GRICEAN CACHE SHARING FOR MULTI-AGENT COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing multi-agent systems rely heavily on natural language for coordination. While flexible, text imposes a strict sequential bottleneck and often transmits redundant tokens, leading to high latency and noise interference. To bypass this limit without incurring the redundancy of naïve broadcasting, we introduce GRICEAN CACHE SHARING, a controllable thought sharing protocol grounded in Grice's Cooperative Principle. Instead of treating the thought as a monolithic block, our framework reformulates inter-agent communication as a constrained optimization problem. We design a transparent interface that exports only carefully selected thoughts, and give a mechanism analysis in which Top-k saliency minimizes discarded attention mass for fixed scores and a per-block budget. Empirically, our method reduces latency by $3.3\times$ on average (up to $5.0\times$) while matching or exceeding the accuracy of text baselines on 12 of 15 model–benchmark settings. Ablation studies show that the three modules are complementary, with the full combination outperforming every single- and dual-module variant, while the communication budget acts as a smooth accuracy–latency control knob.

1 INTRODUCTION

Large language models (LLMs) have become a strong substrate for building agentic systems that decompose complex tasks into coordinated subtasks. A common recipe is to instantiate multiple agents with specialized roles and let them collaborate through message passing. This paradigm has enabled practical multi-agent frameworks in coding, tool-use, and decision making, e.g., role-playing agents (Shanahan et al., 2023; Liu et al., 2024; Chen et al., 2024) and orchestration frameworks (Wu et al., 2024; Zhang et al., 2025b; Li et al., 2023; Hong et al., 2024). However, recent evidence (Wang et al., 2024; Cemri et al., 2025; Zhang et al., 2025a; Shekkizhar et al., 2025) suggests that performance gains over strong single-agent baselines can be fragile, and multi-agent systems exhibit diverse failure modes that often stem from system-level design and inter-agent coordination rather than isolated model weaknesses.

A central bottleneck is communication. Most existing systems (Baroni et al., 2022; Michel et al., 2023; Su et al., 2024; Li et al., 2024; Zhou et al., 2025; Xiong et al., 2026) rely on text-only mediation: each agent must externalize internal reasoning into tokens, which are then appended to the next agent's prompt. Text communication is slow (token-by-token generation), increasingly expensive with long traces, and often lossy or ambiguous—especially when intermediate steps contain hallucinations, redundant details, or role-irrelevant information.

Motivated by these limitations, several recent works (Pham et al., 2024; Zheng et al., 2025) explore communication beyond text, e.g., embedding-level protocols and direct latent/internal-state transfer. However, many of these approaches still follow the same full-trace handoff pattern: in collaboration, an upstream agent forwards its intermediate reasoning wholesale to the downstream agent, either as a long text trace or as raw internal states. However, naïvely "sharing everything" can leak role-specific deliberation, and crucially can transmit irrelevant or unreliable information that harms downstream agents. Yet, a key question remains:

How should agents communicate effectively?

This work argues that effective collaboration requires a *selective* and *role-aware* channel. In human teams, effective collaboration relies on implicit pragmatic norms—teams do not broadcast every

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

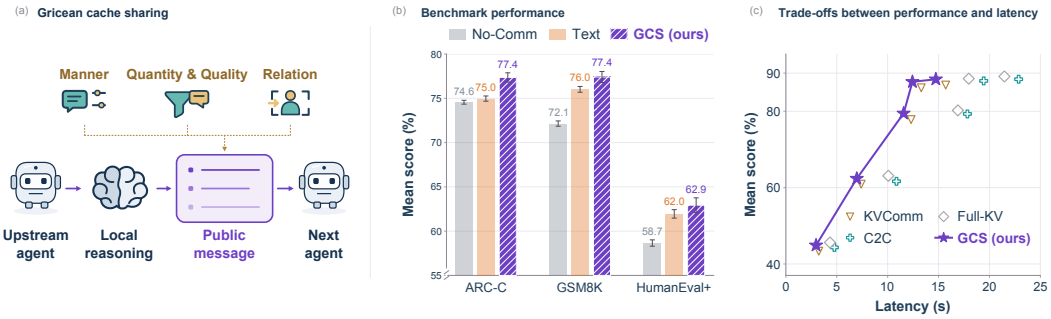


Figure 1: **Gricean Cache sharing.** (a) Agents construct a compact public message guided by Manner (controlled interface), Relation (recipient relevance), and Quantity/Quality (informative content within budget). (b) Benchmark scores for No-Communication, Text, and GCS, averaged over five backbones. (c) Scores and latencies averaged over three benchmarks for four methods across five backbones. Stars denote GCS; straight segments connect its backbone results as a visual guide.

synaptic firing; rather, they allocate limited bandwidth to role-relevant, high-signal information. The classic Grice's Cooperative Principle by Grice (1975) has summarized four maxims for effective collaboration: Manner, Relation, Quantity, and Quality. However, existing agent communication methods use plain text or direct latent state transfer. While text is a universal interface, it is lossy. Direct latent state transfer offers a higher channel capacity but suffers from noise interference.

We thus posit that multi-agent systems should emulate the principle of human cooperation, and propose our approach GRICEAN CACHE SHARING (GCS), using Grice's Principle as a high-level design lens for latent inter-agent communication. The method is previewed in Figure 1. It constructs a compact public message and applies lightweight Gricean filters to export only what is (i) shareable through a public interface (Manner), (ii) aligned to the receiver's role (Relation), and (iii) informative under a strict bandwidth budget via saliency-based signal preservation (Quantity/Quality). We further provide theoretical and empirical analyses of our method, demonstrating its desired properties and effective performance.

We summarize our contributions below:

- We introduce GRICEAN CACHE SHARING, a controllable cache communication protocol for sequential multi-agent collaboration that exports structured KV-cache slices as a public memory interface.
- We provide a theoretical framework for cross-agent latent communication under a fixed bandwidth budget. In particular, we bound attention-induced perturbations under explicit assumptions and prove that Top-k minimizes discarded attention mass for fixed scores and a per-block budget.
- We evaluate our method on multi-agent reasoning benchmarks (math and coding) across multiple model families, demonstrating improved accuracy–latency trade-offs over text communication. We further conduct ablation studies showing the contribution of each component and of the selection rules, and characterize the accuracy–latency trade-off across communication budgets.

2 RELATED WORK

Multi-Agent Systems and Orchestration. A growing body of work builds LLM-based multi-agent systems by instantiating multiple agents with role prompts and coordinating them through message passing and structured workflows, e.g., AutoGen (Wu et al., 2024), CAMEL (Li et al., 2023), MetaGPT (Hong et al., 2024), ChatDev (Qian et al., 2024). Recent studies (Ke et al., 2025; Zhang et al., 2025b; Dang et al., 2025) also explore orchestration policies and organizational design for multi-agent teams, as well as long-context decomposition via agentic collaboration (Zhang et al., 2024). These efforts are largely orthogonal to our focus on the communication channel itself and how internal information should be selectively transferred.

Collaborative Intelligence. Empirical analyses suggest that multi-agent discussion can yield only modest gains over strong single-agent baselines, with failures often attributable to coordination

inefficiencies and communication bottlenecks. Recent work Cemri et al. (2025) analyzes multi-agent failure modes, complementing orchestration strategies like cost-aware routing (Dekoninck et al., 2025; Yue et al., 2025b; Giusti et al., 2025). However, prevailing frameworks (Michel et al., 2023; Su et al., 2024; Li et al., 2024; Zhou et al., 2025; Xiong et al., 2026) rely heavily on NL interfaces, which—despite their flexibility—incur high token overhead and risk error propagation. Motivated by these limitations, several recent works investigate communication beyond text, including embedding-level protocols (Pham et al., 2024) and direct latent/internal-state transfer between language models. Recent studies (Fu et al., 2025; Zheng et al., 2025; Zou et al., 2025) collectively advocate for a shift towards thought-level collaboration, proposing that agents should share internal representations—ranging from latent channels to shared working memories—rather than relying solely on surface text.

3 METHODOLOGY

3.1 GRICE'S COOPERATIVE PRINCIPLE

We motivate our method from Grice's Cooperative Principle (Grice, 1975): in cooperative communication, a speaker does not transmit everything they know, but instead follows conversational norms that balance informativeness, truthfulness, relevance, and clarity. The Gricean maxims are:

- **Quantity:** share as much as needed, but no more.
- **Quality:** avoid passing along low-evidence or speculative content that could mislead others.
- **Relation:** tailor the message to the recipient role.
- **Manner:** communicate through a structured channel that avoids ambiguity and prevents unintended leakage of private deliberation.

As tasks become more complex and labor is divided into more specialized roles, the volume of inter-agent communication, and the number of handoffs typically increase.

In LLM multi-agent pipelines, these maxims become protocol requirements for inter-agent state transfer. Our key design choice is to treat a sender's internal KV-cache as a rich "candidate message," and to construct a small *public* cache slice that better satisfies the above maxims. We map these abstract linguistic norms to concrete architectural constraints in Transformer-based agents:

Table 1: Mapping the Gricean maxims to multi-agent communication requirements and our modules.

Maxim	**Multi-Agent role**	**Our Module**
Manner	controlled public interface	interface scoping gate
Relation	recipient-aware content	semantic alignment
Quantity & Quality	bandwidth-limited message	signal preservation

We use Grice's Cooperative Principle as a high-level design lens for engineering a communication interface. Mechanistically, our method is an explicit and inspectable sparsification policy over a sender's KV-cache along three axes: (i) head subspaces that define what may be exposed through a public interface, (ii) layer depth that controls what is useful to the receiver's role, and (iii) token positions that retain sufficient signal under a bandwidth budget. Thus, the Gricean terminology provides a compact vocabulary for operational constraints on cache-level communication, while our empirical and theoretical claims concern the resulting KV-cache selection policy.

3.2 PRELIMINARIES

Notations. Let $\|\cdot\|_2$ denote ℓ_2 norm, $\mathbb{E}[\cdot]$ denote the expectation. Define $[n] = \{1, 2, \ldots, n\}$ for any integer n.

Agents. We consider a sequential multi-agent system with N agents $\{a_i\}_{i=1}^N$. They can share the same backbone model $\mathcal{M}$ but can have different roles (e.g., specified with different role prompts).[1] Let $\mathcal{L} = \{1, \ldots, L\}$ be the set of Transformer layers, and $\mathcal{H} = \{1, \ldots, H\}$ the set of attention heads. Let $\mathcal{R}$ denote the set of roles. For agent i, its role is denoted as $r_i \in \mathcal{R}$. Its KV-cache at layer ℓ and head h is $(K_{\ell,h}^{(i)}, V_{\ell,h}^{(i)}) \in \mathbb{R}^{T_i \times d_h} \times \mathbb{R}^{T_i \times d_h}$ where $K_{\ell,h}^{(i)} = [k_{\ell,h,t}^{(i)}]_{t=1}^{T_i}$ and $V_{\ell,h}^{(i)} = [v_{\ell,h,t}^{(i)}]_{t=1}^{T_i}$. We denote the full local cache as $C^{(i)} = \{(K_{\ell,h}^{(i)}, V_{\ell,h}^{(i)})\}_{\ell \in \mathcal{L}, h \in \mathcal{H}}$.

Sequential multi-agent system. Let $C_{\text{pub}}^{(i)}$ denote the information shared from agent a_i to the next agent (let $C_{\text{pub}}^{(0)} = \emptyset$). Given an input instance x, the agent a_i takes x with its role r_i and the information $C_{\text{pub}}^{(i-1)}$ from the previous agent a_{i-1}, and computes an intermediate output $y^{(i)}$ and its local cache $C^{(i)}$ as $(y^{(i)}, C^{(i)}) = \mathcal{M}(x, r_i, C_{\text{pub}}^{(i-1)})$. A communication policy ("communicator") g is used to control which cache entries are shared. It computes what information agent a_i should send from its local cache $C^{(i)}$, and concatenates that to the shared information: $C_{\text{send}}^{(i)} = g(C^{(i)})$ and $C_{\text{pub}}^{(i)} = C_{\text{pub}}^{(i-1)} \oplus C_{\text{send}}^{(i)}$, where $\oplus$ denotes concatenation along the token dimension (i.e., appending extra KV entries). $C_{\text{pub}}^{(i)}$ is then sent to the next agent, and so on. At the end, the agent a_N decodes the final answer $\hat{y} := y^{(N)}$. This is the output of the whole system and is denoted as $\hat{y} := F(x; g)$.

Clearly, the key is to define the communicator g, for which we will introduce our Gricean Cache Sharing method.

