LATENTCOLLAB: VISION-LANGUAGE AGENT COLLABORATION BEYOND VERBALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language agents extend vision-language models from isolated multimodal reasoning toward collaborative intelligence, but existing systems primarily exchange natural-language messages. This forces task-conditioned multimodal computation through a verbalization bottleneck that may omit fine-grained visual evidence, while KV-cache communication incurs substantial memory and communication overhead. We introduce LatentCollab, a latent collaboration framework that enables vision-language agents to exchange compact continuous messages without intermediate text generation. The sender recursively constructs a trajectory of final-layer hidden states. A lightweight translator, the only trainable component, maps the trajectory into a receiver-compatible continuous message, which is inserted into the receiver's input-embedding sequence after its visual tokens. Across seven benchmarks and three cross-scale or cross-family sender–receiver settings, LatentCollab outperforms receiver-only and text communication on all 21 pairs, and both KV-cache baselines on 18, with lower latency than all evaluated communication baselines. These results establish latent collaboration as an effective and efficient paradigm for vision-language agents.

1 INTRODUCTION

In recent years, vision-language models (VLMs) have made substantial progress in visual perception, knowledge-intensive reasoning, and multimodal interaction (Liu et al., 2024a; Bai et al., 2025). Beyond serving as standalone prediction models, VLMs are increasingly configured as vision-language agents that observe multimodal inputs, follow role-specific instructions, and produce intermediate analyses. Because models differ in scale, architecture, training data, and visual encoding strategy, different VLM agents may perform distinct task-conditioned computations even when given the same image and question. Input views, role instructions, and information-processing strategies further differentiate agent behaviors, enabling complementary roles in observation, analysis, and final decision-making. This supports joint reasoning among VLM agents with complementary capabilities.

Existing multi-VLM collaboration methods mainly follow two paradigms. One line of work votes on, selects, or aggregates the final answers produced by multiple agents (Alazraki et al., 2023), while another exchanges intermediate analyses through natural language and organizes collaboration through sequential interaction, debate, or structured workflows (Chen et al., 2023; Adhikari & Lapata, 2025). The former exploits prediction diversity but leaves the intermediate computations of individual agents isolated. The latter enables direct information exchange, but requires the sender to decode its continuous computation into discrete text and the receiver to re-encode that text into its own representation space. This process introduces additional generation and context-processing costs and limits communication to information that can be explicitly verbalized. Fine-grained visual attributes, spatial relations, and distributed evidence may therefore be omitted. As illustrated in Fig. 1, the sender captures the task-critical fence–bench relation, whereas the textual message retains only the scene gist, causing the receiver to predict *tennis court* rather than *bench*. Natural language is thus broadly compatible, but may introduce a verbalization bottleneck between collaborating VLM agents.

Continuous representations avoid explicit verbalization. Transferring the sender's KV cache (Fu et al., 2025b; Chen et al., 2026) preserves layer-wise Key–Value states and rich multimodal context. However, its communication payload grows with model depth and cached sequence length, resulting in substantial communication, memory, and bandwidth overhead. As shown in Fig. 1, KV-cache

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

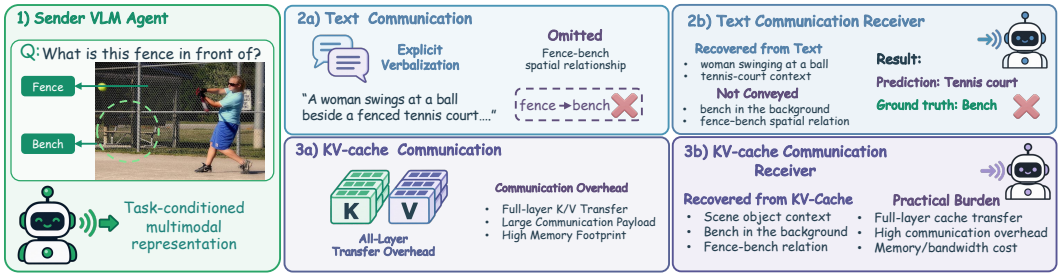


Figure 1: Comparison of text and KV-cache communication. Text is compact but may omit task-relevant visual relations, whereas KV-cache transfer preserves multimodal context at substantially higher communication and memory cost. This trade-off motivates compact latent communication.

communication can preserve evidence omitted by textual messages, but requires full-layer cache transfer. Latent collaboration should retain task-conditioned information without this overhead.

Compared with layer-wise caches, final-layer hidden states offer a more compact interface. They also provide a prediction-oriented multimodal readout, incorporating task-relevant visual information before vocabulary decoding (Neo et al., 2025). Prior work has used such states for continuous reasoning (Hao et al., 2024) and latent communication among language agents (Du et al., 2025; Zou et al., 2025). Building on these formulations, the sender recursively rolls out multiple steps to construct a trajectory jointly conditioned on the image, question, and optional role instruction, externalizing its task-conditioned computation without generating intermediate text or transferring layer-wise caches. Our analyses show that these trajectories vary systematically with the visual and textual inputs and support cross-view retrieval of matching image–question samples. However, sender and receiver states are not natively aligned when the models differ in hidden width and independently learned representation space. A learned transformation is thus needed for effective cross-model use.

