Vision Wormhole: LATENT-SPACE COMMUNICATION IN HETEROGENEOUS MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterogeneous multi-agent systems combine models with different capabilities through a common communication interface. Exchanging internal states directly requires translating between model-specific representations and controlling intermediate computation. We introduce the **Vision Wormhole**, which repurposes the visual input interface of Vision-Language Models (VLMs) for continuous communication between frozen heterogeneous agents. A Universal Visual Codec encodes each sender's latent rollout into a fixed-size message, maps it through a shared reference space, and decodes received messages into the receiver's image-token span. Per-model codecs and affine reference maps form a hub-and-spoke architecture with $O(N)$ components for N models. Each model learns its codec independently through self-distillation on anchor texts, and shared-anchor alignment enables reuse across communication partners. Across four VLM families, six team configurations, and nine reasoning benchmarks, Vision Wormhole improves accuracy by 6.0 percentage points on average over text-mediated MAS and achieves a 1.69× geometric-mean speedup in batch-normalized end-to-end runtime.

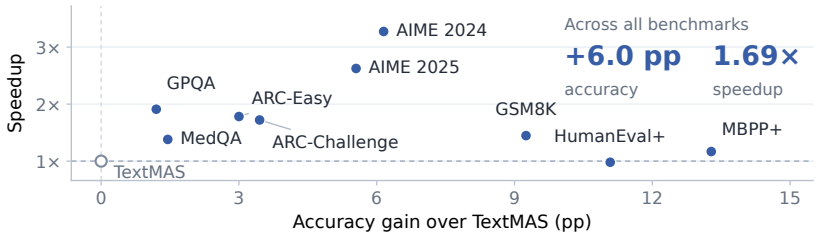

Figure 1: **Joint accuracy and speed gains.** Each point averages all six configurations for one benchmark; speedup uses the geometric mean of TextMAS/VW runtime ratios.

1 INTRODUCTION

Multi-agent systems (MAS) offer a way to bring the capabilities of several pretrained models into a single problem-solving process. Agents plan, critique, and refine each other's work, making the output of one stage the context for the next (Guo et al., 2024; Wu et al., 2024; Hong et al., 2024; Li et al., 2023a; Zhang et al., 2024b). Model diversity gives this composition practical value: different models excel at different agent functions and task domains, and assigning heterogeneous models to the same workflow improves performance (Ye et al., 2025b). Gains from aggregating diverse model responses and heterogeneous debate further demonstrate the value of combining model families (Wang et al., 2024a; Zhang et al., 2025a). The opportunity is to make each model's intermediate computation available to the rest of the team.

Communication determines how that computation reaches another model. In text-mediated teams, each handoff generates a response token by token, which subsequent agents process as additional context (Wu et al., 2024; Wang et al., 2024a). Repeating this cycle across roles adds generation and context-processing costs. Latent communication exposes a direct route through hidden activations and attention caches (Ramesh & Li, 2025; Zou et al., 2025; Fu et al., 2025). For heterogeneous teams,

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.

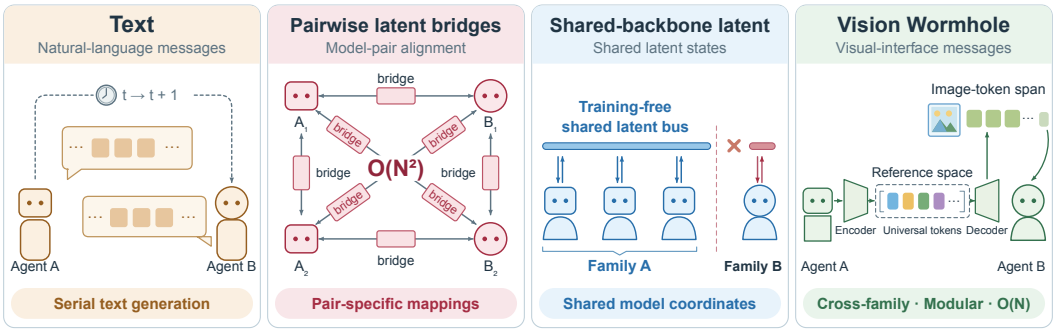

Figure 2: **The Vision Wormhole.** The visual input interface provides a destination for continuous messages between heterogeneous VLM agents. Per-model codecs connect the sender and receiver through a shared reference space.

embedding transformations and cache translation provide ways to connect independently pretrained models (Du et al., 2026; Fu et al., 2025). We build on this direction with a concrete goal: *let a frozen model use another model's internal states as context through an interface that can be reused across model families.* Realizing this goal brings together three challenges.

Representation compatibility. An internal state acquires its meaning through the model that produced it. Across model families, its width, feature coordinates, and role in the computation can differ. Cache translation addresses layer and key–value layouts (Fu et al., 2025; Chen et al., 2026c). Usable communication therefore requires adapting a message to both the structure and the conditioning behavior of its receiver. The choice of where a message enters the receiver is central to this adaptation.

Scalable model integration. Learned bridges (Fu et al., 2025; Du et al., 2026; Yang et al., 2026) usually train mappings for specific sender–receiver pairs. Supporting every directed pair in a pool of N models then entails $N(N-1)$ bridges, tying adaptation work to the set of communication partners. Shared-space approaches make model interfaces reusable through per-model mappings and interaction modules (Fein-Ashley et al., 2025). For heterogeneous teams, this modularity is valuable because a model can serve several roles and communicate with several partners. The challenge is to retain receiver-specific adaptation within a common message format.

Supervision for communication. Continuous messages need a learning signal that connects their contents to the receiver's behavior. Existing approaches obtain this signal from receiver-side reconstruction and generation objectives (Chen et al., 2026c) or optimize multi-agent interaction trajectories (Yu et al., 2026b). A reusable interface has two learning requirements: each module must learn to encode and consume context within its own frozen model, and the message spaces learned by separate modules must support exchange across models. Meeting both requirements calls for local conditioning targets and a common source of cross-model correspondences.

We propose the **Vision Wormhole** (VW), which turns the visual input interface of vision-language models (VLMs) into a communication channel between heterogeneous agents (Figure 2). The key insight is that multimodal pretraining already equips a language backbone to read continuous features supplied by a visual encoder (Alayrac et al., 2022; Li et al., 2023b). Visual conditioning also supports language-rich content, including rendered reasoning traces, documents, and code (Wang et al., 2026b; Wei et al., 2025; Shi et al., 2026b). We repurpose this capability for model-to-model communication: *a learned codec presents one model's internal states as visual context to another frozen model.* The visual pathway supplies a pretrained receiver interface; the codec learns how to express a latent message through it.

Each backbone receives a *Universal Visual Codec* with an encoder and a decoder. An intermediate agent produces a latent rollout by feeding its hidden states back as continuous inputs. The encoder converts this rollout into fixed-size message tokens, and the receiver's decoder expresses incoming messages in its image-token span. This context conditions the frozen receiver's subsequent computation. Intermediate agents exchange these continuous messages, and the final agent generates the answer in text.

To make the codecs reusable across partners, we connect their message spaces through a shared reference space. Per-model affine maps translate messages into this space for exchange and aggregation, then back into a receiver's codec coordinates for decoding. The resulting hub-and-spoke architecture has $O(N)$ components: one codec per model and two maps per non-reference model. Each backbone's visual decoder serves all of its communication partners.

We train this interface in two stages using anchor texts. First, a frozen backbone reads an anchor as text and supplies its prompt-boundary state and next-token distribution as self-distillation targets. Its codec encodes the associated latent rollout and decodes it into visual context. We train the codec so that the same backbone, conditioned on this visual context, reproduces the text-conditioned targets. Each model thereby supplies the supervision for its own encoder and decoder. Second, we give every model the same anchor texts and encode the resulting rollouts to obtain corresponding messages. We fit the affine reference maps to these messages by ridge regression. This separates local codec training from fitting the cross-model correspondences, allowing independently trained codecs to be composed through the reference space. The backbone parameters remain fixed throughout both stages.

Across four VLM families, six team configurations, and nine benchmarks, VW improves accuracy by **6.0 percentage points on average** over text-mediated MAS and achieves a **1.69× geometric-mean speedup** in batch-normalized end-to-end runtime (Figure 1). Both summaries weight all 54 configuration–task pairs equally. The evaluation spans mathematical and scientific reasoning, commonsense reasoning, and code generation, using the same model assignments and role sequence for the two communication channels. These results establish the visual interface as a practical route to composing frozen heterogeneous models through continuous messages.

Contributions. Our contributions are fourfold:

- **Visual conditioning for latent communication.** We introduce a codec that makes sender-side latent rollouts available as continuous visual context to a frozen heterogeneous receiver.
- **Reusable interfaces with $O(N)$ components.** We compose per-model codecs through affine reference maps, allowing the same encoder and decoder to serve multiple communication partners.
- **Independent codec training from text.** We develop a two-stage recipe that uses local self-distillation to learn visual conditioning and shared anchor texts to align the message spaces.
- **Joint accuracy and runtime gains.** Across four VLM families and nine benchmarks, VW improves accuracy by 6.0 percentage points on average and achieves a 1.69× overall speedup over text-mediated MAS.

2 Related Work

Multi-agent coordination. LLM-based MAS combine role specialization, planning, memory, and tool use through orchestration frameworks and collaborative workflows (Guo et al., 2024; Wu et al., 2024; Hong et al., 2024; Li et al., 2023a; Zhang et al., 2024b; Zhao et al., 2025b; Zhou et al., 2025; Zhang et al., 2025c). Model selection adds another design axis: function–domain evaluations guide heterogeneous role assignments, and layered response aggregation composes several models' outputs (Ye et al., 2025b; Wang et al., 2024a). Memory modules retain information across interactions, and workflow optimization allocates computation among successive agent calls (Packer et al., 2023; Zhong et al., 2024; Hu et al., 2025a; Chen et al., 2025; Zhuge et al., 2024). Communication design specifies the participants, message content, timing, and purpose of each exchange (Chen et al., 2026a). Analyses of agent systems study information sharing, coordination, and specialization as factors in task performance (Pezeshkpour et al., 2024; Cemri et al., 2025). Our main comparison evaluates text and visual communication within the same workflow and model assignments.

Latent communication and reasoning. Continuous channels expose several forms of intermediate computation, with corresponding choices of alignment and receiver-side fusion (Liu, 2026). Probability-weighted vocabulary embeddings encode next-token distributions (Pham et al., 2024); activation and state-trajectory interfaces communicate residual-stream vectors or token-wise changes in hidden states (Ramesh & Li, 2025; Tang et al., 2025). Cache-based collaboration exchanges attention state across agents, with selective layer sharing and reusable latent operations for planning, review, and voting (Zou et al., 2025; Shi et al., 2026a; Jin et al., 2026). Execution-oriented approaches reuse caches across contexts and selectively recompute positions affected by prefix changes (Ye et al.,

2025a; Geng et al., 2026). Related work also studies message compression and efficient transport for latent communication (Kriuk & Ng, 2025; Li et al., 2026; Lee et al., 2026; Dai et al., 2026).

Heterogeneous communication learns transformations into a receiver's input embeddings or attention cache (Du et al., 2026; Yang & Eshraghian, 2025; Fu et al., 2025; Rossi et al., 2026; Chen et al., 2026c). Shared representation spaces support composition through per-model mappings, cross-attention, or routed expert interactions (Bansal et al., 2024; Fein-Ashley et al., 2025; Zheng et al., 2025). Training objectives connect these interfaces to receiver behavior through cache reconstruction, generation supervision, or joint optimization over multi-agent trajectories (Chen et al., 2026c; Yu et al., 2026b). Continuous reasoning develops recurrent computation within a model (Hao et al., 2024; Liu et al., 2024; Qu et al., 2025); internalized collaboration distills debate or composes latent role representations for a single model's inference (Yi et al., 2026; Zeng et al., 2026). The Vision Wormhole connects separately trained visual codecs through a shared message space. Appendix F specifies our heterogeneous baseline implementations.

Continuous conditioning and representation alignment. Learned soft prompts and prefixes condition frozen language models through trainable continuous inputs or layer-wise prefix states (Lester et al., 2021; Li & Liang, 2021). Multimodal architectures use projectors and resamplers to connect visual features to language generation (Alayrac et al., 2022; Li et al., 2023b). These architectures establish visual features as a source of continuous context for a language backbone. Cross-modal representation learning also uses paired image–text data to align semantic information across modalities (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023).

Across independently trained networks, representation alignment, model stitching, and permutation-aware merging study correspondences between internal feature spaces (Bansal et al., 2021; Lenc & Vedaldi, 2015; Ainsworth et al., 2022; Wortsman et al., 2022). Anchor-based relative representations support reuse of neural components across spaces and modalities (Moschella, 2024). Our visual codecs combine learned continuous conditioning with affine alignment on shared text anchors, providing a reusable communication interface for each frozen backbone. Appendix A discusses communication methods, model composition, compression, and channel analysis in greater depth.