3.3 Gricean Cache Sharing Method

Our method consists of three modules inspired by the Gricean maxims; see Table 1. By careful design, the interface scoping gate module selects a subset of heads $\mathcal{H}_{\text{pub}} \subseteq \mathcal{H}$, the semantic alignment selects a subset of layers $\mathcal{L}(r)$ for each role $r \in \mathcal{R}$, and the signal preservation module selects a subset of token indices $S_{\ell,h}^{(i)} \subseteq \{1, \ldots, T_i\}$ for (ℓ, h) with $\ell \in \mathcal{L}(r_{i+1})$, $h \in \mathcal{H}_{\text{pub}}$. The communicator g then selects the corresponding cache entries for sharing: $C_{\text{send}}^{(i)} = g(C^{(i)}) := \left\{(K_{\ell,h,S_{\ell,h}^{(i)}}^{(i)}, V_{\ell,h,S_{\ell,h}^{(i)}}^{(i)})\right\}_{\ell \in \mathcal{L}(r_{i+1}),\, h \in \mathcal{H}_{\text{pub}}}$. The communicator design is then reduced to the three modules that select $\mathcal{H}_{\text{pub}}$, $\mathcal{L}(r)$, and $S_{\ell,h}^{(i)}$. Theoretically, this can be formalized as a constrained optimization problem. The communication cost is measured in transmitted scalars: $\text{Cost}(g; x) = \sum_{i=1}^{N-1} \sum_{\ell \in \mathcal{L}(r_{i+1})} \sum_{h \in \mathcal{H}_{\text{pub}}} (2d_h) \cdot |S_{\ell,h}^{(i)}(x)|$, where d_h denotes the per-head dimension, and the factor $2d_h$ accounts for sending both a key and a value vector per selected token index. The communicator g should be chosen to minimize the expected task loss under a communication budget B:

$$\min_g \ \mathbb{E}_{(x,y^*)\sim\mathcal{D}}\big[\ell(F(x;g), y^*)\big] \qquad \text{s.t.} \quad \mathbb{E}_{(x,y^*)\sim\mathcal{D}}[\text{Cost}(g;x)] \le B. \tag{1}$$

Practically, however, it is infeasible to accurately optimize over such a large selection space. We thus turn to proxy approaches following the Gricean maxims: $\mathcal{H}_{\text{pub}}$ are selected as the controlled public interface, $\mathcal{L}(r_{i+1})$ to align with the role r_{i+1} of the next agent, and $S_{\ell,h}^{(i)}$ to preserve as much signals as needed under the limited budget.

We describe the modules and summarize the method in Algorithm 1. Our analysis establishes Top-k optimality for fixed saliency scores and a per-block budget, and bounds the additional receiver tail mass caused by using sender-side proxy scores.

Module I: Interface Scoping Gate. The heads $\mathcal{H}$ are partitioned into public and private subsets as $\mathcal{H}_{\text{pub}} \cup \mathcal{H}_{\text{pri}} = \mathcal{H}$ and $\mathcal{H}_{\text{pub}} \cap \mathcal{H}_{\text{pri}} = \varnothing$. Only heads in $\mathcal{H}_{\text{pub}}$ are eligible for communication, while $\mathcal{H}_{\text{pri}}$ remains internal. Intuitively, this makes $\mathcal{H}_{\text{pub}}$ the "surface" channel for shareable latent facts, and $\mathcal{H}_{\text{pri}}$ the private channel for role-specific deliberation (e.g., scratch work, intermediate hypotheses). Importantly, this head-gate is an architectural *interface constraint* on what the communicator may expose, not a post-hoc filtering of text. A head-level gate is practical because the cache already factorizes by (ℓ, h), and it is inspectable: it constrains communication to a fixed set

[1] Our method can be readily generalized to settings where the agents are based on different backbone models. Here for clarity we present and evaluate our method on the same backbone.

of head subspaces at every layer. This reduces accidental leakage compared to free-form latent export, and it keeps the communication object aligned with the model's own computational units. We view head-gating as an architectural regularization that reduces representational interference or semantic leakage from irrelevant internal states. We construct the public head set using a lightweight calibration procedure. On a small unlabeled calibration set D_{cal}, we compute a magnitude-based head importance score $I_{\ell,h}$ as the average norm of each head's output contribution to the residual stream (Details in Appendix C). We then select the top-m heads with the largest scores to form $\mathcal{H}_{\text{pub}}$. This yields a stable, inspectable public interface while filtering out low-impact heads.

Algorithm 1 GRICEAN CACHE SHARING

Input: agents $\{a_i\}_{i=1}^N$ with same backbone $\mathcal{M}$ but different roles $\{r_i\}_{i=1}^N$, public head set $\mathcal{H}_{\text{pub}}$, role layers $\mathcal{L}(r)$ $(r \in \mathcal{R})$, gate budget k, latent steps ϕ
$C_{\text{pub}}^{(0)} \leftarrow \varnothing$
for $i = 1$ **to** $N-1$ **do**
  $(y^{(i)}, C^{(i)}) \leftarrow \mathcal{M}(x, r_i, C_{\text{pub}}^{(i-1)})$
  $C_{\text{send}}^{(i)} \leftarrow \varnothing$
  for all $h \in \mathcal{H}_{\text{pub}},\ \ell \in \mathcal{L}(r_{i+1})$ **do**
    $\bar{s}_{\ell,h,1:T_i}^{(i)} \leftarrow \text{SALIENCY}(\mathcal{M}, C^{(i)}, x, \ell, h, \mathcal{Q}_i)$
    $S_{\ell,h}^{(i)} \leftarrow \text{TOPK}(\{\bar{s}_{\ell,h,t}^{(i)}\}_{t=1}^{T_i}, k)$
    $C_{\text{send}}^{(i)} \leftarrow C_{\text{send}}^{(i)} \cup \{(K_{\ell,h,S_{\ell,h}^{(i)}}^{(i)}, V_{\ell,h,S_{\ell,h}^{(i)}}^{(i)})\}$
  end for
  $C_{\text{pub}}^{(i)} \leftarrow C_{\text{pub}}^{(i-1)} \oplus C_{\text{send}}^{(i)}$
end for
$(y^{(N)}, C^{(N)}) \leftarrow \mathcal{M}(x, r_N, C_{\text{pub}}^{(N-1)})$
RETURN $\hat{y} = y^{(N)}$

Module II: Semantic Alignment. Different roles typically benefit from different abstraction levels (lower layers are closer to lexical / local patterns; higher layers are closer to global semantics and reasoning features). We implement recipient-aware relevance control via a depth-alignment set $\mathcal{L}(r) \subseteq \{1, \ldots, L\}$, meaning that an agent with role r is allowed to *receive* public cache only on layers $\ell \in \mathcal{L}(r)$. Thus, when agent i sends to agent $i+1$, only $\ell \in \mathcal{L}(r_{i+1})$ are communicable. This makes it a structured sparsification: the receiver sees only the subset of layers deemed relevant to its role, reducing both bandwidth and representational interference. We instantiate $L(r)$ based on semantic profiling, aligning the communication channel with the abstraction level required by the receiver's role. In this work, "semantic profiling" is a lightweight, task-agnostic procedure that uses only the role prompt and the shared backbone, and does not tune $L(r)$ on benchmark-specific validation sets. Concretely, we compute a per-layer role-sensitivity score on a small set of generic, unlabeled prompts, and select depth-aligned layer bands for each role accordingly. We fix $L(r)$ once per backbone and reuse it across tasks. Appendix C provides the exact profiling recipe.

Module III: Signal Preservation. Unlike standard sparse attention analyses within a single model, our setting is cross-agent: the downstream query set is induced by the receiver's role prompt and decoding trajectory and is unknown to the sender at export time. Therefore, token indices $S_{\ell,h}^{(i)}$ should be selected to optimize the downstream task loss with the given budget as formalized in (1). This means token-gating must decide which cached token positions are sufficiently informative (and sufficiently reliable) to share. Similar to the other modules, directly optimizing the loss over discrete subsets $S_{\ell,h}^{(i)}$ is intractable. We adopt a proxy based on *how much the model itself attends to each token position* during generation, and later show that for fixed (ℓ, h) our token selection optimizes the error bound.

Definition 3.1 (Attention Saliency)**.** Fix an agent i, layer ℓ, head h, and a set of query positions $\mathcal{Q}_i$ (e.g., all decoding steps). Let $\alpha_{\ell,h,\tau\to t}^{(i)}$ be the attention weight from query $\tau \in \mathcal{Q}_i$ to key position $t \in \{1, \ldots, T_i\}$. We define the *normalized saliency* of position t as $\bar{s}_{\ell,h,t}^{(i)} := \frac{1}{|\mathcal{Q}_i|} \sum_{\tau \in \mathcal{Q}_i} \alpha_{\ell,h,\tau\to t}^{(i)}$. Then $\sum_{t=1}^{T_i} \bar{s}_{\ell,h,t}^{(i)} = 1$.

We interpret $\bar{s}_{\ell,h,1:T_i}^{(i)}$ as an empirical distribution over token positions: positions that consistently receive attention mass across queries are treated as more salient. We then select the top-k entries as $S_{\ell,h}^{(i)}$. In practice, $\mathcal{Q}_i$ at most ϕ silent latent probe steps run by the sender to approximate the receiver-induced query distribution; Appendix C details the procedure.

Definition 3.2 (Signal Preservation)**.** For each allowed (ℓ, h), let $S_{\ell,h}^{(i)}$ be the set of indices of the top-k entries of $\{\bar{s}_{\ell,h,t}^{(i)}\}_{t=1}^{T_i}$.

3.4 THEORETICAL PROPERTIES

Text-Latent separation. We formalize an information-theoretic separation between text-mediated communication and latent transmission. Even under a strict budget, KV-slice messages can represent a combinatorially large space, while an m'-token text channel can express at most $|V|^{m'}$ distinct messages. This implies a lower bound on the text length required to simulate KV-slice broadcasting losslessly, and yields a strict separation on a constructed task family: under the same token budget, there exist collaboration tasks solvable by KV transmission but not by text-only protocols. This representational-capacity separation constrains token budgets and complements our measured latency results.

Theorem 3.3 (Informal version of Theorem D.8)**.** *Let V be a finite vocabulary. Consider length-m KV-slice messages at a fixed block with head dimension d_h, even restricted to a ternary subset. If a text-only protocol with m' tokens losslessly simulates all such thought messages, then necessarily*

$$m' \geq (2d_h m \log 3)/\log|V|.$$

Conversely, if m' violates this bound, there exists a two-agent collaboration task distribution for which broadcasting the KV-slice achieves zero loss, whereas any m'-token text protocol incurs nonzero expected error.

Error bound. Our second result bounds the task loss in (1) by the tail mass not covered by the selection. Let g_{full} be the reference policy sending all cache entries for allowed (ℓ, h) (i.e., $S^{(i)}_{\ell,h} = \{1, \ldots, T_i\}$). The excess loss of g over g_{full} is

$$\Delta_{\text{task}}(g) := \mathbb{E}\Big[\ell(F(x; g), y^*) - \ell(F(x; g_{\text{full}}), y^*)\Big]. \tag{2}$$

Definition 3.4 (average tail mass)**.** For a transmitted index set $S^{(i)}_{\ell,h} \subseteq \{1, \ldots, T_i\}$, define the discarded attention mass averaged over query set $\mathcal{Q}$ as

$$\varepsilon^{(i)}_{\ell,h}(x) := \frac{1}{|Q|} \sum_{\tau \in Q} \sum_{t \notin S^{(i)}_{\ell,h}} \alpha_{i,\ell,h,\tau \to t}(x).$$

Intuitively, ε measures how much attention probability is removed by token gating.

Theorem 3.5 (Informal version of Theorem E.4)**.** *Assume the value vectors are bounded by ξ, the task loss is L_{task}-Lipschitz, and the logits satisfy the blockwise sensitivity condition of Assumption B.4 with constants $\Gamma^{(i)}_{\ell,h} \geq 0$. Then the excess loss in (2) satisfies*

$$\Delta_{\text{task}}(g) \leq 2L_{\text{task}}\xi \sum_{i=1}^{N-1} \sum_{\substack{\ell \in \mathcal{L}(r_{i+1}) \\ h \in \mathcal{H}_{\text{pub}}}} \Gamma^{(i)}_{\ell,h} \cdot \mathbb{E}_{x \sim \mathcal{D}}\big[\varepsilon^{(i)}_{\ell,h}(x)\big].$$

This theorem indicates that minimizing the "discarded attention mass" guarantees deviation from the full-model performance is small. Based on this insight, we prove that our selection $S^{(i)}_{\ell,h}$ is optimal for the error bound.

