W_a \in \mathbb{R}^{L_h \times D}$, where L_h is the upstream history sequence length. Following prior research [74, 23], we define this projection as $W_a = \left(W_{\text{out}}^\top W_{\text{out}} + \lambda I\right)^{-1} W_{\text{out}}^\top W_{\text{in}}$, where λ is a regularization coefficient, W_{out} is the output layer of the $(i-1)$-th agent, W_{in} is the input embedding layer of the i-th agent, and I is the identity matrix.

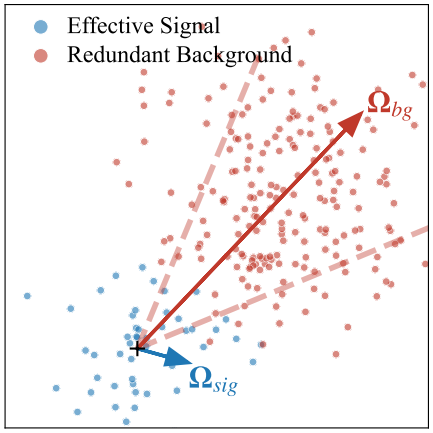


Figure 2: Geometric Anisotropy in Latent Space. We visualize the last hidden states of Qwen3-14B on the GSM8K dataset using t-SNE. The effective signals and redundant background within the inputs are approximated using a GPT-5.4 proxy.

Attention Distraction Dilemma. To examine the limitations of current approaches, we divide the aligned upstream sequence $H^{(i-1)}$ into two disjoint sets: an effective signal subset S (with L_{signal} tokens relevant to the current reasoning step) and a redundant background subset B (with L_{bg} irrelevant tokens). Standard paradigms incorporate the upstream history $H^{(i-1)}$ as context for the downstream agent. After the input stage, the attention update for the m-th downstream token representation, $x_m^{(i)} \in \mathbb{R}^{1\times D}$, forms a superposition of signal and background components: $\tilde{x}_m^{(i)} = x_m^{(i)} + \Omega_{\text{signal}} + \Omega_{\text{bg}}$. Here, Ω_{signal} and Ω_{bg} represent the attention updates originating from S and B, respectively. We quantify the fusion efficacy via the Signal-to-Noise Ratio (SNR) [64, 49], defined by the ratio of the Euclidean norms of these update vectors: $\text{SNR}^{\text{attn}} = \|\Omega_{\text{signal}}\|_2 / \|\Omega_{\text{bg}}\|_2$.

Fig.2 illustrates a geometric phenomenon arising from the interaction between the non-negative Softmax operator and the representation anisotropy of LLMs [12, 18, 60]. Instead of canceling out geometrically, redundant background tokens cluster within a high-density cone. As the upstream context expands, the aggregated background noise rapidly exceeds the signal vector magnitude ($\|\Omega_{\text{bg}}\|_2 \gg \|\Omega_{\text{signal}}\|_2$). Crucially, the distinct roles assigned to agents naturally induce a spatial misalignment between their latent subspaces, which further accelerates this signal dilution. This dilution forces existing approaches into a rigid trade-off: (i) *Dense communication* allows unrestricted sequence divergence, which degrades the SNR as background noise overwhelms the downstream agent; and (ii) *Purified communication* limits noise accumulation but disrupts the upstream chain of thought, potentially causing semantic loss.

3 Methodology

This section details the additive mechanism designed to mitigate the dilemma. By reconceptualizing the upstream representations as a semantic manifold, we introduce a Resonance Filter. This filter projects downstream intentions onto the principal semantic subspace of the upstream agent, regulated by an SNR-Adaptive Spectral Gain. We theoretically motivate this approach by analyzing its asymptotic SNR scaling behavior and empirically assessing the robustness of the mechanism.

3.1 Training-Free Additive Mechanism

Design Motivation. Inspired by linear attention [27, 7], we replace Softmax with linear subspace projection. While Softmax token-level normalization allows background noise to cumulatively dilute the denominator, naive linear projections also fail in MAS because LLMs representational anisotropy amplifies background drift. To resolve this, we conceptualize upstream representations as a semantic manifold, isolating collaborative signals from orthogonal noise along principal geometric directions. Accordingly, we treat $H^{(i-1)}$ as a semantic distribution to construct a Resonance Filter, enabling downstream agents to resonate with principal logical components while suppressing anisotropic drift.

To mitigate the anisotropy of representations in LLMs, we extract the sequence-wise mean row vector $\mu^{(i-1)} \in \mathbb{R}^{1\times D}$ from $H^{(i-1)}$ and perform explicit centralization: $\hat{H}^{(i-1)} = H^{(i-1)} - \mathbf{1}_{L_h} \mu^{(i-1)}$, where $\mathbf{1}_{L_h} \in \mathbb{R}^{L_h \times 1}$ is an all-ones column vector. This centralization acts as a spectral filter, filtering the non-informative common mode associated with the anisotropy cone [12, 56]. Next, we compute

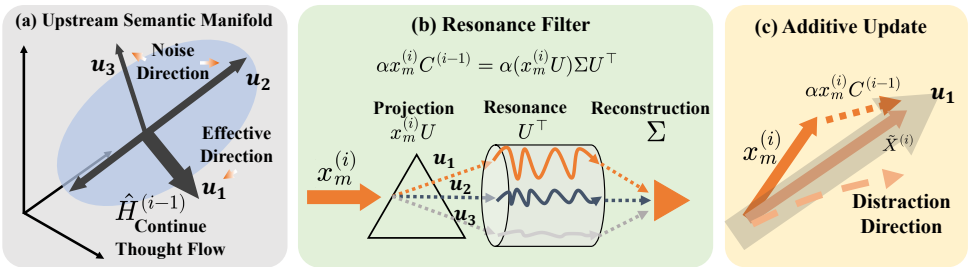


Figure 3: **(a) Upstream Semantic Manifold:** The centralized upstream representations define a semantic subspace spanned by principal eigenvectors (u_1, u_2, u_3). The covariance structure effectively isolates the dominant effective direction (u_1) from the noise directions $(u_2 \& u_3)$. **(b) Resonance Filter:** The downstream intention $x_m^{(i)}$ is projected onto this eigenbasis (U). The eigenvalue matrix Σ acts as a spectral gate, amplifying the high-variance signal (u_1) while suppressing incoherent noise modes $(u_2 \& u_3)$. **(c) Additive Update:** The filtered context is injected via vector interference, correcting the downstream trajectory $\tilde{X}^{(i)}$ towards the effective direction.

the empirical covariance matrix $C^{(i-1)} = \frac{1}{L_h}(\hat{H}^{(i-1)})^\top \hat{H}^{(i-1)}$. It bypasses noise-prone sequential dependencies, extracting sequence-independent principal semantic directional alignments. The upstream representation $H^{(i-1)}$ is extracted from the last hidden states of the upstream agent. The downstream state update is then applied once to the initial input embeddings $X^{(i)}$ of the downstream agent before its Transformer forward pass, formulated as a linear injection:

$$\tilde{X}^{(i)} = X^{(i)} + \alpha X^{(i)} C^{(i-1)}, \tag{1}$$

where α serves as the SNR-Adaptive Spectral Gain (detailed in Section 3.2). By performing the eigendecomposition $C^{(i-1)} = U\Sigma U^\top$, the mechanism operates as a **Resonance Filter**:

$$\alpha x_m^{(i)} C^{(i-1)} = \alpha (x_m^{(i)} U)\Sigma U^\top. \tag{2}$$

As Fig. 3 illustrates, $(x_m^{(i)} U)$ projects the downstream intention $x_m^{(i)}$ onto the principal basis of the upstream agent. Constructing the matrix $C^{(i-1)}$ incurs an $\mathcal{O}(L_h D^2)$ cost. We mitigate this using Randomized SVD [2, 21] to compute a rank-r approximation determined by the maximum spectral gap [73] (Appendix J provides a sensitivity analysis of the rank r). This approach reduces the complexity of upstream aggregation to $\mathcal{O}(L_h D r)$, where $r \ll D$. Integrating additive mechanism requires compatibility with Rotary Position Embeddings [51]. We demonstrate that the proposed mechanism preserves relative positional equivariance, as the linear distributivity of query and key projections ensures that the injected perturbation is independently rotated by the identical position-specific orthogonal matrix as its host token (see Appendix A.5 for the detailed analysis).

3.2 SNR-Adaptive Spectral Gain

Design Motivation. In MAS, distinct roles naturally induce semantic misalignment between upstream and downstream latent spaces, causing ideal geometric assumptions to degrade [58, 5]. To counter this, we introduce an SNR-Adaptive Spectral Gain α. Acting as an alignment-aware safety valve, it dynamically modulates the projection strength based on the geometric coherence between the downstream intent and the upstream subspace.

Macro-Energy Regulation. To prevent upstream bias from overwhelming downstream semantics, we constrain the Frobenius norm of the injected perturbation to a fraction γ of the original energy:

$$\|\alpha X^{(i)} C^{(i-1)}\|_F \leq \gamma \|X^{(i)}\|_F. \tag{3}$$

This yields the global gain limit $\alpha_{\text{global}} = \frac{\gamma \|X^{(i)}\|_F}{\|X^{(i)} C^{(i-1)}\|_F + \epsilon}$, where $\gamma > 0$ denotes the tolerance hyperparameter and ϵ is a small numerical stabilizing constant to prevent division-by-zero.

Sequence-Level Minimum-Response Gate. To suppress unreliable transfer when the upstream subspace poorly matches certain downstream tokens, we gate the shared gain on the weakest normalized projection response across the sequence. The token-wise local gain limit is:

$$\alpha_{\text{local}} = \beta \min_m r_m, \quad r_m = \frac{\|x_m^{(i)} C^{(i-1)}\|_2}{\|x_m^{(i)}\|_2 + \epsilon}, \tag{4}$$

where $\beta > 0$ is the safety threshold and $x_m^{(i)}$ denotes the representation of the m-th downstream token. Because $\alpha \leq \beta\, r_{\min}$, the largest normalized perturbation over the sequence satisfies $\alpha\, r_{\max} \leq \beta\, r_{\min}\, r_{\max}$, where $r_{\max} = \max_m r_m$ and $r_{\min} = \min_m r_m$. The final spectral gain $\alpha = \min(\alpha_{\text{global}}, \alpha_{\text{local}})$ combines two complementary constraints. Eq. 3 bounds the total injected Frobenius energy, while Eq. 4 gates on transfer reliability.

3.3 Theoretical Analysis: SNR Scaling Behavior

To understand the mechanism, we analyze the SNR scaling. We define the SNR of the additive mechanism as the ratio of the Euclidean norm of the effective signal update to the norm of the redundant background update: $\text{SNR}^{\text{add}} = \|\Omega^{\text{add}}_{\text{signal}}\|_2 / \|\Omega^{\text{add}}_{\text{bg}}\|_2$. Using the covariance formulation, the additive update for token $x_m^{(i)}$ decomposes into the signal subset S and background subset B:

$$\alpha x_m^{(i)} C^{(i-1)} = \underbrace{\frac{\alpha}{L_h} \sum_{k \in S} w_k \hat{h}_k^{(i-1)}}_{\Omega^{\text{add}}_{\text{signal}}} + \underbrace{\frac{\alpha}{L_h} \sum_{j \in B} w_j \hat{h}_j^{(i-1)}}_{\Omega^{\text{add}}_{\text{bg}}}, \tag{5}$$

where $w_t = \langle x_m^{(i)}, \hat{h}_t^{(i-1)} \rangle$ represents the projection weight of the downstream intent onto the centralized upstream feature $\hat{h}_t^{(i-1)}$. To bound the growth rates of these terms, we introduce two statistical assumptions modeling the latent space geometry.

Assumption 1. *Drawing on the Task Vector hypothesis in In-Context Learning [22, 57], we posit that the downstream intent $x_m^{(i)}$ resides within the principal semantic subspace of the upstream signal set S. Due to strong semantic correlation, the projection weights w_k exhibit consistent sign alignment, causing constructive interference. Consequently, the effective signal aggregation magnitude scales linearly with the sequence length of the signal: $\|\Omega^{add}_{signal}\|_2 \propto L_{signal}$.*

Assumption 2. *While raw background tokens exhibit anisotropy dominated by a common drift [12, 56], centralization removes this mode [41]. Although local noise clustering may persist in realistic LLM manifolds, implicit covariance truncation via SVD further suppresses these non-orthogonal directions. Crucially, while Softmax structurally prevents destructive interference by rectifying all noise projections into positive probability mass, the additive mechanism permits signed projections. By modeling the residual projections $\langle x_m^{(i)}, \hat{h}_j^{(i-1)} \rangle$ as weakly correlated bounded variables with a near-zero mean, random walk principles in high-dimensional spaces dictate that destructive interference causes the vector sum to exhibit sub-linear growth: $\|\Omega^{add}_{bg}\|_2 \propto \sqrt{L_{bg}}$.*

Under Assumption 1, the signal energy is stable. Under Assumption 2, background noise vector interference grows significantly slower than the linear accumulation observed in Softmax. **As derived in Appendix A**, for any failure probability $\delta \in (0, 1)$, we summarize this behavior:

Theorem 1 (Asymptotic SNR Tipping Point)**.** *Given Assumptions 1 and 2, there exists a critical background threshold L_{crit}. When redundant upstream tokens exceed this threshold, the Additive Mechanism provides a higher SNR compared to Softmax with high probability:*

$$\mathbb{P}\left(SNR^{add} > SNR^{attn}\right) \geq 1 - \delta, \quad \forall L_{bg} > L_{crit}. \tag{6}$$

Theorem 1 establishes that if representations are centered and aligned, the additive mechanism inherently circumvents the linear noise accumulation of Softmax. However, in realistic MAS, distinct agent roles inevitably perturb this perfect geometric alignment, causing Assumptions 1 and 2 to degrade. It is precisely this gap between the idealized tipping point and complex real-world trajectories that necessitates the SNR-Adaptive Spectral Gain α (Section 3.2), which acts as a fallback regularization to prevent variance explosion when ideal subspace conditions are not met.

3.4 Empirical Validation via Controlled Noise Injection

To assess theoretical scaling and its underlying assumptions, we empirically estimate the critical threshold L_{crit} and analyze the robustness of the system against noise accumulation. Because exact values of L_{crit} and intrinsic background noise L_{bg} depend on intractable latent space manifold geometries, we adopt a three-part empirical analysis.

(1) Heuristic Estimation of Intrinsic Noise (L_{bg}). We first quantify task-irrelevant intrinsic noise L_{bg} using a semantic heuristic. Utilizing GPT-5.4, we compute the token length difference between the raw upstream response (R_{raw}) and a compressed summary (R_{clean}), yielding $L_{\text{bg}} \approx \text{len}(R_{\text{raw}}) - \text{len}(R_{\text{clean}})$. As Fig. 4 (Middle) visualizes, intrinsic noise is pervasive. Although this length-based proxy inevitably

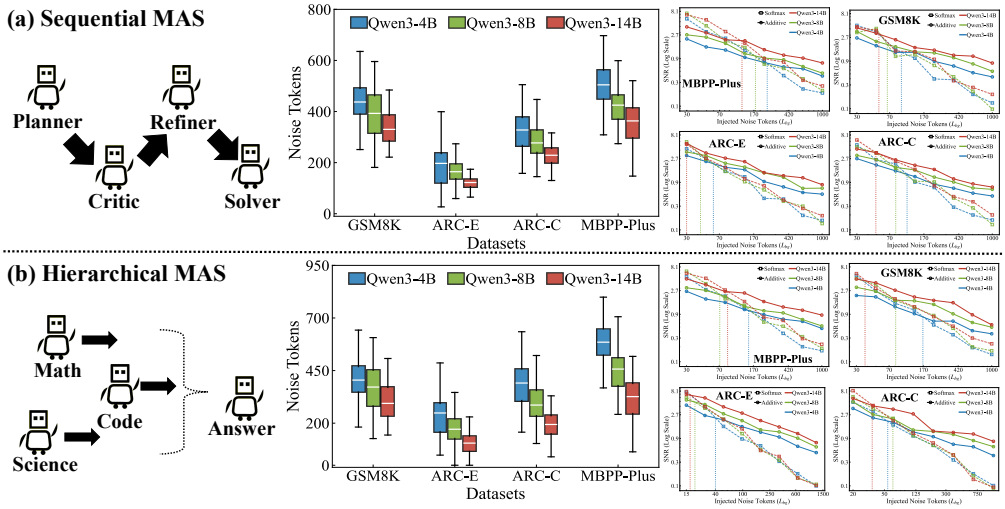


Figure 4: Empirical validation of the Asymptotic SNR Tipping Point. The study utilizes two architectural settings: **(a)** Sequential and **(b)** Hierarchical MAS. **Middle Panel:** Box plots quantify the estimated intrinsic noise (L_{bg}) using the length proxy of GPT-5.4. **Right Panel:** SNR decay curves (log-scale) versus physically injected noise (L_{inj}), illustrating the crossover phenomenon independently of the L_{bg} estimation proxy.

reflects the preferences of the external summarizer alongside true latent noise, it provides a directional estimate of context redundancy (e.g., median noise approaches ~ 400 tokens for GSM8K).

(2) Validation of the Tipping Point (L_{crit}). To circumvent the limitations of the L_{bg} proxy, we locate the empirical tipping point by injecting controlled noise tokens (L_{inj}) sampled from task-irrelevant examples within the same dataset. To ensure a fair comparison, noise is injected precisely at the communication boundary. It is appended to the Text MAS message and fed as tokenized inputs alongside the clean upstream KV cache to the downstream Latent MAS agent, avoiding the unintended corruption of Rotary Position Embeddings [51]. We measure the resulting SNR and emphasize that this decomposition remains an analytical tool for theoretical validation, uncomputed during practical inference. Fig. 4 (Right) presents the SNR decay curves. Consistent with Theorem 1, a crossover occurs: **(i) Low-Noise Regime:** At minimal noise ($L_{\text{inj}} \leq 30$), Softmax (dashed lines) maintains a marginal advantage. **(ii) High-Noise Regime:** As L_{inj} scales to 1000, Softmax SNR collapses rapidly (e.g., decaying by $> 60\%$), whereas the Additive mechanism (solid lines) follows a robust sub-linear decay, retaining over $2\times$ higher SNR. On GSM8K (Sequential), this early crossover occurs at $L_{\text{crit}} \approx 49$. Crucially, the estimated intrinsic background noise ($L_{\text{bg}} \approx 400$) satisfies $L_{\text{bg}} \gg L_{\text{crit}}$. This substantial gap confirms that realistic MAS deployments naturally operate deep within the dominant high-SNR regime of the Additive mechanism.

(3) Validation of the SNR Assumptions. To assess the SNR assumptions, we examine signal coherence under noise and background suppression by centralization and principal-subspace selection. We evaluate Qwen3-8B Sequential MAS on nine datasets using the same protocol to inject L_{inj} noise tokens. Original and injected tokens serve as signal and background proxies, respectively. We compare full-rank covariance (Full), random rank-r directions (Random-r), and top-r principal directions (Top-r), where r denotes the retained subspace dimension.

Table 1: Macro-averages across nine datasets using Qwen3-8B Sequential MAS.

Metric	Setting	Average
Signal coherence	$L_{\text{inj}} = 1000$	0.457
Normalized bg. mean	Raw/Centered	0.409/0.064
Background growth	Full/Random-r/Top-r	4.16/3.97/2.55

Signal coherence measures sign agreement as the absolute sum of sequence-centered signal weights divided by their sum of absolute values. The normalized background mean measures bias as the absolute mean of background weights divided by their root mean square, before and after centering. Background growth is the ratio of unscaled background-update norms at 1000 and 200 injected tokens. Table 1 reports a macro-averaged signal coherence of 0.457 at 1000 injected tokens. Centering reduces the normalized background mean from 0.409 to 0.064, a reduction of 84.4%. Top-r yields background growth of 2.55, compared with 4.16 for Full and 3.97 for Random-r, supporting

Table 2: Comparison of exact match accuracy (%) across three Qwen3 scales. Subscripts indicate standard deviations across three random seeds. The highest accuracy is in **bold**.