To this end, we propose **LatentCollab**, a latent collaboration framework for VLM agents based on final-layer hidden states. Instead of generating a textual message, the sender recursively constructs a final-layer hidden-state trajectory by using its current hidden state as the continuous input to the next rollout step. A lightweight translator maps this trajectory into a receiver-compatible continuous message, which is inserted into the receiver's input-embedding sequence after its visual tokens. Both VLM agents remain frozen, with training limited to the lightweight translator. LatentCollab unifies latent rollout, translation, and continuous message injection for heterogeneous VLM collaboration.

We evaluate LatentCollab on seven vision-language benchmarks under three sender–receiver configurations, covering cross-scale and cross-family collaboration. We compare it with receiver-only inference, natural-language communication, and two KV-cache communication baselines. Across all 21 benchmark–configuration pairs, LatentCollab improves over both receiver-only inference and natural-language communication, and it outperforms both KV-cache communication baselines on 18 of the 21 pairs. These results support latent communication as an effective medium for collaboration through continuous message exchange among VLM agents across model scales and families.

Our contributions are threefold. First, we investigate task-conditioned trajectories of final-layer hidden states as a continuous collaboration medium between vision-language agents and evaluate their input conditionality, cross-view identifiability, and downstream utility. Second, we introduce LatentCollab, a latent collaboration framework that unifies text-free state rollout, latent translation, and continuous message injection. Third, we systematically evaluate LatentCollab across three sender–receiver configurations and seven vision-language benchmarks. Under matched evaluation conditions, it improves over receiver-only inference and natural-language communication on all 21 benchmark–configuration pairs and outperforms both KV-cache baselines on 18 pairs.

2 Related Work

VLM Agents and Collaboration. Vision-language models have increasingly been instantiated as multimodal agents that invoke external visual tools and compose their outputs for complex tasks (Liu et al., 2023). Collaboration among VLM agents has been organized through output-level ensem-

Table 1: Input conditionality of raw latent trajectories under controlled visual and textual interventions. Higher values indicate stronger preservation of sample identity.

Condition	R@1	R@5	Cos.	AUROC
Same pair	**43.75**	**54.17**	**0.554**	**90.18**
Remove image	1.30	5.21	0.061	57.41
Wrong image	0.78	3.12	0.048	55.97
Wrong question	4.69	10.42	0.136	64.42

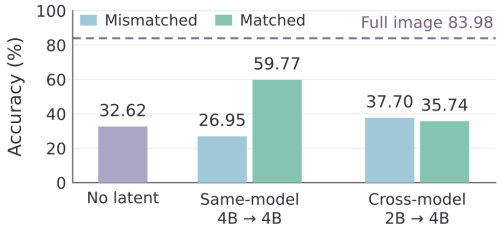


Figure 2: Raw latent transfer with Qwen3-VL. Matched latents help in the same-model setting but not direct cross-model transfer. The dashed line marks the full-image reference.

bling, language-based coordination of complementary models, multimodal debate, disagreement-triggered tool recruitment, and structured workflows with role-specific routing and iterative verification (Alazraki et al., 2023; Chen et al., 2023; Adhikari & Lapata, 2025; Sivakumaran et al., 2026; Lassoued et al., 2026). Although these approaches support task decomposition and decision aggregation, coordination is primarily mediated by decoded answers, textual analyses, or tool outputs rather than direct transfer of agents' internal multimodal states. Consequently, they do not directly exchange the continuous, task-conditioned multimodal representations formed inside VLM agents.

Communication among Language-Model Agents. Language-model agents commonly coordinate through natural-language messages, including role-conditioned dialogue, multi-round debate, and programmable or structured workflows (Li et al., 2023a; Wu et al., 2024; Hong et al., 2024; Du et al., 2023). Beyond text, recent methods communicate continuous signals through output embeddings, intermediate activations, or state-transition trajectories (Pham et al., 2024; Ramesh & Li, 2025; Tang et al., 2025). Other approaches exchange final-layer hidden states or latent working memories (Du et al., 2025; Zou et al., 2025), whereas KV-cache communication transfers token- and layer-specific Key–Value states, with its payload scaling with both cached sequence length and model depth (Fu et al., 2025b; Chen et al., 2026). Continuous latent reasoning within a single model does not itself constitute inter-agent communication (Hao et al., 2024). Existing activation- and hidden-state communication is primarily evaluated on language-only tasks (Ramesh & Li, 2025; Du et al., 2025; Zou et al., 2025), whereas VLM cache transfer communicates full layer-wise states (Chen et al., 2026). We study multimodal latent communication between heterogeneous VLM agents through compact continuous messages without intermediate text generation or full KV-cache transfer.

3 Method

3.1 Empirical Motivation

Motivated by the trade-off between potentially lossy text communication and expensive full KV-cache transfer, we conduct controlled experiments to assess whether final-layer rollout states can support a compact latent collaboration interface between VLM agents. Specifically, we examine whether these states are conditioned on the current multimodal input, whether the model that produces them can use its own rollout trajectory for task prediction, and, after establishing this within-model utility, whether the trajectory can also be used by a different model whose representation distribution and hidden dimensionality differ from those of the sender. We refer to the sequence of final-layer hidden states produced during continuous rollout as the *raw latent trajectory*, or *raw latents* for short.