Theorem 3.6 (Informal Version of Theorem F.1)**.** *Fix (i, ℓ, h, x) and let $p^{(i)}_{\ell,h,t}(x)$ be the receiver saliency scores in Definition 3.1. Under $|S| = k$, any set of k indices attaining the largest k entries of $\{p^{(i)}_{\ell,h,t}(x)\}_{t=1}^{T_i}$ minimizes $\varepsilon^{(i)}_{\ell,h}(x; S)$, with arbitrary tie-breaking. When Top-k is applied to the sender proxy $\bar{s}$ instead, the additional tail mass is at most $\|p - \bar{s}\|_1$.*

Hence, for fixed receiver scores and a per-block budget, Top-k minimizes the tail-mass term of the bound in Theorem 3.5; the sender-side selector used in practice adds a term controlled by the sender & receiver score discrepancy.

4 EXPERIMENTS

We evaluate our method as a communication protocol for sequential multi-agent collaboration. Our experiments aim to answer three questions: **(Q1)** Does Gricean Cache sharing improve the accuracy–latency trade-off over Single Model and Text Communication? **(Q2)** Are the three components

(interface scoping gate, semantic alignment, and signal preservation) necessary and complementary? **(Q3)** Is the communication budget a reliable control knob, and does signal preservation behave consistently with our theory?

Our experiments provide the following findings. **(A1)** As detailed in Table 2, GRICEAN CACHE SHARING reduces latency by $3.3\times$ on average (up to $5.0\times$) while matching or exceeding text-baseline accuracy in 12 of 15 settings; among all communicating methods it is the fastest at every backbone. **(A2)** The full module combination leads the ablations in Table 3; our head- and layer-selection rules lead the alternatives at a matched budget. **(A3)** Figure 2 shows that Top-k attains the highest accuracy at every tested token budget, and Figure 3 shows that the communication budget acts as a smooth, reliable control knob.

4.1 EVALUATION

We evaluate Gricean Cache Sharing on three reasoning-intensive benchmarks that stress different aspects of multi-agent collaboration: ARC-Challenge (Clark et al., 2018) (complex multi-choice reasoning), GSM8K (Cobbe et al., 2021) (multi-step arithmetic reasoning), and HumanEval+ from EvalPlus (Chen et al., 2021; Liu et al., 2023) (code synthesis with unit-test based verification); we report answer accuracy for ARC-C/GSM8K and pass@1 under the expanded unit tests for HumanEval+.

Table 2: Main results: accuracy (%) and per-instance latency (s) under the shared evaluation setup. Payload: exported KV tensors (MiB/hop). ‡: Full-KV reference, excluded from ranking; †: trained projector; adapted mechanisms per §4.1. Best displayed accuracy is bold, second best underlined; latency bolding compares communicating methods.

Model	**Bench.**	No-Comm.		Text		Full-KV‡		KVComm		C2C†		**GCS (ours)**	
		Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time
Qwen3-1.7B	ARC-C	84.4	3.0	82.0	11.5	85.7	6.8	83.6	4.9	84.4	7.3	**84.9**	**4.6**
	GSM8K	76.4	3.2	82.6	16.2	84.9	8.4	82.2	6.4	**84.6**	9.0	83.9	**5.9**
	HE+	66.5	27.7	64.6	42.1	70.1	35.5	67.7	25.6	68.9	37.2	**69.5**	**24.2**
	Avg	75.8	11.3	76.4	23.3	80.2	16.9	77.8	12.3	79.3	17.8	**79.4**	**11.6**
DeepSeek-R1-D. 1.5B	ARC-C	53.2	3.0	**57.2**	16.3	56.4	6.0	54.6	4.5	54.9	6.6	55.7	**4.2**
	GSM8K	70.2	4.2	73.2	20.8	75.0	9.3	72.6	6.7	73.4	9.6	**74.1**	**6.3**
	HE+	47.0	8.9	**58.5**	40.3	57.9	14.9	55.5	11.0	56.7	16.4	57.3	**10.4**
	Avg	56.8	5.4	**63.0**	25.8	63.1	10.1	60.9	7.4	61.7	10.9	62.4	**7.0**
Qwen3-4B	ARC-C	88.7	5.7	88.3	18.6	93.9	9.0	91.6	6.8	92.4	9.6	**92.9**	**6.3**
	GSM8K	86.2	5.9	89.6	21.2	90.6	7.9	88.2	5.7	**90.4**	8.5	89.8	**5.3**
	HE+	75.0	21.8	78.7	92.4	81.1	37.0	78.7	27.4	**81.1**	40.2	80.5	**25.7**
	Avg	83.3	11.1	85.5	44.1	88.5	18.0	86.2	13.3	**88.0**	19.4	87.7	**12.4**
Mistral-7B v0.3	ARC-C	53.2	0.5	57.8	9.3	60.1	3.4	58.1	2.5	58.8	3.7	**59.3**	**2.3**
	GSM8K	39.3	1.1	44.3	8.5	47.5	2.6	45.1	1.9	46.2	2.8	**46.7**	**1.7**
	HE+	27.4	2.0	**29.3**	17.2	29.3	7.0	26.8	5.3	28.0	7.8	28.7	**4.9**
	Avg	40.0	1.2	43.8	11.7	45.6	4.3	43.3	3.2	44.3	4.8	**44.9**	**3.0**
Qwen3-8B	ARC-C	93.2	5.8	89.6	20.1	94.7	10.2	92.7	7.4	93.5	10.9	**93.9**	**6.9**
	GSM8K	88.6	6.8	90.4	24.4	93.4	10.2	91.1	7.7	**93.0**	11.4	92.6	**7.2**
	HE+	77.4	32.8	**78.7**	125.2	79.3	44.0	76.8	32.0	**78.7**	46.3	**78.7**	**30.1**
	Avg	86.4	15.1	86.2	56.6	89.1	21.5	86.9	15.7	**88.4**	22.9	**88.4**	**14.7**
Average	ARC-C	74.6	3.6	75.0	15.2	78.2	7.1	76.1	5.2	76.8	7.6	**77.4**	**4.9**
	GSM8K	72.1	4.2	76.0	18.2	78.3	7.7	75.8	5.7	**77.5**	8.3	77.4	**5.3**
	HE+	58.7	18.6	62.0	63.4	63.5	27.7	61.1	20.3	62.7	29.6	**62.9**	**19.1**
	Avg	68.5	8.8	71.0	32.3	73.3	14.1	71.0	10.4	72.3	15.2	**72.6**	**9.7**
Payload (MiB/hop)		–		<0.1		17.5–90.0		2.5–12.5		17.5–90.0		2.1–10.6	

Backbones and baselines. We instantiate all agents with the same decoder-only backbone and vary only the role prompts. We report results across Qwen3-1.7B/4B/8B (Yang et al., 2025), DeepSeek-R1-

Distill-Qwen-1.5B (Guo et al., 2025), and Mistral-7B-Instruct (Jiang et al., 2023). No-Communication removes collaboration entirely, isolating the benefit of any multi-agent protocol beyond a single forward pass. Text Communication follows the standard Chain-of-Agents (Zhang et al., 2024) style pipeline where agents forward intermediate reasoning in natural language; it is the dominant communication mechanism in prior multi-agent systems and serves as a strong accuracy-oriented reference. All methods use $N{=}3$ agents (Planner→Solver→Verifier role prompts; No-Communication uses a single agent) on one A100-80GB GPU with vLLM at batch size 1, using greedy decoding for visible text. Latency is end-to-end wall-clock per instance with all selection overhead included, following the speed-reporting protocol of latent-collaboration evaluations (Zou et al., 2025). Implementation and accounting details are in Appendix C. The headline $3.3\times$ is the ratio of mean Text and GCS latencies over the 15 settings. We also wanted to know how our selector compares with other ways of moving cache between agents, even though they were designed for different questions than ours: full-cache relay asks what is recoverable when nothing is dropped, Shi et al. (2026) asks which layers to keep, and Fu et al. (2025) asks whether a learned map can bridge two models. We therefore ported all three into our harness, gave the budgeted ones our 12% budget—rounded up to a whole layer for KVComm, the smallest unit it can drop—and timed them the same way, so the comparison reflects the mechanisms rather than the surrounding implementation.

Our main result. Across backbones, our method consistently removes most of the overhead of text-mediated collaboration while retaining (and often improving) accuracy on reasoning and coding benchmarks (Clark et al., 2018; Cobbe et al., 2021; Liu et al., 2023). For example, with Qwen3-4B, we improve ARC-C accuracy to 92.9% (vs. 88.7% no-communication and 88.3% text communication) while reducing text-communication latency from 18.6s to 6.3s (Table 2). On the same backbone, we reach 89.8% on GSM8K at 5.3s (about $4\times$ faster than text) and 80.5% pass@1 on HumanEval+ at 25.7s (about $3.6\times$ faster than text). Similar gains hold for Qwen3-8B, where we achieve 92.6% on GSM8K and match the best HumanEval+ score at $4.2\times$ lower latency than text communication. The largest speedup, $5.0\times$ (8.5s vs. 1.7s), occurs on Mistral-7B GSM8K. Against C2C, GCS has higher accuracy in 10 of 15 cells and matches or exceeds the displayed mean on four of five backbones. On Qwen3-4B, C2C has a 0.3-point higher mean at $1.6\times$ the latency and $8.5\times$ the payload. GCS exceeds KVComm in all 15 cells and has the lowest latency among communicating methods.

4.2 Component Analysis

Table 3 presents the module ablation under both budget protocols. Role-conditioned routing (Module II) is beneficial on its own, and the full combination leads every single- and dual-module variant under both protocols.

For the module variants, free-budget target fractions are 0.50 (I), 0.48 (II), and 0.50 (III), with disabled modules retaining their full axis. The matched protocol holds each row's head and layer counts fixed and adjusts positions using uniform stride when III is off and saliency when III is on. The full configuration is unchanged. Our head- and layer-selection rules lead the strongest alternatives by 0.6 and 0.8 points at the matched budget (Table 5). Saliency must be computed on the public interface: mixing private-head saliency spends the limited budget on role-irrelevant cache entries and degrades with the private-head weight w, while the aligned policy is invariant (Figure 2d,e).

Table 3: **Component ablation analysis.** Separate Qwen3-1.7B/GSM8K ablation. Nom.: free-budget target KV fraction (%); matched budgets target 12%. Δ vs. the baseline; δ: matched − free accuracy; matched means over 8 selector seeds.

Config.	Modules			Free budget			Matched 12%		δ
	I.	II.	III.	Nom.	Acc.	Δ	Acc.	Δ	
Baseline	–	–	–	–	76.7	–	–	–	–
Single	✓			50.0	74.2	-2.5	79.1	+2.4	4.9
		✓		48.0	78.9	+2.2	80.8	+4.1	1.9
			✓	50.0	70.2	-6.5	78.5	+1.7	8.3
Dual	✓	✓		24.0	77.4	+0.7	81.4	+4.7	4.0
	✓		✓	25.0	78.2	+1.4	81.7	+5.0	3.6
		✓	✓	24.0	76.9	+0.2	81.0	+4.2	4.1
Ours	✓	✓	✓	**12.0**	**83.3**	**+6.6**	**83.3**	**+6.6**	–

4.3 Budget Scaling Behavior

Two questions decide whether a communication budget is usable: does the selection rule spend it well, and what does the resulting message cost. Figure 2 answers the first on two levels. Panels (a–c) sweep the token budget k on the three benchmarks and compare our saliency Top-k against random-k

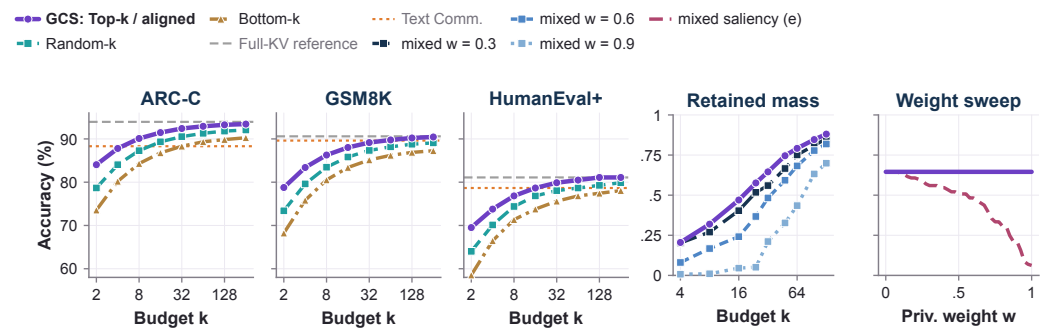


Figure 2: **Budget scaling and public-interface alignment**. (a–c) Top-k saliency versus random-k and bottom-k across token budgets on Qwen3-4B, with Full-KV relay and Text Communication as references. (d) Retained public-head attention mass under aligned and mixed saliency. (e) The aligned policy is invariant to the private-head weight w; mixed saliency degrades monotonically.