3 Method: The Vision Wormhole

The **Vision Wormhole** connects heterogeneous agents by writing continuous messages into the *image-token span* of a VLM. Each backbone has a *Universal Visual Codec* with an encoder and a decoder. The encoder maps a latent rollout to fixed-size universal tokens; affine maps connect these tokens to a shared reference space $\mathcal{U}$; and the receiver's decoder generates a gated perturbation of its image-token embeddings. Codecs are trained separately for each model and frozen at inference, and all VLM backbone parameters remain frozen throughout. Figure 3 illustrates the pipeline. Appendix C provides architectural and procedural details.

3.1 Preliminaries

Multi-agent system (MAS). We define a multi-agent system as $\mathcal{S} = (\mathcal{A}, \pi)$, where $\mathcal{A} = \{a_1, \ldots, a_N\}$ is a set of agents and π is the orchestration policy. Given an input query q, π specifies role execution, interaction order, and routing among agents, and the system outputs a final answer a. This definition is modality-agnostic, in that agents may communicate through discrete text (TextMAS) or through any other interface they expose. In our experiments, each agent a_i is a vision-language model (VLM) with frozen backbone F_i and input embedding dimension d_i. We focus on heterogeneous VLM-based MAS, where agent backbones may come from different VLM families. VLM agents expose a continuous conditioning interface through *visual token embeddings*, which serve as the destination for decoded messages.

The VLM visual interface as a continuous channel. The VLMs in our experiments form an input embedding sequence by concatenating (1) text token embeddings and (2) a dedicated *image-token span*. Given an image $\mathbf{I}$, a vision encoder $E_{\text{vis}}^{(i)}$ and projector ϕ_i produce $X_{\text{img}}^{(i)} = \phi_i(E_{\text{vis}}^{(i)}(\mathbf{I})) \in \mathbb{R}^{L_{\text{img}}^{(i)} \times d_i}$, which is inserted into the language stream at model-specific image positions. During multimodal training, F_i learns to condition its language outputs on these projected image features. Our codec learns a continuous prompt at the same input positions.

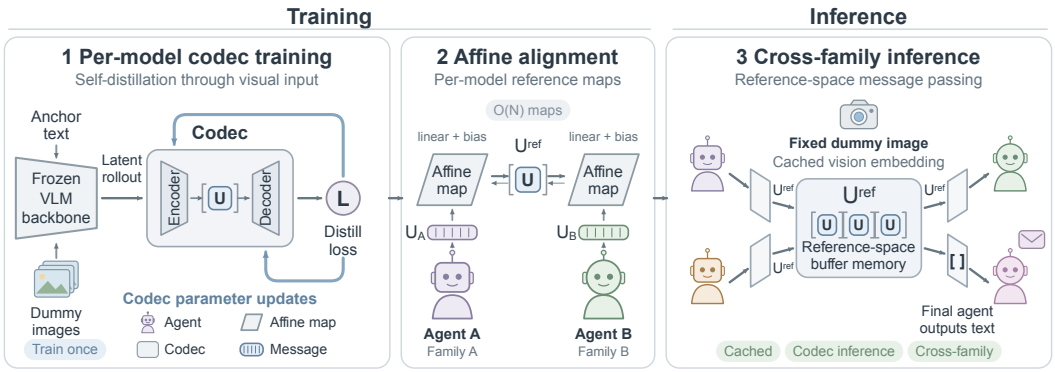

Figure 3: **Vision Wormhole overview.** A sender encodes its latent rollout into universal tokens and maps them to a shared reference space. The receiver maps incoming tokens to its codec coordinates and decodes them into a gated residual on its image-token span.

Latent rollouts. Let a VLM backbone F_i process a prompt and produce the final hidden vector at the prompt boundary, $h_0^{(i)} \in \mathbb{R}^{d_i}$. We define a *latent rollout* by repeatedly feeding back a single continuous pseudo-token embedding derived from the previous hidden state while reusing the prompt's attention cache (detailed in Appendix C.1). At step t, we form an input embedding $x_t^{(i)} = \mathrm{NormMatch}_i(h_t^{(i)})$, where $\mathrm{NormMatch}_i$ rescales vectors to match the typical norm of the model's token embeddings (see Eq. 5 in Appendix C.1). A T-length rollout yields $H_i = [x_0^{(i)}, \ldots, x_{T-1}^{(i)}] \in \mathbb{R}^{T \times d_i}$, which is the input to the sender's codec. Appendix B.2 gives the codec training hyperparameters.

Notation. For agent i, let d_i denote its embedding dimension. We use a universal token dimension D shared across all agents. A message is represented by $K_{\mathrm{u}} = K + 2$ universal tokens (with K message tokens plus two special tokens: a global token and a style token). For vision-span writing, we decode to K_{img} *image query tokens*. We denote by $L_{\mathrm{img}}^{(i)}$ the number of image tokens used by agent i in its prompt.

3.2 Training a Vision Codec for a Specific VLM

We train each codec through self-distillation on anchor texts. The teacher provides a prompt-boundary hidden state and next-token distribution from a text-conditioned forward pass. The student learns to reproduce these two quantities through vision-span conditioning (Appendix C.1).

(1) Sender message extraction via latent rollout. During training, the backbone processes a teacher prompt containing an anchor text m. Its boundary state $h_0^{(i)}$ initializes the length-T_{train} rollout defined in Section 3.1. At MAS inference, the rollout starts from the current role's task prompt and received context, as specified in Section 3.4.

(2) Universal-token encoder. We encode H_i into a fixed-size set of universal tokens using a Perceiver resampler (Alayrac et al., 2022), in which a fixed set of learned queries cross-attends to the rollout (Appendix C.1). The encoder $\mathcal{E}_i$ outputs $U_i = \mathcal{E}_i(H_i) \in \mathbb{R}^{K_{\mathrm{u}} \times D}$. The encoder uses K learned message queries and two special queries for global aggregation and style/statistics cues (Eq. 9 in Appendix C.1).

(3) Universal-to-vision decoder and gated injection. The decoder $\mathcal{D}_i$ maps universal tokens to a vision-span perturbation and a scalar gate: $(\Delta_i, g_i) = \mathcal{D}_i(U_i), \quad \Delta_i \in \mathbb{R}^{K_{\mathrm{img}} \times d_i}, \quad g_i \in (0, 1)$. The perturbation Δ_i is a *continuous prompt* in the agent's embedding space, and the gate g_i scales its strength per example. We parameterize the injected embeddings as a residual on a fixed baseline visual embedding. Let $\bar{X}_{\mathrm{img}}^{(i)} \in \mathbb{R}^{L_{\mathrm{img}}^{(i)} \times d_i}$ denote the image-token embeddings induced by a fixed dummy image under the frozen VLM. We resample Δ_i to the required image-span length via a length-resampling operator $\mathrm{Resample}(\cdot; L)$ and write

$$X_{\mathrm{img}}^{(i)} = \bar{X}_{\mathrm{img}}^{(i)} + g_i \cdot \mathrm{Resample}(\Delta_i; L_{\mathrm{img}}^{(i)}). \tag{1}$$

All downstream computation is performed by the frozen backbone conditioned on $X^{(i)}_{\text{img}}$. Decoder/-gating and dummy-image resampling details are provided in Appendix C.1.

(4) Self-distillation objective. Training uses anchor messages m (short text strings). The **teacher** prompt includes m explicitly as text. The **student** receives the anchor through a dummy image whose image-token span is overwritten by Eq. (1), computed from the teacher-side rollout. We optimize the parameters of $\mathcal{E}_i$ and $\mathcal{D}_i$ against the frozen teacher targets.

Let h^{text} and ℓ^{text} denote the teacher hidden state and next-token logits at the prompt boundary, and let $h^{\text{vis}}, \ell^{\text{vis}}$ be the corresponding student quantities under vision injection. We minimize

$$\begin{aligned}\mathcal{L}_{\text{codec}} = \lambda_h \left\|h^{\text{vis}} - \text{stopgrad}(h^{\text{text}})\right\|_2^2 + \lambda_{\text{kl}}\, \tau^2\, \text{KL}\Big(\text{softmax}\big(\tfrac{\ell^{\text{text}}}{\tau}\big) \,\Big\|\, \text{softmax}\big(\tfrac{\ell^{\text{vis}}}{\tau}\big)\Big) \\ + \lambda_{\text{rms}} \Big(\text{RMS}(\Delta_{\text{inj}}) - \text{RMS}(\bar{X}^{(i)}_{\text{img}})\Big)^2,\end{aligned} \tag{2}$$

where τ is the distillation temperature and Δ_{inj} denotes the gated perturbation before resampling. The first two terms match the boundary representation and next-token distribution. The RMS term regularizes the magnitude of the injected residual.

3.3 Aligning Codecs Across Agents via an Affine Map in Universal Space

Each independently trained codec defines model-specific coordinates for its D-dimensional tokens. We connect N codecs through a hub-and-spoke design, with affine maps to and from a *reference* universal space $\mathcal{U}$. The closed-form ridge-fitting procedure is provided in Appendix C.2.

Reference-space maps. Fix a reference agent r. For each agent i we learn affine maps $A^{\text{out}}_i(U) = UW^{\text{out}}_i + \mathbf{1}(b^{\text{out}}_i)^\top$ (sending: $U_i \to U^{\text{ref}}$) and $A^{\text{in}}_i(U) = UW^{\text{in}}_i + \mathbf{1}(b^{\text{in}}_i)^\top$ (receiving: $U^{\text{ref}} \to U_i$), with $W^{\text{out}}_i, W^{\text{in}}_i \in \mathbb{R}^{D\times D}$ and $b^{\text{out}}_i, b^{\text{in}}_i \in \mathbb{R}^D$. This yields $O(N)$ alignment maps: one map from each model to the hub and one back.

Ridge regression on shared anchors. We fit these affine maps using a subset of the codec-training anchor texts, $\{m_j\}_{j=1}^M$, shared across models. For each anchor m_j, we compute universal tokens $U_i(m_j)$ for every model i using the already trained encoder. We then solve a regularized least-squares problem in closed form (ridge regression) to map each model's tokens to the reference tokens:

$$\min_{W^{\text{out}}_i, b^{\text{out}}_i} \sum_{j=1}^{M} \left\|U_i(m_j)W^{\text{out}}_i + \mathbf{1}(b^{\text{out}}_i)^\top - U_r(m_j)\right\|_F^2 + \lambda \|W^{\text{out}}_i\|_F^2, \tag{3}$$

and analogously for A^{in}_i. The same fitted maps are reused across tasks and role assignments within a model pool.

3.4 Inference: Multi-Agent Collaboration through the Vision Wormhole

At inference time, intermediate agents emit universal tokens in the reference space, and the final agent generates the answer text. Appendix C.3 gives the full protocol.

Message passing operator. A wormhole message from sender s to receiver i is computed as:

$$U^{\text{ref}}_{s\to i} = A^{\text{out}}_s\Big(\mathcal{E}_s(H_s)\Big), \qquad (\Delta_i, g_i) = \mathcal{D}_i\Big(A^{\text{in}}_i(U^{\text{ref}}_{s\to i})\Big), \tag{4}$$

followed by writing Eq. (1) into receiver i's image-token span.

Memory aggregation. Let $\mathcal{M}$ be the set of received messages for an agent, each stored as universal tokens in the reference space. We aggregate memory by concatenation in token dimension: $U^{\text{ref}}_{\text{mem}} = \text{Concat}\big(\{U^{\text{ref}}_m\}_{m\in\mathcal{M}}\big) \in \mathbb{R}^{(|\mathcal{M}|\cdot K_{\text{u}})\times D}$, and decode a single vision-span perturbation from $U^{\text{ref}}_{\text{mem}}$. The decoder cross-attends to $|\mathcal{M}|K_{\text{u}}$ memory tokens and produces a fixed-length image-token span of $L^{(i)}_{\text{img}}$ rows for the backbone.