Observation 1: Rollout trajectories are input-conditioned. For each image–question pair, we extract the final-layer hidden states from continuous rollout by a frozen VLM. We apply controlled interventions to either input modality. A JPEG-compressed version of the image serves as a benign visual perturbation, while the remaining conditions remove the image, replace it with an image from another sample, or replace the question. We measure sample identity preservation via trajectory cosine similarity, cross-view retrieval, and matched-versus-unmatched discrimination.

As shown in Table 1, trajectories from the same image–question pair remain consistent and identifiable under mild JPEG compression. Removing or replacing the image or question substantially degrades

all metrics. In particular, both image and question replacement alter the trajectory, showing that it is jointly conditioned on the visual content and the query rather than determined by generic rollout dynamics. Together, these results show that the trajectory carries a multimodal signal jointly shaped by the image and question, providing an empirical basis for its use as a communication medium.

Observation 2: Raw latents are usable by their source model. We next examine whether a model can use the latents produced by its own rollout. The same frozen VLM serves as both the sender and receiver. The sender produces a raw latent trajectory comprising 32 continuous rollout states from the original image and question, while the receiver answers the question under receiver-side visual ablation, without direct access to the image. This controlled setting isolates the task information carried by the trajectory. We compare matched latents with mismatched latents drawn from other samples and a no-latent baseline.

In the same-model setting in Fig. 2, matched latents achieve 59.77% accuracy, whereas the mismatched and no-latent controls remain near 30%. This result shows that the raw rollout trajectory contains task-relevant information and that the model producing the trajectory can use this information for prediction.

Observation 3: Raw latents are not directly usable across models. After confirming that raw latents can be used by their source model, we further examine their utility in a cross-model setting. We instantiate two distinct frozen VLMs as the sender and receiver and retain the same receiver-side visual ablation and the same matched, mismatched, and no-latent conditions. Because the two models have different hidden dimensions, the sender latents are zero-padded to the receiver hidden dimension and then directly provided to the receiver. This setting tests whether a different model can directly use the sender's raw latents when their representation distributions and hidden dimensions differ.

In the cross-model setting in Fig. 2, matched latents achieve 35.74% accuracy and provide no measurable benefit over mismatched or no-latent inputs. Together with the same-model result, this observation shows that raw rollout trajectories can be used by the model that produces them, but are not directly usable by another model with a different representation distribution and hidden dimensionality. These findings motivate a latent collaboration framework that jointly specifies latent trajectory construction, cross-model translation, and continuous message injection.

3.2 LatentCollab: A Latent Collaboration Framework for VLM Agents

The empirical observations in Sec. 3.1 show that final-layer rollout trajectories retain input-conditioned multimodal information and are usable by their source model, yet are not directly usable by a different receiver. We therefore propose *LatentCollab*, which transfers task-conditioned computation through a compact continuous message. The overall framework is illustrated in Fig. 3.

For instance i, let K be the trajectory length; d_s and d_r the sender and receiver hidden widths; $\mathbf{c}_i^s$ the sender context; $\mathbf{X}_i^r$ the receiver input-embedding sequence before message insertion; and y_i^r a receiver output sequence. One collaboration round is

$$\begin{aligned} \mathbf{H}_i^s = \mathcal{G}_S(\mathbf{c}_i^s) \in \mathbb{R}^{K\times d_s}, \qquad \mathbf{Z}_i = T_\phi(\mathbf{H}_i^s) \in \mathbb{R}^{K\times d_r}, \\ p_{\mathrm{LC}}\left(y_i^r \mid \mathbf{c}_i^s, \mathbf{X}_i^r\right) = p_R\left(y_i^r \mid \widetilde{\mathbf{X}}_i^r(\mathbf{Z}_i)\right), \end{aligned} \tag{1}$$

where $\mathcal{G}_S$ is sender-side continuous rollout, T_ϕ the receiver-grounded translator, $\widetilde{\mathbf{X}}_i^r(\mathbf{Z}_i)$ the receiver sequence after message insertion, p_R the frozen receiver distribution, and p_{LC} the resulting LatentCollab distribution. Together, these operations form a differentiable path from sender-side computation to receiver-side generation without modifying either VLM.

Final-Layer Latent Collaboration

Sender-Side Continuous Rollout. The sender externalizes its task-conditioned computation as a trajectory of final-layer states. Standard decoding projects each state to a vocabulary distribution, selects a discrete token, and re-embeds it for the next step. LatentCollab bypasses this discrete interface: it applies neither vocabulary projection nor token sampling, and directly reuses each state as the next continuous input embedding.