Tasks	Qwen3-4B					Qwen3-8B					Qwen3-14B				
	TM	TM-p	LM	LM-p	Our	TM	TM-p	LM	LM-p	Our	TM	TM-p	LM	LM-p	Our
Sequential MAS Setting															
ARC-E	$97.0_{\pm 1.8}$	$96.7_{\pm 2.8}$	$96.7_{\pm 1.2}$	$\mathbf{98.9}_{\pm 0.5}$	$98.3_{\pm 1.0}$	$96.2_{\pm 1.6}$	$99.0_{\pm 0.6}$	$97.7_{\pm 1.1}$	$98.9_{\pm 1.1}$	$\mathbf{99.3}_{\pm 0.4}$	$\mathbf{99.8}_{\pm 0.1}$	$99.2_{\pm 0.4}$	$\mathbf{99.8}_{\pm 0.1}$	$98.7_{\pm 0.5}$	$99.6_{\pm 0.2}$
ARC-C	$91.6_{\pm 1.7}$	$89.5_{\pm 2.9}$	$90.6_{\pm 1.3}$	$91.9_{\pm 2.2}$	$\mathbf{93.4}_{\pm 2.7}$	$94.3_{\pm 1.7}$	$94.8_{\pm 2.8}$	$92.5_{\pm 1.2}$	$95.1_{\pm 2.3}$	$\mathbf{96.3}_{\pm 0.8}$	$94.8_{\pm 1.6}$	$96.1_{\pm 2.8}$	$95.6_{\pm 1.7}$	$96.0_{\pm 2.2}$	$\mathbf{96.8}_{\pm 2.5}$
GSM8K	$87.9_{\pm 1.9}$	$90.2_{\pm 2.3}$	$90.3_{\pm 1.1}$	$92.1_{\pm 2.4}$	$\mathbf{92.6}_{\pm 2.5}$	$92.9_{\pm 2.3}$	$91.8_{\pm 2.9}$	$94.0_{\pm 1.3}$	$94.2_{\pm 2.2}$	$\mathbf{95.1}_{\pm 0.5}$	$92.7_{\pm 1.7}$	$94.3_{\pm 2.7}$	$95.6_{\pm 1.2}$	$\mathbf{95.7}_{\pm 2.1}$	$94.8_{\pm 2.6}$
MedQA	$66.5_{\pm 2.1}$	$65.8_{\pm 3.5}$	$66.3_{\pm 1.6}$	$69.1_{\pm 2.8}$	$\mathbf{69.5}_{\pm 3.2}$	$76.2_{\pm 2.2}$	$\mathbf{77.4}_{\pm 3.4}$	$76.1_{\pm 1.5}$	$76.2_{\pm 2.1}$	$\mathbf{77.4}_{\pm 0.9}$	$79.6_{\pm 2.8}$	$80.9_{\pm 3.2}$	$79.3_{\pm 1.6}$	$81.5_{\pm 2.7}$	$\mathbf{82.3}_{\pm 3.1}$
MBPP+	$71.6_{\pm 2.0}$	$75.4_{\pm 3.1}$	$72.8_{\pm 1.4}$	$74.2_{\pm 2.5}$	$\mathbf{76.5}_{\pm 2.9}$	$67.9_{\pm 2.0}$	$70.1_{\pm 3.2}$	$73.0_{\pm 1.4}$	$75.3_{\pm 2.5}$	$\mathbf{76.2}_{\pm 0.8}$	$74.2_{\pm 1.9}$	$73.5_{\pm 3.0}$	$76.1_{\pm 1.3}$	$76.4_{\pm 2.3}$	$\mathbf{77.9}_{\pm 2.8}$
HumanEval+†	$80.9_{\pm 1.5}$	$78.0_{\pm 2.7}$	$80.3_{\pm 1.3}$	$82.3_{\pm 2.7}$	$\mathbf{82.7}_{\pm 2.5}$	$80.7_{\pm 1.9}$	$\mathbf{83.9}_{\pm 3.0}$	$82.5_{\pm 1.3}$	$81.3_{\pm 2.5}$	$83.1_{\pm 1.5}$	$80.7_{\pm 1.8}$	$81.7_{\pm 2.8}$	$86.8_{\pm 1.3}$	$87.2_{\pm 2.2}$	$\mathbf{89.6}_{\pm 2.7}$
AIME24	$40.0_{\pm 3.3}$	$38.9_{\pm 3.8}$	$42.2_{\pm 1.9}$	$40.0_{\pm 3.3}$	$\mathbf{46.7}_{\pm 3.3}$	$54.4_{\pm 1.9}$	$50.0_{\pm 3.3}$	$56.7_{\pm 3.3}$	$57.8_{\pm 1.9}$	$\mathbf{58.9}_{\pm 3.8}$	$64.4_{\pm 1.9}$	$64.4_{\pm 3.8}$	$64.4_{\pm 1.9}$	$67.8_{\pm 3.8}$	$\mathbf{73.3}_{\pm 3.3}$
AIME25	$34.4_{\pm 1.9}$	$36.7_{\pm 3.3}$	$38.9_{\pm 1.9}$	$40.0_{\pm 3.3}$	$\mathbf{43.3}_{\pm 3.3}$	$51.1_{\pm 3.8}$	$55.6_{\pm 3.8}$	$52.2_{\pm 1.9}$	$54.4_{\pm 3.8}$	$\mathbf{58.9}_{\pm 3.8}$	$58.9_{\pm 1.9}$	$61.1_{\pm 3.8}$	$62.2_{\pm 1.9}$	$64.4_{\pm 3.8}$	$\mathbf{68.9}_{\pm 1.9}$
GPQA	$28.8_{\pm 2.5}$	$29.0_{\pm 3.6}$	$30.0_{\pm 1.8}$	$31.1_{\pm 2.9}$	$\mathbf{35.0}_{\pm 3.4}$	$42.9_{\pm 2.3}$	$44.1_{\pm 3.7}$	$44.3_{\pm 1.8}$	$46.8_{\pm 3.0}$	$\mathbf{47.5}_{\pm 2.7}$	$50.5_{\pm 2.2}$	$54.4_{\pm 3.5}$	$53.7_{\pm 1.6}$	$52.0_{\pm 2.8}$	$\mathbf{55.4}_{\pm 3.3}$
Hierarchical MAS Setting															
ARC-E	$97.7_{\pm 1.6}$	$97.5_{\pm 1.1}$	$97.2_{\pm 1.1}$	$97.2_{\pm 2.0}$	$\mathbf{98.4}_{\pm 0.7}$	$96.9_{\pm 1.4}$	$98.4_{\pm 1.2}$	$\mathbf{99.2}_{\pm 0.1}$	$98.6_{\pm 0.9}$	$99.1_{\pm 0.3}$	$97.0_{\pm 1.4}$	$98.6_{\pm 0.5}$	$98.2_{\pm 1.0}$	$98.9_{\pm 1.0}$	$\mathbf{99.4}_{\pm 0.4}$
ARC-C	$\mathbf{94.5}_{\pm 1.8}$	$93.0_{\pm 2.8}$	$92.3_{\pm 1.4}$	$90.4_{\pm 2.3}$	$93.9_{\pm 2.9}$	$93.5_{\pm 1.9}$	$93.8_{\pm 2.6}$	$94.6_{\pm 1.2}$	$94.4_{\pm 2.4}$	$\mathbf{95.3}_{\pm 1.0}$	$96.7_{\pm 1.9}$	$95.7_{\pm 3.0}$	$96.6_{\pm 1.3}$	$96.1_{\pm 1.9}$	$\mathbf{97.0}_{\pm 2.5}$
GSM8K	$88.3_{\pm 1.9}$	$90.1_{\pm 2.8}$	$86.5_{\pm 1.5}$	$89.9_{\pm 2.7}$	$\mathbf{91.2}_{\pm 2.5}$	$89.7_{\pm 2.1}$	$91.2_{\pm 3.0}$	$88.6_{\pm 1.7}$	$91.5_{\pm 2.2}$	$\mathbf{92.8}_{\pm 1.0}$	$89.6_{\pm 1.9}$	$91.6_{\pm 3.2}$	$93.4_{\pm 1.3}$	$93.0_{\pm 2.2}$	$\mathbf{94.7}_{\pm 2.6}$
MedQA	$66.5_{\pm 2.5}$	$65.5_{\pm 3.7}$	$66.6_{\pm 1.6}$	$68.0_{\pm 3.0}$	$\mathbf{69.2}_{\pm 2.8}$	$76.9_{\pm 2.1}$	$\mathbf{80.2}_{\pm 3.4}$	$76.6_{\pm 1.8}$	$76.8_{\pm 2.8}$	$78.9_{\pm 1.1}$	$79.7_{\pm 2.4}$	$\mathbf{80.7}_{\pm 3.0}$	$78.1_{\pm 1.6}$	$79.2_{\pm 2.9}$	$80.4_{\pm 3.4}$
MBPP+	$68.4_{\pm 2.0}$	$70.0_{\pm 3.4}$	$69.6_{\pm 1.2}$	$69.8_{\pm 2.9}$	$\mathbf{72.5}_{\pm 2.6}$	$72.1_{\pm 1.9}$	$74.6_{\pm 3.4}$	$71.3_{\pm 1.5}$	$73.5_{\pm 2.6}$	$\mathbf{75.0}_{\pm 1.3}$	$73.3_{\pm 2.2}$	$73.9_{\pm 3.2}$	$75.4_{\pm 1.5}$	$\mathbf{77.1}_{\pm 2.5}$	$76.8_{\pm 3.0}$
HumanEval+†	$75.8_{\pm 1.4}$	$75.0_{\pm 2.8}$	$78.3_{\pm 1.3}$	$80.3_{\pm 2.5}$	$\mathbf{81.1}_{\pm 2.7}$	$78.9_{\pm 1.5}$	$77.6_{\pm 3.0}$	$78.0_{\pm 1.2}$	$77.4_{\pm 2.2}$	$\mathbf{80.9}_{\pm 1.9}$	$82.5_{\pm 1.5}$	$85.0_{\pm 3.0}$	$84.8_{\pm 1.1}$	$83.5_{\pm 1.8}$	$\mathbf{89.4}_{\pm 2.9}$
AIME24	$35.6_{\pm 1.9}$	$42.2_{\pm 3.8}$	$41.1_{\pm 1.9}$	$42.2_{\pm 3.8}$	$\mathbf{44.4}_{\pm 3.8}$	$54.4_{\pm 1.9}$	$53.3_{\pm 3.3}$	$53.3_{\pm 3.3}$	$54.4_{\pm 3.8}$	$\mathbf{56.7}_{\pm 5.8}$	$67.8_{\pm 1.9}$	$67.8_{\pm 3.8}$	$73.3_{\pm 3.3}$	$74.4_{\pm 3.8}$	$\mathbf{75.6}_{\pm 1.9}$
AIME25	$35.6_{\pm 3.8}$	$36.7_{\pm 3.3}$	$37.8_{\pm 1.9}$	$36.7_{\pm 3.3}$	$\mathbf{38.9}_{\pm 3.8}$	$52.2_{\pm 1.9}$	$51.1_{\pm 3.8}$	$50.0_{\pm 3.3}$	$51.1_{\pm 3.8}$	$\mathbf{53.3}_{\pm 3.3}$	$64.4_{\pm 1.9}$	$66.7_{\pm 3.3}$	$67.8_{\pm 1.9}$	$65.6_{\pm 1.9}$	$\mathbf{70.0}_{\pm 3.3}$
GPQA	$28.3_{\pm 2.6}$	$28.1_{\pm 3.7}$	$29.5_{\pm 2.3}$	$30.6_{\pm 3.5}$	$\mathbf{33.5}_{\pm 3.5}$	$42.1_{\pm 2.1}$	$43.6_{\pm 3.6}$	$47.5_{\pm 2.0}$	$\mathbf{50.7}_{\pm 3.1}$	$48.8_{\pm 1.6}$	$50.8_{\pm 2.6}$	$53.0_{\pm 3.9}$	$52.7_{\pm 1.8}$	$\mathbf{57.2}_{\pm 2.8}$	$52.9_{\pm 3.7}$

principal-direction selection beyond rank reduction. Over the fivefold noise increase, this corresponds to an effective exponent of approximately 0.58, close to the square-root reference of 0.5. These proxy measurements support the SNR analysis over the tested range.

4 Experiments

This section evaluates the approach across nine datasets, covering general and reasoning-intensive tasks, to address the following research questions:

- **RQ1:** How does the proposed approach compare to text- or latent-based approaches, particularly regarding purified communication?
- **RQ2:** How does the mechanism balance computational throughput and inference scalability under accumulating context noise?
- **RQ3:** What are the contributions of spectral centralization, resonance filtering, and adaptive spectral gain to overall performance?
- **RQ4:** How does the additive mechanism correct latent trajectories and preserve spectral density in high-noise environments?
- **RQ5:** How does the SNR-Adaptive Spectral Gain modulate information flow?
- **RQ6:** How well does the additive mechanism generalize to heterogeneous agents with different model architectures?
- **RQ7:** How robust is the proposed mechanism to variations in its core hyperparameters, specifically the dual spectral gain thresholds (γ, β) and the subspace truncation rank (r)?

4.1 Experimental Setup

Tasks and Datasets. We evaluate nine benchmarks across three domains: (i) mathematical and scientific reasoning (GSM8K [9], AIME24 [43], AIME25 [43], GPQA [48], MedQA [62]); (ii) commonsense reasoning (ARC-Easy [8], ARC-Challenge [8]); and (iii) code generation (MBPP+ [38], HumanEval+ [38]). Appendices C.1 and L detail the benchmarks and prompt templates.

Baselines. We build the MAS using the Qwen3 series [61] at 4B, 8B, and 14B. We compare the proposed approach against four established approaches: (i) Text MAS **(TM)** using natural language [67]; (ii) Text MAS with Purification using GPT-5.4 **(TM-p)** [26] to refine upstream information; (iii) Latent MAS **(LM)** transmitting aligned latent vectors [74]; and (iv) Latent MAS with Purification **(LM-p)**, utilizing the H_2O heavy-hitter oracle [68] to retain high-importance tokens. We exclusively evaluate training-free paradigms to ensure fair comparisons. **Appendix C.3 provides detailed implementation specifications.** Notably, while numerous purification approaches exist, we select the most representative techniques. Due to space constraints, Appendix F details experiments evaluating **alternative purification approaches** (i.e., snapKV [32], CPC [35], Quest [53]).

Evaluation Metrics. We report exact match accuracy between the outputs of MAS and the ground truth across all datasets. Appendix C.2 provides further details.

4.2 Overall Performance (*RQ1*)

Table 2 demonstrates the consistent advantage of the additive mechanism across three scales and two topologies. In sequential MAS, the proposed approach shows gains on reasoning-intensive tasks.

Table 3: Ablation study of key components in the Additive Mechanism. We report exact match accuracy (%) across three Qwen3 scales under the Sequential MAS setting. The "Single" column represents the single-model baseline, and the "Full" column represents the complete proposed approach. **w/o Cen.**: removing centralization; **w/o Res.**: replacing the covariance filter with mean pooling; **w/o Adapt.**: fixing the spectral gain $\alpha = 1$.

Tasks	Qwen3-4B					Qwen3-8B					Qwen3-14B				
	Single	w/o Cen.	w/o Res.	w/o Adapt.	Full	Single	w/o Cen.	w/o Res.	w/o Adapt.	Full	Single	w/o Cen.	w/o Res.	w/o Adapt.	Full
ARC-E	95.2	95.4	91.7	96.9	**98.3**	95.8	96.7	92.4	98.1	**99.3**	97.1	97.2	94.8	98.5	**99.6**
ARC-C	89.5	89.3	84.6	91.2	**93.4**	90.8	92.8	88.1	94.5	**96.3**	92.3	93.4	89.7	95.1	**96.8**
GSM8K	82.7	88.3	84.1	89.7	**92.6**	81.4	91.8	87.2	90.5	**95.1**	83.5	91.3	88.6	92.4	**94.8**
MedQA	48.1	65.4	61.2	67.1	**69.5**	52.7	73.6	68.9	74.8	**77.4**	64.9	78.7	73.4	80.2	**82.3**
MBPP+	63.2	71.9	66.3	73.8	**76.5**	65.1	71.3	66.8	73.1	**76.2**	68.2	73.6	69.4	75.8	**77.9**
HumanEval+	75.2	78.5	73.6	80.3	**82.7**	74.8	78.5	75.0	80.7	**83.1**	76.4	85.4	79.7	87.2	**89.6**
AIME24	33.3	38.9	34.4	41.1	**46.7**	48.9	53.3	47.8	55.6	**58.9**	61.1	64.4	60.0	66.7	**73.3**
AIME25	31.1	38.9	33.3	41.1	**43.3**	45.6	54.4	48.9	56.7	**58.9**	55.6	63.3	57.8	65.6	**68.9**
GPQA	28.5	30.8	26.4	32.3	**35.0**	40.2	42.1	37.5	44.3	**47.5**	48.1	50.8	44.8	52.5	**55.4**

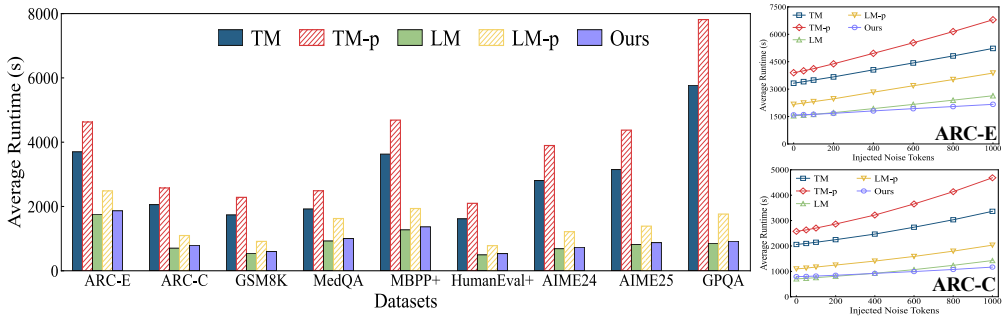


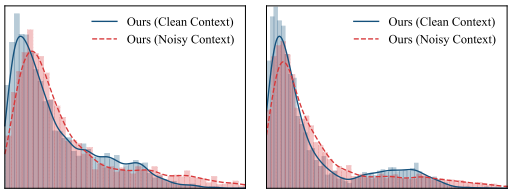
Figure 5: Runtime analysis. The left panel compares average end-to-end inference latency. The right panels illustrate runtime scaling on ARC-C and ARC-E with varying injected noise tokens.

For instance, Qwen3-8B reaches 58.9% exact match accuracy on AIME25, surpassing the 51.1% of the text baseline (TM) and the 52.2% of the latent baseline (LM). This improvement is pronounced in smaller models; Qwen3-4B achieves 43.3% on AIME25, narrowing the capability gap to the 8B class. Although purification strategies occasionally perform competitively (e.g., scoring 83.9% vs. 83.1% on sequential HumanEval+ with Qwen3-8B, and 57.2% vs. 52.9% on hierarchical GPQA with Qwen3-14B), the proposed approach achieves the highest accuracy in 43 of 54 configurations. Furthermore, in the hierarchical setting with noise accumulation, the approach maintains resilience. It improves the accuracy of Qwen3-4B on AIME24 from 35.6% (TM) to 44.4%, indicating the utility of spectral filtering in mitigating information degradation.

4.3 Runtime Efficiency and Scalability Analysis (*RQ2*)

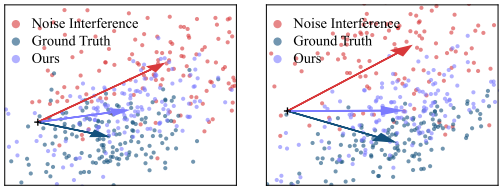
Computational Efficiency Analysis. To assess overhead, we evaluate total end-to-end inference runtime across nine benchmarks using Qwen3-8B (Fig.5, left). Text-based approaches exhibit high latency from autoregressive decoding; on GPQA, Text MAS requires a total of 5771 seconds, whereas the proposed approach requires only 918 seconds (a 6.3× speedup). Because external API calls inflate the latency of TM-p, the primary efficiency comparison is against the fully local baselines (TM, LM, and LM-p). Vanilla Latent MAS holds a marginal edge in low-computation settings (543s vs. 601s on GSM8K) due to the absence of projection overhead. However, the proposed approach consistently outperforms Latent MAS with Purification. This efficiency arises from replacing token generation or heuristic purification with dense matrix multiplication. Consequently, the framework balances latent injection throughput with reasoning performance, achieving runtimes comparable to raw latent models on MBPP+ (1368s vs. 1275s) without explicit filtration costs.