Role interaction loop. We follow the sequential role protocol of Zou et al. (2025). Each intermediate role-agent i performs three operations: *(1) Read:* Decode the current memory $U^{\text{ref}}_{\text{mem}}$ into a vision-span injection for agent i and run the frozen backbone conditioned on this injection; *(2) Think (latent):*

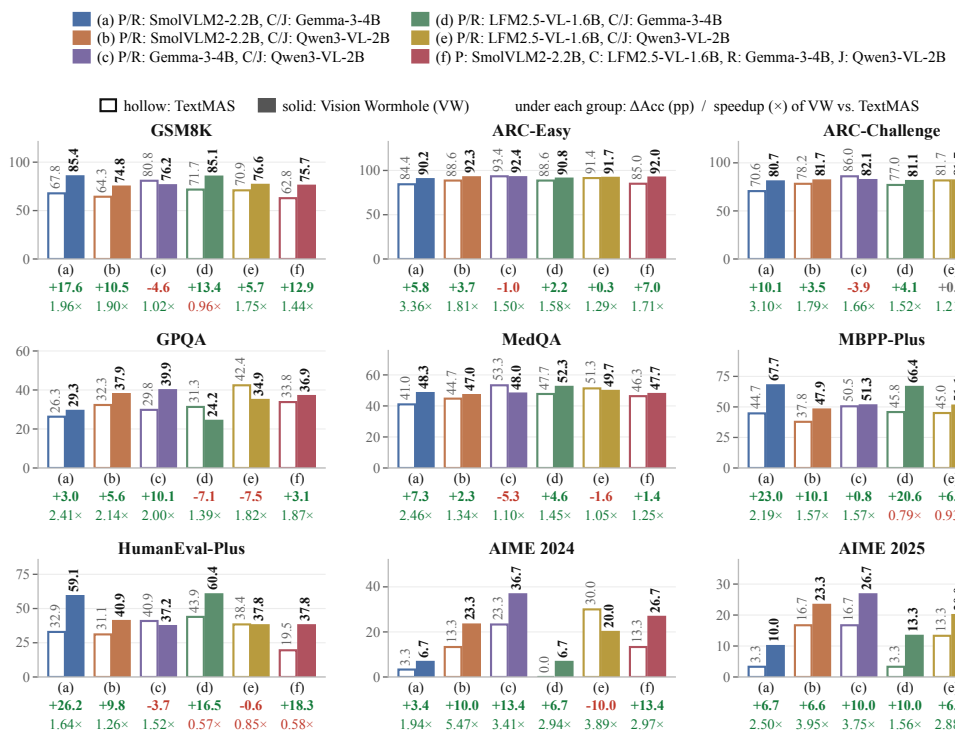

Figure 4: **Main heterogeneous MAS results: accuracy (%).** Each panel shows one benchmark and the six configurations in Table 2. Hollow bars denote TextMAS and solid bars denote VW; labels below each pair give the accuracy difference in percentage points. Panels use independent y-axis ranges. P/C/R/J denote Planner/Critic/Refiner/Judger.

Extract a rollout H_i and encode it into a new universal message $U_i^{\text{ref}} = A_i^{\text{out}}(\mathcal{E}_i(H_i))$; and *(3) Write:* Append U_i^{ref} to the shared memory buffer. The final role generates the answer from its prompt and accumulated memory.

4 Experiments

4.1 Experimental Settings

Tasks and Datasets. We evaluate nine benchmarks using the evaluation sets of LatentMAS (Zou et al., 2025): (i) *Math & Science Reasoning*, including GSM8K (Cobbe et al., 2021) (1,319), AIME 2024 (Maxwell-Jia, 2024) (30), AIME 2025 (Zhang & Math-AI, 2025) (30), GPQA Diamond (Rein et al., 2023) (198), and MedQA (Yang et al., 2025) (300); (ii) *Commonsense Reasoning*, including ARC-Easy (2,376) and ARC-Challenge (1,172) (Clark et al., 2018b;a); and (iii) *Code Generation*, including MBPP-Plus (378) and HumanEval-Plus (164) (Liu et al., 2023b). Parentheses give the number of evaluated examples per configuration. We report accuracy for multiple-choice and short-answer tasks, and pass@1 for code generation.

Models. We use four VLM backbones: Qwen/Qwen3-VL-2B-Thinking (Bai et al., 2025), google/gemma-3-4b-it (Team, 2025), HuggingFaceTB/SmolVLM2-2.2B-Instruct (Marafioti et al., 2025), and LiquidAI/LFM2.5-VL-1.6B (Amini et al., 2025). Both TextMAS and Vision Wormhole use frozen backbones. The main experiments use five two-backbone configurations and one four-backbone pool (model combinations and role assignments in Table 2, Appendix B).

Protocol, Baselines, and Implementation. Following Zou et al. (2025), all main configurations use the sequential role protocol and context accumulation in Appendix C.3. The primary comparison is against text-mediated MAS (TextMAS), with the same role sequence, model assignments,

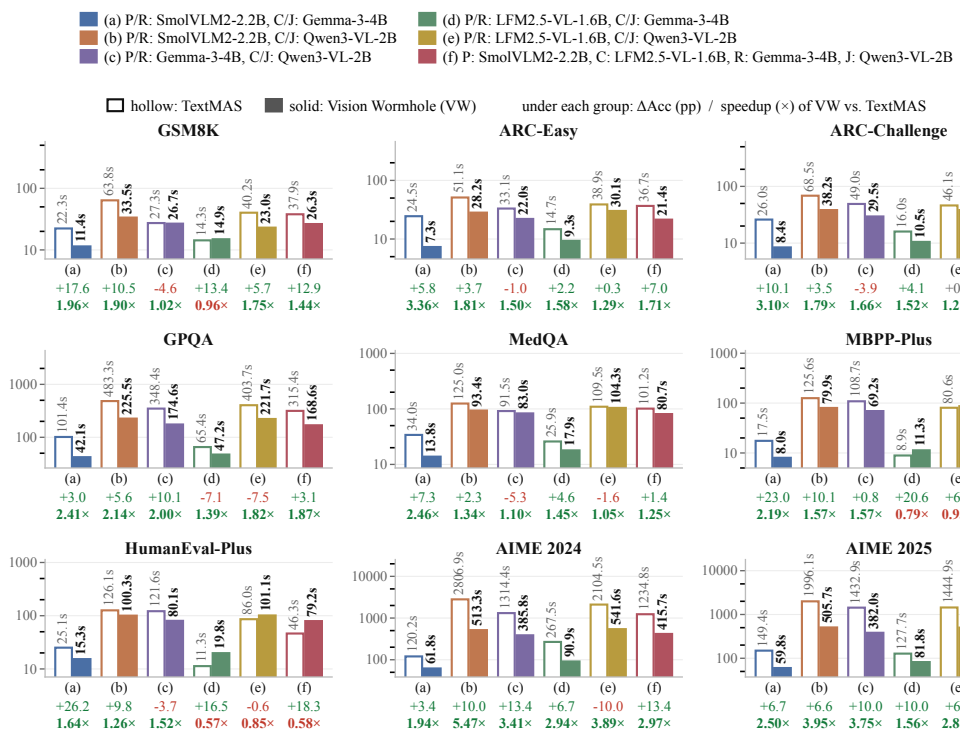

Figure 5: **Main heterogeneous MAS results: batch-normalized runtime (s/query, log scale).** Same layout, colors, and configurations as Figure 4; timing protocol in Appendix B.3. The number under each group is the speedup (×) of VW relative to TextMAS.

evaluation sets, and task-level generation caps. We compare the visual interface with a learned text prefix (Figure 8). Further comparisons cover OCR relay (Appendix D) and two latent baselines on Qwen/Gemma (Appendix F). Role prompts, hardware placement, generation budgets, and batching are in Appendix B.3.

4.2 Results

We report accuracy and batch-normalized end-to-end runtime for TextMAS and the Vision Wormhole (VW). Accuracy gains are in percentage points; speedup is the ratio of TextMAS runtime to VW runtime. Macro-averages weight the nine benchmarks equally. The overall accuracy gain is the arithmetic mean across all 54 configuration–task pairs, and overall speedup is the geometric mean of their runtime ratios.

Main Heterogeneous MAS. VW improves accuracy by 6.0 pp on average and achieves an overall speedup of 1.69× over TextMAS (Figures 1, 4, and 5). Macro-average accuracy increases in five of six configurations, and runtime decreases in 47 of 54 configuration–task pairs. Accuracy and runtime improve together in 36 pairs (Figure 6). Code generation has the largest benchmark-level mean accuracy gains: 13.3 pp on MBPP-Plus and 11.1 pp on HumanEval-Plus, averaging over the six configurations.

Weakly Supervised Setting. Table 1 reports results for the weakly supervised setting defined in Appendix B.2. VW retains its runtime advantage across all 18 configuration–task pairs. The SmolVLM2/Qwen configuration also improves accuracy on every benchmark.

Performance across Model Groups. Figure 7 compares Qwen3-VL-2B and SmolVLM2-2.2B standalone accuracy with their *model-group* MAS accuracy. For each model, we average the scores of all configurations containing that model within each benchmark, then average across benchmarks.

VW improves on TextMAS in both model groups.

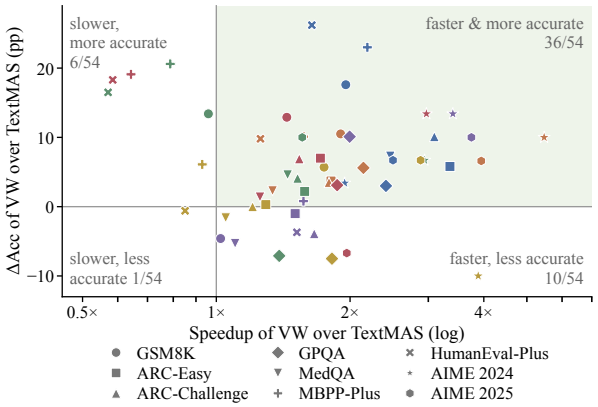

Figure 6: **Accuracy–speed trade-off.** Points denote configuration–benchmark pairs. Colors follow Figure 4; markers identify benchmarks.

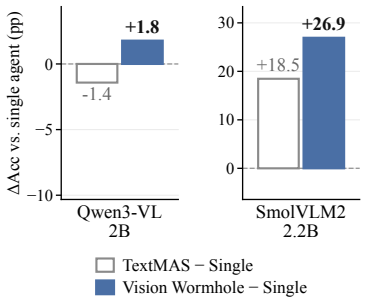

Figure 7: **Model-group performance.** Gains over standalone (Section 4.2). Hollow: TextMAS; solid: VW. Independent y-axes.

Table 1: **Weakly supervised results.** We report accuracy (%), batch-normalized runtime (s/query), and differences relative to TextMAS (ΔAcc in pp; runtime ratio in ×).

	P/R: Gemma-3-4B, C/J: Qwen3-VL-2B						P/R: SmolVLM2-2.2B, C/J: Qwen3-VL-2B					
Dataset	**Text Acc**	**Text Time**	**VW Acc**	**VW Time**	**ΔAcc**	**Speedup**	**Text Acc**	**Text Time**	**VW Acc**	**VW Time**	**ΔAcc**	**Speedup**
GSM8K	80.8%	27.3s	77.6%	22.9s	-3.2pp	1.19×	64.3%	63.8s	77.0%	25.6s	+12.7pp	2.49×
ARC-Easy	93.4%	33.1s	90.3%	23.1s	-3.1pp	1.43×	88.6%	51.1s	91.8%	23.6s	+3.2pp	2.17×
ARC-Challenge	86.0%	49.0s	80.5%	30.3s	-5.5pp	1.62×	78.2%	68.5s	81.3%	30.4s	+3.1pp	2.25×
GPQA	29.8%	348.4s	34.9%	172.9s	+5.1pp	2.02×	32.3%	483.3s	33.3%	172.5s	+1.0pp	2.80×
MedQA	53.3%	91.5s	45.0%	82.0s	-8.3pp	1.12×	44.7%	125.0s	49.0%	75.8s	+4.3pp	1.65×
MBPP-Plus	50.5%	108.7s	46.6%	69.1s	-3.9pp	1.57×	37.8%	125.6s	49.2%	69.4s	+11.4pp	1.81×
HumanEval-Plus	40.9%	121.6s	40.2%	80.7s	-0.7pp	1.51×	31.1%	126.1s	42.7%	80.6s	+11.6pp	1.56×
AIME 2024	23.3%	1314.4s	26.7%	404.8s	+3.4pp	3.25×	13.3%	2806.9s	36.7%	389.8s	+23.4pp	7.20×
AIME 2025	16.7%	1432.9s	23.3%	370.5s	+6.6pp	3.87×	16.7%	1996.1s	23.3%	415.8s	+6.6pp	4.80×

Component Ablations. Visual reception outperforms a learned soft prefix on GSM8K and MedQA with matched Qwen codec training and inputs (Figure 8a). For Qwen/Gemma ablation codecs, aligned messages give lower reconstruction error on held-out texts than cross-text messages or shuffled-anchor maps in both directions (Figure 8b). Appendices G and H.4 give the setups and full results.