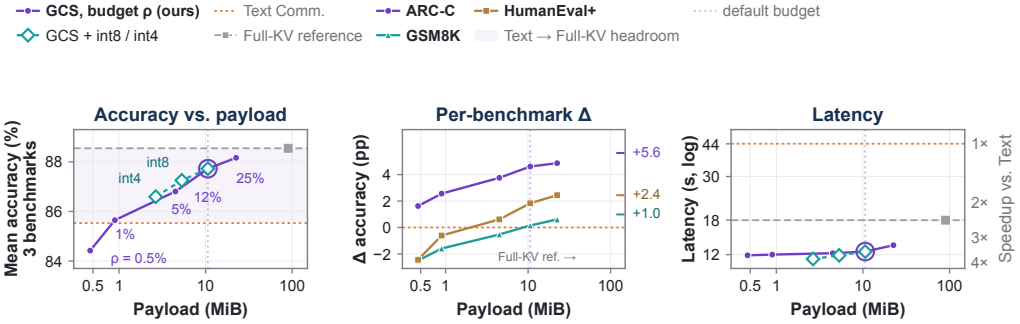


Figure 3: **Operating points on the cost axis** (Qwen3-4B). Accuracy, per-benchmark gain over Text Communication, and latency as functions of the transmitted payload; circled markers indicate the default budget, and open diamonds denote int8/int4 payload quantization. Exact values: Appendix C.

and bottom-k under identical gates, against the full-cache and text references; panels (d,e) instrument the selector itself, tracking how much of the public-head attention mass survives selection as the budget and the private-head weight w vary. Panels (a–c) measure task accuracy, while (d,e) measure retained attention mass. Their observed trends support attention-mass retention as a useful selection mechanism.

Top-k leads bottom-k by 10.5 points on ARC-C at k=2 and remains ahead throughout the measured range. All three policies improve with k; their gaps persist at the largest tested budget. Panels (d,e) show that aligned saliency retains more public mass and is invariant to w, while mixed saliency retains less public mass as w rises. At a matched k, the policies differ in which entries they select.

Figure 3 varies total KV payload, a different budget quantity from the fixed-gate token count k in Figure 2. Mean accuracy increases across the reported operating points and first exceeds Text Communication at 0.9 MiB. The default 10.6 MiB point is 0.8 points below Full-KV relay at 3.5× lower latency than text. Across 0.5–22.5 MiB, latency increases from 11.9 to 13.4 s. At the default selection, int8 and int4 reduce KV tensor storage by factors of 2 and 4 while retaining a positive mean accuracy gain over text.

5 Conclusion

We addressed the inherent information bottleneck in text-mediated agents by introducing Gricean Cache Sharing, which reformulates communication as a bandwidth-constrained optimization problem. Theoretically we showed that at an equal number of transmitted positions KV-slice messages can distinguish strictly more messages than discrete text. Empirically, this framework yields a trade-off, matching text baseline accuracy while reducing latency significantly.

AI USE STATEMENT

In this work, we used generative AI tools for editing assistance (grammar, spelling, and word choice). We have reviewed all AI-assisted content, and we take responsibility for the final content of this work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Marco Baroni, Roberto Dessì, and Angeliki Lazaridou. Emergent language-based coordination in deep multi-agent systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pp. 11–16, 2022.

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*, 2020.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*, 2025.

Guikun Chen, Jin Li, and Wenguan Wang. Scene graph generation with role-playing large language models. *Advances in Neural Information Processing Systems*, 37:132238–132266, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

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.

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.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. Multi-agent collaboration via evolving orchestration. *arXiv preprint arXiv:2505.19591*, 2025.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AAl89VNNy1.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

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.

Lorenzo Giusti, Ole Anton Werner, Riccardo Taiello, Matilde Carvalho Costa, Emre Tosun, Andrea Protani, Marc Molina, Rodrigo Lopes de Almeida, Paolo Cacace, Diogo Reis Santos, et al. Federation of agents: A semantics-aware communication fabric for large-scale agentic ai. *arXiv preprint arXiv:2509.20175*, 2025.

H.P. Grice. Logic and conversation. *Speech acts*, 1975.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Zixuan Ke, Austin Xu, Yifei Ming, Xuan-Phi Nguyen, Ryan Chin, Caiming Xiong, and Shafiq Joty. Mas-zero: Designing multi-agent systems with zero supervision. *arXiv preprint arXiv:2505.14996*, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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.

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.

Jiaheng Liu, Zehao Ni, Haoran Que, Tao Sun, Zekun Wang, Jian Yang, Jiakai Wang, Hongcheng Guo, Zhongyuan Peng, Ge Zhang, et al. Roleagent: Building, interacting, and benchmarking high-quality role-playing agents from scripts. *Advances in Neural Information Processing Systems*, 37:49403–49428, 2024.

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.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Paul Michel, Mathieu Rita, Kory Wallace Mathewson, Olivier Tieleman, and Angeliki Lazaridou. Revisiting populations in multi-agent communication. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=n-UHRIdPju.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.

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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sehRvaIPQQ.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sarath Shekkizhar, Romain Cosentino, Adam Earle, and Silvio Savarese. Echoing: Identity failures when llm agents talk to each other. *arXiv preprint arXiv:2511.09710*, 2025.

Xiangyu Shi, Marco Chiesa, Gerald Q Maguire Jr, and Dejan Kostic. Kvcomm: Enabling efficient llm communication through selective kv sharing. In *International Conference on Learning Representations*, volume 2026, pp. 84546–84573, 2026.

Yu Su, Diyi Yang, Shunyu Yao, and Tao Yu. Language agents: Foundations, prospects, and risks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pp. 17–24, 2024.

Georgios Tzachristas, Lei Deng, Ioannis Tzachristas, Gong Zhang, and Renhai Chen. A mathematical theory of top-k sparse attention via total variation distance. *arXiv preprint arXiv:2512.07647*, 2025.

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.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, 2019.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131, 2024.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions, 2022.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

Nuoya Xiong, Yuhang Zhou, Hanqing Zeng, Zhaorun Chen, Furong Huang, Shuchao Bi, Lizhu Zhang, and Zhuokai Zhao. Token-level llm collaboration via fusionroute. *arXiv preprint arXiv:2601.05106*, 2026.

Shuyao Xu, Cheng Peng, Jiangxuan Long, Weidi Xu, Wei Chu, and Yuan Qi. Harnessing negative signals: Reinforcement distillation from teacher data for llm reasoning. *arXiv preprint arXiv:2505.24850*, 2025.

Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025a.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems. *arXiv preprint arXiv:2502.11133*, 2025b.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. In *Forty-second International Conference on Machine Learning*, 2025a.

Wentao Zhang, Ce Cui, Yilei Zhao, Yang Liu, and Bo An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025b.

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.

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.

Han Zhou, Xingchen Wan, Ruoxi Sun, Hamid Palangi, Shariq Iqbal, Ivan Vulić, Anna Korhonen, and Sercan Ö Arık. Multi-agent design: Optimizing agents with better prompts and topologies. *arXiv preprint arXiv:2502.02533*, 2025.

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.

Appendix

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Methodology **3**

3.1 Grice's Cooperative Principle 3

3.2 Preliminaries 3

3.3 Gricean Cache Sharing Method 4

3.4 Theoretical Properties 6

4 Experiments **6**

4.1 Evaluation 7

4.2 Component Analysis 8

4.3 Budget Scaling Behavior 8

5 Conclusion **9**

A More Related Work on Large Language Models **15**

B Preliminary **15**

B.1 Assumptions 15

B.2 Core Definitions 16

C Additional Details **17**

C.1 Communication Alignment 17

C.2 Latent-thought generation 19

C.3 Head selection 19

C.4 Semantic profiling 19

C.5 Budget, latency, and payload 20

D Proof of Theorem 3.3 **21**

E Proof of Theorem 3.5 **24**

F Proof of Theorem 3.6 **26**

A More Related Work on Large Language Models

Built upon the Transformer architecture (Vaswani et al., 2017), modern LLMs are characterized by their colossal scale in both parameters and pre-training corpora (Achiam et al., 2023; Yang et al., 2025; Comanici et al., 2025; Guo et al., 2025). The standard development lifecycle typically consists of auto-regressive pre-training (Radford et al., 2018; 2019; Brown et al., 2020) followed by a post-training phase. To adapt these base models for diverse applications, extensive research has been dedicated to post-training techniques, including multitask finetuning (Sanh et al., 2022; Wang et al., 2023; Xu et al., 2024), instruction tuning (Chung et al., 2022; Mishra et al., 2022; Wang et al., 2022), and in-context learning (Min et al., 2022; Dong et al., 2022; Yao et al., 2023). Furthermore, alignment strategies such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Rafailov et al., 2023) and RLVR (Guo et al., 2025; Shao et al., 2024; Yue et al., 2025a; Xu et al., 2025) have become essential.

B Preliminary

We analyze token selection with fixed head and layer gates. Each local comparison fixes the candidate sender cache, the reference receiver query, and all retained keys and values. The extension from these local comparisons to final logits is conditional on Assumption B.4.

B.1 Assumptions

In particular, for each sending step $i \in \{1, \dots, N-1\}$, each role-routed layer $\ell \in \mathcal{L}(r_{i+1})$, and each public head $h \in \mathcal{H}_{\text{pub}}$, agent i has local KV-cache $(K^{(i)}_{\ell,h}, V^{(i)}_{\ell,h})$ with $V^{(i)}_{\ell,h} = [v^{(i)}_{\ell,h,t}]_{t=1}^{T_i}$ and $v^{(i)}_{\ell,h,t} \in \mathbb{R}^{d_h}$. For a reference receiver query τ, let $\Omega_{i,\ell,h,\tau}(x)$ index the complete attention memory, including the candidate sender segment $J_i = [T_i]$, local memory, and other public entries. Write $P_{i,\ell,h,\tau\to t}(x)$ for the softmax weights on this complete memory and $w_{i,\ell,h,\tau,t}(x)$ for its values, with $w_{i,\ell,h,\tau,t}(x) = v^{(i)}_{\ell,h,t}$ on J_i.

Assumption B.1 (Bounded value vectors)**.** For every compared block, reference receiver query, and position $t \in \Omega_{i,\ell,h,\tau}(x)$, including local and other public positions, the value vectors satisfy

$$\|w_{i,\ell,h,\tau,t}(x)\|_2 \le \xi. \tag{3}$$

Here $P \ge 0$ sums to one over Ω. Let $\beta_{i,\ell,h,\tau}(x) := \sum_{t\in J_i} P_{i,\ell,h,\tau\to t}(x)$ be the total attention mass on agent i's segment. The reference query set contains queries with $\beta > 0$. On this segment, define $\alpha^{\text{full}}_{i,\ell,h,\tau\to t}(x) := P_{i,\ell,h,\tau\to t}(x)/\beta_{i,\ell,h,\tau}(x)$.

Assumption B.2 (Simplex attention weights on the segment)**.** Fix any input x. For every tuple (i, ℓ, h) and every downstream query position $\tau \in Q^{(\ell,h)}_{i+1}(x)$ (Definition B.6), the conditional attention weights over the T_i keys in agent i's segment satisfy

$$\alpha^{\text{full}}_{i,\ell,h,\tau\to t}(x) \ge 0, \qquad \sum_{t=1}^{T_i} \alpha^{\text{full}}_{i,\ell,h,\tau\to t}(x) = 1. \tag{4}$$

Assumption B.3 (Lipschitz surrogate loss in logits)**.** For every label y, the surrogate loss $\ell_{\text{task}}(z, y)$ is L_{task}-Lipschitz in z with respect to $\|\cdot\|_2$, i.e.,

$$\left|\ell_{\text{task}}(z, y) - \ell_{\text{task}}(z', y)\right| \le L_{\text{task}}\|z - z'\|_2. \tag{5}$$

This condition applies to the surrogate loss used in the analysis, rather than discrete accuracy or pass@1.

Assumption B.4 (Logit sensitivity to block readout perturbations)**.** Let $z(x; g) \in \mathbb{R}^{d_{\text{logit}}}$ denote the logits used by the surrogate loss under communicator g, and let g_{full} retain all sender positions within the same head and layer gates.