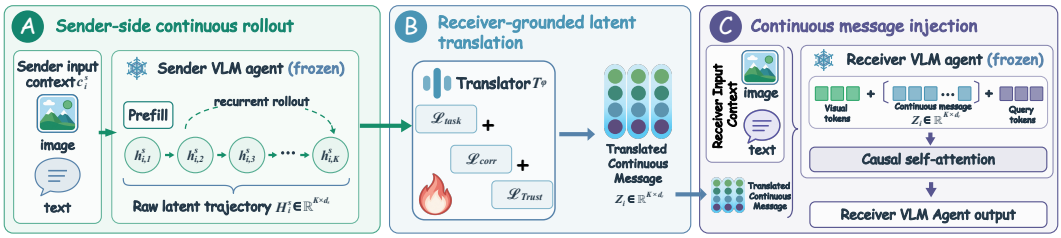


Figure 3: **Overview of LatentCollab.** A frozen sender produces a continuous latent rollout, which a trainable translator maps into the frozen receiver's embedding space and injects between its visual and query tokens. Collaboration avoids intermediate text generation and full KV-cache transfer.

Given the sender context $\mathbf{c}_i^s$, a standard prefill pass produces the final hidden state at the last context position and the corresponding sender-side KV cache:

$$\left(\mathbf{h}_{i,1}^s, \mathcal{C}_{i,0}^s\right) = \mathrm{Prefill}_S\left(\mathbf{c}_i^s\right). \tag{2}$$

Starting from this prompt-terminal state, the sender recursively extends its computation as

$$\left(\mathbf{h}_{i,t+1}^s, \mathcal{C}_{i,t}^s\right) = \mathrm{Step}_S\left(\mathbf{h}_{i,t}^s, \mathcal{C}_{i,t-1}^s; \boldsymbol{\kappa}_{i,t}\right), \qquad t = 1, \ldots, K-1, \tag{3}$$

where $\boldsymbol{\kappa}_{i,t}$ contains the model-native position metadata, keeping cache indexing and positional encoding consistent with the sender's native decoding procedure. Counting the prompt-terminal state as the first position gives

$$\mathbf{H}_i^s = \left[\mathbf{h}_{i,1}^s, \ldots, \mathbf{h}_{i,K}^s\right] \in \mathbb{R}^{K \times d_s}. \tag{4}$$

Thus, K counts the exposed states, including the prompt-terminal state, while Step_S is invoked $K-1$ times. The KV cache supports recurrence but is not communicated; only $\mathbf{H}_i^s$ is exposed. Consequently, the communicated payload scales with the trajectory length and hidden width, rather than with the number of layers and the full cached sequence length as in layer-wise KV-cache transfer.

Receiver-Grounded Latent Translation. Because the two frozen agents may have different hidden widths and representation bases, the sender states cannot be injected directly. A shared two-layer MLP maps each sender state independently into the receiver input-embedding space:

$$\mathbf{z}_{i,t} = T_\phi\left(\mathbf{h}_{i,t}^s\right), \qquad T_\phi : \mathbb{R}^{d_s} \to \mathbb{R}^{d_r}. \tag{5}$$

Applying it across the trajectory gives

$$\mathbf{Z}_i = \left[\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,K}\right] \in \mathbb{R}^{K \times d_r}. \tag{6}$$

The translator preserves length and order but does not mix positions; the frozen receiver performs joint modeling through its native self-attention. It is trained through receiver-side prediction, without hidden-state reconstruction or coordinate-wise alignment.

Continuous Message Injection. The translated trajectory is inserted into the receiver input sequence as

$$\widetilde{\mathbf{X}}_i^r(\mathbf{Z}_i) = \left[\mathbf{X}_{\mathrm{sys},i}^r; \mathbf{X}_{\mathrm{vis},i}^r; \mathbf{Z}_i; \mathbf{X}_{\mathrm{qry},i}^r; \mathbf{X}_{\mathrm{asst}}^r\right]. \tag{7}$$

Here, $\mathbf{X}_{\mathrm{sys},i}^r$, $\mathbf{X}_{\mathrm{vis},i}^r$, $\mathbf{X}_{\mathrm{qry},i}^r$, and $\mathbf{X}_{\mathrm{asst}}^r$ denote the system, visual, query, and assistant-prefix embeddings, respectively. The K message positions therefore follow the visual tokens and precede the query, assistant prefix, and generated answer. Under the native causal mask, subsequent text positions can jointly attend to the receiver's multimodal context and the sender-provided message without an additional cross-agent attention module.

Learning the Latent Collaboration Interface

The three training objectives provide complementary supervision for T_ϕ. The receiver task objective makes the translated message predictive of the target answer. The correspondence objective encourages a higher receiver score for the input-matched trajectory than for a mismatched trajectory,

rather than learning an input-agnostic continuous prompt. The confidence-gated trust objective uses a position-matched null receiver as a token-level soft reference where that reference is reliable. It requires neither binary rollout-level validity labels nor assumptions about the sign of an individual rollout's receiver-side effect. Only ϕ is optimized; both agents remain frozen.

Receiver Target Score. For target answer $(y^r_{i,1}, \ldots, y^r_{i,M_i})$, define the length-normalized teacher-forced score

$$s^r_i(\mathbf{Z}; \mathbf{A}) = \frac{1}{M_i} \sum_{j=1}^{M_i} \log p_R\left(y^r_{i,j} \mid \widetilde{\mathbf{X}}^r_i(\mathbf{Z}), y^r_{i,<j}; \mathbf{A}\right). \tag{8}$$

Only answer positions contribute. System, visual, message, query, and control positions are excluded, and length normalization makes scores comparable across different answer lengths. Let $\mathbf{A}^{\mathrm{col}}$ denote the native causal mask with the message enabled.