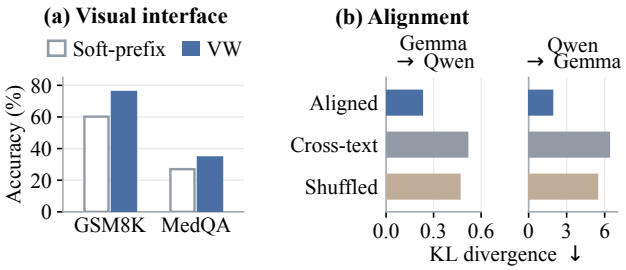

Figure 8: **Interface and alignment ablations.** (a) Qwen accuracy. (b) Held-out reconstruction KL.

5 Conclusion

Vision Wormhole turns the visual input interface of a VLM into a channel for receiving another model's internal states. Per-model codecs encode latent rollouts into universal messages and decode incoming messages into visual context. Self-distillation trains each codec with its backbone frozen; affine reference maps connect the codecs with $O(N)$ components for N models. Across four VLM families and nine benchmarks, VW improves accuracy by 6.0 percentage points on average and achieves a 1.69× overall speedup over TextMAS. These results establish visual conditioning as a practical interface for continuous communication among frozen heterogeneous models.

AI USE STATEMENT

We used large language models as general-purpose assistants for literature search, manuscript drafting and editing, experimental design, code implementation, result analysis, and figure preparation. Language models also supported the evaluations in Appendices E and H.6. The authors reviewed the AI-assisted work and take responsibility for the final manuscript and results.

ETHICS STATEMENT

The experiments study communication among publicly released language and vision-language models using public reasoning and code-generation benchmarks. The system inherits the capabilities and behavioral limitations of those checkpoints. Evaluation is confined to the benchmark tasks described in Section 4.1.

REPRODUCIBILITY STATEMENT

The codec architecture, training objective, and alignment procedure are specified in Section 3, with training hyperparameters, anchor-corpus construction, and optimizer settings in Appendix B.2 and architectural details in Appendix C. Hardware placement, generation budgets, decoding, prompt sources, and batching are given in Appendix B.3. Model configurations and role assignments are listed in Table 2. We use publicly released benchmarks and model checkpoints, with the LatentMAS evaluation sets listed in Section 4.1.

REFERENCES

Deepak Bhaskar Acharya, Karthigeyan Kuppan, and Divya Bhaskaracharya. Agentic ai: Autonomous intelligence for complex goals – a comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. doi: 10.1109/ACCESS.2025.3532853.

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, Ramin Hasani, Saniya Karwa, Yuri Khrustalev, Maxime Labonne, Mathias Lechner, Valentine Lechner, Simon Lee, Zetian Li, Noel Loo, Jacob Marks, Edoardo Mosca, Samuel J. Paech, Paul Pak, Rom N. Parnichkun, Alex Quach, Ryan Rogers, Daniela Rus, Nayan Saxena, Bettina Schlager, Tim Seyde, Jimmy T. H. Smith, Aditya Tadimeti, and Neehal Tumma. Lfm2 technical report, 2025. URL https://arxiv.org/abs/2511.23404.

Raviteja Anantha, Bortik Bandyopadhyay, Anirudh Kashi, Sayantan Mahinder, Andrew W Hill, and Srinivas Chappidi. Protip: Progressive tool retrieval improves planning, 2023. URL https://arxiv.org/abs/2312.10332.

Sadia Asif, Mohammad Mohammadi Amiri, Momin Abbas, Prasanna Sattigeri, and Karthikeyan Natesan Ramamurthy. Lcguard: Latent communication guard for safe kv sharing in multi-agent systems, 2026. URL https://arxiv.org/abs/2605.22786.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren,

Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Shikhar Vashishth, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. Llm augmented llms: Expanding capabilities through composition, 2024. URL https://arxiv.org/abs/2401.02412.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/01ded4259d101feb739b06c399e9cd9c-Paper.pdf.

Lina Berrayana, Ahmed Heakl, Abdullah Sohail, Thomas Hofmann, Salman Khan, and Wei Chen. Latent-darm: Bridging discrete diffusion and autoregressive models for reasoning, 2026. URL https://arxiv.org/abs/2603.09184.

Zhuohang Bian, Feiyang Wu, Chengrui Zhang, Hangcheng Dong, Yun Liang, and Youwei Zhuo. Tokendance: Scaling multi-agent llm serving via collective kv cache sharing, 2026. URL https://arxiv.org/abs/2604.03143.

Timo Birr, Christoph Pohl, Abdelrahman Younes, and Tamim Asfour. Autogpt+p: Affordance-based task planning with large language models, 2024. URL https://arxiv.org/abs/2402.10778.

Luís Brito and Carlos Baquero. When latent agents lie: Kv-cache integrity in multi-agent llm collaboration, 2026. URL https://arxiv.org/abs/2606.28958.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya G. Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*, 2025.

Jingdi Chen, Hanqing Yang, Zongjun Liu, and Carlee Joe-Wong. The five ws of multi-agent communication: Who talks to whom, when, what, and why – a survey from marl to emergent language and llms, 2026a. URL https://arxiv.org/abs/2602.11583.

Kerui Chen, Jinglu Wang, Jianrong Zhang, Ming Li, Yan Lu, and Hehe Fan. Scaling video understanding via compact latent multi-agent collaboration, 2026b. URL https://arxiv.org/abs/2605.00444.

Siyi Chen, Xiaoyan Zhang, Meng Wu, Jonathan Tremblay, Valts Blukis, Stan Birchfield, Rene Vidal, Alvaro Velasquez, Sijia Liu, and Qing Qu. See what i see, know what i think: Dense latent communication across heterogeneous agents, 2026c. URL https://arxiv.org/abs/2606.13594.

Tianhao Chen, Yuheng Wu, and Dongman Lee. Laco: Adaptive latent communication for collaborative driving, 2026d. URL https://arxiv.org/abs/2605.22504.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. *arXiv preprint arXiv:2410.08115*, 2024.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11534–11557, 2025.

Jiaming Cheng, Subhransu Das, and Rajiv Ramnath. When does latent communication pay? a causal audit of relayed kv caches in multi-agent llms, 2026. URL https://arxiv.org/abs/2608.04893.

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*, 2018a.

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*, 2018b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Julian Coda-Forno, Zhuokai Zhao, Qiang Zhang, Dipesh Tamboli, Weiwei Li, Xiangjun Fan, Lizhu Zhang, Eric Schulz, and Hsiao-Ping Tseng. Exploring system 1 and 2 communication for latent reasoning in llms. *arXiv preprint arXiv:2510.00494*, 2025.

Lipeng Dai, Luping Xiang, and Kun Yang. A token/kv-cache communication media selection and resource allocation strategy for multi-agent collaboration, 2026. URL https://arxiv.org/abs/2605.25422.

C.m. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. In Duygu Ataman (ed.), *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pp. 268–281, Singapore, dec 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.mrl-1.20. URL https://aclanthology.org/2023.mrl-1.20/.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. Rethinking memory in llm based agents: Representations, operations, and emerging topics. *arXiv preprint arXiv:2505.00675*, 2025. doi: 10.48550/arXiv.2505.00675.

Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27106–27129, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1248. URL https://aclanthology.org/2026.acl-long.1248/.

Darius Feher, Ivan Vulić, and Benjamin Minixhofer. Retrofitting large language models with dynamic tokenization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Benjamin Fein-Ashley and Jacob Fein-Ashley. Bridging hidden states in vision-language models. *arXiv preprint arXiv:2511.11526*, 2025.

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

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied ai: Advances and future directions. *arXiv preprint arXiv:2505.05108*, 2025.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang (Eric) Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024. URL https://arxiv.org/abs/2411.04468. Submitted Nov 7, 2024.

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.

Yingsheng Geng, Yuchong Gao, Weihong Wu, Guyue Liu, and Jiang Liu. Relaycaching: Accelerating llm collaboration via decoding kv cache reuse, 2026. URL https://arxiv.org/abs/2603.13289.

C Goddard and FF Neto. Training-free tokenizer transplantation via orthogonal matching pursuit. *arXiv preprint arXiv:2506.06607*, 2025.

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

Guangfu Hao, Yuming Dai, Xianzhe Qin, and Shan Yu. Brain-inspired graph multi-agent systems for llm reasoning, 2026. URL https://arxiv.org/abs/2603.15371.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhipeng Hou, Junyi Tang, and Yipeng Wang. Halo: Hierarchical autonomous logic-oriented orchestration for multi-agent llm systems. *arXiv preprint arXiv:2505.13516*, 2025.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, 2025a.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, Zeyu Zhang, Yifeng Wang, Qianshuo Ye, Bernard Ghanem, Ping Luo, and Guohao Li. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025b.

Hyesung Jeon, Hyeongju Ha, and Jae-Joon Kim. Lragent: Efficient kv cache sharing for multi-lora llm agents, 2026. URL https://arxiv.org/abs/2602.01053.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 2021.

Haibo Jin, Peng Kuang, Ye Yu, Xiaopeng Yuan, and Haohan Wang. Agent primitives: Reusable latent building blocks for multi-agent systems, 2026. URL https://arxiv.org/abs/2602.03695.

Boris Kriuk and Logic Ng. Q-kvcomm: Efficient multi-agent communication via adaptive kv cache compression, 2025. URL https://arxiv.org/abs/2512.17914.

Seunghun Lee, Jihong Park, Ce Zheng, and Hyuncheol Park. Low-latency edge llm handover via joint kv cache transfer and token prefill, 2026. URL https://arxiv.org/abs/2603.28018.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *CVPR*, 2015.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025a.

Bojie Li and Noah Shi. The latent bridge: A continuous slow-fast channel for real-time game agents, 2026. URL https://arxiv.org/abs/2606.24470.

Chong Li, Jiajun Zhang, and Chengqing Zong. Tokalign: Efficient vocabulary adaptation via token alignment. *arXiv preprint arXiv:2506.03523*, 2025b. URL https://arxiv.org/abs/2506.03523.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS 2023)*, 2023a. URL https://proceedings.neurips.cc/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Yiping Li, Zhiyu An, and Wan Du. When less latent leads to better relay: Information-preserving compression for latent multi-agent llm collaboration, 2026. URL https://arxiv.org/abs/2604.13349.

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

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

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency, 2023a. URL https://arxiv.org/abs/2304.11477.

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, 2023b.

Luyang Liu, Jonas Pfeiffer, Jiaxing Wu, Jun Xie, and Arthur Szlam. Deliberation in latent space via differentiable cache augmentation. *arXiv preprint arXiv:2412.17747*, 2024.

Xiaoze Liu, Weichen Yu, Matt Fredrikson, Xiaoqian Wang, and Jing Gao. When the same coefficients reach different places: Asymmetric realizability in transplanting tokenizers across large language models, 2026. URL https://arxiv.org/abs/2601.00065.

Yingzhuo Liu. Beyond tokens: a unified framework for latent communication in llm-based multi-agent systems, 2026. URL https://arxiv.org/abs/2606.05711.

Wenhao Lu and Spursh Deshpande. On the role of learned alignment matrices in LatentMAS. In *Second Workshop on Compositional Learning: Safety, Interpretability, and Agents at ICML*, 2026a. URL https://openreview.net/forum?id=manXhfpRH3.

Wenhao Lu and Spursh Deshpande. What do latent agents actually represent? interpreting hidden-state communication in multi-agent systems. In *Second Workshop on Compositional Learning: Safety, Interpretability, and Agents at ICML*, 2026b. URL https://openreview.net/forum?id=UJdeo8SM7F.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. URL https://arxiv.org/abs/2504.05299.

Maxwell-Jia. AIME 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, 2024.

Benjamin Minixhofer, Edoardo Maria Ponti, and Ivan Vulić. Zero-shot tokenizer transfer. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. *arXiv preprint arXiv:2503.20083*, 2025.

Luca Moroni, Giovanni Puccetti, Pere-Lluís Huguet Cabot, Andrei Stefan Bejgu, Alessio Miaschi, Edoardo Barba, Felice Dell'Orletta, Andrea Esuli, and Roberto Navigli. Optimizing llms for italian: Reducing token fertility and enhancing efficiency through vocabulary adaptation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6661–6675, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.371. URL https://aclanthology.org/2025.findings-naacl.371/.

Luca Moschella. Latent communication in artificial neural networks, 2024. URL https://arxiv.org/abs/2406.11014.

Nandini Mundra, Aditya Nanda Kishore Khandavally, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, and Mitesh M. Khapra. An empirical comparison of vocabulary expansion and initialization approaches for language models. In *Proceedings of the 28th Conference on Computational Natural Language Learning (CoNLL)*, pp. 84–104, Miami, FL, USA, nov 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.conll-1.8. URL https://aclanthology.org/2024.conll-1.8/.

nhminle. Latentmas-hybrid. https://github.com/nhminle/LatentMAS-Hybrid, 2026. Public fork of LatentMAS. Cross-model realignment implementation at commit 6db9551. Accessed September 10, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

Aneesh Pappu, Batu El, Hancheng Cao, Carmelo di Nolfo, Yanchao Sun, Meng Cao, and James Zou. Multi-agent teams hold experts back, 2026. URL https://arxiv.org/abs/2602.01011.