Noise Sensitivity and Scalability. To investigate stability under varying context loads, Fig.5 (right) analyzes the runtimes on ARC-E and ARC-C against increasing injected noise. Latent MAS exhibits steep latency degradation (1542s to 2638s on ARC-E) because attention costs scale with expanding context windows. Conversely, the Resonance Filter projects variable-length upstream history into a fixed-dimension spectral basis. While the upstream covariance aggregation retains an $\mathcal{O}(L_h D^2)$ cost, this mechanism decouples the downstream autoregressive generation cost from the accumulated noise. The approach surpasses the efficiency of Latent MAS beyond approximately 200 noise tokens, achieving a 22% speedup at 1000 tokens. Consistent scalability profiles across other benchmarks (Appendix G) indicate these efficiency gains remain stable across scales and tasks.



(a) ARC-C (b) ARC-E

Figure 6: Spectral robustness analysis.



(a) ARC-C (b) ARC-E

Figure 7: Latent manifold alignment.

4.4 Ablation Study (*RQ3*)

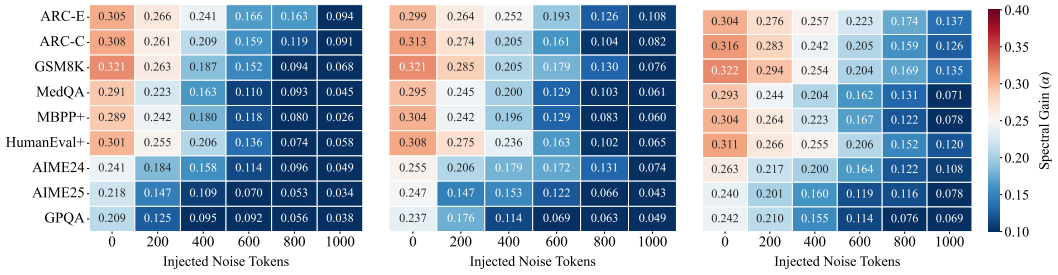
(1) Effectiveness over Single Baseline. To validate the overall framework, we compare the full approach against the single baseline. Table 3 demonstrates substantial performance gains across all scales. Specifically, the accuracy of Qwen3-8B on HumanEval+ rises from 74.8% to 83.1%, confirming that our mechanism successfully elicits multi-agent synergy beyond standalone capabilities. **(2) Impact of Feature Centralization.** To investigate isotropy, we remove mean subtraction (**w/o Cen.**), projecting representations onto the subspace. This modification causes consistent performance degradation across all scales. For example, the accuracy of Qwen3-14B on AIME25 drops from 68.9% to 63.3%, and that of Qwen3-8B on GSM8K falls from 95.1% to 91.8%. These results indicate that without explicit centralization, the dominant common mean vector contaminates principal components, hindering task-specific semantic extraction. **(3) Role of Resonance Filter.** To evaluate the spectral approach against standard aggregation, we replace the covariance-based projection with mean pooling (**w/o Res.**). Consequently, the accuracy of Qwen3-8B on AIME24 decreases from 58.9% to 47.8%, and that of Qwen3-4B on MBPP+ drops from 76.5% to 66.3%. This gap highlights that simple averaging fails to disentangle effective reasoning from background noise. **(4) Role of Adaptive Spectral Gain.** To assess SNR-aware modulation, we fix the spectral coefficient α to 1 (**w/o Adapt.**), disabling dynamic noise gating. This configuration degrades performance in noise-sensitive tasks: the accuracy of Qwen3-8B drops from 95.1% to 90.5% on GSM8K, and from 47.5% to 44.3% on GPQA. These declines show that static injection amplifies noise when upstream signals are weak or incoherent. Appendix H details similar patterns under the hierarchical setting.

4.5 Representation-Level Analysis of Additive Mechanism (*RQ4*)

To examine representation-level mechanics, we conduct geometric and spectral analyses using Qwen3-8B on ARC-C and ARC-E. We contrast clean inputs against polluted inputs injected with 500 task-irrelevant tokens. Extracting last-layer hidden states, we visualize spectral density distributions and map directional alignments of latent trajectories. **(1) Fig.6** illustrates projected spectral density for clean and noisy contexts. The noisy context density curve closely tracks the clean baseline. Overlapping high-density peaks and identical long-tail decay rates indicate the Resonance Filter preserves the principal semantic subspace. **(2) Fig.7** presents a latent space projection mapping contextual directional vectors. The noise interference vector exhibits angular deviation, pulling representations toward noise clusters and causing semantic drift. However, the additive update trajectory resists this deviation. Projecting downstream intent onto the stabilized upstream eigenspace yields a resultant vector aligned with the ground truth. This geometric proximity aligns with the Resonance Filter's theoretical premise: inducing destructive interference along noise directions while constructively amplifying signals, thus maintaining logic chains in high-noise environments.

4.6 Dynamics Behavior of the SNR-Adaptive Spectral Gain (*RQ5*)

To evaluate the SNR-Adaptive Spectral Gain, we conduct controlled noise injection experiments (Fig.8). We inject 0 to 1000 noise tokens across nine benchmarks. Recording the gain $\alpha = \min(\alpha_{\text{global}}, \alpha_{\text{local}})$ applied during downstream decoding quantifies noise sensitivity and adaptability across model capacities and task complexities. At zero noise, the mechanism exhibits scaling in initial spectral resonance. Crucially, this initial state directly captures inherent semantic misalignment across tasks. For instance, tasks with higher agent role divergence, such as GPQA, naturally yield lower initial gains than GSM8K (e.g., 0.209 vs. 0.321 on Qwen3-4B). Furthermore, larger models establish stronger geometric coherence with the upstream subspace, permitting higher initial gains. On GSM8K, the initial gain increases monotonically from 0.321 (4B) to 0.322 (14B). This suggests increased capacity reduces latent anisotropy, allowing the filter to extract principal components with a wider aperture. As noise increases, the spectral gain acts as a safety valve, decaying to



(a) Qwen3-4B (b) Qwen3-8B (c) Qwen3-14B

Figure 8: Heatmaps of the SNR-Adaptive Spectral Gain across varying injected noise tokens.

Table 4: Accuracy (%) for heterogeneous sender–receiver pairs in Sequential MAS.

Dataset	Llama 3.1-8B → Qwen3-8B					Ministral 3-8B → Llama 3.1-8B					Qwen3-4B → Llama 3.1-8B				
	TM	TM-p	LM	LM-p	Ours	TM	TM-p	LM	LM-p	Ours	TM	TM-p	LM	LM-p	Ours
ARC-E	$95.8_{\pm 3.1}$	$98.4_{\pm 1.1}$	$95.5_{\pm 2.3}$	$97.1_{\pm 1.1}$	**98.5**$_{\pm 0.7}$	$93.9_{\pm 0.8}$	**97.5**$_{\pm 0.6}$	$92.8_{\pm 1.9}$	$95.9_{\pm 1.4}$	$96.5_{\pm 0.7}$	$96.0_{\pm 0.3}$	$97.3_{\pm 1.7}$	$93.0_{\pm 2.2}$	$95.3_{\pm 0.7}$	**97.9**$_{\pm 0.9}$
ARC-C	$93.7_{\pm 1.7}$	$94.8_{\pm 1.8}$	$89.8_{\pm 2.9}$	$93.3_{\pm 4.0}$	**95.3**$_{\pm 3.9}$	$93.3_{\pm 2.7}$	**93.7**$_{\pm 2.1}$	$87.8_{\pm 0.8}$	$92.3_{\pm 1.2}$	$93.0_{\pm 3.4}$	$92.3_{\pm 1.5}$	$92.2_{\pm 2.2}$	$86.9_{\pm 3.5}$	$90.2_{\pm 4.7}$	**92.7**$_{\pm 1.2}$
GSM8K	$92.6_{\pm 1.1}$	$92.1_{\pm 3.3}$	$91.5_{\pm 0.8}$	$92.7_{\pm 2.8}$	**94.2**$_{\pm 1.4}$	$92.4_{\pm 0.4}$	$91.0_{\pm 1.1}$	$89.9_{\pm 1.8}$	$91.7_{\pm 3.0}$	**92.5**$_{\pm 1.5}$	$90.1_{\pm 0.6}$	**91.4**$_{\pm 3.5}$	$87.7_{\pm 1.3}$	$90.3_{\pm 2.9}$	**91.4**$_{\pm 1.8}$
MedQA	$76.4_{\pm 2.8}$	$77.2_{\pm 2.5}$	$73.4_{\pm 3.7}$	$75.3_{\pm 3.2}$	**77.7**$_{\pm 1.6}$	$75.0_{\pm 1.1}$	**76.8**$_{\pm 0.7}$	$71.9_{\pm 1.2}$	$73.3_{\pm 0.7}$	$75.9_{\pm 2.3}$	$71.6_{\pm 3.2}$	$71.4_{\pm 1.0}$	$66.5_{\pm 0.5}$	$70.3_{\pm 1.3}$	**72.2**$_{\pm 3.2}$
AIME24	**60.0**$_{\pm 3.3}$	$50.0_{\pm 3.3}$	$50.0_{\pm 3.3}$	$51.1_{\pm 1.9}$	$55.6_{\pm 1.9}$	$53.3_{\pm 3.3}$	$52.2_{\pm 5.1}$	$56.7_{\pm 3.3}$	$55.6_{\pm 1.9}$	**60.0**$_{\pm 3.3}$	$41.1_{\pm 1.9}$	$45.6_{\pm 5.1}$	$41.1_{\pm 1.9}$	$43.3_{\pm 3.3}$	**50.0**$_{\pm 6.7}$
AIME25	$46.7_{\pm 3.3}$	$54.4_{\pm 1.9}$	$48.9_{\pm 1.9}$	**57.8**$_{\pm 3.8}$	$50.0_{\pm 3.3}$	$50.0_{\pm 3.3}$	$52.2_{\pm 1.9}$	$50.0_{\pm 3.3}$	$46.7_{\pm 5.8}$	**54.4**$_{\pm 1.9}$	$42.2_{\pm 1.9}$	$47.8_{\pm 5.1}$	**51.1**$_{\pm 1.9}$	$37.8_{\pm 1.9}$	$41.1_{\pm 1.9}$
GPQA	$42.9_{\pm 0.5}$	$43.4_{\pm 3.9}$	$42.3_{\pm 2.5}$	$44.9_{\pm 1.7}$	**46.0**$_{\pm 2.7}$	$42.4_{\pm 0.5}$	$42.4_{\pm 4.3}$	$40.9_{\pm 1.3}$	$43.4_{\pm 2.2}$	**45.8**$_{\pm 2.4}$	$35.7_{\pm 1.5}$	$35.0_{\pm 4.2}$	$33.8_{\pm 1.3}$	$36.4_{\pm 2.2}$	**38.4**$_{\pm 1.5}$
MBPP+	$67.5_{\pm 1.7}$	$69.6_{\pm 3.4}$	$71.2_{\pm 1.7}$	$73.0_{\pm 2.4}$	**75.1**$_{\pm 1.0}$	$66.1_{\pm 0.5}$	$68.5_{\pm 2.5}$	$70.1_{\pm 1.0}$	$72.2_{\pm 2.1}$	**73.8**$_{\pm 2.6}$	$69.3_{\pm 1.2}$	$72.2_{\pm 1.2}$	$69.0_{\pm 0.3}$	$70.9_{\pm 1.7}$	**73.5**$_{\pm 1.6}$
HumanEval+	$79.7_{\pm 1.9}$	$80.5_{\pm 2.8}$	$81.1_{\pm 1.2}$	$82.1_{\pm 1.5}$	**85.0**$_{\pm 1.9}$	$78.5_{\pm 1.9}$	$79.1_{\pm 5.5}$	$77.6_{\pm 2.7}$	$76.6_{\pm 1.5}$	**83.5**$_{\pm 1.2}$	$80.5_{\pm 1.6}$	**81.7**$_{\pm 3.7}$	$77.4_{\pm 1.6}$	$79.3_{\pm 3.2}$	$79.9_{\pm 3.8}$

protect downstream semantics. This persistent non-zero gain suggests the filter continuously extracts constructive interference, outperforming fixed α baselines (Table 3).

4.7 Generalization to Heterogeneous Agents (*RQ6*)

To evaluate heterogeneous collaboration, we are conducting experiments on Sequential MAS, averaging scores over three random seeds. The sender–receiver pairs are Llama 3.1-8B [19] → Qwen3-8B, Ministral 3-8B [36] → Llama 3.1-8B, and Qwen3-4B → Llama 3.1-8B. The hidden dimensions of sender u and receiver d are $D_u = D_d = 4096$ for the first two pairs and $D_u = 2560$, $D_d = 4096$ for the third. We adopt prior work [74] as preprocessing, with a fixed map $W_a^{u\to d} \in \mathbb{R}^{D_u \times D_d}$ shared by LM, LM-p, and Ours for each pair. For M lexical-anchor pairs, $A \in \mathbb{R}^{M \times D_u}$ contains sender output-layer token vectors. Each row of $B \in \mathbb{R}^{M \times D_d}$ averages receiver input embeddings obtained by decoding the corresponding sender token and retokenizing it with the receiver tokenizer. Ridge regression yields $W_a^{u\to d} = \left(A^\top A + \lambda I_{D_u}\right)^{-1} A^\top B$, where $\lambda > 0$ is the regularization coefficient and I_{D_u} the identity matrix of size D_u. **The Resonance Filter operates after alignment and requires only sender states mapped into the receiver input-embedding space.** Table 4 shows that Ours ranks best in 19 settings at the reported precision. Its macro-average accuracy of 73.7% exceeds TM-p, the strongest aggregate baseline (72.2%), by 1.54 percentage points. For the $2560 \to 4096$ pair, Ours is best or tied-best on seven of nine datasets, averaging 70.8% versus 70.5% for TM-p, but trails TM-p on HumanEval+ (79.9% versus 81.7%). These results support generalization to the tested heterogeneous pairs under a shared alignment interface, with gains varying by task.

4.8 Generalizability, Case Studies, Sensitivity, Related Work & Limitations

Due to space constraints, Appendix I details experiments evaluating the applicability of the approach across different LLM architectures (e.g., **Llama** [19] and **Ministral** [36]). Furthermore, Appendix K provides case studies analyzing problem-solving tasks, Appendix J evaluates hyperparameter robustness, Appendix D reviews related literature, and Appendix E discusses the limitations.

5 Conclusion

This work addresses a bottleneck in LLM-driven MAS: the Attention Distraction Dilemma. By conceptualizing upstream representations as a semantic manifold, we propose a training-free additive mechanism that shifts the interaction paradigm from Softmax competition to linear subspace projection. Theoretical analysis via the Asymptotic SNR Tipping Point Theorem and empirical results indicate that the proposed approach outperforms baselines in 43 of 54 settings, maintaining performance in high-noise regimes. These findings illustrate the utility of adapting attention structures for continuous agent communication, providing a basis for long-context collaborative reasoning.

Acknowledgments and Disclosure of Funding

This work was supported by the Ant Group Research Fund. The authors received no other third-party funding or support for this work during the 36 months prior to submission. The authors declare no competing interests beyond the affiliations listed in this paper.

References

[1] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B Divya. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEe Access*, 13:18912–18936, 2025.

[2] David Anderson, Simon Du, Michael Mahoney, Christopher Melgaard, Kunming Wu, and Ming Gu. Spectral gap error bounds for improving cur matrix decomposition and the nyström method. In *Artificial intelligence and statistics*, pages 19–27. PMLR, 2015.

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] M Cemri, MZ Pan, S Yang, LA Agrawal, B Chopra, R Tiwari, K Keutzer, A Parameswaran, D Klein, K Ramchandran, et al. Why do multi-agent llm systems fail? arxiv preprint. *arXiv preprint arXiv:2503.13657*, 2025.

[6] Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025.

[7] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pages 2793–2803. PMLR, 2021.

[11] Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. *arXiv preprint arXiv:2511.09149*, 2025.

[12] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 55–65, 2019.

[13] Jacob Fein-Ashley, Dhruv Parikh, Rajgopal Kannan, and Viktor Prasanna. Mixture of thoughts: Learning to aggregate what experts think, not just what they say. *arXiv preprint arXiv:2509.21164*, 2025.

[14] Z Feng, R Xue, L Yuan, Y Yu, N Ding, M Liu, B Gao, J Sun, X Zheng, and G Wang. Multi-agent embodied ai: Advances and future directions. arxiv 2025. *arXiv preprint arXiv:2505.05108*.

[15] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[16] Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. *arXiv preprint arXiv:2510.03215*, 2025.

[17] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019.

[18] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6894–6910, 2021.

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[20] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[21] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[22] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, 2023.

[23] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[24] Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024.

[25] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[26] Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

[27] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[28] Boris Kriuk and Logic Ng. Q-kvcomm: Efficient multi-agent communication via adaptive kv cache compression. In *2026 Second International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, pages 15–21. IEEE, 2026.

[29] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[30] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

[31] Huao Li, Hossein Nourkhiz Mahjoub, Behdad Chalaki, Vaishnav Tadiparthi, Kwonjoon Lee, Ehsan Moradi Pari, Charles Lewis, and Katia Sycara. Language grounded multi-agent reinforcement learning with human-interpretable communication. *Advances in Neural Information Processing Systems*, 37:87908–87933, 2024.

[32] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

[33] Zhuo Li, Weiran Wu, Yunlong Guo, Jian Sun, and Qing-Long Han. Embodied multi-agent systems: A review. *IEEE/CAA Journal of Automatica Sinica*, 12(6):1095–1116, 2025.

[34] Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. *arXiv preprint arXiv:2510.05592*, 2025.

[35] Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24595–24604, 2025.

[36] Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

[37] Hung Ming Liu. Ai mother tongue: Self-emergent communication in marl via endogenous symbol systems. *arXiv preprint arXiv:2507.10566*, 2025.

[38] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

[39] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[40] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[41] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

[42] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[43] Bhrij Patel, Souradip Chakraborty, Wesley A Suttle, Mengdi Wang, Amrit Singh Bedi, and Dinesh Manocha. Aime: Ai system optimization via multiple llm evaluators. *arXiv preprint arXiv:2410.03131*, 2024.

[44] Pouya Pezeshkpour, Eser Kandogan, Nikita Bhutani, Sajjadur Rahman, Tom Mitchell, and Estevam Hruschka. Reasoning capacity in multi-agent systems: Limitations, challenges and human-centered solutions. *arXiv preprint arXiv:2402.01108*, 2024.

[45] Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. *arXiv preprint arXiv:2310.06272*, 2023.

[46] Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024.

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[48] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First conference on language modeling*, 2024.

[49] Keitaro Sakamoto and Issei Sato. Benign overfitting in token selection of attention mechanism. *arXiv preprint arXiv:2409.17625*, 2024.

[50] Xiangyu Shi, Marco Chiesa, Gerald Q Maguire Jr, and Dejan Kostic. Kvcomm: Enabling efficient llm communication through selective kv sharing. *arXiv preprint arXiv:2510.03346*, 2025.

[51] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[52] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

[53] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024.

[54] Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. Augmenting multi-agent communication with state delta trajectory. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10230–10251, 2025.

[55] Tadahiro Taniguchi, Ryo Ueda, Tomoaki Nakamura, Masahiro Suzuki, and Akira Taniguchi. Generative emergent communication: Large language model is a collective world model. *arXiv preprint arXiv:2501.00226*, 2024.

[56] William Timkey and Marten Van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, 2021.

[57] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

[58] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[60] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[61] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[62] Hang Yang, Hao Chen, Hui Guo, Yineng Chen, Ching-Sheng Lin, Shu Hu, Jinrong Hu, Xi Wu, and Xin Wang. Llm-medqa: Enhancing medical question answering through case studies in large language models. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

[63] Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, et al. Kvcomm: Online cross-context kv-cache communication for efficient llm-based multi-agent systems. *arXiv preprint arXiv:2510.12872*, 2025.