For the communicator under analysis, there exist finite constants $\Gamma^{(i)}_{\ell,h} \geq 0$, independent of x, such that for every input x,

$$\big\|z(x;g) - z(x;g_{\text{full}})\big\|_2 \leq \sum_{i=1}^{N-1} \sum_{\ell \in \mathcal{L}(r_{i+1})} \sum_{h \in \mathcal{H}_{\text{pub}}} \Gamma^{(i)}_{\ell,h}\, \Delta^{(i)}_{\ell,h}(x), \tag{6}$$

where $\Delta^{(i)}_{\ell,h}(x)$ is the blockwise average joint-memory readout perturbation (Definition B.11).

This is an additional sensitivity condition on the complete computation, including changes in subsequent queries and decoding trajectories. It is not implied by the local softmax truncation identity.

B.2 Core Definitions

We now introduce several auxiliary quantities that will be used to formalize the notion of communication-induced degradation, and to connect discarded attention mass with excess task loss.

Definition B.5 (Excess task loss)**.** Let $\mathcal{D}$ be the data distribution over $(x, y^\star)$. Define the excess task loss of communicator g relative to full-sending as

$$\Delta_{\text{task}}(g) := \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[\ell_{\text{task}}(z(x;g), y^\star) - \ell_{\text{task}}(z(x;g_{\text{full}}), y^\star)\Big]. \tag{7}$$

Note that $\Delta_{\text{task}}(g)$ can be negative; we upper-bound it via the absolute loss difference using Lipschitzness. The reference retains all positions within the fixed gates and is distinct from the unrestricted Full-KV experimental baseline.

Definition B.6 (Downstream query set for a block)**.** Fix a sending step $i \in \{1, \dots, N-1\}$, a routed layer $\ell \in \mathcal{L}(r_{i+1})$, and a public head $h \in \mathcal{H}_{\text{pub}}$. For an input x, let $Q^{(\ell,h)}_{i+1}(x)$ be a fixed set of reference receiver query positions with positive attention mass on agent i's candidate segment. The queries are held fixed across the local pruning comparisons; they are not the sender probe set $\mathcal{Q}_i$.

We assume $Q^{(\ell,h)}_{i+1}(x)$ is finite. If $Q^{(\ell,h)}_{i+1}(x) = \emptyset$, then by convention we set all block-averaged quantities defined as "average over $Q^{(\ell,h)}_{i+1}(x)$" to be 0 (e.g., Definitions B.10 and B.11).

Definition B.7 (Transmitted index set)**.** Fix (i, ℓ, h) and input x. Let $S^{(i)}_{\ell,h}(x) \subseteq \{1, \dots, T_i\}$ be the set of token indices transmitted by communicator g for this block.

Definition B.8 (Full joint-memory attention readout)**.** Fix (i, ℓ, h, x) and $\tau \in Q^{(\ell,h)}_{i+1}(x)$. The full readout includes the sender segment and the background memory:

$$a^{\text{full}}_{i,\ell,h,\tau}(x) := \sum_{t \in \Omega_{i,\ell,h,\tau}(x)} P_{i,\ell,h,\tau\to t}(x)\, w_{i,\ell,h,\tau,t}(x). \tag{8}$$

Definition B.9 (Joint attention weights and gated readout)**.** Fix (i, ℓ, h, x, τ) and abbreviate $\Omega = \Omega_{i,\ell,h,\tau}(x)$, $P_t = P_{i,\ell,h,\tau\to t}(x)$ and $S = S^{(i)}_{\ell,h}(x)$. Deleting only $J_i \setminus S$ leaves $R = \Omega \setminus (J_i \setminus S)$. Let $\eta = \sum_{t \in J_i \setminus S} P_t$ and require $P(R) = 1 - \eta > 0$. With the query and retained K/V fixed, the joint weights after deletion are

$$P^{\text{gate}}_{i,\ell,h,\tau\to t}(x) := \begin{cases} P_t/(1-\eta) & t \in R, \\ 0 & t \notin R. \end{cases} \tag{9}$$

The positive-mass condition holds for finite softmax logits when at least one visible key is retained. The joint gated readout is

$$a^{\text{gate}}_{i,\ell,h,\tau}(x) := \sum_{t \in R} \frac{P_t}{1-\eta}\, w_{i,\ell,h,\tau,t}(x). \tag{10}$$

Definition B.10 (Per-query tail mass and average tail mass)**.** Fix (i, ℓ, h) and input x. For any $\tau \in Q^{(\ell,h)}_{i+1}(x)$, define the discarded (tail) mass

$$\delta^{(i)}_{\ell,h}(\tau; x) := \sum_{t \in J_i \setminus S^{(i)}_{\ell,h}(x)} \alpha^{\text{full}}_{i,\ell,h,\tau\to t}(x). \tag{11}$$

Define the average tail mass over the query set as

$$\varepsilon^{(i)}_{\ell,h}(x) := \frac{1}{|Q^{(\ell,h)}_{i+1}(x)|} \sum_{\tau \in Q^{(\ell,h)}_{i+1}(x)} \delta^{(i)}_{\ell,h}(\tau; x), \tag{12}$$

with the convention $\varepsilon^{(i)}_{\ell,h}(x) = 0$ if $Q^{(\ell,h)}_{i+1}(x) = \emptyset$.

Definition B.11 (Average joint-memory readout perturbation)**.** Fix (i, ℓ, h) and input x. Using the full and gated joint-memory readouts, define

$$\Delta^{(i)}_{\ell,h}(x) := \frac{1}{|Q^{(\ell,h)}_{i+1}(x)|} \sum_{\tau \in Q^{(\ell,h)}_{i+1}(x)} \left\| a^{\text{full}}_{i,\ell,h,\tau}(x) - a^{\text{gate}}_{i,\ell,h,\tau}(x) \right\|_2, \tag{13}$$

with the convention $\Delta^{(i)}_{\ell,h}(x) = 0$ if $Q^{(\ell,h)}_{i+1}(x) = \emptyset$.

Definition B.12 (Reference receiver saliency)**.** Fix (i, ℓ, h), input x, and the reference receiver queries. Define the receiver saliency scores

$$p^{(i)}_{\ell,h,t}(x) := \frac{1}{|Q^{(\ell,h)}_{i+1}(x)|} \sum_{\tau \in Q^{(\ell,h)}_{i+1}(x)} \alpha^{\text{full}}_{i,\ell,h,\tau \to t}(x), \qquad t \in \{1, \dots, T_i\}, \tag{14}$$

with $p^{(i)}_{\ell,h,t}(x) = 0$ if $Q^{(\ell,h)}_{i+1}(x) = \emptyset$.

Under Assumption B.2 and $Q^{(\ell,h)}_{i+1}(x) \neq \emptyset$, we have $\sum_{t=1}^{T_i} p^{(i)}_{\ell,h,t}(x) = 1$. These reference receiver scores are distinct from the sender-side proxy $\widehat{s}$ in Definition 3.1.

Definition B.13 (Tail mass induced by a candidate transmitted set)**.** Fix (i, ℓ, h) and input x. For any candidate set $S \subseteq \{1, \dots, T_i\}$, define

$$\varepsilon^{(i)}_{\ell,h}(x; S) := \sum_{t \in [T_i] \setminus S} p^{(i)}_{\ell,h,t}(x). \tag{15}$$

If $S = S^{(i)}_{\ell,h}(x)$ (Definition B.7), then $\varepsilon^{(i)}_{\ell,h}(x; S) = \varepsilon^{(i)}_{\ell,h}(x)$ (Definition B.10).

C Additional Details

C.1 Communication Alignment

We represent the public memory $C^{(i)}_{\text{pub}}$ as a per-layer, per-head KV container:

$$C^{(i)}_{\text{pub}} = \left\{ \left(K^{\text{pub}}_{\ell,h}, V^{\text{pub}}_{\ell,h} \right) \right\}_{\ell \in \{1,\dots,L\},\, h \in \{1,\dots,H\}}, \tag{16}$$

where $K^{\text{pub}}_{\ell,h} \in \mathbb{R}^{m_{\ell,h} \times d_h}$ and $V^{\text{pub}}_{\ell,h} \in \mathbb{R}^{m_{\ell,h} \times d_h}$ store the concatenated keys/values of all previously shared tokens on layer ℓ, head h. For heads not in $\mathcal{H}_{\text{pub}}$, we set $m_{\ell,h} = 0$ by construction (interface scoping gate).

After agent i finishes, we first take its local cache $C^{(i)} = \{(K^{(i)}_{\ell,h}, V^{(i)}_{\ell,h})\}_{\ell,h}$. We apply: (i) head-gating by restricting to $h \in \mathcal{H}_{\text{pub}}$, (ii) layer routing by restricting to $\ell \in \mathcal{L}(r_{i+1})$, and (iii) token selection by taking indices $S^{(i)}_{\ell,h} \subseteq \{1, \dots, T_i\}$. The transmitted cache is therefore:

$$C^{(i)}_{\text{send}} = \left\{ \left(K^{(i)}_{\ell,h,S^{(i)}_{\ell,h}}, V^{(i)}_{\ell,h,S^{(i)}_{\ell,h}} \right) \right\}_{\ell \in \mathcal{L}(r_{i+1}),\, h \in \mathcal{H}_{\text{pub}}}. \tag{17}$$

We update public memory by concatenation along the token dimension:

$$K^{\text{pub}}_{\ell,h} \leftarrow K^{\text{pub}}_{\ell,h} \oplus K^{(i)}_{\ell,h,S^{(i)}_{\ell,h}}, \qquad V^{\text{pub}}_{\ell,h} \leftarrow V^{\text{pub}}_{\ell,h} \oplus V^{(i)}_{\ell,h,S^{(i)}_{\ell,h}}. \tag{18}$$

Here $\oplus$ denotes concatenation on the first (token) dimension.

When running agent $i{+}1$, we inject $C_{\text{pub}}^{(i)}$ by extending the receiver's attention memory. Concretely, for each layer ℓ and head $h \in \mathcal{H}_{\text{pub}}$, we concatenate the public KV tensors to the receiver's per-layer past_key_values (or equivalently, to the layer's attention cache) so that the receiver can attend to them as additional "past" memory entries. Heads in $\mathcal{H}_{\text{pri}}$ are not extended.

To preserve the interface scoping gate, we ensure that private heads never attend to public memory entries. Practically, this can be enforced either structurally (by not concatenating public KV into private heads) or by masking (setting attention logits to $-\infty$ for memory positions on private heads).

Position-id bookkeeping. Most modern decoder-only LLMs use RoPE-like positional encodings, where the cached keys are already position-rotated. When transferring keys/values across agents, it is important that the receiver continues decoding with position IDs that are consistent with the injected cache. A simple, robust strategy is to maintain a global position counter $p_{\max}$ that tracks the maximum absolute position used by the public memory. When agent $i{+}1$ encodes its prompt tokens, we offset its `position_ids` to start from $p_{\max} + 1$, i.e.,

$$\texttt{position_ids} = (p_{\max} + 1) + [0, 1, \ldots, T_{i+1}^{\text{prompt}} - 1]. \tag{19}$$

This keeps RoPE rotations consistent without modifying the backbone. In our implementation, we also store (or reconstruct) the absolute positions of transmitted tokens so that the public memory can be appended with monotonically increasing positions across agents.

Communication Alignment. Our protocol relies on manipulating intermediate representations rather than only exchanging decoded tokens. Concretely, (i) the public memory $\mathcal{C}_{\text{pub}}$ is formed by concatenating selected KV slices along the token dimension (so the receiver can attend to them as if they were part of its own past), and (ii) our saliency estimator uses a small number of latent steps (parameter ϕ in Section 4) to estimate sender-side cache relevance as a proxy for the receiver. These probes still require sequential forward passes, but do not transmit an intermediate text trace. While (i) guarantees structural compatibility in homogeneous settings (same backbone implies matching (L, H, d_h)), (ii) introduces a subtler issue: a model's output-side representation (e.g., the final-layer hidden state used to produce logits) is generally not distributed like the model's input-side representation (token embeddings consumed at the first layer). Feeding an uncorrected hidden state back as a "pseudo-token embedding" can therefore push the model off the manifold it was trained on, leading to distribution drift across latent steps and unstable attention profiles (a failure mode also reported in latent-space reasoning/collaboration systems).

For latent rollouts, the fixed alignment map is computed from the pretrained input and output embedding tables (Zou et al., 2025).