Receiver Task Objective. For minibatch size B,

$$\mathbf{Z}^+_i = T_\phi(\mathbf{H}^s_i), \quad s^+_i = s^r_i\left(\mathbf{Z}^+_i; \mathbf{A}^{\mathrm{col}}\right), \quad \mathcal{L}_{\mathrm{task}} = -\frac{1}{B}\sum_{i=1}^{B} s^+_i. \tag{9}$$

Gradients pass through the frozen receiver into T_ϕ. Because the receiver is frozen, this objective changes how the trajectory is expressed in its embedding space rather than modifying the receiver.

Input-Conditioned Correspondence Objective. A useful interface should preserve the relation between a trajectory and the multimodal input that produced it. We therefore use another instance in the same minibatch as an input-mismatched reference. For a fixed-point-free minibatch permutation π with $\pi(i) \neq i$, the stop-gradient mismatched score is

$$s^-_i = \mathrm{sg}\left[s^r_i\left(T_\phi\left(\mathbf{H}^s_{\pi(i)}\right); \mathbf{A}^{\mathrm{col}}\right)\right], \tag{10}$$

where sg denotes stop-gradient. We then define

$$\mathcal{L}_{\mathrm{corr}} = \frac{1}{B}\sum_{i=1}^{B}\left[m_{\mathrm{corr}} - s^+_i + s^-_i\right]_+, \tag{11}$$

where $[x]_+ = \max(x, 0)$ and $m_{\mathrm{corr}} > 0$. The hinge is one-sided: stop-gradient makes the mismatch a fixed reference, so an active term raises the matched score rather than depressing an unrelated trajectory. We resample π at each update and omit this term for $B < 2$.

Confidence-Gated Receiver Trust Regularization. The one-hot task objective rewards the observed token but provides no reference for the relative structure among the remaining token probabilities. We use a position-matched null receiver as a soft reference, weighted more strongly where its prediction is label-verified and confident. The null branch retains K zero placeholders at the message positions but masks them from subsequent attention, preserving the remaining positional layout. Retaining these placeholders avoids confounding message removal with a shift in the downstream positional layout. Let $\mathbf{Z}^0_i = \mathbf{0}_{K\times d_r}$, $\mathbf{A}^+ = \mathbf{A}^{\mathrm{col}}$, and let $\mathbf{A}^0$ denote the null mask. For $a \in \{0, +\}$, let $\boldsymbol{\ell}^a_{i,j}$ be the frozen receiver logits conditioned on $\widetilde{\mathbf{X}}^r_i(\mathbf{Z}^a_i)$ and $y^r_{i,<j}$ under $\mathbf{A}^a$. For temperature $\tau > 0$, define

$$p^{a,\tau}_{i,j}(\cdot) = \mathrm{softmax}\left(\boldsymbol{\ell}^a_{i,j}/\tau\right), \qquad p^a_{i,j} = p^{a,1}_{i,j}. \tag{12}$$

The continuous trust weight is

$$\omega_{i,j} = \left(\mathrm{sg}\left[p^0_{i,j}(y^r_{i,j})\right]\right)^\gamma, \qquad \gamma > 0. \tag{13}$$

The gate is computed from the frozen null branch and detached from optimization. Because it uses the observed-token probability, it varies continuously from strong trust in a label-consistent, confident reference to weak trust in an unreliable one; a confidently incorrect null prediction therefore receives low weight. The weights are not renormalized across target positions. We define

$$\mathcal{L}_{\mathrm{trust}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{M_i}\sum_{j=1}^{M_i}\omega_{i,j}\tau^2 D_{\mathrm{KL}}\left(\mathrm{sg}\left[p^{0,\tau}_{i,j}(\cdot)\right] \middle\| p^{+,\tau}_{i,j}(\cdot)\right). \tag{14}$$

Table 2: Communication results across sender–receiver configurations. Parentheses report signed differences from the corresponding Receiver-only baseline; bold and underlining mark the best and second-best results within each benchmark–configuration pair.