[64] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

[65] Ye Yu, Heming Liu, Haibo Jin, Xiaopeng Yuan, Peng Kuang, and Haohan Wang. Learning to communicate: Toward end-to-end optimization of multi-agent language systems. *arXiv preprint arXiv:2604.21794*, 2026.

[66] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024.

[67] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237, 2024.

[68] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

[69] Yujia Zheng, Zhuokai Zhao, Zijian Li, Yaqi Xie, Mingze Gao, Lizhu Zhang, and Kun Zhang. Thought communication in multiagent collaboration. *arXiv preprint arXiv:2510.20733*, 2025.

[70] Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. Aim: Adaptive inference of multi-modal llms via token merging and pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20180–20192, 2025.

[71] Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15990–16009, 2025.

[72] Mingjie Zhu, Mingcheng Li, Mingzhang Cui, Ziyun Qian, Yue Jiang, Minghao Han, Mingyang Sun, Zizhi Chen, Yizhou Liu, Jinghang Han, et al. A survey on large language model-based multi-agent systems: Paradigms, applications, and challenges. *Authorea Preprints*, 2026.

[73] Mu Zhu and Ali Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006.

[74] Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, et al. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*, 2025.

A Detailed Theoretical Derivations

In this appendix, we provide the rigorous mathematical derivations supporting the *Asymptotic Signal-to-Noise Ratio Tipping Point Theorem* presented in Section 3.3. We first formalize the geometric bounds for standard Softmax attention, then derive the scaling laws for the proposed Additive Mechanism, and finally prove the existence of the critical tipping point.

A.1 Preliminaries and Notation

Throughout this appendix, we employ standard mathematical and probabilistic notations. $\exp(\cdot)$ denotes the natural exponential function, and $\sup(\cdot)$ represents the supremum. For random variables, $\mathbb{E}[\cdot]$, $\mathrm{Var}(\cdot)$, and $\mathbb{P}(\cdot)$ denote the expected value, variance, and probability measure, respectively. For vector and matrix operations, $\|\cdot\|_2$ denotes the Euclidean (L_2) norm, $\|\cdot\|_F$ denotes the Frobenius norm, and $\langle\cdot,\cdot\rangle$ denotes the standard inner product. $\mathbf{0}$ represents the zero vector.

In the context of multi-agent interactions, let the superscript $(i-1)$ denote the upstream agent and (i) denote the downstream agent. We define D as the hidden dimension and d_{head} as the attention head dimension. For the downstream agent, $x_m^{(i)}, x_n^{(i)} \in \mathbb{R}^{1\times D}$ denote the input intent representations at absolute token positions m and n, respectively. For the upstream agent, let L_h denote the total history sequence length. The aligned continuous trajectory is denoted by $H^{(i-1)} \in \mathbb{R}^{L_h\times D}$. We explicitly define $\hat{h}_t^{(i-1)} \in \mathbb{R}^{1\times D}$ as the centralized hidden state corresponding to token $t \in \{1,\dots,L_h\}$.

For the additive mechanism, $C^{(i-1)} \in \mathbb{R}^{D\times D}$ represents the empirical covariance matrix of the upstream representations, $r \ll D$ denotes the SVD truncation rank, and α denotes the SNR-Adaptive Spectral Gain. In probabilistic bounds, $\delta \in (0,1)$ represents the allowed failure probability.

For attention mechanisms, let $v_t \in \mathbb{R}^{1\times D}$ denote the Value vector corresponding to token t, and q_t denote the unnormalized pre-Softmax attention logit. The attention probability is given by $P_t = \exp(q_t)/Z$, where $Z = \sum_{\tau\in S\cup B}\exp(q_\tau)$ is the partition function. $W_q, W_k \in \mathbb{R}^{D\times d_{\text{head}}}$ denote the query and key projection matrices, and R_m denotes the orthogonal Rotary Position Embedding (RoPE) matrix at position m.

A.2 Proof of the SNR Decay Bound for Softmax Attention

1. Geometric Bounds on Signal and Noise. In standard cross-attention, the signal and background updates are nonnegative weighted sums of Value vectors: $\Omega_{\text{signal}} = \sum_{k\in S} P_k v_k$ and $\Omega_{\text{bg}} = \sum_{j\in B} P_j v_j$. We first upper-bound the signal magnitude. Since hidden states are bounded, let $R_v = \max_{k\in S}\|v_k\|_2$. By the triangle inequality, we obtain:

$$\|\Omega_{\text{signal}}\|_2 \le \sum_{k\in S} P_k\|v_k\|_2 \le P_{\text{signal}} R_v, \tag{7}$$

where $P_{\text{signal}} = \sum_{k\in S} P_k$ is the total probability mass allocated to the signal.

Next, we lower-bound the noise magnitude. Let $P_{\text{bg}} = \sum_{j\in B} P_j$. The background update can be rewritten as $\Omega_{\text{bg}} = P_{\text{bg}}\hat{v}_{\text{bg}}$, where $\hat{v}_{\text{bg}} = \frac{1}{P_{\text{bg}}}\sum_{j\in B} P_j v_j$ is the weighted centroid of the background tokens. As discussed in Section 2, due to the inherent representation anisotropy of LLMs [12], background vectors cluster within a high-density cone rather than cancelling out geometrically. Consequently, the norm of this centroid is strictly positive, defined as $r_v = \|\hat{v}_{\text{bg}}\|_2 > 0$. Thus:

$$\|\Omega_{\text{bg}}\|_2 = P_{\text{bg}}\|\hat{v}_{\text{bg}}\|_2 = P_{\text{bg}} r_v. \tag{8}$$

Substituting these bounds yields the SNR upper bound. Notice that the shared partition function Z cancels out:

$$\text{SNR}^{\text{attn}} \le \frac{P_{\text{signal}} R_v}{P_{\text{bg}} r_v} = \frac{Z_{\text{signal}}}{Z_{\text{bg}}}\frac{R_v}{r_v}, \tag{9}$$

where Z_{signal} and Z_{bg} denote the unnormalized probability sums.

2. Probabilistic Analysis of Partition Functions. For the signal component, let $M_{\text{sig}} = \sup_{k\in S}(q_k)$. Then, $Z_{\text{signal}} \le L_{\text{signal}}\exp(M_{\text{sig}})$. For the background component, we model the unnormalized scores q_j as weakly correlated random variables. Let $\mu_E = \mathbb{E}[\exp(q_j)] > 0$ and $\sigma_E^2 = \mathrm{Var}(\exp(q_j))$. By linearity of expectation, $\mathbb{E}[Z_{\text{bg}}] = L_{\text{bg}}\mu_E$. Because the variables are modeled as weakly correlated, the variance of their sum mathematically includes cross-covariance terms. To rigorously justify linear variance scaling, we explicitly assume that the sum of these cross-covariances is bounded by $\mathcal{O}(L_{\text{bg}})$, such that $\mathrm{Var}(Z_{\text{bg}}) \le L_{\text{bg}}\tilde{\sigma}_E^2$, where $\tilde{\sigma}_E^2$ absorbs the bounded cross-correlations. Applying Chebyshev's inequality, for any $\epsilon \in (0, \mu_E)$, the probability that Z_{bg} deviates significantly below its mean is bounded by:

$$\mathbb{P}(Z_{\text{bg}} \le L_{\text{bg}}(\mu_E - \epsilon)) \le \frac{L_{\text{bg}}\tilde{\sigma}_E^2}{(L_{\text{bg}}\epsilon)^2} = \frac{\tilde{\sigma}_E^2}{L_{\text{bg}}\epsilon^2}. \tag{10}$$

Thus, with high probability, $Z_{\text{bg}} \geq L_{\text{bg}} C_{\text{bg}}$, where $C_{\text{bg}} = \mu_E - \epsilon$.

3. Final Decay Bound. Combining the geometric and probabilistic bounds, we derive:

$$\text{SNR}^{\text{attn}} \leq \frac{L_{\text{signal}} \exp(M_{\text{sig}})}{L_{\text{bg}} C_{\text{bg}}} \cdot \frac{R_v}{r_v} = \frac{A}{L_{\text{bg}}}, \tag{11}$$

where we formally define the constant $A = \frac{L_{\text{signal}} \exp(M_{\text{sig}}) R_v}{C_{\text{bg}} r_v}$. This gives an $\mathcal{O}(L_{\text{bg}}^{-1})$ upper bound on Softmax SNR under the stated assumptions.

A.3 Proof of the SNR Lower Bound for Additive Mechanism

1. Decomposition of Background Disturbance. The actual additive update is scaled by the adaptive gain α and the length factor $1/L_h$. Because α operates as a global scalar applied equally to both the signal and background components, it factors out and has no influence on the SNR. Furthermore, as the intrinsic per-dimension variance σ_{res}^2 is evaluated on the unscaled residual vectors, its boundedness inherently stems from the bounded domain of the network's hidden states, independent of α. Therefore, we can safely omit α in this scaling behavior analysis and focus exclusively on the unscaled raw background vector $V_{\text{bg}}^{\text{raw}} = \sum_{j \in B} w_j \hat{h}_j^{(i-1)}$, where $w_j = \langle x_m^{(i)}, \hat{h}_j^{(i-1)} \rangle$. Decomposing the weights w_j into a mean μ_w and residual Δw_j, we obtain:

$$V_{\text{bg}}^{\text{raw}} = \mu_w \sum_{j \in B} \hat{h}_j^{(i-1)} + \sum_{j \in B} \Delta w_j \hat{h}_j^{(i-1)} = \underbrace{-\mu_w \sum_{k \in S} \hat{h}_k^{(i-1)}}_{V_{\text{offset}}} + \underbrace{\sum_{j \in B} \Delta w_j \hat{h}_j^{(i-1)}}_{V_{\text{res}}}. \tag{12}$$

By explicitly enforcing the centralization constraint $\sum_{t=1}^{L_h} \hat{h}_t^{(i-1)} = \mathbf{0}$, we have $\sum_{j \in B} \hat{h}_j^{(i-1)} = -\sum_{k \in S} \hat{h}_k^{(i-1)}$. Because the projection weight is a linear inner product $w_t = \langle x_m^{(i)}, \hat{h}_t^{(i-1)} \rangle$, the sum of the background weights evaluates exactly to:

$$\sum_{j \in B} w_j = \langle x_m^{(i)}, \sum_{j \in B} \hat{h}_j^{(i-1)} \rangle = \langle x_m^{(i)}, -\sum_{k \in S} \hat{h}_k^{(i-1)} \rangle = -\sum_{k \in S} w_k. \tag{13}$$

Consequently, the sample mean is deterministically $\mu_w = -\frac{1}{L_{\text{bg}}} \sum_{k \in S} w_k$. Because the signal sum is geometrically bounded by the constant signal length $|S| = L_{\text{signal}}$, we have $\mu_w \sim \mathcal{O}(L_{\text{signal}}/L_{\text{bg}})$. The offset magnitude therefore scales as $\|V_{\text{offset}}\|_2 \sim \mathcal{O}(L_{\text{signal}}^2/L_{\text{bg}})$, which deterministically converges to zero as $L_{\text{bg}} \to \infty$.

2. Vector Interference and SVD Truncation of Linear Drift. By the triangle inequality, $\|V_{\text{bg}}^{\text{raw}}\|_2 \leq \|V_{\text{offset}}\|_2 + \|V_{\text{res}}\|_2$. Since $\|V_{\text{offset}}\|_2$ vanishes asymptotically, the growth is strictly dominated by the residual vector $V_{\text{res}} = \sum_{j \in B} Z_j$, where $Z_j = \Delta w_j \hat{h}_j^{(i-1)}$. The expected squared norm is:

$$\mathbb{E}[\|V_{\text{res}}\|_2^2] = \sum_{j \in B} \mathbb{E}[\|Z_j\|_2^2] + \sum_{j \neq l} \mathbb{E}[\langle Z_j, Z_l \rangle]. \tag{14}$$

In a full, un-truncated dense space D, the downstream query is not perfectly orthogonal to the background covariance, meaning the expectation of the cross-terms $\mathbb{E}[\langle Z_j, Z_l \rangle]$ would be strictly non-zero, theoretically leading to a linear covariance drift ($\mathcal{O}(L_{\text{bg}}^2)$ for the squared norm). However, the explicit low-rank SVD truncation ($C^{(i-1)} \approx U_r \Sigma_r U_r^\top$) introduced in Section 3.1 projects the representations onto the top-r principal components ($r \ll D$). This projection formally eliminates the dense, non-orthogonal noise dimensions. Consequently, under **Assumption 2**, the projected residual vectors in the rank-r subspace are rigorously decorrelated, causing the cross-terms to vanish:

$$\mathbb{E}[\|V_{\text{res}}\|_2^2] \approx \sum_{j \in B} \mathbb{E}[\|Z_j\|_2^2] \leq L_{\text{bg}} \sigma_{\text{res}}^2 r, \tag{15}$$

where r is the truncated subspace dimension and σ_{res}^2 is the bounded variance per dimension. Applying Markov's inequality to the squared norm for a failure probability $\delta/2$, we bound the noise magnitude:

$$\|V_{\text{res}}\|_2 \leq \frac{\sigma_{\text{res}} \sqrt{r}}{\sqrt{\delta/2}} \sqrt{L_{\text{bg}}}. \tag{16}$$

3. Derivation of the SNR Lower Bound and Union Bound. Under **Assumption 1**, the constructive interference of the signal subset ensures that the signal update magnitude grows linearly and is strictly positive ($\|\Omega_{\text{signal}}^{\text{add}}\|_2 \geq S_{\min} > 0$). Because $\|V_{\text{offset}}\|_2 \to 0$, asymptotically we have $\|V_{\text{bg}}^{\text{raw}}\|_2 \leq \|V_{\text{res}}\|_2$. Combining this with the square-root noise growth, we obtain that with probability at least $1 - \delta/2$:

$$\text{SNR}^{\text{add}} \geq \frac{S_{\min}}{\frac{\sigma_{\text{res}} \sqrt{r}}{\sqrt{\delta/2}} \sqrt{L_{\text{bg}}}} = \frac{B_\delta}{\sqrt{L_{\text{bg}}}}, \tag{17}$$

where we define the system constant $B_\delta = \frac{S_{\min}\sqrt{\delta/2}}{\sigma_{\text{res}}\sqrt{r}}$.

To rigorously establish the joint condition $\text{SNR}^{\text{add}} > \text{SNR}^{\text{attn}}$ for Theorem 1 in the finite-L_{bg} regime, we apply a union bound over the separate failure events. Assuming the Softmax upper bound holds with probability at least $1-\delta/2$ and the Additive lower bound holds with probability at least $1-\delta/2$, the union bound guarantees that both bounds hold simultaneously with a combined success probability of at least $1-\delta$. Because $\text{SNR}^{\text{attn}} \sim \mathcal{O}(L_{\text{bg}}^{-1})$ decays faster than $\text{SNR}^{\text{add}} \sim \mathcal{O}(L_{\text{bg}}^{-1/2})$, there strictly exists a critical background threshold L_{crit} beyond which the Additive Mechanism dominates, formally validating the theorem.

A.4 Proof of Asymptotic SNR Tipping Point Theorem

Comparing the derived bounds from Appendices A.2 and A.3,

$$\frac{\text{SNR}^{\text{add}}}{\text{SNR}^{\text{attn}}} \geq \frac{B_\delta/\sqrt{L_{\text{bg}}}}{A/L_{\text{bg}}} = \frac{B_\delta}{A}\sqrt{L_{\text{bg}}}. \tag{18}$$

The Additive Mechanism strictly dominates when this ratio exceeds 1. Solving for L_{bg}, we obtain the critical threshold $L_{\text{crit}} = (A/B_\delta)^2$. Thus, for all $L_{\text{bg}} > L_{\text{crit}}$, the condition $\text{SNR}^{\text{add}} > \text{SNR}^{\text{attn}}$ holds with high probability, concluding the proof.

A.5 Analysis of Relative Positional Equivariance under Additive Mechanism

Integrating the Additive Mechanism at the input embedding layer requires evaluating its compatibility with Rotary Position Embeddings (RoPE) [51]. Because RoPE injects absolute positional information directly into query and key vectors via complex rotations, global perturbations on input embeddings could theoretically disrupt the relative positional manifold. This section demonstrates that the additive formulation inherently maintains the relative dependency of RoPE due to the linearity of projection operators. Furthermore, we analyze how the SNR-Adaptive Spectral Gain restricts both the magnitude and the semantic direction of the perturbation to prevent the disruption of native attention distributions post-softmax.

1. Native RoPE Attention and Positional Dependency.
Omitting the agent index (i), let $x_m, x_n \in \mathbb{R}^{1\times D}$ denote the native input embeddings of the downstream agent at absolute positions m and n, where D represents the embedding dimension. The linear projections for queries and keys use weight matrices $W_q, W_k \in \mathbb{R}^{D\times d_{\text{head}}}$, with d_{head} indicating the attention head dimension. RoPE applies orthogonal rotation matrices $R_m, R_n \in \mathbb{R}^{d_{\text{head}}\times d_{\text{head}}}$ to encode absolute positions.

The query and key vectors after RoPE are $\mathbf{q}_m = x_m W_q R_m$ and $\mathbf{k}_n = x_n W_k R_n$. The native pre-softmax attention score between positions m and n relies on the inner product:

$$\text{Score}(m,n) = \mathbf{q}_m \mathbf{k}_n^\top = (x_m W_q R_m)(x_n W_k R_n)^\top = x_m W_q R_m R_n^\top W_k^\top x_n^\top. \tag{19}$$

Exploiting the orthogonal property of RoPE matrices in the complex plane, $R_m R_n^\top = R_{m-n}$, the absolute positions translate exclusively into a relative positional rotation:

$$\text{Score}(m,n) = x_m W_q \mathbf{R_{m-n}} W_k^\top x_n^\top. \tag{20}$$

In this formulation, the explicit positional rotation depends on the relative distance $m-n$.

2. Structural Consistency under Additive Injection.
The proposed mechanism modulates downstream intent using the filtered upstream covariance matrix $C^{(i-1)}$. Let $\Delta x_t = \alpha x_t C^{(i-1)} \in \mathbb{R}^{1\times D}$ denote the injected contextual perturbation for any position t. The modified input representations for the Transformer become $\tilde{x}_m = x_m + \Delta x_m$ and $\tilde{x}_n = x_n + \Delta x_n$.

Applying the identical RoPE transformation to these modulated vectors yields the updated attention score:

$$\widetilde{\text{Score}}(m,n) = (\tilde{x}_m W_q R_m)(\tilde{x}_n W_k R_n)^\top = \tilde{x}_m W_q (R_m R_n^\top) W_k^\top \tilde{x}_n^\top. \tag{21}$$

Substituting $R_m R_n^\top = R_{m-n}$ and expanding the terms yields four distinct components:

$$\begin{aligned}\widetilde{\text{Score}}(m,n) = &\underbrace{x_m W_q \mathbf{R_{m-n}} W_k^\top x_n^\top}_{\text{Native Relative Attention}} \\ &+ \underbrace{\Delta x_m W_q \mathbf{R_{m-n}} W_k^\top x_n^\top}_{\text{Query-Injected Interference}} + \underbrace{x_m W_q \mathbf{R_{m-n}} W_k^\top \Delta x_n^\top}_{\text{Key-Injected Interference}} \\ &+ \underbrace{\Delta x_m W_q \mathbf{R_{m-n}} W_k^\top \Delta x_n^\top}_{\text{High-Order Interaction}}.\end{aligned} \tag{22}$$

Algorithm 1 Training-Free Additive Mechanism

Input: Upstream last-layer hidden states $H^{(i-1)} \in \mathbb{R}^{L_h \times D}$, downstream initial input embeddings $X^{(i)} \in \mathbb{R}^{L_x \times D}$, energy tolerance γ, safety threshold β, truncation rank r.
Output: Updated downstream input embeddings $\tilde{X}^{(i)}$.