Let $W_{\text{in}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the token embedding matrix (mapping a discrete token to a d-dimensional input embedding), and let $W_{\text{out}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the output projection (the LM head, mapping a d-dimensional hidden state to vocabulary logits).[2] Even when weight tying is used, W_{in} and W_{out} need not be perfect inverses, and the statistics of final-layer hidden states can differ substantially from those of input embeddings. We therefore seek a linear map $W_a \in \mathbb{R}^{d \times d}$ that brings output-side representations into better agreement with the input embedding geometry by aligning the two vocab-aligned matrices:

$$W_a \;=\; \arg\min_{W \in \mathbb{R}^{d \times d}} \|W_{\text{in}} - W_{\text{out}}\, W\|_F^2 \;+\; \alpha\, \|W\|_F^2\,, \tag{20}$$

where $\alpha > 0$ is a small ridge regularizer. This is a standard least-squares alignment between two embedding tables indexed by the same vocabulary, and it admits a closed-form solution:

$$W_a \;=\; \left(W_{\text{out}}^\top W_{\text{out}} + \alpha I\right)^{-1} W_{\text{out}}^\top W_{\text{in}}. \tag{21}$$

The $d \times d$ map is computed once per backbone.

During latent rollouts, we take the model's final hidden state $h_t \in \mathbb{R}^d$ (after the final normalization) and form a pseudo-embedding using the fixed alignment map:

$$\tilde{e}_{t+1} \;=\; h_t\, W_a. \tag{22}$$

We then append $\tilde{e}_{t+1}$ as the next "continuous token" for the subsequent forward step. Operationally, this means we update the model's KV cache exactly as in standard decoding, except that the next-step input is $\tilde{e}_{t+1}$ rather than an embedding lookup of a discrete token. The resulting attention weights are used for saliency estimation. The alignment map remains fixed during inference.

[2] Some implementations store the LM head as $W_{\text{out}}^\top \in \mathbb{R}^{d \times |\mathcal{V}|}$; in that case, transpose accordingly.

C.2 LATENT-THOUGHT GENERATION

We define the sender probe protocol using continuous feedback as in equation 22. For each non-final agent, ϕ caps the number of deterministic probe steps; the default is $\phi = 50$. The final hidden state after prompt prefill supplies the first pseudo-embedding; after each completed step, $h_t W_a$ supplies the next input. An argmax token from $h_t W_{\text{out}}^{\top}$ is used only to check the EOS stopping condition and is not fed back as a token embedding. Ordinary greedy token decoding is used for the final visible answer.

Let T_i^{prompt} be the number of prompt tokens for agent i, and let $n_i \le \phi$ count completed probe steps, including the step whose EOS check terminates the loop. The local query index set is

$$\mathcal{Q}_i = \{T_i^{\text{prompt}} + 1,\ T_i^{\text{prompt}} + 2,\ \ldots,\ T_i^{\text{prompt}} + n_i\}, \tag{23}$$

corresponding to the completed continuous probe positions. For each layer ℓ and head h, we collect $\alpha^{(i)}_{\ell,h,\tau\to t}$ for $\tau \in \mathcal{Q}_i$, assigning zero to key positions not yet available at query τ.

Given a candidate cache at (ℓ, h), we estimate token saliency by averaging attention mass from the latent queries:

$$u^{(i)}_{\ell,h,t} \ :=\ \frac{1}{n_i} \sum_{\tau \in \mathcal{Q}_i} \alpha^{(i)}_{\ell,h,\tau\to t}, \qquad t \in \{1, \ldots, T_i\}. \tag{24}$$

Let $\widehat{s}$ be its normalization over candidate positions as in Definition 3.1; this rescaling does not change the ranking. We select

$$S^{(i)}_{\ell,h} = \textsc{TopK}\big(\widehat{s}^{(i)}_{\ell,h,1:T_i},\ k_i\big). \tag{25}$$

These sender-side scores are proxies for receiver relevance, distinct from the reference receiver saliency p in Definition B.12. Lemma F.2 bounds the resulting selection gap by their ℓ_1 discrepancy.

Remark C.1. The EOS test is applied after a completed probe step, so $1 \le n_i \le \phi$ and the averaging denominator is the actual number of completed steps. The selected position count satisfies $1 \le k_i \le T_i$.

C.3 HEAD SELECTION

We rank heads by the magnitude of their residual-stream contribution on the calibration set (Michel et al., 2019; Voita et al., 2019).

Selection Protocol. Let $W_O^{(l,h)} \in \mathbb{R}^{d \times d_h}$ denote the output projection matrix for head h at layer l. For a given input x and token position t, the output contribution of this head to the residual stream is given by:

$$o_{l,h,t}(x) = W_O^{(l,h)} \text{Attention}(x)_{l,h,t}$$

We define the Global Importance Score $I_{l,h}$ for each head as the expected Euclidean norm of its contribution over a small calibration dataset $\mathcal{D}_{cal}$:

$$I_{l,h} = \mathbb{E}_{x \sim \mathcal{D}_{cal}} \left[\frac{1}{T} \sum_{t=1}^{T} \| o_{l,h,t}(x) \|_2 \right] \tag{26}$$

We compute the scores offline and select the highest-ranked heads:

$$\mathcal{H}_{\text{send}} = \text{Top-k}_{(l,h) \in \mathcal{L} \times \mathcal{H}}(I_{l,h}) \tag{27}$$

C.4 SEMANTIC PROFILING

Role profiling scores candidate layers by their representation similarity to the receiver role features.

Let $\mathbf{h}_\ell(x)$ denote the sender's activation at layer ℓ for input x, and $\mathbf{z}_r(x)$ be the receiver's shallow feature representation under role r (e.g., input embeddings). We define the layer-wise affinity score $s_{r,\ell}$ as the expected similarity over a held-out distribution $\mathcal{D}$:

$$s_{r,\ell} = \mathbb{E}_{x \sim \mathcal{D}} \left[\Phi\left(\mathbf{h}_\ell(x), \mathbf{z}_r(x)\right) \right], \tag{28}$$

where Φ is the representation-similarity measure, implemented using linear CKA (Kornblith et al., 2019).

The affinity profile is smoothed, and the highest-scoring layer indices are selected subject to the layer budget. Profiling uses approximately 100 generic unlabeled prompts and is performed once per backbone. Linear CKA is computed on centered feature matrices (Kornblith et al., 2019).

C.5 Budget, latency, and payload

Table 4: **Budget operating points** (Qwen3-4B). Per-benchmark and mean accuracy, payload (MiB/handoff), latency, and speedup vs. Text Communication; Δ (pp) vs. its 85.5 mean accuracy. The shaded row is the default configuration.

Configuration	Payload	ARC-C	GSM8K	HE+	Avg	Δ	Lat. (s)	Speedup
Text Comm. (reference)	<0.1	88.3	89.6	78.7	85.5	–	44.1	1.0×
Full-KV relay	90.0	93.9	90.6	81.1	88.5	+3.0	18.0	2.5×
GCS, $\rho = 0.5\%$	0.5	89.9	87.1	76.2	84.4	−1.1	11.9	3.7×
GCS, $\rho = 1\%$	0.9	90.9	88.0	78.0	85.6	+0.1	12.0	3.7×
GCS, $\rho = 5\%$	4.5	92.1	89.1	79.3	86.8	+1.3	12.2	3.6×
GCS, $\rho = 12\%$ (default)	10.6	92.9	89.8	80.5	87.7	+2.2	12.4	3.5×
GCS, $\rho = 25\%$	22.5	93.2	90.2	81.1	88.2	+2.6	13.4	3.3×
GCS, 12% + int8	5.3	92.5	89.4	79.9	87.3	+1.7	11.9	3.7×
GCS, 12% + int4	2.7	91.8	88.7	79.3	86.6	+1.1	11.4	3.9×

Table 5: Head- and layer-selection rule alternatives at the matched nominal 12% budget (three-task mean, Qwen3-4B); Δ vs. our rule; means over 8 selector seeds.

Rule	Acc. (%)	Δ
(i) Head-selection rule		
Random groups	85.6	-2.1
All heads, sparser positions	86.8	-1.0
Gradient probe	87.1	-0.6
Magnitude (ours)	**87.7**	**0.0**
(ii) Layer-selection rule		
Random layers	85.2	-2.5
Uniform depth	86.4	-1.3
Depth prior	86.9	-0.8
Role-conditioned (ours)	**87.7**	**0.0**

C.5.1 Latency breakdown

Table 6 splits the per-instance latency of Table 2 into the stages each protocol executes, measured on Qwen3-4B under the setup of §4.1. Selection—saliency scoring plus the top-k gather—costs 1.07 s, or 8.6% of the 12.43 s that our method spends end to end. Text Communication has no selection stage but pays 34.50 s to generate the intermediate traces that our probe steps replace. The column totals agree with Table 2 at its displayed precision. The 41.18 s of Text token generation comprises 34.50 s of intermediate traces and 6.68 s of final decoding; all protocol overhead is included.

Table 6: Per-instance latency by stage (s, Qwen3-4B, mean over the three benchmarks). "–": the protocol has no such stage. Totals match the Qwen3-4B rows of Table 2.

Stage	GCS	Text Comm.	No-Comm.
Prompt prefill	2.16	2.46	2.78
Latent probe steps (ϕ)	5.45	–	–
Saliency and top-k gather	1.07	–	–
Serialization and injection	0.71	0.43	–
Intermediate text generation	–	34.50	–
Final answer decoding	3.04	6.68	8.35
Total	**12.43**	**44.07**	**11.13**

C.5.2 Payload accounting

The Payload column of Table 2 and the horizontal axis of Figure 3 count the K/V tensor bytes newly exported at a handoff. This is the byte counterpart of $\mathrm{Cost}(g; x)$: previously exported entries are counted once. Position indices and quantization metadata are outside this tensor-storage measure; forwarding a cumulative cache would incur additional transport bytes. For a backbone with L layers, n_{kv} key/value heads and head dimension d_h, storing one token's keys and values in `fp16` costs

$$2 \cdot L \cdot n_{\text{kv}} \cdot d_h \cdot 2 \text{ bytes} = 2 \cdot 36 \cdot 8 \cdot 128 \cdot 2 = 147{,}456 \text{ bytes} = 144 \text{ KiB} \tag{29}$$

on Qwen3-4B (L=36, n_{kv}=8, d_h=128). At a mean sender cache length of $T \approx 640$ tokens the full cache occupies 90.0 MiB. The total retained fraction ρ combines head, layer, and position retention; k specifies only the position count for fixed heads and layers.

Our three modules cut this along independent axes, so the transmitted fraction is their product. The default Qwen3-4B configuration keeps 4 of 8 key/value head groups, 17 of 36 layers, and 320 of 640 token positions:

$$\rho = \frac{4}{8} \cdot \frac{17}{36} \cdot \frac{320}{640} \approx 0.1181, \tag{30}$$

that is 11.81% of the cache, or 10.63 MiB per hop — 8.47× less than a full-cache relay. Quantizing the selected keys and values divides their tensor-storage cost directly, with the selected entries unchanged:

Table 7: Transmitted payload per hop at the default budget (Qwen3-4B), for the three precisions reported in Figure 3.

Precision	Payload (MiB/hop)	vs. full cache
`fp16` (default)	10.63	8.5× less
`int8`	5.31	16.9× less
`int4`	2.66	33.9× less

The baseline rows of Table 2 follow the same accounting. A full-cache relay and C2C both transmit every entry, so their payload is the full cache; C2C's projector changes the representation, not the number of entries sent. KVComm keeps whole layers, so its payload is a multiple of $1/L$ and cannot land on 12% exactly; we round its layer count up rather than down, which gives it 12.5–14.3% of the cache across the listed backbones. On Qwen3-4B, KVComm retains $5/36$ of the cache (13.89%), compared with the 11.81% default GCS configuration.

D Proof of Theorem 3.3

It suffices to consider a discrete ternary subset of KV values, since a lower bound on this subset implies a lower bound for the full continuous space.

Definition D.1 (Ternary KV-slice message space)**.** Let $\mathcal{A} := \{-1, 0, +1\}$. Fix a head dimension $d_h \in \mathbb{N}$. Define the *single-position KV symbol alphabet*

$$\mathcal{Q} := \mathcal{A}^{2d_h} = \{(k, v) : k \in \mathcal{A}^{d_h},\ v \in \mathcal{A}^{d_h}\}. \tag{31}$$

A length-m ternary KV-slice is any sequence of m such KV pairs:

$$\mathcal{M}_m := \mathcal{Q}^m. \tag{32}$$

We will view an element $M \in \mathcal{M}_m$ as $M = ((k_1, v_1), \ldots, (k_m, v_m))$ with each $k_t, v_t \in \mathcal{A}^{d_h}$.