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

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023b.

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Unks everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10186–10203, 2021.

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 (ICLR)*, 2024. URL https://openreview.net/forum?id=sehRvaIPQQ.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/ramesh25a.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp. *arXiv preprint arXiv:2408.04303*, 2024. URL https://arxiv.org/abs/2408.04303. Accepted at COLM 2024.

François Remy, Pieter Delobelle, Bettina Berendt, Kris Demuynck, and Thomas Demeester. Tik-to-tok: Translating language models one token at a time: An embedding initialization strategy for efficient language adaptation. *arXiv preprint arXiv:2310.03477*, 2023.

Maximillian Rossi, Prajwal Raghunath, and Eugene Wu. Latent cache flow: Model-to-model communication without text, 2026. URL https://arxiv.org/abs/2605.22863.

Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Xingyu Zeng, and Rui Zhao. Tptu: large language model-based ai agents for task planning and tool usage. *arXiv preprint arXiv:2308.03427*, 2023.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

Arup Kumar Sarker, Alexander James Halpern, Mills Staylor, Aymen Alsaadi, Gregor von Laszewski, Yue Cheng, Shantenu Jha, and Geoffrey Fox. [aaflow+] stateful operator abstraction with zero-copy distributed kv cache orchestration for multi-agent workflows, 2026. URL https://arxiv.org/abs/2607.10987.

Shaurya Sharthak, Vinayak Pahalwan, Adithya Kamath, and Adarsh Shirawalmath. Achieving tokenizer flexibility in language models through heuristic adaptation and supertoken learning. *arXiv preprint arXiv:2505.09738*, 2025. URL https://arxiv.org/abs/2505.09738.

Xiangyu Shi, Marco Chiesa, Gerald Q. Maguire Jr., and Dejan Kostic. Kvcomm: Enabling efficient llm communication through selective kv sharing, 2026a. URL https://arxiv.org/abs/2510.03346.

Yuling Shi, Chaoxiang Xie, Zhensu Sun, Yeheng Chen, Chenxu Zhang, Longfei Yun, Chengcheng Wan, Hongyu Zhang, David Lo, and Xiaodong Gu. Codeocr: On the effectiveness of vision language models in code understanding, 2026b. URL https://arxiv.org/abs/2602.01785.

Yakov Pyotr Shkolnikov. Agent memory below the prompt: Persistent q4 kv cache for multi-agent llm inference on edge devices, 2026. URL https://arxiv.org/abs/2603.04428.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. exbert: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1433–1439, Online, nov 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.129. URL https://aclanthology.org/2020.findings-emnlp.129/.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. Augmenting multi-agent communication with state delta trajectory, 2025. URL https://arxiv.org/abs/2506.19209.

Wei Tao, Yucheng Zhou, Yanlin Wang, Wenqiang Zhang, Hongyu Zhang, and Yu Cheng. Magis: Llm-based multi-agent framework for github issue resolution. *Advances in Neural Information Processing Systems*, 37:51963–51993, 2024.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

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

Giorgos Vernikos and Andrei Popescu-Belis. Subword mapping and anchoring across languages. *arXiv preprint arXiv:2109.04556*, 2021.

Chenxi Wang, Ruiyang Huang, Jiayan Sun, Lei Wei, and Yifan Wu. Out of sight, not out of mind: Unveiling latent attack in latent-based multi-agent systems, 2026a. URL https://arxiv.org/abs/2605.28214.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities, 2024a. URL https://arxiv.org/abs/2406.04692.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025a.

Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. *arXiv preprint arXiv:2503.24377*, 2025b.

Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Yang Tang, and Zheng Wei. Render-of-thought: Rendering textual chain-of-thought as images for visual latent reasoning, 2026b. URL https://arxiv.org/abs/2601.14750.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. MEMORYLLM: towards self-updatable large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50453–50466. PMLR, 2024c.

Zhao Wang, Sota Moriyama, Wei-Yao Wang, Briti Gangopadhyay, and Shingo Takamatsu. Talk structurally, act hierarchically: A collaborative framework for llm multi-agent systems. *arXiv preprint arXiv:2502.11098*, 2025c.

Ziqing Wang, Lili Zhao, and Kaize Ding. Medlatentdx: Latent multi-agent communication for cross-hospital rare-disease diagnosis, 2026c. URL https://arxiv.org/abs/2606.13945.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression, 2025. URL https://arxiv.org/abs/2510.18234.

Markus Wenzel. Latent communication between language model agents: Channels, alignment, and the limits of text, 2026. URL https://arxiv.org/abs/2607.14103.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Di Wu and Xiaohui Zhu. Post-hoc sparse coding of latent communication between vision-language model agents, 2026. URL https://arxiv.org/abs/2608.10198.

Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. Talk to right specialists: Routing and planning in multi-agent system for question answering. *arXiv preprint arXiv:2501.07813*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation. In *First Conference on Language Modeling*, 2024.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. An empirical study on cross-lingual vocabulary adaptation for efficient language model inference. *arXiv preprint arXiv:2402.10712*, 2024.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. How can we effectively expand the vocabulary of LLMs with 0.01GB of target language text? *Computational Linguistics*, 52(1): 295–330, 2026.

Bingyu Yan, Zhibo Zhou, Litian Zhang, Lian Zhang, Ziyi Zhou, Dezhuang Miao, Zhoujun Li, Chaozhuo Li, and Xiaoming Zhang. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*, 2025.

Fu-Chun Yang and Jason Eshraghian. Direct semantic communication between large language models via vector translation, 2025. URL https://arxiv.org/abs/2511.03945.

Hang Yang, Hao Chen, Hui Guo, Yineng Chen, Ching-Sheng Lin, Shu Hu, Jinrong Hu, Xi Wu, and Xin Wang. Llm-medqa: Enhancing medical question answering through case studies in large language models. *arXiv preprint arXiv:2501.05464*, 2025. Submitted 31 Dec 2024 (v1); revised 18 Jan 2025 (v2).

Wooseong Yang, Wei-Chieh Huang, Weizhi Zhang, Yu Wang, Philip S. Yu, and Junhyun Lee. Xbridge: Entity-grounded latent bridge for heterogeneous llm communication, 2026. URL https://arxiv.org/abs/2608.11676.

Yingxuan Yang, Qiuying Peng, Jun Wang, Ying Wen, and Weinan Zhang. Llm-based multi-agent systems: Techniques and business perspectives. *arXiv preprint arXiv:2411.14033*, 2024.

Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, and Dacheng Tao. A survey on agentic multimodal large language models. *arXiv preprint arXiv:2510.10991*, 2025.

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

Rui Ye, Xiangrui Liu, Qimin Wu, Xianghe Pang, Zhenfei Yin, Lei Bai, and Siheng Chen. X-mas: Towards building multi-agent systems with heterogeneous llms, 2025b. URL https://arxiv.org/abs/2505.16997.

John Seon Keun Yi, Aaron Mueller, and Dokyun Lee. Latent agents: A post-training procedure for internalized multi-agent debate, 2026. URL https://arxiv.org/abs/2604.24881.

Xinlei Yu, Chengming Xu, Zhangquan Chen, Bo Yin, Cheng Yang, Yongbo He, Yihao Hu, Jiangning Zhang, Cheng Tan, Xiaobin Hu, and Shuicheng Yan. Dual latent memory for visual multi-agent system, 2026a. URL https://arxiv.org/abs/2602.00471.

Ye Yu, Heming Liu, Haibo Jin, Xiaopeng Yuan, Peng Kuang, and Haohan Wang. Learning to communicate: Toward end-to-end optimization of multi-agent language systems, 2026b. URL https://arxiv.org/abs/2604.21794.

Xiaopeng Yuan, Haibo Jin, Ye Yu, Peng Kuang, Lijun Yu, Yushun Dong, and Haohan Wang. Closing the loop on latent reasoning via test-time reconstruction, 2026. URL https://arxiv.org/abs/2606.06252.

Zhichen Zeng, Huiyuan Chen, Jingru Cheng, Juan Zha, Ming Liu, Ying Chen, Xiyuan Yang, Chaosheng Dong, Haiyang Zhang, and Hanghang Tong. One model, many minds: Unlocking multi-agent synergy in a single agent via mixture of roles, 2026. URL https://arxiv.org/abs/2608.27338.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023. doi: 10.1109/ICCV51070.2023.01100.

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large language model-brained gui agents: A survey. *arXiv preprint arXiv:2411.18279*, 2024a.

Hangfan Zhang, Zhiyao Cui, Jianhao Chen, Xinrun Wang, Qiaosheng Zhang, Zhen Wang, Dinghao Wu, and Shuyue Hu. Stop overvaluing multi-agent debate – we must rethink evaluation and embrace model heterogeneity, 2025a. URL https://arxiv.org/abs/2502.08788.

Huixiang Zhang and Mahzabeen Emu. Do latent channels actually communicate? a causal audit of latent multi-agent llm, 2026. URL https://arxiv.org/abs/2607.26773.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025b.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025. https://huggingface.co/datasets/math-ai/aime25, 2025. Dataset card on Hugging Face (license: Apache-2.0).

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, 2024b.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):155:1–155:47, 2025c. doi: 10.1145/3748302.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*, 2025d.

Bingxi Zhao, Lin Geng Foo, Ping Hu, Christian Theobalt, Hossein Rahmani, and Jun Liu. Llm-based agentic reasoning frameworks: A survey from methods to scenarios. *arXiv preprint arXiv:2508.17692*, 2025a.

Jiaxing Zhao, Hongbin Xie, Yuzhen Lei, Xuan Song, Zhuoran Shi, Lianxin Li, Shuangxue Liu, Linguo Xie, and Haoran Zhang. Cochain: Balancing insufficient and excessive collaboration in llm agent workflows. *arXiv preprint arXiv:2505.10936*, 2025b.

Wanjia Zhao, Mert Yuksekgonul, Shirley Wu, and James Zou. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. *arXiv preprint arXiv:2502.04780*, 2025c.

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.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2023. URL https://arxiv.org/abs/2310.04406.

Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. *arXiv preprint arXiv:2503.02390*, 2025.

Wenxuan Zhou, Junyi Du, and Xiang Ren. Improving bert fine-tuning with embedding normalization. *arXiv preprint arXiv:1911.03918*, 2019.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. *arXiv preprint arXiv:2505.12514*, 2025.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. Toolchain*: Efficient action space navigation in large language models with a* search, 2023. URL https://arxiv.org/abs/2310.13227.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*, 2025.

A Extended Related Work

A.1 Multi-Agent Workflows, Memory, and Tools

Surveys of LLM-based multi-agent systems catalogue role assignments, coordination patterns, and evaluation practices (Guo et al., 2024; Tran et al., 2025; Yan et al., 2025; Yang et al., 2024; Acharya et al., 2025; Wang et al., 2024b; Yao et al., 2025; Li et al., 2025c; Feng et al., 2025; Zhang et al., 2024a; Zhao et al., 2025a). Communication surveys connect these systems to learned communication in multi-agent reinforcement learning and emergent language through the choice of participants, message content, timing, and purpose (Chen et al., 2026a). Agent frameworks implement natural-language interaction together with routing, delegation, and tool calls (Wu et al., 2024; Hong et al., 2024; Li et al., 2023a; Fourney et al., 2024; Hou et al., 2025; Hu et al., 2025b; Tao et al., 2024; Park et al., 2023a; Wu et al., 2025; Wang et al., 2025c). Workflow research studies multi-step collaboration, self-improvement, and allocation of inference computation (Zhang et al., 2024b; Zhao et al., 2025b;c; Chen et al., 2024; 2025; Zhou et al., 2025; Li et al., 2025d; Zhuge et al., 2024). Graph-based orchestration also coordinates specialized agents through a centralized workspace (Hao et al., 2026). Analyses of agent systems identify information sharing, coordination, and specialization as contributors to task outcomes (Pezeshkpour et al., 2024; Cemri et al., 2025; Pappu et al., 2026; Zhang et al., 2025a).

Memory and planning modules maintain information across interactions and support long-horizon execution (Zhang et al., 2025c; Du et al., 2025; Packer et al., 2023; Zhong et al., 2024; Wang et al., 2024c; Hu et al., 2025a; Pan et al., 2025). Tool-use methods add retrieval and action selection to these workflows (Ruan et al., 2023; Zhou et al., 2023; Liu et al., 2023a; Birr et al., 2024; Anantha et al., 2023; Zhuang et al., 2023). Our experiments use a sequential four-role workflow to evaluate a continuous communication interface among heterogeneous VLMs.