Phase 1: Upstream Subspace Extraction
Compute sequence-wise mean: $\mu^{(i-1)} \leftarrow \frac{1}{L_h} \sum_{t=1}^{L_h} H_t^{(i-1)}$
Perform explicit centralization: $\hat{H}^{(i-1)} \leftarrow H^{(i-1)} - \mathbf{1}_{L_h} (\mu^{(i-1)})^\top$
// Extract top-r eigen-components via Randomized SVD to achieve $\mathcal{O}(L_h D r)$ cost
$(P_r, S_r, V_r) \leftarrow \text{RandomizedSVD}(\hat{H}^{(i-1)}, \text{rank} = r)$
$U_r \leftarrow V_r, \quad \Sigma_r \leftarrow S_r^2 / L_h$
Phase 2: SNR-Adaptive Spectral Gain Calculation
Compute global gain limit: $\alpha_{\text{global}} \leftarrow \frac{\gamma \|X^{(i)}\|_F}{\|(X^{(i)} U_r)\Sigma_r U_r^\top\|_F}$
Initialize local gain limit: $\alpha_{\text{local}} \leftarrow \infty$
for each token position $m \in \{1, \ldots, L_x\}$ **do**
    Extract token representation $x_m^{(i)}$ from $X^{(i)}$
    Compute token-wise constraint: $v_m \leftarrow \beta \frac{\|(x_m^{(i)} U_r)\Sigma_r U_r^\top\|_2}{\|x_m^{(i)}\|_2}$
    Update local limit: $\alpha_{\text{local}} \leftarrow \min(\alpha_{\text{local}}, v_m)$
end for
Determine final spectral gain: $\alpha \leftarrow \min(\alpha_{\text{global}}, \alpha_{\text{local}})$
Phase 3: Additive Resonance Injection
// Associative multiplication avoids materializing the dense $D \times D$ covariance matrix
Update downstream trajectory: $\tilde{X}^{(i)} \leftarrow X^{(i)} + \alpha (X^{(i)} U_r)\Sigma_r U_r^\top$
return $\tilde{X}^{(i)}$

Equation 22 shows that every interference term contains the same relative rotation matrix $\mathbf{R_{m-n}}$. Under the linear query–key model, the perturbation Δx_m receives the same positional rotation as the host token x_m, preserving this relative-rotation structure.

3. Magnitude and Directional Bounding against Softmax Sensitivity.
While Equation 22 establishes structural consistency in the logits, the non-linear softmax operation remains highly sensitive to dot-product anomalies. An unbounded or adversarial perturbation could generate abnormal alignments between distant queries and keys, effectively washing out the high-frequency local diagonal structures typical of RoPE post-softmax.

This vulnerability necessitates the SNR-Adaptive Spectral Gain (α). Following Equation 3, the global gain limit α_{global} bounds the Frobenius norm ($\|\cdot\|_F$) of the injection matrix $\Delta X^{(i)}$ relative to the native input matrix $X^{(i)}$:

$$\|\Delta X^{(i)}\|_F = \|\alpha X^{(i)} C^{(i-1)}\|_F \leq \gamma \|X^{(i)}\|_F, \tag{23}$$

where $\gamma > 0$ denotes the macro-energy tolerance hyperparameter. A conservative value of γ, such as $\gamma \ll 1$, strictly constrains the expected perturbation magnitude across the sequence. Beyond this magnitude constraint, the perturbation direction further mitigates attention shifts. The covariance matrix $C^{(i-1)}$ captures the principal statistical components of the upstream representations. Thus, the perturbation Δx_m shifts the token embedding along smooth semantic principal directions, rather than injecting random high-frequency adversarial noise. Together, directional alignment and strict norm control make the interference terms function as localized semantic shifts. In standard autoregressive architectures, the initial input embeddings pass through normalization operations, such as RMSNorm, and preceding MLP layers before reaching the query and key projections. Because normalization is nonlinear, with $f(x_m + \Delta x_m) \neq f(x_m) + f(\Delta x_m)$, the linear decomposition in Equation 22 should be interpreted as an idealized first-order approximation. **Nevertheless, the SNR-Adaptive Gain restricts the perturbation to a small fraction of the total energy through $\gamma \ll 1$ and aligns it with the principal semantic subspace**. This design preserves the local geometry through the smooth normalization manifold and empirically maintains relative positional structural consistency.

B Algorithm

To explicitly illustrate the deployment of our approach, Algorithm 1 summarizes the forward pass intervention of the Additive Mechanism. Operating entirely training-free, this procedure intervenes precisely at the agent

boundary, transforming the raw upstream hidden states into a compact covariance filter and dynamically injecting it into the downstream agent's initial embeddings.

C Experiment Setting

C.1 Evaluation Datasets

To thoroughly evaluate the robustness, scalability, and efficiency of our proposed Additive Mechanism, we select nine comprehensive benchmarks spanning distinct reasoning domains. Below, we provide detailed descriptions of each dataset utilized in our experiments.

- **GSM8K [9]:** GSM8K is a widely used benchmark consisting of 8.5K grade-school math word problems designed to evaluate multi-step numerical reasoning. Each problem requires decomposing a natural-language description into structured arithmetic steps, making it a standard testbed for assessing chain-of-thought reasoning ability.
- **AIME24 [43]:** AIME24 consists of 30 competition-level problems from the 2024 American Invitational Mathematics Examination. These questions span algebra, geometry, number theory, and combinatorics, and require precise numeric answers with typically 1-3 digits.
- **AIME25 [43]:** AIME25 provides 30 additional problems from the 2025 AIME exam, maintaining the same answer format and difficulty profile. Compared with AIME24, this benchmark includes more multi-phase derivations and intricate combinatorial constructions, offering a complementary stress test for mathematical robustness.
- **GPQA [48]:** We use the GPQA-Diamond split, which is the most difficult subset featuring 198 graduate-level multiple-choice questions written by domain experts in physics, biology, and chemistry. The dataset emphasizes conceptual depth, cross-disciplinary reasoning, and the ability to synthesize multi-step scientific arguments under rigorous distractor settings.
- **MedQA [62]:** MedQA contains real medical licensing exam questions that assess biomedical knowledge, clinical reasoning, and diagnostic decision-making. Problems require integrating textual context with domain-specific medical understanding, making the benchmark a representative testbed for professional-level scientific reasoning.
- **ARC-Easy [8]:** ARC-Easy consists of grade-school science questions from the AI2 Reasoning Challenge that test foundational factual knowledge and straightforward commonsense reasoning. As a simplified subset of ARC, it serves as a baseline measure of basic scientific understanding without requiring complex multi-step inference.
- **ARC-Challenge [8]:** ARC-Challenge includes the most difficult items from the AI2 Reasoning Challenge. These questions are intentionally adversarial, requiring multi-hop reasoning, causal and counterfactual inference, and systematic elimination of distractor choices.
- **MBPP+ [38]:** MBPP+ extends the original MBPP benchmark with broader input coverage, additional hidden test cases, and stricter execution-based evaluation. Each problem requires generating a self-contained Python function that satisfies a comprehensive unit-test suite, making the benchmark a robust measure of code synthesis reliability and correctness.
- **HumanEval+ [38]:** HumanEval+ augments HumanEval with denser, more challenging test suites, significantly increasing the rigor of functional correctness evaluation. The benchmark emphasizes generalization beyond prompt examples and tests a model's ability to produce semantically precise, executable Python code under more demanding verification settings.

C.2 Evaluation Metrics

To ensure rigorous and reproducible assessment, we employ task-specific evaluation criteria [74]. For all natural language benchmarks, the raw generated outputs undergo a standard text normalization pipeline (lowercasing, whitespace trimming, and removal of extraneous punctuation) prior to matching. The specific evaluation protocols for each task category are as follows:

Multiple-Choice Question Answering. For multiple-choice datasets (GPQA-Diamond, MedQA, and the ARC Easy and Challenge variants), we apply an exact matching criterion. The terminal answer string is parsed from the trajectory of the model and strictly compared against the ground-truth option letter.

Mathematical and Numeric Reasoning. For numerical problem-solving tasks (GSM8K, AIME24, and AIME25), we evaluate performance based on strict mathematical equivalence. The predicted answer is isolated, parsed into a numerical float or integer, and compared to the reference solution. Any generation failing to yield a valid, parsable numeric format is classified as incorrect.

Code Generation. For programmatic benchmarks (MBPP-Plus and HumanEval-Plus), we assess functional correctness via test-case execution rather than static string matching. The generated code snippet is extracted, appended with the ground-truth unit tests of the benchmark, and executed within a secure, isolated sandbox environment with a 10-second timeout. A generated solution is deemed successful if and only if it passes all associated unit assertions without runtime errors or timeouts.

C.3 Detailed Implementation Specifications

General Setup. To ensure reproducibility and fair comparisons across training-free paradigms, we conduct experiments using the Qwen3-4B, Qwen3-8B, and Qwen3-14B instruction-tuned models [61]. Text generation across baselines employs a consistent autoregressive decoding strategy with a temperature of 0.6 and a top-p value of 0.95 to balance reasoning diversity and output determinism. Where applicable, evaluation environments are accelerated using the vLLM backend [29]. All matrix manipulations for latent and additive mechanisms execute on NVIDIA A100 GPUs.

Text MAS (TM). The TM baseline follows the standard chain-of-agents paradigm [67]. Upstream agents receive the initial task prompt alongside concatenated natural language responses from preceding agents. These agents perform standard autoregressive decoding to generate textual reasoning steps. The generated text is subsequently appended to the global context window and transmitted as prompt tokens to the downstream agent.

Text MAS With GPT-Driven Semantic Purification (TM-p). To mitigate the accumulation of hallucinated noise in the text-based cascade, TM-p introduces an explicit purification step. Before transmission to downstream agents, the textual response of the upstream agent is processed by the GPT-5.4 API [26]. GPT-5.4 is prompted to act as a semantic filter, summarizing the reasoning trajectory to extract task-relevant core logic while discarding redundant or contradictory statements. Downstream generation is strictly conditioned on this purified summary.

Latent MAS (LM). The LM baseline [74] implements a pure latent-space collaboration framework. Instead of decoding explicit text, upstream agents generate autoregressive latent thoughts using last-layer hidden states. To maintain distributional consistency and prevent activation drift, these hidden states are projected back to the input embedding space via a pre-computed alignment matrix. The reasoning trajectory of the agent is preserved as a latent working memory, comprising layer-wise key and value caches. This cache is directly concatenated to the past key values of the downstream agent, enabling lossless latent information transfer without text-based mediation. Implementation details and hyperparameter configurations for latent communication strictly follow the optimal settings of the original paper [74].

Latent MAS With Attention-Guided Token Pruning (LM-p). While LM avoids the text bottleneck, it remains susceptible to latent noise accumulation. LM-p augments the LM baseline by applying the Heavy-Hitter Oracle [68] as a latent purification mechanism. Before transferring the cache to downstream agents, cumulative attention scores for each latent token are computed across all attention heads and layers. Tokens with the lowest accumulated scores, predominantly representing task-irrelevant noise, are systematically pruned from the cache. The pruning ratio is empirically set to 30 percent, ensuring only the most salient semantic representations propagate downstream. Attention tracking and cache eviction mechanisms follow the default settings of the original H_2O paper [68].

Our approach avoids text decoding for communication and heuristic token dropping. Given the upstream continuous sequence, we compute the sequence-wise mean vector and perform explicit centralization to eliminate representation anisotropy. The empirical covariance matrix is then computed to extract principal semantic directions. For the additive update, the dynamic SNR-adaptive spectral gain is determined on the fly. We set the macro-energy tolerance hyperparameter to 0.15 to constrain the perturbation norm, and the safety threshold to 0.7 to guarantee individual token alignment. For the low-rank approximation of the covariance matrix using Randomized SVD, we dynamically determine the rank r based on the maximum spectral gap, capped at a maximum rank of $r_{\max} = 256$ for efficiency. Empirically, we observed that the intrinsic dimensionality of a single reasoning trajectory is highly low-rank, with the primary semantic directions typically concentrated within the top 32 to 64 singular values. Setting $r_{\max} = 256$ ensures that we retain over 95% of the effective variance (energy) while achieving significant computational speedups over the full $\mathcal{O}(D^2)$ covariance projection.

D Related Work

Dense Latent Communication. This line of research advocates for the direct transmission of raw hidden states or Key-Value (KV) caches across agent boundaries, bypassing the quantization loss and decoding bottlenecks inherent in standard text-based interactions [20, 58, 1, 63, 50]. Early conceptualizations explored aggregating expert hidden states to form unified continuous representations [13]. More recently, frameworks have operationalized this concept at scale; for instance, LatentMAS [74] and Cache-to-Cache [16] enable agents to auto-regressively generate latent thoughts through last-layer embeddings, transferring them as a shared latent working memory [11]. While these dense methods preserve the complete trajectory of the internal reasoning of upstream agents without semantic degradation, they introduce a severe scalability bottleneck. They compel

downstream agents to ingest an ever-expanding context window of uncompressed latent vectors, stretching the reasoning capacity limits of LLMs [44]. This unrestricted accumulation exacerbates the noise sensitivity of standard attention mechanisms, forcing downstream agents to process the entirety of the upstream thought process. This vulnerability aligns with rank-loss failures in long-context reasoning [39, 10, 67].

Purified Latent Communication. To mitigate noise and redundancy in dense feature transfer, several studies propose purified or compressed latent communication [54, 69, 45, 68, 60, 32, 35, 53, 28]. Methods such as ThoughtComm [69] employ trained encoder-decoder modules or feature decoupling techniques to distill the continuous thoughts of upstream agents into a compact set of highly salient latent vectors. Concurrently, recent advancements leverage KV cache eviction strategies, such as SnapKV [32] and Quest [53], or prompt compression techniques like CPC [35], to dynamically retain high-impact tokens and minimize memory overhead. Although purification strategies reduce the absolute length of the transmitted context and alleviate computational burdens, they often rely on heuristic token-dropping or trained routing modules that risk discarding crucial semantic nuances [3, 70]. Furthermore, downstream agents still process these purified vectors via standard concatenation-based Softmax interaction, leaving vulnerabilities to attention hijacking.

RL-Optimized Continuous Communication. An alternative trajectory explores optimizing continuous communication protocols through Reinforcement Learning (RL). Foundational works in Multi-Agent RL (MARL) pioneered this paradigm by treating continuous vectors as communicative actions, utilizing global task rewards to train communication channels end-to-end via backpropagation [52, 15, 40, 65]. Building upon this, recent advancements in LLM-driven MAS adapt these principles to evolve self-emergent, task-specific latent languages among foundation models [37, 55] or to ground communication in human-interpretable semantic spaces [31]. While RL-optimized protocols discover highly efficient communication manifolds, they require extensive multi-agent joint training or iterative fine-tuning. This optimization process restricts their applicability in modern zero-shot MAS paradigms, which demand training-free, plug-and-play frameworks capable of accommodating diverse downstream tasks without retraining overheads [46, 24].

The Attention Distraction Dilemma in MAS. Despite advancements in MAS communication, a structural limitation persists across existing training-free frameworks: the reliance on standard concatenation-based Softmax interactions to integrate upstream information. This paradigm suffers from two interconnected issues. First, in training-free MAS scenarios, agents are typically assigned distinct roles via system prompts. Diverging contextual conditions create a geometric mismatch between the task-specific latent subspaces of upstream and downstream agents, impeding precise Query-Key alignment [58, 5]. Second, due to the anisotropic nature of representations in LLMs, token embeddings tend to occupy a narrow cone in the vector space [12, 18, 17, 41]. Consequently, even task-irrelevant background tokens exhibit artificially high cosine similarities with the downstream query. In the Softmax operation, these high similarities exponentiate and accumulate in the denominator. As the interaction chain deepens, the true semantic signal is diluted by growing background noise [10, 39]. This phenomenon creates the Attention Distraction Dilemma, which our proposed Additive Mechanism addresses.

E Limitations and Future Work

While the proposed Additive Mechanism effectively mitigates the Attention Distraction Dilemma in long-context collaborations, it presents certain limitations that warrant future investigation.

Computational Overhead in Short Contexts. The extraction of the principal semantic subspace relies on empirical covariance estimation and Singular Value Decomposition (SVD). Although this mechanism achieves $\mathcal{O}(1)$ downstream autoregressive complexity relative to the upstream context length, it introduces a fixed mathematical overhead. In scenarios involving extremely short upstream contexts (e.g., fewer than 200 tokens) or minimal interaction turns, this fixed cost may marginally exceed the latency of standard cross-attention. In practical deployments, this can be seamlessly mitigated via a length-aware back-off mechanism, defaulting to standard attention until the accumulated context approaches the theoretical SNR tipping point.

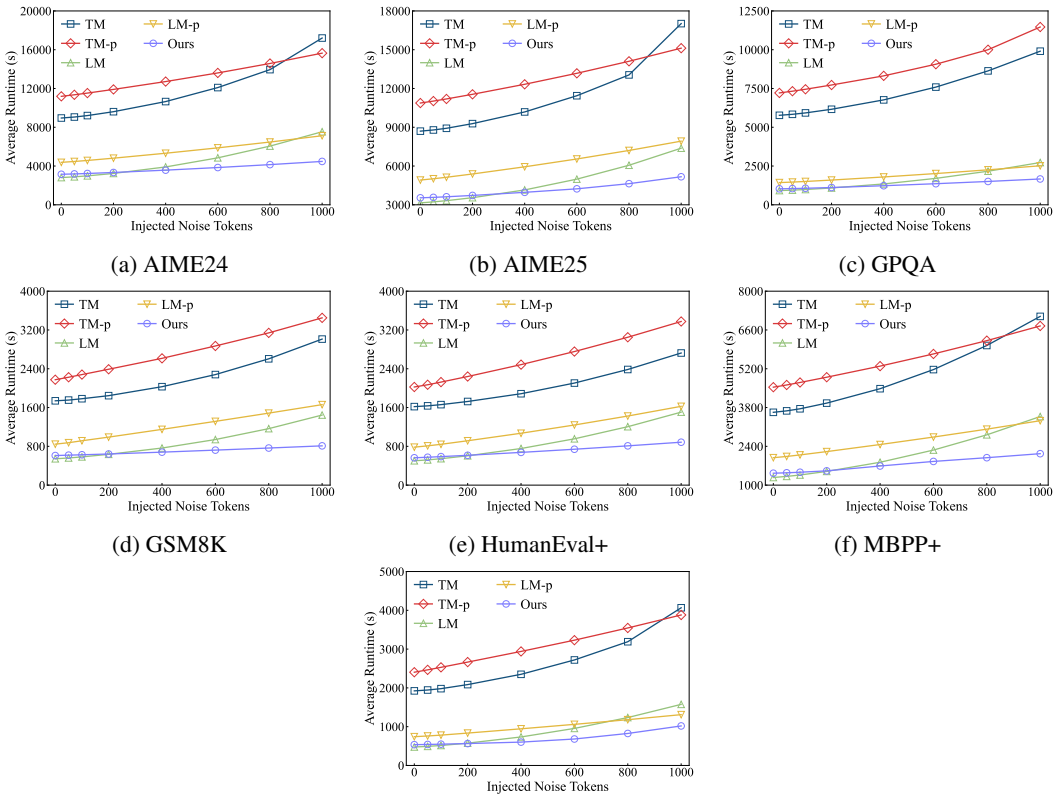
Linearity of Spectral Filtering. Our method conceptualizes the upstream semantic manifold through linear principal components. While empirical results confirm this efficiently isolates task-relevant signals from orthogonal noise, highly non-linear or deeply entangled semantic representations might not be perfectly disentangled by linear spectral filtering alone. Future work could explore kernelized covariance or non-linear manifold projections to capture more complex semantic topologies, albeit potentially at a higher computational cost.

F Purified Baselines Performance

To ensure rigorous evaluation, the experimental setup for the three alternative purification baselines, SnapKV [32], CPC [35], and Quest [53], strictly adheres to the default hyperparameter configurations, compression ratios, and eviction policies detailed in the respective original publications. This aligns with intended inference conditions. Quantitative results across six models yield the following empirical insights.

Table 5: Exact match accuracy (%) of alternative state-of-the-art purification methods (SnapKV, CPC, Quest) across all six evaluated models. Standard deviations across three random seeds for the Qwen3, Llama 3.1 and Ministral 3 families, are reported as subscripts.