Benchmark	Receiver-only	Text communication	C2C-VLM	DenseAlign-VLM	LatentCollab
Qwen3-VL-2B → Qwen3-VL-4B					
GQA	<u>62.80</u> (0.00)	59.32 (−3.48)	58.00 (−4.80)	55.47 (−7.33)	**67.24** (+4.44)
TextVQA$_{val}$	<u>81.84</u> (0.00)	70.83 (−11.01)	74.31 (−7.53)	74.02 (−7.82)	**85.38** (+3.54)
POPE	89.81 (0.00)	<u>91.07</u> (+1.26)	85.17 (−4.64)	85.99 (−3.82)	**91.73** (+1.92)
VizWiz	41.58 (0.00)	45.35 (+3.77)	**50.41** (+8.83)	39.06 (−2.52)	<u>48.88</u> (+7.30)
SQA	80.43 (0.00)	85.08 (+4.65)	88.20 (+7.77)	<u>89.04</u> (+8.61)	**90.63** (+10.20)
MME	<u>2342.50</u> (0.00)	2256.17 (−86.33)	2305.55 (−36.95)	2266.86 (−75.64)	**2399.15** (+56.65)
MMB	80.36 (0.00)	83.69 (+3.33)	<u>87.21</u> (+6.85)	84.20 (+3.84)	**87.91** (+7.55)
Qwen3-VL-4B → Qwen3-VL-8B					
GQA	<u>63.68</u> (0.00)	59.91 (−3.77)	58.95 (−4.73)	59.60 (−4.08)	**69.75** (+6.07)
TextVQA$_{val}$	<u>82.07</u> (0.00)	70.05 (−12.02)	74.68 (−7.39)	74.17 (−7.90)	**85.44** (+3.37)
POPE	88.06 (0.00)	<u>89.93</u> (+1.87)	84.21 (−3.85)	85.36 (−2.70)	**91.57** (+3.51)
VizWiz	49.37 (0.00)	50.27 (+0.90)	<u>58.62</u> (+9.25)	54.08 (+4.71)	**63.85** (+14.48)
SQA	<u>94.40</u> (0.00)	90.68 (−3.72)	93.41 (−0.99)	92.81 (−1.59)	**95.22** (+0.82)
MME	2363.05 (0.00)	2336.71 (−26.34)	<u>2398.14</u> (+35.09)	2332.79 (−30.26)	**2428.56** (+65.51)
MMB	82.96 (0.00)	88.10 (+5.14)	88.32 (+5.36)	**88.60** (+5.64)	<u>88.46</u> (+5.50)
InternVL3.5-2B → Qwen3-VL-4B					
GQA	<u>62.80</u> (0.00)	58.56 (−4.24)	59.31 (−3.49)	60.57 (−2.23)	**66.42** (+3.62)
TextVQA$_{val}$	<u>81.84</u> (0.00)	68.29 (−13.55)	74.42 (−7.42)	73.97 (−7.87)	**84.70** (+2.86)
POPE	89.81 (0.00)	<u>90.77</u> (+0.96)	86.25 (−3.56)	86.06 (−3.75)	**91.13** (+1.32)
VizWiz	41.58 (0.00)	37.00 (−4.58)	43.66 (+2.08)	<u>49.86</u> (+8.28)	**53.11** (+11.53)
SQA	80.43 (0.00)	85.29 (+4.86)	85.29 (+4.86)	<u>89.19</u> (+8.76)	**91.52** (+11.09)
MME	<u>2342.50</u> (0.00)	2216.76 (−125.74)	2298.69 (−43.81)	2291.73 (−50.77)	**2370.71** (+28.21)
MMB	80.36 (0.00)	84.03 (+3.67)	**87.96** (+7.60)	<u>87.82</u> (+7.46)	84.85 (+4.49)

Unlike the one-hot task objective, this term regularizes the full receiver distribution. The KL term uses the detached null distribution as a reference to regularize the collaboration distribution. Gradients pass only through the collaboration branch into T_ϕ. The regularization is stronger where the null receiver is correct and confident and weaker where task supervision should dominate.

Overall Objective. The three objectives are combined with equal weight:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{corr}} + \mathcal{L}_{\text{trust}}. \tag{15}$$

The mismatched and null branches are training-only. At inference, the sender produces $\mathbf{H}_i^s$, the translator maps it to $\mathbf{Z}_i$, and the receiver generates the output through the single collaboration path, with no auxiliary receiver forwards.

4 Experiments

4.1 Empirical Evaluations

Tasks and Evaluation. We evaluate LatentCollab on seven vision-language benchmarks: GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), POPE (Li et al., 2023b), VizWiz (Gurari et al., 2018), ScienceQA (Lu et al., 2022), MME (Fu et al., 2025a), and MMBench (Liu et al., 2024b). These benchmarks cover compositional visual question answering, scene-text understanding, object hallucination, low-quality visual inputs, multimodal science reasoning, and general visual understanding. All experiments are conducted with the LMMS-Eval framework (Zhang et al., 2025) using its standardized task definitions and metric implementations.

Models and Baselines. We evaluate two same-family cross-scale configurations, Qwen3-VL-2B (Bai et al., 2025) → Qwen3-VL-4B and Qwen3-VL-4B → Qwen3-VL-8B, and one cross-family configuration, InternVL3.5-2B (Wang et al., 2025) → Qwen3-VL-4B. Baselines include *Receiver-only* without sender information, *Text Communication* with sender-generated natural-language analysis, and C2C-VLM (Fu et al., 2025b) and DenseAlign-VLM (Chen et al., 2026), which transfer KV caches through learned cross-model representation alignment. All methods within each configuration use the same models and receiver-side evaluation protocol.

Implementation Details. We use $K = 32$ by default and train a separate translator per configuration on 25K randomly sampled LLaVA training examples (Liu et al., 2024a). AdamW (Loshchilov & Hutter, 2019) optimizes only T_ϕ: first with the task objective at a learning rate of 10^{-3} and effective global batch size of 128, then jointly with all three objectives at 10^{-4} and 64.