Heterogeneous model selection connects these workflows to differences in agent capabilities. Function–domain evaluations compare models on question answering, revision, aggregation, planning, and evaluation, then use those measurements to assign models within existing agent topologies (Ye et al., 2025b). Layered response aggregation studies separate proposer and aggregator roles and finds benefits from diverse model proposals in its generation benchmarks (Wang et al., 2024a). These results motivate communication interfaces that can serve several model families within one workflow.

A.2 Continuous Messages and Hidden-State Interfaces

Latent communication exposes intermediate model computations as messages. A recent survey organizes these interfaces by the representation transferred, the sender–receiver alignment, and the receiver's fusion operation (Liu, 2026). These choices connect the message format to concrete implementation requirements: its token axis, feature dimension, layer correspondence, and insertion point. Probability-weighted vocabulary embeddings provide a continuous representation of the sender's next-token distribution (Pham et al., 2024). Activation communication inserts a sender's intermediate vector into the receiver's residual stream, with arithmetic fusion and an optional model-pair alignment matrix (Ramesh & Li, 2025). State-delta communication augments generated text with token-wise changes in internal states (Tang et al., 2025). Together, these approaches establish several ways to expose information from the sender's forward computation. Latent communication has also been examined in dedicated evaluation studies (Wenzel, 2026; Lu & Deshpande, 2026b;a; Zhang & Emu, 2026; Cheng et al., 2026).

Input-level interfaces place translated messages alongside the receiver's prompt embeddings. Learned adapters support the transfer and compression of last-layer state sequences (Du et al., 2026), and vector translators align representations between independently trained language models (Yang & Eshraghian, 2025). Hybrid lexical–latent interfaces combine receiver-native embeddings of the sender's context tokens with gated cross-attention over its hidden states (Yang et al., 2026). These designs give concrete roles to contextual representations, lexical identity, and input-space calibration.

A.3 KV-Cache Communication and Heterogeneous Translation

KV-cache interfaces transfer attention state across model calls. Selective sharing chooses sender layers using attention importance and a layer prior (Shi et al., 2026a). Latent collaboration accumulates

intermediate computation in a shared working-memory cache and passes this state between role-specific agents (Zou et al., 2025). Reusable latent primitives compose recurring review, voting, and planning operations with internal KV-cache communication (Jin et al., 2026). These methods link the granularity of the transmitted state to the organization of the collaborative workflow.

Heterogeneous cache translation learns how sender tensors condition a receiver with its own representation geometry. Cache-fusion networks project source and target states into a receiver-side update and learn gates for the participating layers (Fu et al., 2025). Joint key–value translation and pooling produce compact updates for both shared-context and different-context exchanges (Rossi et al., 2026). Dense cache alignment uses reconstruction followed by generation-oriented training to transfer source context across models with different depths and KV layouts (Chen et al., 2026c). Its evaluation includes both receivers that observe the original context and receivers whose source information arrives through the cache. This line of work develops explicit transformations at the receiver's attention-state interface.

A.4 Shared Spaces, Model Composition, and Communication Training

Learned shared spaces make model composition a reusable operation. Cross-attention modules connect frozen language models so that one model can draw on representations produced by another (Bansal et al., 2024). Shared-space expert collaboration combines query-dependent routing with interaction layers that project selected experts' hidden states into a common space (Fein-Ashley et al., 2025). Latent-variable approaches use sparsity-regularized reconstruction to infer shared and private components of agent states and route them through prefix adaptation (Zheng et al., 2025). These works establish shared representations as a design for composing model capabilities.

Communication objectives also vary with the intended interaction. Parameter-efficient supervised training over multi-agent latent trajectories jointly optimizes how information is produced and consumed across successive roles (Yu et al., 2026b). Reconstruction-guided test-time optimization uses a question–latent–question cycle to update latent states before answer generation (Yuan et al., 2026). Hybrid diffusion–autoregressive systems train a projector from a diffusion planner's representations to an autoregressive executor's input embeddings (Berrayana et al., 2026). The Vision Wormhole trains each visual codec from its backbone's prompt-boundary state and next-token distribution, then fits reference maps on shared anchor texts. Its shared-space messages connect separately trained codecs at the visual input interface.

A.5 Message Compression and Cache Execution

Communication compression controls the amount of state stored and transported. Adaptive quantization allocates precision across KV layers and combines compressed caches with explicit fact extraction and model calibration (Kriuk & Ng, 2025). Cache-relay compression retains selected KV entries and folds a low-rank residual from discarded states into the retained representation (Li et al., 2026). Post-hoc sparse coding has also been applied to frozen Vision Wormhole messages, measuring payload size, reconstruction, downstream accuracy, and feature reuse (Wu & Zhu, 2026). These approaches operate on different message representations and compression stages; their relevant budgets include scalar precision, token or layer count, and the storage required for the coding mechanism.

Execution systems reuse attention state across agents and conversation phases. Online cross-context reuse estimates cache offsets from an adaptive anchor pool (Ye et al., 2025a); decoding-cache relay selectively recomputes positions affected by prefix changes (Geng et al., 2026). Multi-LoRA caching decomposes attention state into shared backbone and low-rank adapter components (Jeon et al., 2026). Collective cache sharing exploits repeated output blocks across synchronized agent rounds and stores differences relative to a shared copy (Bian et al., 2026). Persistent quantized caches support agent-context restoration on memory-constrained devices (Shkolnikov, 2026), while distributed workflow operators manage cache materialization, transfer, composition, and eviction (Sarker et al., 2026). Network-aware studies jointly schedule cache delivery and token prefill, or select between token and KV transport under computation and bandwidth constraints (Lee et al., 2026; Dai et al., 2026). This systems work makes transport and receiver execution explicit parts of the communication cost.

A.6 Multimodal and Domain-Specific Collaboration

Latent communication is also being developed for agents with distinct observations and execution budgets. Visual agent systems maintain separate perception and reasoning memories with entropy-dependent access (Yu et al., 2026a). Long-video collaboration assigns video segments to local agents, compresses their observations into shared-space tokens, and trains a coordinator through a staged curriculum (Chen et al., 2026b). Continuous slow–fast coupling projects a deliberative model's residuals into a fast model's input embeddings for real-time action selection (Li & Shi, 2026). Collaborative driving uses iterative latent deliberation, saliency-based state selection, and structured distillation to support decisions from partial observations (Chen et al., 2026d). In cross-hospital diagnosis, local agents distill clinical evidence into compact KV blocks, with separate alignment procedures for shared and heterogeneous backbones (Wang et al., 2026c). These applications connect message design to the division of observations, roles, and computation among agents.

A.7 Security of Latent Channels

Security research studies the handling of communicated state. Latent interventions test how attack-associated information propagates through hidden states and KV handoffs (Wang et al., 2026a). Cache-integrity studies bind transmitted tensors to sender, session, and message metadata through authenticated manifests (Brito & Baquero, 2026). Adversarial representation learning trains cache transformations to preserve task information while reducing reconstruction of sensitive source inputs (Asif et al., 2026). This work develops auditing, integrity, and disclosure controls for latent communication interfaces.

A.8 Continuous Reasoning and Internalized Collaboration

Continuous reasoning methods study how a model can perform intermediate computation through recurrent hidden states or continuous token representations (Hao et al., 2024; Zhang et al., 2025d; Zhu et al., 2025; Liu et al., 2024; Coda-Forno et al., 2025; Qu et al., 2025; Sui et al., 2025; Wang et al., 2025b; Zhang et al., 2025b). Internalized collaboration distills multi-agent debate into a single model and studies the resulting agent-specific activation directions (Yi et al., 2026). Related steering approaches learn a codebook of latent roles and compose them with a query-dependent router (Zeng et al., 2026). The Vision Wormhole uses a cached latent rollout as the input to a communication encoder. Its receiver conditions on the decoded message while performing its own role-specific computation. Appendix F specifies the implementation sources, tokenizer handling, and role sequences used to evaluate earlier communication channels.

A.9 Cross-Model Representation Alignment

Tokenizer transfer, vocabulary alignment, and cross-tokenizer distillation study compatibility between models with different token inventories (Minixhofer et al., 2024; Feher et al., 2025; Minixhofer et al., 2025; Li et al., 2025b; Remy et al., 2023; 2024; Tai et al., 2020; Pfeiffer et al., 2021; Rust et al., 2021; Vernikos & Popescu-Belis, 2021; Mundra et al., 2024; Yamaguchi et al., 2024; 2026; Moroni et al., 2025; Downey et al., 2023; Goddard & Neto, 2025; Sharthak et al., 2025; Liu et al., 2026). At the representation level, linear alignment, model stitching, and permutation-aware merging investigate correspondences between independently trained networks (Bansal et al., 2021; Lenc & Vedaldi, 2015; Ainsworth et al., 2022; Wortsman et al., 2022; Park et al., 2023b; Zou et al., 2023; Zhou et al., 2019). Research on relative representations and latent communication studies how semantic anchors support reuse across neural spaces and modalities (Moschella, 2024). We use affine regression on corresponding anchor representations to connect each codec to a chosen reference model. The encoder, decoder, and reference maps form a reusable interface for each backbone within the model pool.

A.10 Continuous Prompts and Multimodal Conditioning

Prompt Tuning and Prefix-Tuning demonstrate learned continuous conditioning of frozen language models (Lester et al., 2021; Li & Liang, 2021). Vision-language architectures condition language generation on image features processed by learned projectors or resamplers (Alayrac et al., 2022;

Li et al., 2023b). Contrastive image-text training studies shared representations across modalities (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023), and recent work examines alignment and reasoning within multimodal hidden spaces (Fein-Ashley & Fein-Ashley, 2025; Li et al., 2025a; Wang et al., 2025a).

The Vision Wormhole generates a continuous prompt in a receiver's image-token span from a sender's latent rollout. Its per-model self-distillation objective matches the text-conditioned backbone at the prompt boundary, and reference maps compose the resulting codecs for heterogeneous communication. This combination links continuous conditioning, cross-model alignment, and multi-agent inference in one modular architecture.

B Implementation Details

B.1 Main Heterogeneous MAS Model Configurations

Table 2 lists the heterogeneous MAS model configurations used in the main experiments.

Table 2: **Main heterogeneous MAS model configurations.** Two-backbone configurations alternate backbones across roles.

Backbone Setup	Backbones	Role assignment (P/C/R/J)
Two-backbone configurations		
2 backbones	Gemma-3-4B + SmolVLM2-2.2B	(SmolVLM2, Gemma-3-4B, SmolVLM2, Gemma-3-4B)
2 backbones	Qwen3-VL-2B + SmolVLM2-2.2B	(SmolVLM2, Qwen3-VL-2B, SmolVLM2, Qwen3-VL-2B)
2 backbones	Qwen3-VL-2B + Gemma-3-4B	(Gemma-3-4B, Qwen3-VL-2B, Gemma-3-4B, Qwen3-VL-2B)
2 backbones	LFM2.5-VL-1.6B + Gemma-3-4B	(LFM2.5-VL-1.6B, Gemma-3-4B, LFM2.5-VL-1.6B, Gemma-3-4B)
2 backbones	LFM2.5-VL-1.6B + Qwen3-VL-2B	(LFM2.5-VL-1.6B, Qwen3-VL-2B, LFM2.5-VL-1.6B, Qwen3-VL-2B)
Four-backbone pool (1.6B–4B)		
4 backbones	SmolVLM2-2.2B + LFM2.5-VL-1.6B + Gemma-3-4B + Qwen3-VL-2B	(SmolVLM2, LFM2.5-VL-1.6B, Gemma-3-4B, Qwen3-VL-2B)

B.2 Codec Training Setup (Shared Across Runs)

Objective. We train a latent-to-vision injection codec for each backbone model. Multi-model inference combines these per-model codecs with fitted reference-space maps.

Anchor corpora. We consider two anchor settings: (i) the **default** setting uses 3,000 texts drawn equally from `cos_e`, `OpenCodeReasoning`, and `PRM800K`; and (ii) the **weakly supervised** setting uses 90 texts, with equal contributions from the same sources.

Anchor construction. For each data source, we cap the number of sampled examples, concatenate sources, and shuffle deterministically.

Step-based random sampling. At each optimization step, we sample a mini-batch uniformly with replacement from the anchor pool.

Codec architecture hyperparameters. We use universal dimension $D = 512$ and $K = 1024$ learned message queries. A global query and a style query give $K_{\mathrm{u}} = 1026$ total tokens per message. The decoder uses $K_{\mathrm{img}} = 256$ image queries, with its output resampled to the receiver's image-span length. The codec uses 6 transformer layers with 8 attention heads and dropout 0.10. Codec training uses $T_{\mathrm{train}} = 1024$ latent steps.