Tasks	Qwen3-4B			Qwen3-8B			Qwen3-14B			Llama 3.1 8B			Ministral 3 8B			Ministral 3 14B		
	SnapKV	CPC	Quest	SnapKV	CPC	Quest	SnapKV	CPC	Quest	SnapKV	CPC	Quest	SnapKV	CPC	Quest	SnapKV	CPC	Quest
Sequential MAS Setting																		
ARC-E	$98.9_{\pm 0.2}$	$97.1_{\pm 2.5}$	$99.2_{\pm 0.1}$	$98.9_{\pm 0.3}$	$97.0_{\pm 2.8}$	$98.9_{\pm 0.1}$	$98.5_{\pm 0.2}$	$99.4_{\pm 0.4}$	$99.0_{\pm 0.4}$	$97.7_{\pm 0.3}$	$96.0_{\pm 0.6}$	$98.0_{\pm 0.4}$	$97.5_{\pm 1.4}$	$95.0_{\pm 2.0}$	$98.2_{\pm 0.5}$	$96.4_{\pm 1.2}$	$98.0_{\pm 0.9}$	$96.1_{\pm 1.3}$
ARC-C	$92.0_{\pm 1.8}$	$91.0_{\pm 2.5}$	$91.7_{\pm 1.8}$	$95.2_{\pm 1.3}$	$95.0_{\pm 2.9}$	$95.7_{\pm 1.4}$	$96.0_{\pm 1.9}$	$95.3_{\pm 2.1}$	$96.3_{\pm 1.4}$	$94.1_{\pm 0.6}$	$93.8_{\pm 0.9}$	$94.6_{\pm 0.5}$	$93.5_{\pm 2.1}$	$92.4_{\pm 2.5}$	$94.1_{\pm 2.4}$	$95.7_{\pm 1.7}$	$95.0_{\pm 3.3}$	$96.3_{\pm 1.3}$
GSM8K	$92.1_{\pm 1.5}$	$88.6_{\pm 1.8}$	$92.2_{\pm 1.5}$	$93.3_{\pm 2.6}$	$92.3_{\pm 3.6}$	$94.4_{\pm 2.3}$	$95.2_{\pm 1.3}$	$93.8_{\pm 2.5}$	$95.1_{\pm 2.1}$	$94.6_{\pm 0.7}$	$92.4_{\pm 0.8}$	$94.4_{\pm 0.6}$	$92.3_{\pm 1.4}$	$90.2_{\pm 2.0}$	$93.4_{\pm 2.0}$	$95.5_{\pm 2.0}$	$93.2_{\pm 1.5}$	$96.4_{\pm 1.8}$
MedQA	$69.4_{\pm 2.6}$	$66.6_{\pm 3.2}$	$69.2_{\pm 1.7}$	$76.5_{\pm 2.4}$	$76.7_{\pm 2.7}$	$76.4_{\pm 1.8}$	$82.0_{\pm 2.2}$	$79.7_{\pm 2.4}$	$81.5_{\pm 2.3}$	$77.7_{\pm 1.0}$	$77.0_{\pm 1.6}$	$77.9_{\pm 1.1}$	$72.8_{\pm 3.7}$	$75.5_{\pm 3.3}$	$73.5_{\pm 2.0}$	$80.6_{\pm 2.1}$	$78.4_{\pm 3.1}$	$81.1_{\pm 2.5}$
MBPP+	$73.7_{\pm 1.8}$	$73.5_{\pm 2.3}$	$74.5_{\pm 2.0}$	$75.7_{\pm 2.2}$	$69.0_{\pm 2.4}$	$75.8_{\pm 2.2}$	$76.0_{\pm 2.5}$	$73.8_{\pm 2.4}$	$77.1_{\pm 2.7}$	$73.7_{\pm 1.0}$	$67.5_{\pm 1.7}$	$74.0_{\pm 0.9}$	$73.7_{\pm 3.5}$	$67.0_{\pm 3.9}$	$74.8_{\pm 2.8}$	$74.6_{\pm 2.0}$	$72.8_{\pm 2.9}$	$74.8_{\pm 2.2}$
HumanEval+	$82.7_{\pm 2.0}$	$79.5_{\pm 3.7}$	$82.3_{\pm 1.2}$	$81.3_{\pm 2.1}$	$82.7_{\pm 3.0}$	$82.3_{\pm 1.8}$	$87.2_{\pm 2.1}$	$81.1_{\pm 1.8}$	$87.6_{\pm 2.1}$	$81.9_{\pm 3.0}$	$82.9_{\pm 2.1}$	$81.9_{\pm 2.0}$	$78.7_{\pm 2.1}$	$79.7_{\pm 3.1}$	$79.1_{\pm 2.3}$	$84.1_{\pm 2.8}$	$80.7_{\pm 2.3}$	$85.0_{\pm 1.8}$
AIME24	$38.9_{\pm 5.1}$	$37.8_{\pm 5.1}$	$41.1_{\pm 1.9}$	$56.7_{\pm 3.3}$	$53.3_{\pm 3.3}$	$57.8_{\pm 1.9}$	$67.8_{\pm 1.9}$	$64.4_{\pm 3.8}$	$67.8_{\pm 5.1}$	$57.8_{\pm 1.9}$	$54.4_{\pm 1.9}$	$57.8_{\pm 1.9}$	$54.4_{\pm 1.9}$	$52.2_{\pm 3.8}$	$55.6_{\pm 1.9}$	$66.7_{\pm 3.3}$	$62.2_{\pm 3.8}$	$68.9_{\pm 3.8}$
AIME25	$41.1_{\pm 3.8}$	$34.4_{\pm 5.1}$	$41.1_{\pm 3.8}$	$55.6_{\pm 5.1}$	$53.3_{\pm 3.3}$	$56.7_{\pm 0.0}$	$63.3_{\pm 3.3}$	$58.9_{\pm 3.8}$	$64.4_{\pm 3.8}$	$55.6_{\pm 1.9}$	$54.4_{\pm 3.8}$	$55.6_{\pm 1.9}$	$51.1_{\pm 5.1}$	$51.1_{\pm 3.8}$	$52.2_{\pm 1.9}$	$64.4_{\pm 5.1}$	$58.9_{\pm 1.9}$	$64.4_{\pm 3.8}$
GPQA	$30.8_{\pm 3.1}$	$28.6_{\pm 3.4}$	$31.6_{\pm 1.6}$	$46.1_{\pm 3.2}$	$43.9_{\pm 3.3}$	$47.1_{\pm 2.4}$	$51.5_{\pm 2.2}$	$53.0_{\pm 3.6}$	$52.0_{\pm 2.2}$	$46.8_{\pm 2.5}$	$43.3_{\pm 2.1}$	$47.1_{\pm 0.8}$	$43.6_{\pm 3.8}$	$42.3_{\pm 2.4}$	$43.9_{\pm 2.3}$	$53.2_{\pm 3.8}$	$51.9_{\pm 3.9}$	$53.2_{\pm 2.0}$
Hierarchical MAS Setting																		
ARC-E	$96.8_{\pm 2.1}$	$98.1_{\pm 0.8}$	$96.8_{\pm 2.2}$	$98.4_{\pm 0.3}$	$98.3_{\pm 0.3}$	$98.6_{\pm 0.1}$	$98.6_{\pm 0.3}$	$98.3_{\pm 0.4}$	$99.0_{\pm 0.4}$	$98.6_{\pm 0.1}$	$97.5_{\pm 1.0}$	$98.9_{\pm 0.1}$	$96.2_{\pm 2.2}$	$94.7_{\pm 3.6}$	$97.6_{\pm 1.7}$	$98.1_{\pm 1.3}$	$97.0_{\pm 2.0}$	$98.5_{\pm 0.2}$
ARC-C	$89.5_{\pm 1.6}$	$93.2_{\pm 2.2}$	$90.2_{\pm 2.7}$	$93.6_{\pm 2.0}$	$93.1_{\pm 3.4}$	$95.2_{\pm 1.5}$	$95.7_{\pm 2.3}$	$96.2_{\pm 2.0}$	$96.5_{\pm 1.7}$	$92.7_{\pm 1.2}$	$93.2_{\pm 1.3}$	$93.2_{\pm 0.4}$	$91.2_{\pm 2.7}$	$91.6_{\pm 3.3}$	$91.0_{\pm 2.4}$	$95.2_{\pm 1.6}$	$94.9_{\pm 3.2}$	$95.3_{\pm 2.9}$
GSM8K	$88.9_{\pm 2.8}$	$88.7_{\pm 2.0}$	$89.8_{\pm 2.3}$	$91.4_{\pm 2.0}$	$90.0_{\pm 3.6}$	$91.3_{\pm 1.5}$	$93.2_{\pm 1.7}$	$90.9_{\pm 2.9}$	$93.9_{\pm 1.4}$	$91.1_{\pm 0.4}$	$90.0_{\pm 1.2}$	$91.8_{\pm 0.6}$	$89.6_{\pm 3.2}$	$87.9_{\pm 3.3}$	$90.0_{\pm 1.8}$	$91.1_{\pm 1.7}$	$88.7_{\pm 2.2}$	$90.8_{\pm 2.4}$
MedQA	$67.9_{\pm 2.4}$	$65.8_{\pm 3.7}$	$67.8_{\pm 2.8}$	$76.3_{\pm 2.9}$	$78.4_{\pm 2.5}$	$77.6_{\pm 2.4}$	$78.8_{\pm 2.2}$	$79.9_{\pm 3.0}$	$79.3_{\pm 2.2}$	$76.5_{\pm 0.6}$	$77.5_{\pm 1.6}$	$76.5_{\pm 1.3}$	$74.3_{\pm 3.3}$	$76.0_{\pm 3.7}$	$75.0_{\pm 3.8}$	$78.2_{\pm 2.3}$	$79.1_{\pm 3.7}$	$77.3_{\pm 1.7}$
MBPP+	$69.5_{\pm 3.5}$	$69.0_{\pm 3.4}$	$69.3_{\pm 1.7}$	$72.9_{\pm 1.8}$	$73.1_{\pm 3.1}$	$73.2_{\pm 2.6}$	$77.0_{\pm 3.0}$	$73.9_{\pm 2.7}$	$77.8_{\pm 2.1}$	$72.5_{\pm 1.1}$	$72.7_{\pm 1.7}$	$72.3_{\pm 1.1}$	$71.3_{\pm 4.3}$	$71.5_{\pm 3.0}$	$71.5_{\pm 2.9}$	$76.8_{\pm 2.1}$	$73.1_{\pm 2.8}$	$78.0_{\pm 3.1}$
HumanEval+	$80.3_{\pm 3.1}$	$75.2_{\pm 3.9}$	$80.7_{\pm 1.8}$	$77.4_{\pm 2.1}$	$78.0_{\pm 2.8}$	$77.6_{\pm 1.9}$	$83.5_{\pm 2.2}$	$84.1_{\pm 2.8}$	$83.3_{\pm 2.5}$	$76.4_{\pm 1.9}$	$77.8_{\pm 2.8}$	$76.6_{\pm 1.8}$	$75.0_{\pm 3.4}$	$76.8_{\pm 2.8}$	$74.8_{\pm 2.5}$	$82.1_{\pm 2.3}$	$82.7_{\pm 3.9}$	$83.5_{\pm 2.1}$
AIME24	$42.2_{\pm 1.9}$	$38.9_{\pm 1.9}$	$43.3_{\pm 3.3}$	$53.3_{\pm 3.3}$	$52.2_{\pm 1.9}$	$54.4_{\pm 1.9}$	$74.4_{\pm 1.9}$	$66.7_{\pm 5.8}$	$74.4_{\pm 1.9}$	$52.2_{\pm 3.8}$	$55.6_{\pm 1.9}$	$52.2_{\pm 1.9}$	$52.2_{\pm 5.1}$	$51.1_{\pm 5.1}$	$54.4_{\pm 3.8}$	$74.4_{\pm 5.1}$	$68.9_{\pm 3.8}$	$73.3_{\pm 3.3}$
AIME25	$35.6_{\pm 1.9}$	$35.6_{\pm 1.9}$	$35.6_{\pm 1.9}$	$51.1_{\pm 3.8}$	$51.1_{\pm 3.8}$	$51.1_{\pm 5.1}$	$64.4_{\pm 1.9}$	$64.4_{\pm 3.8}$	$64.4_{\pm 5.1}$	$50.0_{\pm 3.3}$	$53.3_{\pm 3.3}$	$52.2_{\pm 3.8}$	$50.0_{\pm 3.3}$	$50.0_{\pm 3.3}$	$50.0_{\pm 3.3}$	$64.4_{\pm 3.8}$	$64.4_{\pm 5.1}$	$65.6_{\pm 1.9}$
GPQA	$30.1_{\pm 2.1}$	$27.8_{\pm 3.6}$	$30.3_{\pm 3.5}$	$50.3_{\pm 4.2}$	$43.3_{\pm 2.5}$	$51.3_{\pm 3.0}$	$56.6_{\pm 3.5}$	$52.0_{\pm 4.1}$	$57.4_{\pm 3.5}$	$51.3_{\pm 1.8}$	$42.4_{\pm 2.6}$	$51.3_{\pm 1.6}$	$49.5_{\pm 3.3}$	$41.9_{\pm 3.8}$	$49.5_{\pm 3.3}$	$55.9_{\pm 3.0}$	$50.2_{\pm 4.1}$	$56.6_{\pm 2.2}$



(a) AIME24 (b) AIME25 (c) GPQA

(d) GSM8K (e) HumanEval+ (f) MBPP+

(g) MedQA

Figure 9: Average end-to-end inference runtime (in seconds) against varying injected noise tokens across seven benchmark datasets for the Qwen3-14B model.

Although latent purification methods such as Quest and SnapKV significantly outperform discrete prompt compression (CPC) in preserving critical context, they remain susceptible to the attention distraction dilemma inherent in standard Softmax architectures. Quantitative comparisons reveal that the proposed training-free additive mechanism consistently yields performance improvements across both sequential and hierarchical topologies. For instance, in the illustrative comparison under the hierarchical MAS setting using Qwen3-14B on AIME24, Quest and SnapKV both achieve 74.4% exact match accuracy, compared with 75.6% for the proposed approach. On sequential GPQA using Ministral 3 14B, the corresponding reference values are 53.2% for both Quest and SnapKV and 58.4% for the proposed method. These results indicate that evicting or compressing historical tokens via heuristic policies remains limited, and the proposed resonance filter alongside the SNR-adaptive additive projection more mitigates noise accumulation.

Table 6: Ablation study of key components in the Additive Mechanism under the **Hierarchical MAS** setting. We report exact match accuracy (%) across three Qwen3 scales. The "Full" column represents our complete method. **w/o Cen.**: removing centralization; **w/o Res.**: replacing the covariance filter with mean pooling; **w/o Adapt.**: fixing the spectral gain $\alpha = 1$.

Tasks	Qwen3-4B				Qwen3-8B				Qwen3-14B			
	w/o Cen.	w/o Res.	w/o Adapt.	Full	w/o Cen.	w/o Res.	w/o Adapt.	Full	w/o Cen.	w/o Res.	w/o Adapt.	Full
ARC-E	94.6	89.3	96.1	**98.4**	95.8	91.2	97.4	**99.1**	96.5	92.8	97.9	**99.4**
ARC-C	88.7	82.1	90.4	**93.9**	91.1	85.6	92.5	**95.3**	92.7	87.4	94.2	**97.0**
GSM8K	85.3	79.4	86.9	**91.2**	88.3	81.7	89.1	**92.8**	89.6	84.1	91.5	**94.7**
MedQA	63.1	57.8	65.0	**69.2**	72.4	66.5	74.6	**78.9**	74.2	68.3	76.8	**80.4**
MBPP+	65.8	59.6	68.4	**72.5**	68.7	62.3	71.2	**75.0**	70.5	64.2	73.1	**76.8**
HumanEval+	74.2	67.5	76.8	**81.1**	74.6	68.9	77.2	**80.9**	82.1	75.6	85.4	**89.4**
AIME24	37.8	31.1	40.0	**44.4**	50.0	43.3	53.3	**56.7**	68.9	61.1	72.2	**75.6**
AIME25	33.3	26.7	35.6	**38.9**	50.0	43.3	50.0	**53.3**	63.3	56.7	66.7	**70.0**
GPQA	28.5	22.2	31.0	**33.5**	41.6	35.7	44.8	**48.8**	46.0	38.7	49.2	**52.9**

G Runtime Efficiency and Scalability Analysis

To evaluate computational efficiency under accumulating noise, we analyze end-to-end inference runtimes across seven datasets. As shown in Figure 9, injecting 0 to 1000 noise tokens into the upstream context isolates the scalability of different communication paradigms.

Inefficiency of Text-Based Cascades. Textual paradigms exhibit severe bottlenecks. The standard text baseline shows a steep non-linear latency increase due to the quadratic cost of expanding context windows, reaching 17205 seconds on AIME24 at 1000 noise tokens. Although text purification truncates context via external APIs, inherent overhead establishes a high latency floor, requiring 7214 seconds on GPQA even at zero noise.

Latent Bottlenecks Under Noise Accumulation. The pure latent baseline avoids decoding overhead, achieving the lowest initial runtime (543 seconds on GSM8K at zero noise). However, direct cache concatenation causes quadratic degradation, reaching 1445 seconds at 1000 noise tokens. Latent purification mitigates this degradation via token pruning, yielding a linear growth curve. Nevertheless, calculating attention scores imposes an initial computational penalty, starting at 1990 seconds on MBPP+, lagging behind the unpruned baseline (1275 seconds).

Flat Scalability of the Additive Mechanism. The proposed approach compresses variable-length upstream history into a fixed-dimension spectral basis, maintaining a constant downstream context window. This yields a flat runtime curve. Despite a marginal initial overhead for covariance projection (3150 seconds versus 2808 seconds for the latent baseline on AIME24), the proposed method overtakes the pure latent baseline once noise exceeds a 200-token threshold. At 1000 noise tokens on AIME24, the approach requires 4475 seconds, compared to 7535 seconds for the latent baseline. This indicates that substituting quadratic attention computation with constant-overhead dense matrix multiplication effectively mitigates long-context latency bottlenecks.

H Ablation Study with Hierarchical MAS

To validate the robustness of the proposed mechanisms, the ablation study extends to the Hierarchical MAS setting (Table 6). In this topology, downstream agents receive concurrent inputs from multiple upstream reasoning branches, a structure amplifying multi-source representation interference and uncoordinated latent noise. The study isolates the contributions of feature centralization, the resonance filter, and the adaptive spectral gain across the Qwen3 model family to assess effectiveness in mitigating these challenges.

Failure of Simple Aggregation. Replacing the covariance-based resonance filter with standard mean pooling results in severe performance degradation, highlighting issues with naive aggregation in multi-source communication. As concurrent upstream branches inject heterogeneous noise, simple averaging blurs semantic boundaries. On the AIME24 benchmark using the 14B model, accuracy decreases by 14.4% (from 75.6% to 61.1%). Similarly, on the HumanEval+ dataset with the 4B model, performance regresses from 81.1% to 67.5%.

Amplified Anisotropy in Hierarchical Topologies. Removing the explicit centralization step forces downstream agents to project raw representations containing compounded common-mode drift. The hierarchical concatenation of multiple upstream representations exacerbates intrinsic representation anisotropy, obscuring task-specific semantic directions. This omission reduces performance: on the GPQA dataset with the 8B model, accuracy falls from 48.8% to 41.6% (a 7.2% decline). On the AIME25 benchmark using the 4B model, accuracy decreases by 5.6% (from 38.9% to 33.3%).

Crucial Role of Dynamic Noise Gating. Disabling the signal-to-noise ratio adaptive modulation by fixing the spectral coefficient to exactly 1.0 impairs the ability of the system to reject low-quality upstream branches. This static injection forces the downstream model to absorb all spectral projections regardless of localized geometric coherence. Consequently, on the GSM8K benchmark with the 8B model, accuracy decreases from 92.8% to

Table 7: Comparison of exact match accuracy (%) across the Llama 3.1 and Ministral 3 families to demonstrate cross-architecture generalizability. Standard deviations across three random seeds are reported as subscripts. We highlight the highest accuracy in **bold**.