4.2 Main Results

Table 2 shows that LatentCollab outperforms both *Receiver-only* and *Text Communication* on all 21 benchmark–configuration pairs, ranking first on 18 and within the top two on 20. Gains cover same-family cross-scale and cross-family collaboration, including 14.48 points on VizWiz with Qwen3-VL-4B → Qwen3-VL-8B and 11.09 points on ScienceQA in the cross-family configuration.

LatentCollab also provides more consistent gains than the competing KV-cache communication baselines, outperforming both C2C-VLM and DenseAlign-VLM on 18 of the 21 benchmark–configuration pairs. In the InternVL3.5-2B → Qwen3-VL-4B configuration, it achieves the best result on six of the seven benchmarks. The remaining exceptions are limited to individual metrics: C2C-VLM performs best on VizWiz for Qwen3-VL-2B → Qwen3-VL-4B and on MMBench for the cross-family configuration, while DenseAlign-VLM slightly leads on MMBench for Qwen3-VL-4B → Qwen3-VL-8B. Overall, LatentCollab provides a reliable collaboration interface across model scales and families.

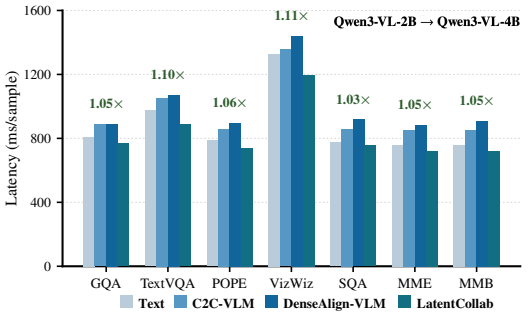


Figure 4: Per-sample inference latency for Qwen3-VL-2B → Qwen3-VL-4B across seven benchmarks. Lower is better. Values above each benchmark denote the speedup of LatentCollab over Text Communication.

4.3 Inference Time

As shown in Fig. 4, LatentCollab achieves the lowest latency among the compared communication methods across all seven benchmarks. It provides consistent speedups of $1.03\times$–$1.11\times$ over *Text Communication* and is also faster than C2C-VLM and DenseAlign-VLM. These results show that its accuracy improvements do not come at the cost of increased inference latency. The modest but consistent speedups support a favorable accuracy–latency trade-off for the evaluated model pair.

4.4 Ablation Studies

Effect of Rollout Length. Under Qwen3-VL-2B → Qwen3-VL-4B, all tested lengths ($K = 8$ to 128) in Table 3 outperform Receiver-only on all four benchmarks, showing that gains do not depend on a specific length. Performance is nonmonotonic: GQA and MME peak at $K = 64$ and 16, respectively, while POPE and ScienceQA peak at $K = 32$. We adopt $K = 32$ to balance overall performance, message length, and rollout cost across all four evaluated multimodal benchmarks.

Effect of Training Objectives. In Table 4, the full objective performs best on all four benchmarks. Removing $\mathcal{L}_{\text{corr}}$ has the largest effect: GQA, POPE, ScienceQA, and MME drop by 3.65, 2.98, 9.97, and 32.80 points, respectively; POPE falls below Receiver-only, while GQA and ScienceQA retain only marginal gains. This highlights the importance of correspondence supervision, which encourages the translator to preserve trajectory–input associations rather than collapse to an input-agnostic continuous prompt that is insensitive to the sender's specific image–question pair.

Table 3: Effect of the sender-side trajectory length K under the Qwen3-VL-2B $\rightarrow$ Qwen3-VL-4B configuration. The best and second-best results are shown in bold and underlined.

K	GQA	POPE	SQA	MME
8	66.34	90.26	84.60	2366.47
16	66.90	<u>91.14</u>	84.82	**2407.35**
32	<u>67.24</u>	**91.73**	**90.63**	<u>2399.15</u>
64	**68.05**	90.36	<u>85.71</u>	2385.29
128	66.48	90.51	84.70	2390.16

Table 4: Ablation of the training objectives under the Qwen3-VL-2B $\rightarrow$ Qwen3-VL-4B configuration. The best and second-best results are shown in bold and underlined.

Objective	GQA	POPE	SQA	MME
Receiver-only	62.80	89.81	80.43	2342.50
LatentCollab	**67.24**	**91.73**	**90.63**	**2399.15**
w/o $\mathcal{L}_{\text{corr}}$	63.59	88.75	80.66	2366.35
w/o $\mathcal{L}_{\text{trust}}$	<u>65.91</u>	<u>90.84</u>	<u>87.26</u>	<u>2380.90</u>

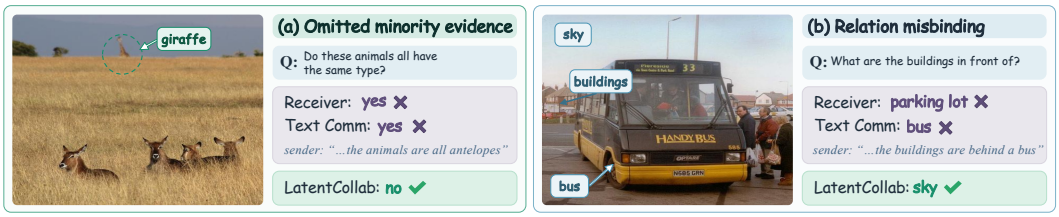


Figure 5: Qualitative case study comparing Receiver-only, Text Communication, and LatentCollab. For readability, we quote the richer-evidence qualitative sender messages; the official text protocol produces the same incorrect answers in both examples.