Optimization and losses. We optimize with AdamW at a learning rate of 2×10^{-4} for 400 steps with batch size 2. The training objective is a weighted sum of three terms:

- Hidden-state MSE (weight 1.0).
- Logit alignment via KL divergence (weight 0.25; temperature 1.0).
- Injection-statistics regularizer (weight 0.1).

We apply gradient clipping with max-norm 1.0 and use standard numerical stabilization, including clipping for latent/logit/injection values and non-finite guards.

Rollout computation. Each training example produces one latent rollout from its teacher prompt. The frozen backbone feeds normalized hidden states back as continuous pseudo-tokens.

What changes across variants. Across codec variants, we vary the backbone model, the anchor corpus size, and the merge pairing used at inference. We keep the core codec architecture, optimizer/loss weights, step count, batch size, and the step-based sampling strategy fixed.

Merged codecs. For multi-model inference, we assemble the trained per-model codecs and fit universal-space alignment by ridge regression on texts from the codec-training pool. The default setting randomly selects 300 alignment texts; the weakly supervised setting reuses the entire training pool. The Qwen3-VL/Gemma-3 configuration uses Qwen3-VL as reference and ridge coefficient $\lambda = 10^{-3}$.

B.3 Experiment Runtime and Generation Budgets

Decoding and evaluation. The main experiments use sampling with temperature 0.6, top-p 0.95, and seed 42. Both methods use the task-specific generation caps in Table 3.

Role instructions. We use the text- and latent-context role prompts of Zou et al. (2025), with backbone-specific system messages and benchmark-specific answer formats. VW carries the latent context through the visual interface.

Hardware and model placement. All experiments are conducted on NVIDIA A6000 GPUs. For two-backbone runs in the main setting, we colocate both backbones on a single GPU. For the four-backbone pool, we use two GPUs and place half of the agents on each GPU.

Per-dataset token budgets. Table 3 lists the shared generation caps and default batch sizes.

Table 3: **Per-dataset generation budgets and default batch sizes.** We adopt the same maximum generation budgets as Zou et al. (2025).

Dataset	Max. new tokens	Default batch size
GSM8K	2048	12
ARC-Easy	2048	12
ARC-Challenge	2048	12
MedQA	4096	8
MBPP-Plus	4096	8
HumanEval-Plus	4096	8
GPQA	8192	4
AIME 2024	20000	4
AIME 2025	20000	4

Dynamic batching and time reporting. All methods in the main experiments use the default batch sizes in Table 3 and the same out-of-memory (OOM) backoff schedule, {12, 8, 4, 2, 1}. Starting from the dataset-specific default, we reduce the batch size along this schedule until the run succeeds. We report batch-normalized runtime (seconds/query) as the total wall-clock time across evaluation batches divided by the number of evaluated examples. Timing includes intermediate-role computation, message processing, and final-answer generation.

C Additional Details: Codec Training, Alignment, and Inference

This appendix specifies the codec architecture, alignment fit, and inference procedure.

C.1 Training a Codec for a Specific VLM

Pseudo-token normalization. Latent rollouts feed back continuous pseudo-token embeddings derived from hidden states. A per-model normalization operator sets their magnitude relative to the token-embedding scale:

$$\mathrm{NormMatch}_i(h) \;=\; \alpha_i \cdot \frac{h}{\|h\|_2 + \epsilon}, \tag{5}$$

where α_i is the typical token-embedding norm for model i, e.g., $\alpha_i = \mathbb{E}_{w\sim\mathcal{V}_i}\|E_i(w)\|_2$, and ϵ is a small constant. The normalized pseudo-token has norm approximately α_i.

Latent rollout with cached context. Let the prompt produce cached attention keys/values (equivalently, a fixed conditioning context). A rollout step appends a single pseudo-token embedding x_t and computes the next hidden state at the new position. Repeating for T steps yields $H_i \in \mathbb{R}^{T\times d_i}$. The rollout length T is a fixed hyperparameter that bounds message extraction cost.

Perceiver resampler encoder. The encoder $\mathcal{E}_i$ maps H_i to K_{u} tokens of dimension D. We first project the rollout into the universal dimension:

$$Z = H_i P_i \in \mathbb{R}^{T\times D}, \tag{6}$$

for a learned matrix $P_i \in \mathbb{R}^{d_i\times D}$. We then maintain a fixed set of learned queries $Q^{(0)} \in \mathbb{R}^{K_{\mathrm{u}}\times D}$ and update them through L cross-attention blocks:

$$\tilde{Q}^{(\ell+1)} = Q^{(\ell)} + \mathrm{MHA}\Big(\mathrm{LN}(Q^{(\ell)}), \mathrm{LN}(Z), \mathrm{LN}(Z)\Big), \tag{7}$$

$$Q^{(\ell+1)} = \tilde{Q}^{(\ell+1)} + \mathrm{FFN}\Big(\mathrm{LN}(\tilde{Q}^{(\ell+1)})\Big), \tag{8}$$

for $\ell = 0, \ldots, L-1$. This is the Perceiver resampler pattern (Alayrac et al., 2022): a constant number of queries attends to a variable-length latent sequence.

Global and style tokens. In addition to the K message tokens, we include a global token (for pooling) and a style token (for scale/statistics cues). The style statistic vector is

$$s(H_i) = \left[\mathrm{mean}(H_i),\, \mathrm{std}(H_i),\, \frac{1}{T}\sum_{t=1}^{T}\|H_{i,t}\|_2\right] \in \mathbb{R}^3, \tag{9}$$

which is mapped by a small MLP into $\mathbb{R}^D$ and added to the style token. The style token thus includes the rollout's mean, standard deviation, and average vector norm.

Universal-to-vision decoder. The decoder $\mathcal{D}_i$ mirrors the resampler pattern. A learned set of K_{img} image queries attends to the (possibly concatenated) universal tokens to produce K_{img} vectors, which are linearly projected into $\mathbb{R}^{d_i}$ to form Δ_i. A gate $g_i \in (0, 1)$ is predicted from a pooled representation of the universal tokens. The gate sets a learned, example-dependent scale for the residual.

Dummy-image baseline and length resampling. Different VLMs use different image-token lengths $L^{(i)}_{\mathrm{img}}$. We compute once per model a baseline image-span embedding $\bar{X}^{(i)}_{\mathrm{img}}$ using a fixed dummy image. At inference, we resample the decoded $\Delta_i \in \mathbb{R}^{K_{\mathrm{img}}\times d_i}$ to the required span length $\mathrm{Resample}(\Delta_i; L^{(i)}_{\mathrm{img}}) \in \mathbb{R}^{L^{(i)}_{\mathrm{img}}\times d_i}$ using linear interpolation along the token index. The injected span is written as in Eq. (1).

Distillation signals and where they are taken. The teacher processes the anchor text m and provides two targets at the prompt boundary: a hidden-state vector and a distribution over the next token. The teacher-side rollout, initialized from this boundary, supplies the encoder input. The student processes the corresponding vision-conditioned prompt, and the loss matches its boundary state and next-token distribution to these targets.

C.2 Affine Alignment in Universal Space

Reference-space construction. Each encoder produces tokens in its learned coordinates. We fit affine maps between each model's tokens and those of the reference model, using the same subset of codec-training texts and matching query positions.

Closed-form ridge regression. Given anchor texts $\{m_j\}_{j=1}^{M}$, we obtain token matrices $U_i(m_j)$ and $U_r(m_j)$ for model i and reference r. We flatten across anchors and token positions to form $X_i \in \mathbb{R}^{(MK_u)\times D}$ and $Y_r \in \mathbb{R}^{(MK_u)\times D}$, and solve

$$\min_{W,b} \quad \|X_i W + \mathbf{1}b^\top - Y_r\|_F^2 + \lambda\|W\|_F^2, \tag{10}$$

Let X_c and Y_c denote the column-centered matrices, with column means $\bar{x}$ and $\bar{y}$. The solution is

$$W = (X_c^\top X_c + \lambda I)^{-1} X_c^\top Y_c, \qquad b = \bar{y} - W^\top \bar{x}. \tag{11}$$

We fit the forward map A_i^{out} and reverse map A_i^{in} separately, exchanging the regression inputs and targets.

Component count. For a fixed reference, each additional model contributes one encoder, one decoder, and two affine maps. The reference model uses identity maps. Thus, the architecture contains N codecs and $2(N-1)$ fitted maps.

C.3 Inference Protocols and Role Interaction

Unified "read–think–write" abstraction. The sequential workflow applies the following operations to each intermediate role:

1. **Read memory:** decode aggregated $U_{\text{mem}}^{\text{ref}}$ into an injected image span for the current agent.
2. **Think:** run the frozen backbone conditioned on the injection to produce either (a) a latent rollout H_i (for intermediate roles), or (b) an output token sequence (for the final role).
3. **Write memory:** encode H_i into a new universal message U_i^{ref} and add it to memory.

Accumulated context. We follow Zou et al. (2025) for sequential context accumulation. The Planner, Critic, and Refiner execute in order, each appending one message to memory; the Judger generates the final answer after reading the accumulated messages.

Message processing. Reading $|\mathcal{M}|$ messages involves decoder cross-attention over $|\mathcal{M}|K_u$ tokens, followed by a fixed-length image span of $L_{\text{img}}^{(i)}$ tokens in the backbone. Sender computation includes prompt prefill, a cached latent rollout, and codec encoding.

D OCR-Based Image Relay Baseline

The OCR relay sends natural-language messages as rendered images within the four-role workflow. Each sender generates a text message, a renderer converts it to an image, and the receiver reads the image through its standard visual input.

Table 4 compares the OCR relay with the TextMAS and VW runs from Figures 4 and 5 on Gemma/Qwen and SmolVLM2/Qwen. VW achieves higher accuracy than the OCR relay on all nine benchmarks in both configurations, together with lower macro-average runtime in each configuration.

E Rollout-Length Sweep for LatentMAS-Hybrid

We vary the rollout length of *LatentMAS-Hybrid*, a public fork of LatentMAS supporting heterogeneous role assignments (Zou et al., 2025; nhminle, 2026), and measure task accuracy and answer perplexity.

Table 4: **OCR-based image-relay baseline.** OCR renders the sender's generated text as an image, which the receiver reads through its native visual input. Each cell reports accuracy (%) and average wall-clock time (s/query).

	P/R: Gemma-3-4B C/J: Qwen3-VL-2B			P/R: SmolVLM2-2.2B C/J: Qwen3-VL-2B		
Dataset	**Text**	**VW**	**OCR**	**Text**	**VW**	**OCR**
GSM8K	80.8% / 27.3s	76.2% / 26.7s	72.8% / 112.9s	64.3% / 63.8s	74.8% / 33.5s	58.0% / 115.0s
ARC-Easy	93.4% / 33.1s	92.4% / 22.0s	82.0% / 113.9s	88.6% / 51.1s	92.3% / 28.2s	73.7% / 113.9s
ARC-Challenge	86.0% / 49.0s	82.1% / 29.5s	72.4% / 113.9s	78.2% / 68.5s	81.7% / 38.2s	64.0% / 112.7s
GPQA	29.8% / 348.4s	39.9% / 174.6s	28.8% / 216.2s	32.3% / 483.3s	37.9% / 225.5s	28.3% / 210.1s
MedQA	53.3% / 91.5s	48.0% / 83.0s	43.3% / 137.4s	44.7% / 125.0s	47.0% / 93.4s	37.0% / 133.6s
MBPP-Plus	50.5% / 108.7s	51.3% / 69.2s	23.0% / 132.8s	37.8% / 125.6s	47.9% / 79.9s	15.9% / 131.9s
HumanEval-Plus	40.9% / 121.6s	37.2% / 80.1s	15.8% / 136.4s	31.1% / 126.1s	40.9% / 100.3s	7.9% / 132.6s
AIME 2024	23.3% / 1314.4s	36.7% / 385.8s	3.3% / 535.6s	13.3% / 2806.9s	23.3% / 513.3s	6.7% / 518.8s
AIME 2025	16.7% / 1432.9s	26.7% / 382.0s	10.0% / 532.7s	16.7% / 1996.1s	23.3% / 505.7s	16.7% / 519.6s
Macro Avg.	52.7% / 391.9s	54.5% / 139.2s	39.1% / 225.8s	45.2% / 649.6s	52.1% / 179.8s	34.2% / 220.9s

Sweep setup. We run a rollout-length sweep on GSM8K using a fixed 200-example subset. We use the public implementation described in Appendix F. The rollout-length grid is shown in Figure 9. The sweep uses greedy decoding, a 256-token answer budget, and batch size 4. We compare the Qwen/LFM and Qwen/Gemma configurations in Table 2. Table 5 reports representative rollout lengths and VW's main-experiment predictions on these same questions and role assignments.