Tasks	Llama 3.1 8B					Ministral 3 8B					Ministral 3 14B				
	TM	TM-p	LM	LM-p	Our	TM	TM-p	LM	LM-p	Our	TM	TM-p	LM	LM-p	Our
Sequential MAS Setting															
ARC-E	$94.9_{\pm 0.6}$	$97.8_{\pm 0.8}$	$97.4_{\pm 0.0}$	$97.7_{\pm 0.4}$	$\mathbf{99.6}_{\pm 0.1}$	$92.8_{\pm 1.9}$	$97.1_{\pm 1.5}$	$93.0_{\pm 1.0}$	$\mathbf{98.0}_{\pm 1.6}$	$96.3_{\pm 4.0}$	$98.5_{\pm 1.6}$	$97.9_{\pm 0.1}$	$98.2_{\pm 0.5}$	$96.2_{\pm 0.9}$	$\mathbf{98.6}_{\pm 0.2}$
ARC-C	$93.1_{\pm 0.6}$	$95.1_{\pm 0.5}$	$91.2_{\pm 0.6}$	$94.5_{\pm 0.4}$	$\mathbf{96.1}_{\pm 0.2}$	$93.5_{\pm 0.9}$	$91.6_{\pm 1.8}$	$89.4_{\pm 1.0}$	$\mathbf{94.0}_{\pm 1.2}$	$92.9_{\pm 1.7}$	$95.5_{\pm 1.3}$	$95.6_{\pm 2.0}$	$94.0_{\pm 0.5}$	$96.4_{\pm 2.9}$	$\mathbf{97.0}_{\pm 2.4}$
GSM8K	$92.4_{\pm 0.4}$	$92.7_{\pm 1.7}$	$93.1_{\pm 1.3}$	$94.5_{\pm 0.5}$	$\mathbf{95.2}_{\pm 1.7}$	$91.8_{\pm 1.1}$	$89.2_{\pm 1.5}$	$91.7_{\pm 0.9}$	$93.0_{\pm 1.1}$	$\mathbf{93.1}_{\pm 3.5}$	$92.6_{\pm 0.9}$	$94.0_{\pm 2.5}$	$95.4_{\pm 0.9}$	$\mathbf{96.3}_{\pm 1.6}$	$94.0_{\pm 2.0}$
MedQA	$76.7_{\pm 1.5}$	$77.0_{\pm 1.7}$	$74.7_{\pm 0.8}$	$77.4_{\pm 1.2}$	$\mathbf{79.9}_{\pm 2.0}$	$73.3_{\pm 2.4}$	$\mathbf{76.6}_{\pm 3.5}$	$74.6_{\pm 1.4}$	$73.2_{\pm 3.3}$	$74.9_{\pm 4.1}$	$78.5_{\pm 1.8}$	$78.7_{\pm 4.8}$	$78.4_{\pm 2.1}$	$\mathbf{81.0}_{\pm 3.4}$	$\mathbf{81.0}_{\pm 2.8}$
MBPP+	$67.0_{\pm 0.2}$	$68.9_{\pm 0.9}$	$73.4_{\pm 0.8}$	$73.8_{\pm 0.9}$	$\mathbf{75.8}_{\pm 2.0}$	$65.1_{\pm 2.6}$	$68.2_{\pm 5.4}$	$71.7_{\pm 3.0}$	$74.5_{\pm 1.8}$	$\mathbf{74.6}_{\pm 2.4}$	$71.6_{\pm 1.3}$	$74.4_{\pm 2.5}$	$75.8_{\pm 1.8}$	$74.7_{\pm 2.0}$	$\mathbf{77.1}_{\pm 3.4}$
HumanEval+	$79.5_{\pm 1.8}$	$\mathbf{85.6}_{\pm 1.8}$	$82.1_{\pm 0.7}$	$82.1_{\pm 2.5}$	$81.7_{\pm 1.1}$	$77.4_{\pm 1.2}$	$\mathbf{81.1}_{\pm 2.8}$	$79.7_{\pm 0.7}$	$79.1_{\pm 1.8}$	$80.5_{\pm 2.7}$	$79.5_{\pm 0.9}$	$80.9_{\pm 3.4}$	$84.6_{\pm 0.9}$	$85.0_{\pm 1.8}$	$\mathbf{88.2}_{\pm 3.6}$
AIME24	$53.3_{\pm 3.3}$	$54.4_{\pm 3.8}$	$55.6_{\pm 1.9}$	$\mathbf{57.8}_{\pm 1.9}$	$\mathbf{57.8}_{\pm 7.7}$	$53.3_{\pm 3.3}$	$52.2_{\pm 1.9}$	$\mathbf{56.7}_{\pm 3.3}$	$54.4_{\pm 1.9}$	$\mathbf{56.7}_{\pm 3.3}$	$64.4_{\pm 3.8}$	$63.3_{\pm 3.3}$	$63.3_{\pm 3.3}$	$67.8_{\pm 1.9}$	$\mathbf{70.0}_{\pm 3.3}$
AIME25	$53.3_{\pm 3.3}$	$55.6_{\pm 1.9}$	$53.3_{\pm 3.3}$	$55.6_{\pm 3.8}$	$\mathbf{58.9}_{\pm 1.9}$	$48.9_{\pm 1.9}$	$52.2_{\pm 5.1}$	$51.1_{\pm 1.9}$	$51.1_{\pm 5.1}$	$\mathbf{57.8}_{\pm 1.9}$	$57.8_{\pm 1.9}$	$58.9_{\pm 5.1}$	$62.2_{\pm 1.9}$	$64.4_{\pm 1.9}$	$\mathbf{68.9}_{\pm 1.9}$
GPQA	$42.6_{\pm 2.0}$	$42.9_{\pm 2.3}$	$45.6_{\pm 2.5}$	$46.8_{\pm 1.5}$	$\mathbf{48.0}_{\pm 2.2}$	$41.8_{\pm 1.6}$	$41.8_{\pm 4.2}$	$41.1_{\pm 2.0}$	$43.8_{\pm 3.2}$	$\mathbf{46.0}_{\pm 4.4}$	$49.3_{\pm 1.3}$	$54.9_{\pm 5.1}$	$53.9_{\pm 1.1}$	$53.0_{\pm 2.7}$	$\mathbf{58.4}_{\pm 3.6}$
Hierarchical MAS Setting															
ARC-E	$96.3_{\pm 0.1}$	$99.3_{\pm 0.5}$	$97.8_{\pm 0.3}$	$98.7_{\pm 0.2}$	$\mathbf{99.6}_{\pm 0.1}$	$94.9_{\pm 1.2}$	$95.3_{\pm 4.1}$	$\mathbf{97.2}_{\pm 0.6}$	$96.7_{\pm 1.7}$	$96.7_{\pm 1.5}$	$97.6_{\pm 1.9}$	$97.3_{\pm 0.7}$	$96.7_{\pm 1.1}$	$\mathbf{98.2}_{\pm 0.3}$	$97.6_{\pm 0.6}$
ARC-C	$93.0_{\pm 0.9}$	$93.5_{\pm 1.7}$	$\mathbf{95.9}_{\pm 0.4}$	$93.1_{\pm 1.1}$	$94.4_{\pm 1.1}$	$90.6_{\pm 1.5}$	$93.3_{\pm 2.6}$	$\mathbf{93.8}_{\pm 0.3}$	$91.5_{\pm 1.9}$	$93.5_{\pm 2.5}$	$94.5_{\pm 1.3}$	$94.3_{\pm 2.7}$	$95.3_{\pm 0.3}$	$95.5_{\pm 1.1}$	$\mathbf{97.8}_{\pm 0.2}$
GSM8K	$90.6_{\pm 0.6}$	$91.3_{\pm 1.1}$	$89.5_{\pm 1.1}$	$91.5_{\pm 0.6}$	$\mathbf{93.5}_{\pm 0.5}$	$88.5_{\pm 1.6}$	$88.5_{\pm 3.5}$	$87.6_{\pm 0.4}$	$89.7_{\pm 1.0}$	$\mathbf{90.8}_{\pm 3.1}$	$88.5_{\pm 1.3}$	$89.9_{\pm 2.8}$	$92.7_{\pm 0.2}$	$90.9_{\pm 2.0}$	$\mathbf{94.4}_{\pm 2.1}$
MedQA	$78.1_{\pm 1.0}$	$\mathbf{81.1}_{\pm 0.7}$	$76.9_{\pm 0.2}$	$76.8_{\pm 1.0}$	$79.5_{\pm 1.5}$	$73.8_{\pm 2.1}$	$76.9_{\pm 2.8}$	$75.7_{\pm 1.6}$	$74.4_{\pm 2.2}$	$\mathbf{77.4}_{\pm 2.4}$	$78.7_{\pm 1.2}$	$79.5_{\pm 3.7}$	$78.3_{\pm 1.4}$	$77.9_{\pm 2.6}$	$\mathbf{80.5}_{\pm 2.6}$
MBPP+	$73.1_{\pm 0.9}$	$73.6_{\pm 1.9}$	$72.7_{\pm 0.4}$	$72.2_{\pm 1.4}$	$\mathbf{74.7}_{\pm 1.4}$	$70.1_{\pm 3.1}$	$\mathbf{73.6}_{\pm 4.6}$	$68.3_{\pm 1.5}$	$71.7_{\pm 1.6}$	$72.6_{\pm 4.8}$	$74.0_{\pm 1.5}$	$71.8_{\pm 3.4}$	$73.8_{\pm 1.4}$	$\mathbf{77.5}_{\pm 2.1}$	$75.4_{\pm 3.0}$
HumanEval+	$80.1_{\pm 1.8}$	$76.2_{\pm 1.2}$	$77.8_{\pm 0.9}$	$76.2_{\pm 2.1}$	$\mathbf{81.1}_{\pm 3.2}$	$78.3_{\pm 1.5}$	$75.0_{\pm 3.2}$	$76.4_{\pm 0.7}$	$74.8_{\pm 1.4}$	$\mathbf{78.7}_{\pm 3.0}$	$81.9_{\pm 1.4}$	$83.3_{\pm 1.9}$	$84.8_{\pm 1.1}$	$82.9_{\pm 1.1}$	$\mathbf{90.2}_{\pm 2.8}$
AIME24	$53.3_{\pm 3.3}$	$54.4_{\pm 6.9}$	$52.2_{\pm 1.9}$	$53.3_{\pm 5.8}$	$\mathbf{56.7}_{\pm 3.3}$	$53.3_{\pm 3.3}$	$51.1_{\pm 1.9}$	$50.0_{\pm 3.3}$	$\mathbf{54.4}_{\pm 5.1}$	$\mathbf{54.4}_{\pm 3.8}$	$68.9_{\pm 3.8}$	$68.9_{\pm 5.1}$	$71.1_{\pm 1.9}$	$74.4_{\pm 1.9}$	$\mathbf{75.6}_{\pm 1.9}$
AIME25	$53.3_{\pm 3.3}$	$52.2_{\pm 6.9}$	$52.2_{\pm 1.9}$	$51.1_{\pm 3.8}$	$\mathbf{56.7}_{\pm 6.7}$	$51.1_{\pm 1.9}$	$47.8_{\pm 1.9}$	$48.9_{\pm 3.8}$	$50.0_{\pm 5.8}$	$\mathbf{53.3}_{\pm 5.8}$	$63.3_{\pm 3.3}$	$64.4_{\pm 3.8}$	$66.7_{\pm 3.3}$	$64.4_{\pm 1.9}$	$\mathbf{70.0}_{\pm 3.3}$
GPQA	$42.1_{\pm 1.9}$	$42.6_{\pm 2.0}$	$46.1_{\pm 1.8}$	$50.8_{\pm 2.5}$	$\mathbf{53.7}_{\pm 2.5}$	$41.4_{\pm 2.3}$	$42.6_{\pm 1.8}$	$44.9_{\pm 1.5}$	$\mathbf{49.8}_{\pm 1.5}$	$46.8_{\pm 3.0}$	$49.3_{\pm 3.4}$	$50.7_{\pm 2.5}$	$51.7_{\pm 2.9}$	$\mathbf{56.6}_{\pm 2.2}$	$51.3_{\pm 3.1}$

89.1%. A similar trend occurs on the HumanEval+ task using the 14B model, where performance drops from 89.4% to 85.4%.

I Cross-Architecture Generalizability

To evaluate whether the proposed Additive Mechanism is independent of specific pre-training biases, the study extends beyond the Qwen3 backbone to evaluate three distinct architectures: Llama 3.1-8B [19], Ministral 3-8B [36], and Ministral 3-14B [36]. Following the primary experimental protocol, these models are assessed across nine benchmarks in both sequential and hierarchical multi-agent settings (Table 7).

Implementation Details. All generation processes using Llama 3.1-8B, Ministral 3-8B, and Ministral 3-14B employ an identical autoregressive decoding strategy with a temperature of 0.6 and a top-p value of 0.95. All hyperparameters governing the Additive Mechanism remain strictly identical to those established in Appendix C.

Consistent Effectiveness Across Distinct Architectures. Results in Table 7 demonstrate that the additive mechanism maintains robust effectiveness across diverse LLM families. For example, on the AIME24 benchmark, the proposed method increases the performance of Llama 3.1-8B to 56.7%, surpassing both the text baseline (53.3%) and the latent baseline (52.2%). Similar trends occur within the Ministral 3 family; on the ARC-C hierarchical task, the Ministral 3-14B model achieves 97.8% with the additive mechanism, outperforming both text (94.5%) and latent (95.3%) communication methods. This consistent performance delta indicates that the Attention Distraction Dilemma is an architectural artifact inherent to attention-based interactions, and that covariance-based subspace projection functions as a generalizable geometric remedy.

Resilience and Stability in Reasoning-Intensive Scenarios. Empirical data reveal a widening performance gap in tasks demanding logical coherence, such as AIME25, GPQA, and HumanEval+. On the Llama 3.1-8B model under the sequential setting, standard dense latent interaction achieves 53.3% on AIME25, trailing the 58.9% of the proposed mechanism. Furthermore, the Ministral 3-8B model exhibits an improvement on GPQA, rising from 41.1% via standard latent transmission to 46.0% with the additive filter. While purification strategies occasionally yield competitive peak accuracies—such as the text purification baseline achieving 85.6% on HumanEval+ with Llama 3.1-8B—they simultaneously exhibit higher variance (± 1.8 compared to ± 0.7 for standard latent transmission). This high standard deviation indicates that aggressive discrete pruning or external text summarization introduces systemic instability across different random seeds. Conversely, the additive filter establishes an optimal trade-off, neutralizing background drift via destructive interference while avoiding the high-variance risks associated with discrete semantic truncation.

J Hyperparameter Sensitivity Study

J.1 Sensitivity Analysis Of Safety Threshold And Energy Tolerance

We investigate the impact of the Safety Threshold (β) and Energy Tolerance (γ) on generation quality and reasoning capabilities. To evaluate these effects, we conduct grid search experiments across diverse benchmarks by varying β from 0.1 to 0.9 and γ from 0.05 to 0.9. Tracking the accuracy under each hyperparameter combination allows us to analyze the sensitivity of the model to these constraints.

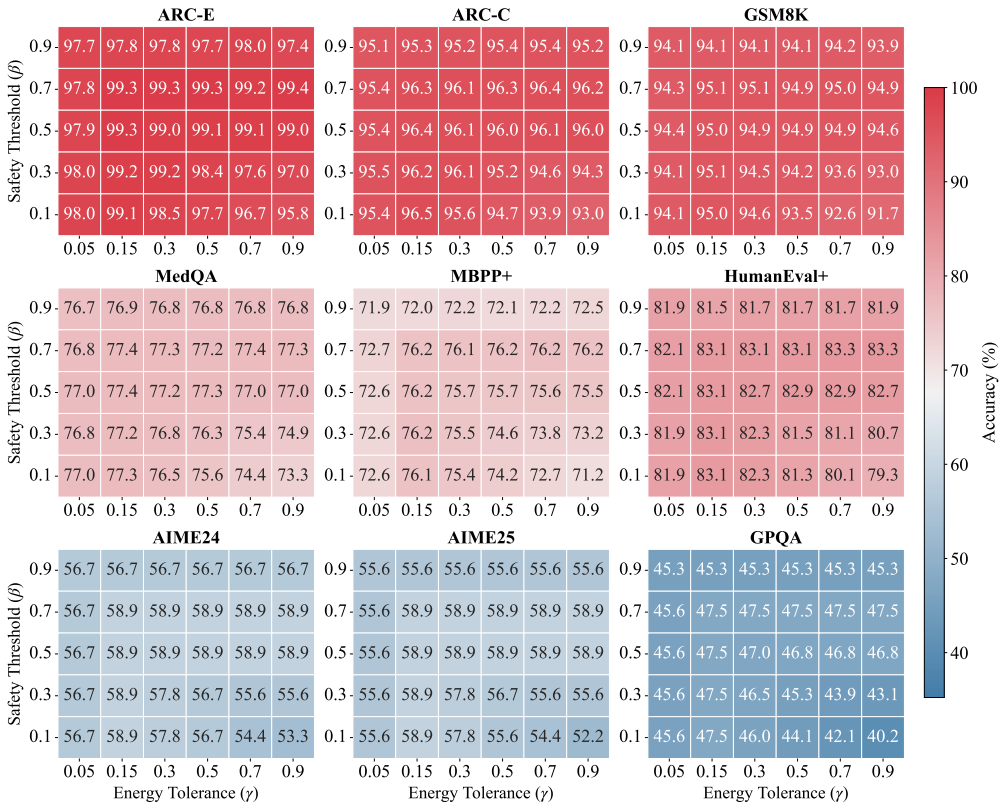
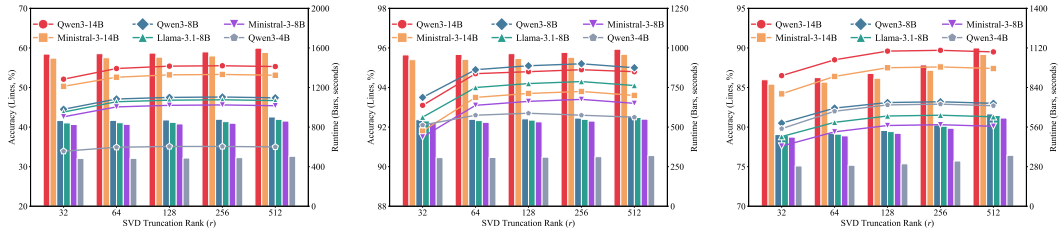


Figure 10: Sensitivity analysis of Safety Threshold (β) and Energy Tolerance (γ) across various reasoning and coding benchmarks with Qwen3-8B.



(a) GPQA (b) GSM8K (c) HumanEval+

Figure 11: Impact of SVD Truncation Rank (r) on model accuracy (lines) and computational runtime (bars) across GPQA, GSM8K, and HumanEval+ benchmarks.

As shown in Fig. 10, the proposed method exhibits a broad **stability plateau** rather than a narrow sweet spot. Specifically, performance remains highly consistent across all tasks for $\beta \in [0.5, 0.9]$ and $\gamma \in [0.05, 0.3]$. For instance, the optimal configuration ($\beta = 0.7, \gamma = 0.15$) yields robust results across diverse domains, spanning from algorithmic coding (MBPP+, **76.2%**) to competitive mathematics (AIME25, **58.9%**). This task-invariant property confirms that β and γ represent the intrinsic semantic safety and energy properties of the latent space of the model, rather than task-specific empirical parameters. Furthermore, the performance degradation observed at extreme boundary conditions (e.g., $\beta = 0.1, \gamma = 0.9$, where AIME25 drops to **52.2%** and GPQA to **40.2%**) serves as a **stress test** to delineate the operational boundaries of the method. This relative degradation may be associated with a more restrictive local gain constraint ($\beta = 0.1$), limiting the contribution of useful upstream information to the reasoning chain. The transition from the stability plateau to the boundary zone is gradual in the evaluated settings, suggesting that an exhaustive grid search may not be necessary. A single task-agnostic default setting (e.g., $\beta = 0.7, \gamma = 0.15$) can serve as a starting point for evaluation on unseen tasks.

J.2 Effect Of SVD Truncation Rank On Performance And Efficiency

We investigate the influence of the SVD Truncation Rank (r) on the balance between model performance and computational efficiency. We evaluate six models across three benchmarks, varying r among discrete values of

32, 64, 128, 256, and 512 (similar phenomena exist in other datasets). By jointly monitoring accuracy (line plots, primary y-axis) and execution runtime (bar charts, secondary y-axis), we aim to identify the optimal rank that maximizes reasoning capability while minimizing computational overhead.