Removing $\mathcal{L}_{\text{trust}}$ reduces GQA, POPE, ScienceQA, and MME by 1.33, 0.89, 3.37, and 18.25 points relative to the full objective, respectively, but all remain above Receiver-only and the degradations are smaller than without $\mathcal{L}_{\text{corr}}$. Both auxiliary objectives contribute, with correspondence supervision having the larger effect. This is consistent with the trust term limiting deviations from reliable receiver predictions while favoring task supervision when the null reference is unreliable for the target token.

4.5 Case Study

Figure 5 illustrates two limitations of text-mediated collaboration: omission of task-relevant visual evidence and distortion of spatial relations. In (a), the distant giraffe provides a decisive counterexample to the dominant foreground category, but this evidence is absent from the sender's textual description. In (b), the sender's description mentions relevant scene entities but frames a different spatial relation from the one queried, leading text communication to an incorrect referent. Both errors suggest that message fidelity depends on preserving evidence relevant to the query, beyond naming salient objects in the scene. LatentCollab answers both questions correctly. These outcomes are consistent with our motivation to communicate task-conditioned multimodal states before verbalization, reducing reliance on the sender's explicit textual account. Together, the cases highlight the potential of latent collaboration to support judgments involving subtle visual evidence and precise relational distinctions.

5 Conclusion

We introduce LatentCollab, a latent collaboration framework that enables vision-language agents to exchange task-conditioned computation through compact continuous messages. Across seven benchmarks and three same-family or cross-family sender–receiver configurations, LatentCollab outperforms Receiver-only and Text Communication on all 21 comparisons and both KV-cache communication baselines on 18, while achieving lower inference latency than all evaluated communication baselines. These results show that final-layer latent trajectories can provide an effective interface for transferring multimodal computation without intermediate text or KV-cache exchange. LatentCollab therefore offers a practical foundation for continuous collaboration among heterogeneous vision-language agents across model scales and families through compact continuous messages.

AI USE STATEMENT

In preparing this manuscript, we used AI tools for language polishing, including improving grammar, wording, clarity, and organization. The authors reviewed all AI-assisted text, checking technical descriptions, mathematical notation, and statements about experimental results for accuracy and consistency with the underlying work. We take full responsibility for the final content of this paper, including all text, claims, and artifacts produced with the assistance of AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, Sec. 3.2 describes the LatentCollab framework, including latent trajectory construction, translation, message injection, and training objectives. Sec. 4.1 documents the training data, evaluation benchmarks, sender–receiver configurations, comparison methods, and evaluation protocol, together with implementation settings such as the trainable components, default trajectory length, optimizer, learning rates, and batch sizes. All datasets used in our experiments are publicly available, with their sources cited in the main text.

REFERENCES

Ashutosh Adhikari and Mirella Lapata. Debating for Better Reasoning in Vision-Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15766–15784, 2025.

Lisa Alazraki, Lluis Castrejon, Mostafa Dehghani, Fantine Huot, Jasper Uijlings, and Thomas Mensink. How (not) to ensemble LVLMs for VQA. In *Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops*, volume 239 of *Proceedings of Machine Learning Research*, pp. 1–20. PMLR, 2023. URL https://proceedings.mlr.press/v239/alazraki23a.html.

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.

Liangyu Chen, Bo Li, Sheng Shen, Jingkang Yang, Chunyuan Li, Kurt Keutzer, Trevor Darrell, and Ziwei Liu. Large language models are visual reasoning coordinators. *Advances in Neural Information Processing Systems*, 36:70115–70140, 2023.

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, 2026. URL https://arxiv.org/abs/2606.13594.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

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

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2025a.

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*, 2025b.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz Grand Challenge: Answering Visual Questions From Blind People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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, Steven Yau, Zijuan Lin, Liyang Zhou, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Aymen Lassoued, Mohamed Ali Souibgui, and Yousri Kessentini. ORCA: Orchestrated Reasoning with Collaborative Agents for Document Visual Question Answering. *arXiv preprint arXiv:2603.02438*, 2026.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for" mind" exploration of large language model society. *arXiv preprint arXiv:2303.17760*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024a.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. LLaVA-Plus: Learning to use tools for creating multimodal agents. *arXiv preprint arXiv:2311.05437*, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards Interpreting Visual Information Processing in Vision-Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chanJGoa7f.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In *International Conference on Learning Representations*, volume 2024, pp. 51899–51928, 2024.

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. *arXiv preprint arXiv:2501.14082*, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Nithin Sivakumaran, Justin Chen, David Wan, Yue Zhang, Jaehong Yoon, Elias Stengel-Eskin, and Mohit Bansal. DART: Leveraging multi-agent disagreement for tool recruitment in multimodal reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5445–5464, 2026.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. Augmenting Multi-Agent Communication with State Delta Trajectory. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10230–10251, 2025.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First conference on language modeling*, 2024.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.51. URL https://aclanthology.org/2025.findings-naacl.51/.

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