Definition D.2 (Text message space with a token budget)**.** Let $\mathcal{V}$ be a finite vocabulary (token alphabet), $|\mathcal{V}| > 1$. For an integer $m' \geq 0$, define the *length-m' text message space* as

$$\mathcal{T}_{m'} := \mathcal{V}^{m'}. \tag{33}$$

(Allowing variable length $\leq m'$ only increases the size by at most a constant factor; using fixed length keeps the counting exact. One can always pad with a dedicated $\langle \mathrm{PAD} \rangle \in \mathcal{V}$ token.)

Definition D.3 (Text-only protocol and lossless simulation of KV-slices)**.** A *text-only communication protocol* with token budget m' for transmitting messages in $\mathcal{M}_m$ is a pair of functions

$$\mathrm{Enc} : \mathcal{M}_m \to \mathcal{T}_{m'}, \qquad \mathrm{Dec} : \mathcal{T}_{m'} \to \mathcal{M}_m. \tag{34}$$

We say the protocol *losslessly simulates* $\mathcal{M}_m$ if

$$\mathrm{Dec}(\mathrm{Enc}(M)) = M \qquad \text{for all } M \in \mathcal{M}_m. \tag{35}$$

Equivalently, Enc is injective on $\mathcal{M}_m$.

Definition D.4 (cache identification)**.** Fix m and define a distribution $\mathcal{D}_m$ as follows: sample $M \sim \mathrm{Unif}(\mathcal{M}_m)$, and set the target label to be $Y^\star := M$. The task is: a sender observes M and communicates with a receiver, after which the receiver outputs $\widehat{M}$. We evaluate with 0–1 loss by

$$\ell_{\mathrm{id}}(\widehat{M}, M) := \mathbf{1}\{\widehat{M} \neq M\}. \tag{36}$$

Lemma D.5 (Cardinality of ternary KV-slices)**.** *With the notation in Definition D.1,*

$$|\mathcal{M}_m| = 3^{2d_h m}.$$

Proof. By definition, $|\mathcal{A}| = 3$ and $\mathcal{Q} = \mathcal{A}^{2d_h}$, hence $|\mathcal{Q}| = 3^{2d_h}$. Then $\mathcal{M}_m = \mathcal{Q}^m$ implies $|\mathcal{M}_m| = |\mathcal{Q}|^m = (3^{2d_h})^m = 3^{2d_h m}$. □

Lemma D.6 (Text length lower bound for lossless simulation)**.** *If there exists a lossless text-only protocol (Definition D.3) transmitting $\mathcal{M}_m$ with budget m', then $|\mathcal{T}_{m'}| \geq |\mathcal{M}_m|$. Equivalently,*

$$|\mathcal{V}|^{m'} \geq 3^{2d_h m}, \qquad \textit{and hence} \qquad m' \geq \frac{2d_h m \log 3}{\log |\mathcal{V}|}.$$

Proof. Lossless simulation means $\mathrm{Enc} : \mathcal{M}_m \to \mathcal{T}_{m'}$ is injective. An injective map from a finite set of size $|\mathcal{M}_m|$ into another finite set requires the codomain to have size at least $|\mathcal{M}_m|$, i.e., $|\mathcal{T}_{m'}| \geq |\mathcal{M}_m|$.

By using Definition D.2, we have $|\mathcal{T}_{m'}| = |\mathcal{V}|^{m'}$. By Lemma D.5, we got $|\mathcal{M}_m| = 3^{2d_h m}$. Thus $|\mathcal{V}|^{m'} \geq 3^{2d_h m}$.

Taking logarithms in any consistent base yields $m' \log |\mathcal{V}| \geq 2d_h m \log 3$, hence $m' \geq \frac{2d_h m \log 3}{\log |\mathcal{V}|}$. □

Lemma D.7 (Unavoidable identification error under a token budget)**.** *Consider the identification task in Definition D.4 with $M \sim \mathrm{Unif}(\mathcal{M}_m)$. For any text-only protocol* (Enc, Dec) *with budget m' (Definition D.3),*

$$\mathbb{P}\big[\mathrm{Dec}(\mathrm{Enc}(M)) = M\big] \leq \frac{|\mathcal{T}_{m'}|}{|\mathcal{M}_m|} = \frac{|\mathcal{V}|^{m'}}{3^{2d_h m}}.$$

Equivalently, its expected 0–1 *loss satisfies*

$$\mathbb{E}\big[\ell_{\mathrm{id}}(\mathrm{Dec}(\mathrm{Enc}(M)), M)\big] \geq 1 - \frac{|\mathcal{V}|^{m'}}{3^{2d_h m}}.$$

Proof. Define the *correctly decoded set*

$$\mathcal{C} := \{M \in \mathcal{M}_m : \mathrm{Dec}(\mathrm{Enc}(M)) = M\}.$$

We claim $|\mathcal{C}| \le |\mathcal{T}_{m'}|$. Indeed, for each text message $t \in \mathcal{T}_{m'}$, the decoder output $\mathrm{Dec}(t)$ is a single element of $\mathcal{M}_m$. Therefore, among all M that satisfy $\mathrm{Enc}(M) = t$, at most one can be correctly decoded, namely $M = \mathrm{Dec}(t)$. Hence each t can contribute at most one element to $\mathcal{C}$, so $|\mathcal{C}| \le |\mathcal{T}_{m'}|$.

Since M is uniform over $\mathcal{M}_m$,

$$\mathbb{P}[\mathrm{Dec}(\mathrm{Enc}(M)) = M] = \frac{|\mathcal{C}|}{|\mathcal{M}_m|} \le \frac{|\mathcal{T}_{m'}|}{|\mathcal{M}_m|}.$$

Substituting $|\mathcal{T}_{m'}| = |\mathcal{V}|^{m'}$ and $|\mathcal{M}_m| = 3^{2d_h m}$ (Lemma D.5) gives the stated bound. The expected 0–1 loss is $1 - \mathbb{P}[\text{correct}]$. □

Theorem D.8. *Let $\mathcal{V}$ be the (finite) text vocabulary with $|\mathcal{V}| > 1$. For any integer $m \ge 1$ with fixed r_2, $\ell \in \mathcal{L}(r_2), d_h, h \in \mathcal{H}_{pub}$, consider the family $\mathcal{M}_m$ of length-m* ternary *KV-slices at block (ℓ, h) (Definition D.1), i.e., cache slices consisting of m key–value pairs whose coordinates lie in $\{-1, 0, +1\}$, we have*

(i) Text length lower bound. *If a text-only communication protocol of length m' tokens* losslessly simulates *every message in $\mathcal{M}_m$ (Definition D.3), then necessarily*

$$m' \ge \frac{2d_h\, m\, \log 3}{\log |\mathcal{V}|}. \tag{37}$$

(ii) Strict domination under a token budget. *If m' violates equation 37, then there exists a two-agent collaboration task distribution (Definition D.4) for which a KV-cache broadcast communicator that transmits those m KV pairs (and thus has scalar communication cost $2d_h m$ under the cost definition in Section 3.4) achieves zero task loss, whereas* every *text-based protocol restricted to m' tokens incurs expected* 0–1 *loss at least*

$$1 - \frac{|\mathcal{V}|^{m'}}{3^{2d_h m}} > 0. \tag{38}$$

Consequently, in this regime there exists a cache-based algorithm strictly better than token-limited text transmission; hence text communication is not universally optimal.

Proof. Building upon the theoretical foundation of latent collaboration established by Zou et al. (2025), we extend the analysis to constrained communication budgets. We first assume there exists a text-only protocol of length m' that losslessly simulates $\mathcal{M}_m$ in the sense of Definition D.3. By Lemma D.6, we must have

$$m' \ge \frac{2d_h m \log 3}{\log |\mathcal{V}|}.$$

This is exactly equation 37.

Suppose m' violates equation 37. Then, equivalently, $|\mathcal{V}|^{m'} < 3^{2d_h m} = |\mathcal{M}_m|$.

We consider the identification task $\mathcal{D}_m$ from Definition D.4. A KV-cache broadcast communicator can achieve *zero* task loss on $\mathcal{D}_m$ by transmitting the message M itself as the public-memory KV-slice at block (ℓ, h). In the terminology of Section 3.4, this corresponds to choosing a communicator g_{KV} that sends exactly the m token positions of that slice, so its scalar communication cost is

$$\mathrm{Cost}(g_{\mathrm{KV}}; x) = (2d_h) \cdot m,$$

matching the paper's cost definition (each position transmits one key and one value of dimension d_h). Because the receiver can read the transmitted KV-slice, it can output $\widehat{M} = M$, achieving $\ell_{\mathrm{id}}(\widehat{M}, M) = 0$ almost surely.

On the other hand, take any text-only protocol $(\mathrm{Enc}, \mathrm{Dec})$ with budget m'. Lemma D.7 gives

$$\mathbb{E}\big[\ell_{\mathrm{id}}(\mathrm{Dec}(\mathrm{Enc}(M)), M)\big] \ge 1 - \frac{|\mathcal{V}|^{m'}}{3^{2d_h m}}.$$

Since $|\mathcal{V}|^{m'} < 3^{2d_h m}$ in the current regime, the right-hand side is strictly positive, proving equation 38. This yields a strict separation: cache broadcast attains zero loss while any m'-token text protocol suffers nonzero expected loss on $\mathcal{D}_m$. □

E PROOF OF THEOREM 3.5

We first show and prove some important results.

Lemma E.1 (Head–tail identity for truncation+renormalization)**.** *Fix any $T \geq 1$, any probability mass function P on $\{1, \ldots, T\}$, any nonempty set $S \subseteq \{1, \ldots, T\}$ with $P(S) > 0$, and any vectors $\{v_t\}_{t=1}^{T} \subset \mathbb{R}^d$.*

Let the tail mass be $\eta := P(S^c) = \sum_{t \notin S} P(t)$, so that $P(S) = 1 - \eta$. Define the truncated-renormalized distribution $\widehat{P}$ by

$$\widehat{P}(t) := \begin{cases} \dfrac{P(t)}{P(S)} & t \in S, \\ 0 & t \notin S. \end{cases} \tag{39}$$

We define the full and gated expectations and conditional means,

$$a := \sum_{t=1}^{T} P(t) v_t, \qquad \widehat{a} := \sum_{t=1}^{T} \widehat{P}(t) v_t = \sum_{t \in S} \frac{P(t)}{P(S)} v_t \tag{40}$$

$$\mu_{\text{head}} := \mathbb{E}[v_t \mid t \in S] = \sum_{t \in S} \frac{P(t)}{P(S)} v_t, \qquad \mu_{\text{tail}} := \mathbb{E}[v_t \mid t \in S^c] = \begin{cases} \sum_{t \notin S} \frac{P(t)}{\eta} v_t & \eta > 0, \\ 0 & \eta = 0. \end{cases} \tag{41}$$

Then the following identity holds:

$$a - \widehat{a} = \eta\big(\mu_{\text{tail}} - \mu_{\text{head}}\big). \tag{42}$$

Proof. We use the head–tail decomposition (Tzachristas et al., 2025) and give its algebraic derivation for a general finite memory.

We first expand a by splitting the sum over S and S^c:

$$a = \sum_{t \in S} P(t) v_t + \sum_{t \notin S} P(t) v_t. \tag{43}$$

Rewrite each term as "(total mass) $\times$ (conditional mean)":

$$\sum_{t \in S} P(t) v_t = P(S) \sum_{t \in S} \frac{P(t)}{P(S)} v_t = (1 - \eta) \mu_{\text{head}}, \tag{44}$$

and if $\eta > 0$,

$$\sum_{t \notin S} P(t) v_t = \eta \sum_{t \notin S} \frac{P(t)}{\eta} v_t = \eta \mu_{\text{tail}}. \tag{45}$$

(If $\eta = 0$ then $P(S^c) = 0$ and the tail sum is 0; our convention $\mu_{\text{tail}} = 0$ makes the same equality hold.) Therefore,

$$a = (1 - \eta) \mu_{\text{head}} + \eta \mu_{\text{tail}}. \tag{46}$$

By definition of $\widehat{P}$, we also have

$$\widehat{a} = \sum_{t \in S} \frac{P(t)}{P(S)} v_t = \mu_{\text{head}}. \tag{47}$$

Subtracting yields

$$a - \widehat{a} = \big((1 - \eta) \mu_{\text{head}} + \eta \mu_{\text{tail}}\big) - \mu_{\text{head}} = \eta(\mu_{\text{tail}} - \mu_{\text{head}}). \tag{48}$$

□

This identity isolates the contribution of the discarded mass.