Accuracy Saturation And Optimal Sweet Spot Results reveal a non-linear relationship between r and model accuracy across all tasks. Increasing r from 32 to 128 yields substantial performance improvements. On the HumanEval benchmark, the accuracy of Qwen3-14B climbs from approximately 86.5% at $r = 32$ to a near-peak of 89.5% at $r = 128$. Models exhibit a similar ascent on GSM8K and GPQA, rapidly recovering performance as r increases to 64 and 128. However, between $r = 128$ and $r = 256$, accuracy trajectories saturate and form a plateau. Concurrently, runtime monotonically increases with rank. This indicates that intermediate ranks ($r = 128$ and $r = 256$) provide an optimal configuration. At these values, the model captures the critical singular values necessary for high-fidelity generation without incurring the severe latency penalties of higher-rank computations.

Diminishing Returns And High Rank Degradation Contrary to the intuition that retaining more singular values strictly benefits performance, experiments demonstrate diminishing returns and performance degradation at $r = 512$. Although $r = 512$ incurs the highest computational cost, the corresponding accuracy fails to improve and frequently declines compared to $r = 256$. This degradation is observable on the GSM8K dataset, where models such as Qwen3-14B and Qwen3-8B show an accuracy drop when transitioning from $r = 256$ to $r = 512$. A similar downward trend occurs in HumanEval at $r = 512$. This phenomenon suggests that exceptionally high truncation ranks incorporate noisy or less informative singular values. These redundant values drastically inflate runtime and interfere with reasoning precision, leading to suboptimal performance.

K Case Study

Following previous work [74], we conduct a detailed case study on the GSM8K benchmark using the Qwen3-14B backbone under the sequential multi-agent setting to thoroughly understand the dynamics of different communication paradigms.

Error Compounding in Textual Communication The explicit text-based pipeline, represented by the TextMAS baseline [74], relies on lengthy natural language exchanges that inherently amplify early reasoning errors. In the provided example, the planner and critic agents misinterpret the chronological relationship of the leaving bees. This linguistic ambiguity propagates through the refiner agent, which entirely discards the sixty bees from the third period. Consequently, the solver agent inherits this compounded textual error and produces a severely constrained search space, ultimately yielding an incorrect final answer of fifteen. This phenomenon highlights the fundamental fragility of using discrete tokens as the sole medium for complex multi-agent coordination.

Noise Suppression and Mathematical Precision While the standard Latent MAS baseline [74] bypasses the textual bottleneck to successfully predict the correct answer of seventy-five, its decoded reasoning path remains highly verbose and rigidly tied to the upstream textual plan. In contrast, our proposed Additive Mechanism elevates the reasoning quality by systematically filtering out this structural verbosity. As shown in the generated output, our method digests the multi-agent latent context and distills it into a highly concentrated, mathematically rigorous logic chain. The solver agent directly formulates algebraic variables for intermediate states, sequentially calculating the net number of bees outside the hive without redundant linguistic descriptions. This qualitative shift demonstrates that the additive update mechanism successfully induces destructive interference against contextual noise while constructively amplifying the core semantic signal, leading to a significantly more stable and efficient problem-solving trajectory.

Case Study (TextMAS on GSM8K)

Input Question: Debra is monitoring a beehive to see how many bees come and go in a day. She sees 30 bees leave the hive in the first 6 hours, and then she sees 1/2 that many bees return in the next 6 hours. She sees two times as many bees as she saw first leave the hive fly from the hive and leave in the next 6 hours. Then every bee that left before that hadn't already returned returns to the hive in the next 6 hours. How many bees did Debra see return to the hive in the last 6 hours of the day?

TextMAS Output

Planner Agent:
Okay, let's see. I need to figure out how many bees Debra saw return to the hive in the last 6 hours. Let me break down the problem step by step. First, the problem mentions different time intervals: first 6 hours, next 6 hours, then the next 6 hours, and finally the last 6 hours. So there are four 6-hour periods in a day, making up 24 hours total. Let me note each period and what happens in each. Starting with the first 6 hours: Debra sees 30 bees leave the hive. Let me write that down as the first data point. So, Period 1 (0-6 hours): 30 bees leave. Then, the next 6 hours (Period 2: 6-12 hours), she sees half that number return. Half

of 30 is 15. So, 15 bees return in Period 2. Next, the third period (Period 3: 12-18 hours), she sees two times as many bees as she saw first leave the hive fly from the hive and leave. The first time she saw 30 bees leave, so two times that is 60. So, 60 bees leave in Period 3. Then, the fourth period (Period 4: 18-24 hours), every bee that left before that hadn't already returned returns to the hive.

Critic Agent:

The reasoning plan correctly identifies all four 6-hour periods and reaches the right final conclusion, but the intermediate analysis contains several issues that reduce clarity and efficiency. First, the interpretation of the phrase "before that" is revisited multiple times, even though it simply refers to the event in the previous sentence (the 60 bees leaving), meaning only bees that left in Period 1 are relevant. Second, the plan temporarily entertains the possibility that bees may have left in Period 2, despite the problem statement explicitly describing only returns during this interval. Third, the plan repeatedly questions whether the 15 returning bees in Period 2 originate from the initial 30, even though the text directly states they are "half that many" and therefore unambiguously tied to Period 1. These detours introduce redundancy and confusion. Overall, while the conclusion of 15 returning bees is correct, the path to it is unnecessarily convoluted and would benefit from a more concise timeline-based analysis.

Refiner Agent:

I will provide a refined plan based on previous agents' information.

1. Identify the Time Periods and Events. We divide the day into four 6-hour intervals and record the events in each:

- Period 1 (0–6 hours): 30 bees leave the hive.
- Period 2 (6–12 hours): Half of those bees return, i.e., 15 bees.
- Period 3 (12–18 hours): Twice the number of bees that left in Period 1 leave the hive: $2 \times 30 = 60$ bees.
- Period 4 (18–24 hours): Every bee that left *before Period 3* and has not yet returned comes back to the hive.

2. Track Bees Leaving and Returning.

- Period 1: 30 bees leave.
- Period 2: 15 of these 30 bees return, leaving 15 still outside the hive.
- Period 3: 60 new bees leave (these are not relevant for the final return count).
- Period 4: Only bees that left *before Period 3* and have not returned will come back.

3. Determine Which Bees Return in Period 4.

The only bees that left before Period 3 are the 30 from Period 1. Of these, 15 returned in Period 2. Thus, the remaining 15 bees from Period 1 return in Period 4.

4. Return the Final Answer.

Solver Agent:

Okay, let's see. I need to figure out how many bees Debra saw return to the hive in the last 6 hours. Let me break down the problem step by step.

First, the problem mentions different time intervals: first 6 hours, next 6 hours, then the next 6 hours, and finally the last 6 hours. So there are four 6-hour periods in a day, making up 24 hours total. Let me note each period and what happens in each. Starting with the first 6 hours: Debra sees 30 bees leave the hive. Let me write that down as the first data point. So, Period 1 (0-6 hours): 30 bees leave. Then, the next 6 hours (Period 2: 6-12 hours), she sees half that number return. Half of 30 is 15. So, 15 bees return in Period 2. Next, the third period (Period 3: 12-18 hours), she sees two times as many bees as she saw first leave the hive fly from the hive and leave. The first time she saw 30 bees leave, so two times that is 60. So, 60 bees leave in Period 3. Then, the fourth period (Period 4: 18-24 hours), every bee that left before that hadn't already returned returns to the hive. So, bees that left in Period 1 and Period 2. But some of those bees may have already returned in Period 2.

Final Answer

Debra saw $\boxed{15}$ bees return to the hive in the last 6 hours of the day. (*Wrong Final Answer*)

Case Study (Latent MAS on GSM8K)

Input Question: Debra is monitoring a beehive to see how many bees come and go in a day. She sees 30 bees leave the hive in the first 6 hours, and then she sees 1/2 that many bees return in the next 6 hours. She sees two times as many bees as she saw first leave the hive fly from the hive and leave in the next 6 hours. Then every bee that left before that hadn't already returned returns to the hive in the next 6 hours. How many bees did Debra see return to the hive in the last 6 hours of the day?

Latent MAS Output

We are given a sequence of events involving bees leaving and returning to the hive over the course of a day, broken into 6-hour intervals.

According to the refiner's plan, we need to track how many bees leave and return during each 6-hour interval, noting that 30 leave first, 15 return next, and 60 more leave afterward. In the final interval, all bees

that had left and not yet returned came back, giving a total of 15 remaining from the first group, plus 60 from the third group, for a total of 75 returning bees.
Let's break it down step by step based on the plan:
Step 1: Bees that leave in the first 6 hours.
Debra sees 30 bees leave the hive during the first 6-hour period.
Step 2: Bees that return in the next 6 hours.
In the second 6-hour period, she sees half as many bees return: $\frac{1}{2} \times 30 = 15$ bees return.
Step 3: Bees that leave in the third 6-hour period.
In the next 6 hours, twice the number from Step 1 leave: $2 \times 30 = 60$ bees leave.
Step 4: Bees that return in the final 6 hours.
During the final 6 hours, every bee that had left earlier and had not yet returned comes back to the hive.
We track the bees that left and determine which have already returned:
- From the first 6 hours: 30 bees left; 15 returned in Step 2, leaving 15 bees still out.
- From the third 6 hours: 60 bees left; none have returned yet.

Thus, the total number of bees returning in the final 6 hours is $15 + 60 = 75$ bees.
###Final Answer:
$\boxed{75}$ (*Correct Final Answer*)

Case Study (Ours on GSM8K)

Input Question: Debra is monitoring a beehive to see how many bees come and go in a day. She sees 30 bees leave the hive in the first 6 hours, and then she sees 1/2 that many bees return in the next 6 hours. She sees two times as many bees as she saw first leave the hive fly from the hive and leave in the next 6 hours. Then every bee that left before that hadn't already returned returns to the hive in the next 6 hours. How many bees did Debra see return to the hive in the last 6 hours of the day?

Ours Output
To determine how many bees Debra saw return to the hive in the last 6 hours, we calculate the net number of bees outside the hive throughout the day.
1. During the first 6 hours, 30 bees leave, making the current number of bees outside $N_1 = 30$.
2. In the subsequent 6 hours, half of this initial number returns, meaning $R_2 = \frac{1}{2} \times 30 = 15$ bees come back. This reduces the number of bees outside to $N_2 = 30 - 15 = 15$.
3. Over the third 6-hour interval, twice the initial number of bees leave the hive, so $L_3 = 2 \times 30 = 60$ bees depart.
The total number of bees outside becomes $N_3 = 15 + 60 = 75$. Debra sees 30 bees leave the hive.
###Final Answer:
Debra saw $\boxed{75}$ bees return to the hive in the last 6 hours of the day. (*Correct Final Answer*)

L Prompt Templates

To ensure a fair and consistent evaluation, we adopt the multi-agent prompt templates established in prior work [74] for all our baseline and proposed methods. The detailed templates for the Sequential and Hierarchical settings across different tasks, alongside the semantic summarization prompt used for intrinsic noise estimation, are provided below.

Prompt for Intrinsic Noise Estimation

System Prompt:
You are an expert AI editor and summarizer, specialized in distilling complex reasoning processes into concise logical steps.

Prompt for Noise Reduction:
You are given an original question and a raw reasoning response generated by an upstream AI agent. Your task is to compress this response by removing all redundant, conversational, and task-irrelevant background noise.
Original Question: {question}
Raw Upstream Response: {raw_response}
Please rewrite the raw response into a highly compressed, clean summary. Requirements: 1. Retain all core logical steps, mathematical deductions, and essential semantic information necessary to solve the problem. 2. Strictly eliminate verbose explanations, repetitive filler, and meta-commentary (e.g., "Let's think step by step", "Sure, I can help with that", "In conclusion"). 3. Do not alter the original reasoning trajectory or solve the problem independently; only compress the provided text.
Now, output your clean summary below:

Sequential LatentMAS Prompts on Numeric Tasks (GSM8K / AIME2024 / AIME2025)

System Prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Planner Agent:
You are a Planner Agent. Given an input question, design a clear, step-by-step plan for how to solve the question. Question: {question}
Your outlined plan should be concise with a few bulletpoints for each step. Do not produce the final answer. Now output your plan to solve the question below:

Prompt for Critic Agent:
You are a Critic Agent to evaluate the correctness of the input plan for the given question and provide helpful feedback for improving the plan. Question: {question}
The plan information is provided in latent KV representation format. Review the plan and question and output: (1) original plan contents (2) constructive feedback on the original plan. Format your response as follows: Original Plan: [Copy the provided Planner Agent's plan here] Feedback: [Your detailed feedback to improve the plan here] Now, output your response below:

Prompt for Refiner Agent:
You are a Refiner Agent to provide a refined step-by-step plan for solving the given question. Question: {question}
You are provided with: (1) latent-format information: a previous plan with feedback (2) text-format information: the input question you need to solve. Based on the input, write a refined and improved plan to solve the question. Make sure your output plan is correct and concise. Now, output your refined plan below:

Prompt for Judger Agent:
You are a helpful assistant. You are provided with latent information for reference and a target question to solve. Target Question: {question}
The latent information might contain irrelevant contents. Ignore it if it is not helpful for solving the target question. Now, reason step by step and output the final answer inside \boxed{YOUR_FINAL_ANSWER}:

Sequential LatentMAS prompts for multiple-choice tasks (ARC-E, ARC-C, GPQA, MedQA)

System Prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Planner Agent:
You are a Planner Agent. Given an input question, design a clear, step-by-step plan for how to solve the question. Question: {question}
Your outlined plan should be concise with a few bulletpoints for each step. Do not produce the final answer. Now output your plan to solve the question below:

Prompt for Critic Agent:
You are a Critic Agent to evaluate the correctness of the input plan for the given question and provide helpful feedback for improving the plan. Question: {question}
The plan information is provided in latent KV representation format. Review the plan and question and output: (1) original plan contents (2) constructive feedback on the original plan. Format your response as follows: Original Plan: [Copy the provided Planner Agent's plan here] Feedback: [Your detailed feedback to improve the plan here] Now, output your response below:

Prompt for Refiner Agent:
You are a Refiner Agent to provide a refined step-by-step plan for solving the given question. Question: {question}
You are provided with: (1) latent-format information: a previous plan with feedback (2) text-format information: the input question you need to solve. Based on the input, write a refined and improved plan to solve the question. Make sure your output plan is correct and concise. Now, output your refined plan below:

Prompt for Judger Agent:
You are a helpful assistant. You are provided with latent information for reference and a target question to solve. Target Question: {question}
The latent information might contain irrelevant contents. Ignore it if it is not helpful for solving the target question. Your final answer must be selected from A,B,C,D. For example \boxed{A}. Do not add any other contents inside the box. Now, reason step by step and output the final answer inside \boxed{YOUR_FINAL_ANSWER}:

Sequential LatentMAS prompts for python coding tasks (MBPP+, HumanEval+)

System Prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Planner Agent:
You are a Planner Agent. Given an input question, design a clear, step-by-step plan for how to solve the question. Question: {question}
Your outlined plan should be concise with a few bulletpoints for each step. Do not produce the final answer. Now output your plan to solve the question below:

Prompt for Critic Agent:
You are a Critic Agent to evaluate the correctness of the input plan for the given question and provide helpful feedback for improving the plan. Question: {question}
The plan information is provided in latent KV representation format. Review the plan and question and output: (1) original plan contents (2) constructive feedback on the original plan. Format your response as follows: Original Plan: [Copy the provided Planner Agent's plan here] Feedback: [Your detailed feedback to improve the plan here] Now, output your response below:

Prompt for Refiner Agent:
You are a Refiner Agent to provide a refined step-by-step plan for solving the given question. Question: {question}
You are provided with: (1) latent-format information: a previous plan with feedback (2) text-format information: the input question you need to solve. Based on the input, write a refined and improved plan to solve the question. Make sure your output plan is correct and concise. Now, output your refined plan below:

Prompt for Judger Agent:
You are a helpful assistant. You are provided with latent information for reference and a target question to solve. Target Question: {question}
The latent information might contain irrelevant contents. Ignore it if it is not helpful for solving the target question. Your must reason step-by-step to solve the **provided Target Question** without outputing other irrelevant information. You must put all python code as self-contained Python function in markdown code blocks. For example
‘‘‘python
import math
def add(a, b):
return a + b‘‘‘.
Do not add any other contents inside the markdown code block. Now, reason step by step and output the final answer inside ‘‘‘python YOUR_PYTHON_CODE‘‘‘:

Hierarchical LatentMAS prompts for numeric-answer tasks (GSM8K, AIME2024, AIME2025)

System prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Math Agent:
You are a math agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Question: {question}
Your response:

Prompt for Science Agent:
You are a science agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Question: {question}
Your response:

Prompt for Code Agent:
You are a code agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Question: {question}
Your response:

Prompt for Task Summarizer Agent:
You are a task summarizer. Given the input question and responses from previous agents as reference, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}.
Question: {question}
Your response:

Hierarchical LatentMAS prompts for multiple-choice tasks (ARC-E, ARC-C, GPQA, MedQA)

System Prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Math Agent:
You are a math agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Your final answer must be selected from A,B,C,D. For example \boxed{A}. Do not add any other contents inside the box. Question: {question}

Your response:

Prompt for Science Agent:
You are a science agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Your final answer must be selected from A,B,C,D. For example \boxed{A}. Do not add any other contents inside the box. Question: {question}
Your response:

Prompt for Code Agent:
You are a code agent. Given the input question, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Your final answer must be selected from A,B,C,D. For example \boxed{A}. Do not add any other contents inside the box. Question: {question}
Your response:

Prompt for Task Summarizer Agent:
You are a task summarizer. Given the input question and responses from previous agents as reference, reason step-by-step and put the final answer inside \boxed{YOUR_FINAL_ANSWER}. Your final answer must be selected from A,B,C,D. For example \boxed{A}. Do not add any other contents inside the box. Question: {question}
Your response:

Hierarchical LatentMAS prompts for python coding tasks (MBPP+, HumanEval+)

System Prompt for All Agents:
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Prompt for Math Agent:
You are a math agent. Given the input question, reason step by step and provide an efficient and self-contained Python function that solves the following problem in a markdown code block. You must put all python code as self-contained Python function in markdown code blocks. For example
‘‘‘python
import math
def add(a, b):
return a + b‘‘‘. Do not add any other contents inside the markdown code block.
Question: {question}
Your response:

Prompt for Science Agent:
You are a science agent. Given the input question, reason step by step and provide an efficient and self-contained Python function that solves the following problem in a markdown code block.
You must put all python code as self-contained Python function in markdown code blocks. For example
‘‘‘python
import math
def add(a, b):
return a + b‘‘‘. Do not add any other contents inside the markdown code block.
Question: {question}
Your response:

Prompt for Code Agent:
You are a code agent. Given the input question, reason step by step and provide an efficient and self-contained Python function that solves the following problem in a markdown code block.
You must put all python code as self-contained Python function in markdown code blocks. For example
‘‘‘python
import math
def add(a, b):
return a + b‘‘‘. Do not add any other contents inside the markdown code block.
Question: {question}
Your response:

Prompt for Task Summarizer Agent:
You are a task summarizer. Given the input question and responses from previous agents as reference, reason step by step and provide an efficient and self-contained Python function that solves the following problem in a markdown code block.
You must put all python code as self-contained Python function in markdown code blocks. For example
‘‘‘python
import needed_library
def FUNC_NAME(a, b):
return a + b‘‘‘. Do not add any other contents inside the markdown code block.
Input Question: {question}
Your response:

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we clearly express the contributions and scope of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a limitation part to discuss the limitations for the work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the complete proof and assumptions for our theoretical result in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide enough implementation details for reproducing the results of the proposed method in the main paper and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code is undergoing a mandatory internal review at our institution and will be released upon approval.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide adequate training and test details in the main paper (Section 4) and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We introduced the statistical variance of the experiment in the Table 2

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the computer resources in the Appendix C.3

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on fundamental MAS architecture and does not have direct, foreseeable negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release high-risk models requiring specific safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models and data used in this paper are open-sourced and authorized.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce or release any new datasets or assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research does not involve human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used for language polishing and text editing, and did not serve as a component of the core methods in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.