Table 5: **LatentMAS-Hybrid and VW on the same GSM8K subset.** Accuracy (%), with correct counts in parentheses. VW uses the main-run generation settings in Appendix B.3.

Method	Gemma / Qwen	LFM / Qwen
LatentMAS-Hybrid, 16 steps	0.5	23.5
48 steps	0.0	26.0
128 steps	0.0	16.5
192 steps	0.5	18.0
256 steps	0.0	20.0
512 steps	0.0	12.5
1024 steps	0.5	14.0
VW	**79.0** (158)	**78.0** (156)

Metrics. At each rollout length, we record GSM8K exact-match accuracy and compute answer perplexity with three language models from the Llama, Qwen, and Gemma families. We also report the mean of their perplexities.

Results. For Qwen3-VL-2B + Gemma-3-4B, accuracy never exceeds 0.5%, and mean judge PPL rises from 8.1×10^5 at the shortest evaluated rollout to 8.6×10^7 at the longest. For Qwen3-VL-2B + LFM2.5-VL-1.6B, mean judge PPL is around 10–12 for shorter rollouts, then rises to 8.1×10^5 and remains in the 10^6–10^7 range for longer rollouts while accuracy stays around 12–20%.

F Implementations of Prior Latent Communication Channels

We compare Vision Wormhole (VW) with CIPHER (Pham et al., 2024) and LatentMAS-Hybrid (nhminle, 2026) in the four-role Qwen3-VL-2B-Thinking/Gemma-3-4B workflow. Related interfaces study pairwise activation and cache transfer (Ramesh & Li, 2025; Fu et al., 2025).

Setup. Evaluation uses the first 256 GSM8K test problems and the full HumanEval-Plus test set. Final-answer caps follow Table 3. Cross-tokenizer handling follows LatentMAS-Hybrid (nhminle, 2026). Table 6 gives the results and channel settings.

Appendix E reports the LatentMAS-Hybrid rollout-length sweep.

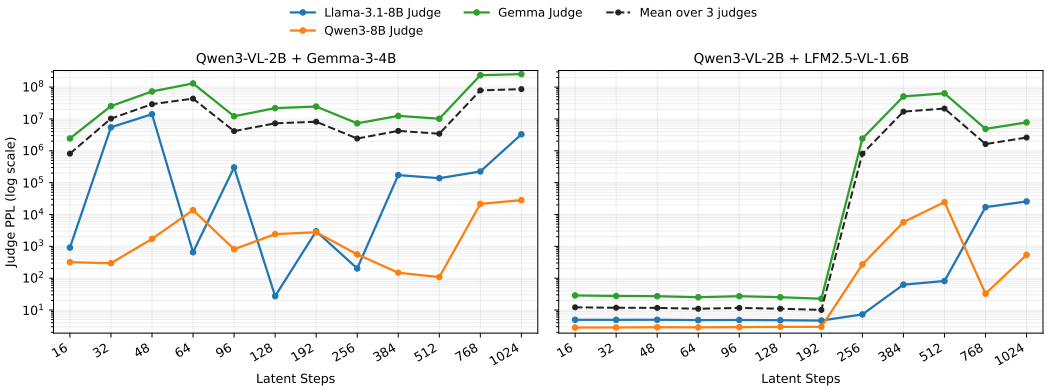

Figure 9: **Answer perplexity across rollout lengths for the evaluated LatentMAS-Hybrid configurations.** Curves show perplexity assigned to the generated answers by the external judge models and their mean (dashed black).

Table 6: **System accuracy for heterogeneous latent communication channels.** Accuracy (%), with correct counts in parentheses. All methods use the Gemma/Qwen role assignment in Table 2.

Channel	GSM8K	HumanEval+
Vision Wormhole	**77.0%** (197)	**37.2%** (61)
CIPHER	65.6% (168)	21.3% (35)
LatentMAS-Hybrid	0.0% (0)	0.0% (0)

Channel settings: 256 embedding tokens (CIPHER); 32 latent steps (LatentMAS-Hybrid).

G Visual Interface Ablation on Qwen

We compare the visual interface with a learned text-prefix interface for Qwen on mathematical and medical reasoning (Table 7). Both interfaces consume the same intermediate-role trajectories and decode their accumulated messages into a 64-token input span. The visual interface uses the model's native image-token span; the text-prefix interface adds its decoded residual to ordinary space-token embeddings. Each retains its native token structure and attention behavior.

Table 7: **Visual interface ablation on Qwen.** Accuracy (%); differences in pp.

Method	GSM8K	MedQA
Qwen + VW	**75.8**	**34.4**
Qwen + soft-prefix	60.2	27.0
Δ (pp)	+15.6	+7.4

Setup. We train separate visual and text-prefix codecs with matched architectures, paired initialization, identical anchor schedules, and the same prompt-boundary hidden-state and KL objectives. The update count and batch size follow Appendix B.2; initialization uses the evaluation seed in Appendix B.3. The RMS regularizer targets each interface's own carrier embeddings. Evaluation uses 256 questions per benchmark, with rollout length $T = T_{\text{train}}$ per intermediate role (Appendix B.2). The two interfaces share the final role instructions and question set; decoding parameters and generation caps follow Appendix B.3. Scoring follows Section 4.1.

Table 7 shows accuracy gains of 15.6 percentage points on GSM8K and 7.4 points on MedQA with visual reception.

H Additional Communication Experiments

H.1 Intermediate Text Budgets

We vary the maximum generation length of each intermediate TextMAS role in the Gemma/Qwen configuration of Table 2. The question sets follow Appendix F; final-answer budgets and sampling settings follow Appendix B.3. All runs use batch size 4. Table 8 shows that GSM8K accuracy varies little across these budgets, while HumanEval+ achieves its highest score with the full task budget.

Table 8: **TextMAS with capped intermediate messages.** Accuracy (%); "Task cap" uses the per-task limit in Table 3.

Intermediate token cap	GSM8K	HumanEval+
256	79.3	34.1
512	**80.5**	—
1024	78.1	32.9
Task cap	**80.5**	**40.9**

Concise intermediate messages. VW outperforms both standard and concise TextMAS in five of the six main configurations (Table 9). We evaluate the first 128 GSM8K test questions per configuration, using the shortest intermediate cap in Table 8 and instructing concise TextMAS roles to keep their messages brief. Final-answer generation follows Appendix B.3; standard TextMAS and VW use the corresponding main-experiment predictions.

Table 9: **Standard and concise text communication across all six configurations.** GSM8K accuracy (%), with correct counts in parentheses. The pooled row aggregates 768 configuration–question evaluations.

Configuration	TextMAS	Concise TextMAS	VW
SmolVLM2 / Gemma	65.6 (84)	69.5 (89)	**82.8** (106)
SmolVLM2 / Qwen	64.8 (83)	50.8 (65)	**72.7** (93)
Gemma / Qwen	**78.9** (101)	75.0 (96)	74.2 (95)
LFM / Gemma	77.3 (99)	74.2 (95)	**85.2** (109)
LFM / Qwen	68.0 (87)	64.8 (83)	**74.2** (95)
Four-model pool	57.8 (74)	63.3 (81)	**72.7** (93)
Pooled	68.8 (528)	66.3 (509)	**77.0** (591)

H.2 Parallel Multi-Agent Topology

We also evaluate a parallel topology with three independent specialists and a final aggregator, using Gemma-3-4B and Qwen3-VL-8B-Thinking. Gemma supplies two specialists; Qwen supplies the third specialist and the aggregator. Aggregation instructions are adapted to each communication channel, and VW reuses the trained codec. Table 10 reports accuracy on a prefix of the GSM8K test set and the full HumanEval+ test set, with sample counts in the table and final-answer caps from Table 3. VW achieves the highest accuracy on both tasks.

Table 10: **Communication channels in a parallel MAS.** Accuracy (%), with correct counts in parentheses.

Task	n	Text	OCR relay	VW
GSM8K	128	90.6 (116)	92.2 (118)	**96.9** (124)
HumanEval+	164	64.0 (105)	44.5 (73)	**65.2** (107)

H.3 Cross-Model Key–Value Retrieval

We evaluate Qwen-to-Gemma retrieval from key–value memories built from GSM8K question–answer pairs (Table 11). Qwen reads the target pair and 16 distractors. Gemma receives the message and query question, then selects the answer from shuffled candidate values. VW uses the frozen main-experiment codec and alignment with rollout length $T = T_{\text{train}}$ (Appendix B.2); text communication uses a sender-generated memory note. We evaluate 100 queries per seed over three seeds, with shared queries, candidates, receiver decoding, and a 256-token answer cap.

Table 11: **Key–value retrieval through the Qwen → Gemma interface.** Exact-selection accuracy (%), reported as mean ± standard deviation across seeds.

Communication	Retrieval accuracy
Text message	34.0 ± 8.7
Vision Wormhole	$\mathbf{36.0} \pm 4.6$

Retrieval with larger memories. VW retrieves more accurately than text communication in larger memories (Table 12). We vary the number of distractors for Gemma-to-Gemma communication using its frozen codec, with 30 queries per seed for each condition over the same seeds. Text senders use the shortest intermediate cap in Table 8.

Table 12: **Gemma retrieval as memory size increases.** Exact-selection accuracy (%), reported as mean ± standard deviation across seeds.

Distractor entries	Text message	Vision Wormhole
0	$\mathbf{100.0} \pm 0.0$	88.9 ± 5.7
1	$\mathbf{91.1} \pm 4.2$	64.4 ± 7.9
4	$\mathbf{66.7} \pm 7.2$	53.3 ± 7.2
16	17.8 ± 9.6	$\mathbf{34.4} \pm 6.8$
64	3.3 ± 4.7	$\mathbf{26.7} \pm 9.8$

H.4 Representation Alignment

Aligned messages achieve lower reconstruction error than cross-text substitutions and shuffled-anchor maps in both directions (Table 13). We train Qwen and Gemma ablation codecs using the visual-interface recipe in Appendix G and fit affine maps on paired calibration anchors. Evaluation uses 354 held-out texts across three training seeds, measuring prompt-boundary KL divergence from the receiver's text-conditioned teacher to its message-conditioned output. The controls substitute messages across texts or shuffle the anchor correspondences when fitting the maps.

Table 13: **Aligned messages reduce reconstruction error.** Mean KL divergence (↓) on held-out texts.

Direction	Aligned message	Cross-text message	Shuffled-anchor map
Gemma → Qwen	**0.227**	0.513	0.464
Qwen → Gemma	**1.864**	6.344	5.408

Affine calibration of pooled representations. We additionally examine a larger Qwen/Gemma codec with 1024-dimensional messages on 1,000 Flickr30K captions. A ridge-affine readout is fitted to the paired mean-pooled representations, with ridge coefficient 0.1. Table 14 reports calibration-set centered kernel alignment (CKA) and caption-identity retrieval, averaged over both directions. Calibration increases representation agreement and cross-model retrieval accuracy; shuffling caption correspondences after calibration gives CKA 0.003.

Table 14: **Affine calibration of cross-model representations.** CKA and retrieval accuracy (%) on the paired calibration captions.

Readout	CKA	Top-1	Top-5	Top-10
Raw pooled representations	0.261	0.20	1.15	2.45
Affine-calibrated	**0.840**	**10.05**	**17.15**	**22.15**
Chance retrieval	—	0.10	0.50	1.00

H.5 Input-Dependent Decoder Gates

The frozen Qwen decoder assigns substantially larger gates to caption-derived messages than to Gaussian inputs (Table 15). We compare messages produced from 20 Flickr30K captions with 200 Gaussian message tensors of the same shape, sampled from $\mathcal{N}(0, 0.5^2)$. The decoder weights are shared across both conditions.

Table 15: **Qwen decoder gate response.** Mean gate value and standard deviation across inputs.

Decoder input	Gate value
Gaussian message	0.0185 ± 0.0001
Caption-derived message	0.6551 ± 0.0066

H.6 Completeness of Intermediate Role Outputs

We examine intermediate outputs in the sequential Gemma-3-4B/Qwen3-VL-8B TextMAS configuration. Claude Opus grades each planner, critic, and refiner output against the reference answer as correct, partial, incorrect, or no attempt. Table 16 counts roles graded fully correct for each question. The prevalence of partial solutions motivates preserving intermediate reasoning across roles.

Table 16: **Completeness of sender outputs.** Percentage of questions with the indicated number of fully correct intermediate roles.

Fully correct senders	GSM8K ($n = 100$)	HumanEval+ ($n = 100$)	AIME 2024 ($n = 30$)
Three	39.0	8.0	0.0
Two	45.0	24.0	13.3
One	12.0	53.0	13.3
Zero	4.0	15.0	73.3