Lemma E.2 (Per-query output error bound in terms of tail mass)**.** *Under the setting of Lemma E.1, assume additionally that* $\|v_t\|_2 \le \xi$ *for all* t*. Then*

$$\|a - \widehat{a}\|_2 \le 2\xi\,\eta, \qquad \text{where} \quad \eta = \sum_{t \notin S} P(t). \tag{49}$$

Proof. By Lemma E.1,

$$\|a - \widehat{a}\|_2 = \eta \|\mu_{\text{tail}} - \mu_{\text{head}}\|_2 \le \eta\big(\|\mu_{\text{tail}}\|_2 + \|\mu_{\text{head}}\|_2\big). \tag{50}$$

We bound $\|\mu_{\text{head}}\|_2$ and $\|\mu_{\text{tail}}\|_2$ separately. Since each is a convex combination of $\{v_t\}$, we have (by triangle inequality / Jensen)

$$\|\mu_{\text{head}}\|_2 = \left\|\sum_{t \in S} \frac{P(t)}{P(S)} v_t\right\|_2 \le \sum_{t \in S} \frac{P(t)}{P(S)} \|v_t\|_2 \le \sum_{t \in S} \frac{P(t)}{P(S)} \xi = \xi. \tag{51}$$

Similarly, if $\eta > 0$,

$$\|\mu_{\text{tail}}\|_2 = \left\|\sum_{t \notin S} \frac{P(t)}{\eta} v_t\right\|_2 \le \sum_{t \notin S} \frac{P(t)}{\eta} \|v_t\|_2 \le \xi, \tag{52}$$

and if $\eta = 0$ then $\mu_{\text{tail}} = 0$ so the bound still holds. Plugging back gives

$$\|a - \widehat{a}\|_2 \le \eta(\xi + \xi) = 2\xi\eta. \tag{53}$$

□

We next translate the identity into an ℓ_2 perturbation bound using bounded values.

Lemma E.3 (Blockwise average readout perturbation is controlled by average tail mass)**.** *For any fixed* (i, ℓ, h) *and input* x*,*

$$\Delta^{(i)}_{\ell,h}(x) \le 2\xi\, \varepsilon^{(i)}_{\ell,h}(x), \tag{54}$$

where $\Delta^{(i)}_{\ell,h}(x)$ *is from Definition B.11 and* $\varepsilon^{(i)}_{\ell,h}(x)$ *is from Definition B.10.*

Proof. If $Q^{(\ell,h)}_{i+1}(x) = \emptyset$, both sides are zero. Otherwise, fix a query τ and use the full memory $\Omega = \Omega_{i,\ell,h,\tau}(x)$ and retained set $R = \Omega \setminus (J_i \setminus S^{(i)}_{\ell,h}(x))$ from Definition B.9. The removed mass under joint attention is

$$\eta_{i,\ell,h,\tau}(x) = \sum_{t \in J_i \setminus S^{(i)}_{\ell,h}(x)} P_{i,\ell,h,\tau \to t}(x) \tag{55}$$

$$= \beta_{i,\ell,h,\tau}(x)\, \delta^{(i)}_{\ell,h}(\tau; x). \tag{56}$$

Apply Lemma E.2 to the distribution P on Ω, the values $w_{i,\ell,h,\tau,t}(x)$, and the retained set R. Assumption B.1 bounds all these values, and Definition B.9 ensures $P(R) > 0$. Hence

$$\big\|a^{\text{full}}_{i,\ell,h,\tau}(x) - a^{\text{gate}}_{i,\ell,h,\tau}(x)\big\|_2 \le 2\xi\, \eta_{i,\ell,h,\tau}(x) \tag{57}$$

$$= 2\xi\, \beta_{i,\ell,h,\tau}(x)\, \delta^{(i)}_{\ell,h}(\tau; x) \tag{58}$$

$$\le 2\xi\, \delta^{(i)}_{\ell,h}(\tau; x), \tag{59}$$

where the last inequality uses $0 < \beta \le 1$. Averaging over the reference queries gives

$$\Delta^{(i)}_{\ell,h}(x) \le \frac{2\xi}{|Q^{(\ell,h)}_{i+1}(x)|} \sum_{\tau \in Q^{(\ell,h)}_{i+1}(x)} \delta^{(i)}_{\ell,h}(\tau; x) = 2\xi\, \varepsilon^{(i)}_{\ell,h}(x). \tag{60}$$

□

The local readout bound extends to surrogate loss conditionally via Assumptions B.3 and B.4.

Here we first restate Theorem 3.5.

Theorem E.4 (Conditional tail-mass bound on excess surrogate loss)**.** *Under Assumptions B.1, B.2, B.3, and B.4, the excess loss in Definition B.5 satisfies*

$$\Delta_{\text{task}}(g) \le 2L_{\text{task}}\xi \sum_{i=1}^{N-1} \sum_{\ell\in\mathcal{L}(r_{i+1})} \sum_{h\in\mathcal{H}_{pub}} \Gamma_{\ell,h}^{(i)} \cdot \mathbb{E}_{x\sim\mathcal{D}}\big[\varepsilon_{\ell,h}^{(i)}(x)\big], \tag{61}$$

where $\varepsilon_{\ell,h}^{(i)}(x)$ is the average tail mass from Definition B.10.

Proof of Theorem E.4. By Definition B.5 and Assumption B.3, we first have

$$\begin{aligned}\Delta_{\text{task}}(g) &= \mathbb{E}\big[\ell_{\text{task}}(z(x;g),y^\star) - \ell_{\text{task}}(z(x;g_{\text{full}}),y^\star)\big] &(62)\\ &\le \mathbb{E}\big[|\ell_{\text{task}}(z(x;g),y^\star) - \ell_{\text{task}}(z(x;g_{\text{full}}),y^\star)|\big] &(63)\\ &\le L_{\text{task}}\,\mathbb{E}\|z(x;g) - z(x;g_{\text{full}})\|_2. &(64)\end{aligned}$$

Applying Assumption B.4 yields

$$\|z(x;g) - z(x;g_{\text{full}})\|_2 \le \sum_{i=1}^{N-1} \sum_{\ell\in\mathcal{L}(r_{i+1})} \sum_{h\in\mathcal{H}_{\text{pub}}} \Gamma_{\ell,h}^{(i)}\,\Delta_{\ell,h}^{(i)}(x). \tag{65}$$

Then by Lemma E.3, for each block (i,ℓ,h) we have

$$\Delta_{\ell,h}^{(i)}(x) \le 2\xi\,\varepsilon_{\ell,h}^{(i)}(x). \tag{66}$$

Combining the above inequalities and taking expectation over $x\sim\mathcal{D}$ gives

$$\Delta_{\text{task}}(g) \le 2L_{\text{task}}\xi \sum_{i=1}^{N-1} \sum_{\ell\in\mathcal{L}(r_{i+1})} \sum_{h\in\mathcal{H}_{\text{pub}}} \Gamma_{\ell,h}^{(i)}\,\mathbb{E}_{x\sim\mathcal{D}}[\varepsilon_{\ell,h}^{(i)}(x)]. \tag{67}$$

□

F Proof of Theorem 3.6

We restate Theorem 3.6 here.

Theorem F.1 (Top-k saliency minimizes average tail mass)**.** *Fix (i,ℓ,h,x), the reference receiver queries, and a budget $1\le k\le T_i$. Let $p_{\ell,h,t}^{(i)}(x)$ be the receiver saliency scores in Definition B.12. Under the budget constraint $|S|=k$, any set consisting of k indices attaining the largest k entries of $\{p_{\ell,h,t}^{(i)}(x)\}_{t=1}^{T_i}$ minimizes the tail mass objective in Definition B.13:*

$$S^\star = \text{Top-}k\Big(p_{\ell,h,1:T_i}^{(i)}(x)\Big) \quad\Longrightarrow\quad S^\star \in \arg\min_{S\subseteq[T_i],\,|S|=k} \varepsilon_{\ell,h}^{(i)}(x;S),$$

with arbitrary tie-breaking.

Proof of Theorem F.1. Fix (i,ℓ,h,x), and abbreviate $p_t := p_{\ell,h,t}^{(i)}(x)$. If the reference query set is empty, all scores and tail masses are zero, so every k-subset is optimal. Otherwise the scores sum to one.

By Definition B.13, for any $S\subseteq[T_i]$ with $|S|=k$,

$$\varepsilon_{\ell,h}^{(i)}(x;S) = \sum_{t\notin S} p_t. \tag{68}$$

When $Q_{i+1}^{(\ell,h)}(x)\neq\emptyset$, Definition B.12 gives $\sum_{t=1}^{T_i} p_t = 1$, hence

$$\varepsilon_{\ell,h}^{(i)}(x;S) = 1 - \sum_{t\in S} p_t. \tag{69}$$

Therefore minimizing $\varepsilon^{(i)}_{\ell,h}(x;S)$ over all k-subsets is equivalent to maximizing $\sum_{t\in S} p_t$ over all k-subsets:

$$\arg\min_{S:|S|=k} \varepsilon^{(i)}_{\ell,h}(x;S) = \arg\max_{S:|S|=k} \sum_{t\in S} p_t. \tag{70}$$

Let $p_{(1)} \geq p_{(2)} \geq \cdots \geq p_{(T_i)}$ be the saliency values sorted in nonincreasing order (breaking ties arbitrarily but consistently), and let

$$S^\star := \{(1), (2), \ldots, (k)\} \tag{71}$$

be a set of indices of the k largest entries.

We prove $S^\star$ is optimal via an exchange argument. Take any $S \subseteq [T_i]$ with $|S| = k$. If $S = S^\star$, it is already optimal. Otherwise there exists $u \in S \setminus S^\star$ and $v \in S^\star \setminus S$. By construction of $S^\star$, $p_v \geq p_u$. Define the swapped set $S' := (S \setminus \{u\}) \cup \{v\}$, which still satisfies $|S'| = k$. Then

$$\sum_{t\in S'} p_t = \sum_{t\in S} p_t - p_u + p_v \geq \sum_{t\in S} p_t. \tag{72}$$

So the swap never decreases the maximization objective. Repeating this swap finitely many times yields $S^\star$ while monotonically increasing (or preserving) $\sum_{t\in S} p_t$. Hence $S^\star$ maximizes $\sum_{t\in S} p_t$, and therefore minimizes $\varepsilon^{(i)}_{\ell,h}(x;S)$.

This proves that any Top-k set of indices of $\{p_t\}$ is optimal (with arbitrary tie-breaking). □

Lemma F.2 (Selection gap from proxy saliency)**.** *Fix the candidate positions, reference receiver queries, head and layer gates, and budget $1 \leq k \leq T_i$. Abbreviate the receiver scores in Definition B.12 as p and the sender proxy in Definition 3.1 as $\widehat{s}$. Let $S^\star = \text{Top-}k(p)$ and $\widehat{S} = \text{Top-}k(\widehat{s})$. Then*

$$0 \leq \varepsilon^{(i)}_{\ell,h}(x;\widehat{S}) - \varepsilon^{(i)}_{\ell,h}(x;S^\star) \leq \|p - \widehat{s}\|_1. \tag{73}$$

Proof. Suppress the fixed block indices, and write $q(A) = \sum_{t\in A} q_t$ for any score vector q. Put $A = S^\star \setminus \widehat{S}$ and $B = \widehat{S} \setminus S^\star$. Proxy Top-$k$ optimality gives $\widehat{s}(A) \leq \widehat{s}(B)$, so

$$\begin{aligned}
\varepsilon(x;\widehat{S}) - \varepsilon(x;S^\star) &= p(A) - p(B) && (74)\\
&= (p-\widehat{s})(A) - (p-\widehat{s})(B) + \widehat{s}(A) - \widehat{s}(B) && (75)\\
&\leq \sum_{t\in A\cup B} |p_t - \widehat{s}_t| \leq \|p - \widehat{s}\|_1. && (76)
\end{aligned}$$

The lower bound follows from Theorem F.1. The argument also applies to zero score vectors and arbitrary tie-breaking. □

Combining Lemma F.2 with Theorem E.4 gives, for the communicator $\widehat{g}$ using the proxy selections and satisfying the same sensitivity assumption,

$$\Delta_{\text{task}}(\widehat{g}) \leq 2L_{\text{task}}\xi \sum_b \Gamma_b\, \mathbb{E}_x[\varepsilon_b(x;S^\star_b) + \|p_b(x) - \widehat{s}_b(x)\|_1], \tag{77}$$

where $b = (i,\ell,h)$ ranges over the sending steps and allowed blocks. The discrepancy term is not assumed to be small. These results concern fixed per-block budgets and gates; they do not establish optimal allocation across heads, layers, or blocks.